tesis defendida por Leonardo Trujillo Reyes

Y aprobada por el siguiente comité:

Dr. Gustavo Olague Caballero

Director del Comité

Dr. Oscar Castillo López Miembro del Comité Dr. Cesar Cruz Hernández Miembro del Comité

Dr. Jésus Favela Vara Miembro del Comité

Dr. Pedro Gilberto López Mariscal

Coordinador del Posgrado en Ciencias en Ciencias de la Computación Dr. David Hilario Covarrubias Rosales

Director de Estudios de Posgrado

21 de Noviembre del 2008.

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA



PROGRAMA DE POSGRADO EN CIENCIAS EN CIENCIAS DE LA COMPUTACIÓN

Cómputo evolutivo aplicado en el diseño de métodos para la detección y descripción de regiones de interés

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

DOCTOR EN CIENCIAS

Presenta:

Leonardo Trujillo Reyes

Ensenada, Baja California, México. Noviembre del 2008.

RESUMEN de la tesis de **Leonardo Trujillo Reyes**, presentada como requisito parcial para obtener el grado de DOCTOR EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, B. C. Noviembre del 2008.

Cómputo evolutivo aplicado en el diseño de métodos para la detección y descripción de regiones de interés

Resumen aprobado por:

Dr. Gustavo Olague Caballero

Director de Tesis

En la última década, una parte significativa de la comunidad en visión por computadora ha adoptado un enfoque basado en la detección y descripción de rasgos locales para abordar una variedad amplia de problemas, entre ellos esta el reconocimiento o detección de objetos, el indexado de imágenes, la recuperación de imágenes en base al contenido, y la visión estéreo. Este enfoque depende de dos tareas básicos: la *detección* y la *descripción* de puntos interesantes dentro de la imagen. Estas tareas se realizan utilizando operadores, algoritmos y procesos especializados.

En particular, los métodos propuestos han sido diseñados de manera directa por expertos humanos, el enfoque clásico para la solución de problemas científicos y tecnológicos. En esta tesis, la meta principal es proponer un enfoque basado en el paradigma del cómputo evolutivo que nos permita estudiar y sintetizar operadores y técnicas computacionales que faciliten la detección y descripción de regiones de interés en imágenes digitales. Si consideramos que la visión es una de las habilidades más prominentes en los seres cognitivos, entonces podemos apreciar la motivación subyacente que existe en proponer soluciones a los problemas de visión utilizando técnicas que permitan reproducir ciertos aspectos de la inteligencia humana a través de procesos computacionales. También, la propuesta central de esta tesis esta basada en incorporar a estos problemas de visión procesos inspirados en la evolución natural. Es importante notar que muchos teóricos proponen que este mecanismo de la naturaleza ha tenido un papel central en la generación, una y otra vez, de las mismas capacidades visuales que ahora queremos reproducir en una computadora. Por otro lado, los modelos y abstracciones de los trabajos que aquí se presentan, y otros similares en el área del cómputo evlutivo, aparentan ser muy básicos en comparación con los procesos naturales de evolución. Sin embargo, estos algoritmos han demostrado la habilidad de competir

con las mejores soluciones propuestas directamente por investigadores humanos en una variedad amplia de dominios de problemas.

Para lograr la meta general y los objetivos específicos de esta tesis, se formulan planteamientos nuevos para los problemas de *detección* y *descripción* de regiones locales, utilizando procesos de búsqueda y optimización evolutiva, los cuales no son comunes en visión artificial. Sin embargo, nuestros planteamientos estan basados en criterios de evaluación bien establecidos y ampliamente aceptados por la comunidad de visión.

Los resultados que se presentan en esta tesis ofrecen tres tipos de contribuciones. Primero, desde un punto de vista pragmático se proponen varios operadores y algoritmos que se comparan de manera favorable con el estado-del-arte en estas líneas de investigación.

Segundo, las soluciones propuestas combinan los productos de la evolución artificial con el análisis humano, y se integran con técnicas clásicas basadas en métodos matemáticos.

Tercero, desde una perspectiva conceptual esta tesis hace un cambio de enfoque importante durante el proceso de diseño para posibles soluciones a estos problemas. Se comprueba experimentalmente que los algoritmos evolutivos ofrecen la capacidad de generar soluciones nuevas, diferentes y poco ortodoxas para problemas bien conocidos que se consideran complejos por la comunidad de visión.

Así, se puede concluir que bajo ciertas restricciones, el enfoque propuesto provee una perspectiva diferente que complementa al trabajo que históricamente se ha realizado en el área de visión.

Palabras clave: Visión por computadora, cómputo evolutivo, algoritmos genéticos, programación genética, puntos y regiones de interés, descriptores locales, exponente Hölder, reconocimiento de objetos y lugares, modelos de mezclas de Gaussianas..

ABSTRACT of the thesis presented by **Leonardo Trujillo Reyes**, as a partial requirement to obtain the DOCTOR degree in COMPUTER SCIENCES. Ensenada, B. C. November 2008.

Evolutionary computation applied to the design of detection and description methods for interest regions

Abstract approved by:

Dr. Gustavo Olague Caballero

Thesis director

In recent years, a significant part of the computer vision community has adopted an approach based on local features for a variety of problems, which include object detection and recognition, image indexing, content based image retrieval, and stereo vision. This approach relies on two basic tasks: the *detection* and *description* of interesting points within an image. These tasks are carried out using specially designed operators, algorithms and processes.

In particular, all of the proposed methods have been designed directly by human experts in the field, the classic approach to problem solving in scientific and technological domains. In this thesis, the main goal is to propose an approach based on the evolutionary computation paradigm which can allow us to study and synthesize operators and computational techniques that facilitate the detection and description of interesting regions within digital images. If we consider that vision is one of the most prominent abilities exhibited by cognitive beings, then it is possible to appreciate the underlying motivation in developing solutions to vision problems using techniques that allow us to reproduce certain aspects of human intelligence through computational processes. Moreover, the main proposals in this work are based on incorporating to these vision tasks processes which are inspired by natural evolution. It is important to note that many researchers believe that this natural process has played a central in the development, time and again, of the same visual capacities that we are now attempting to reproduce in machines. On the other hand, some of the models used, and abstractions made, in the contributions presented here, and in other works in evolutionary computation, appear to be very basic when compared to their biological counterparts. Nevertheless, these algorithms have shown that they can generate solutions that compete on equal terms with man-made proposals in a variety of problem domains.

In order to achieve each objective in this thesis, new formulations of the detection and description processes are proposed, based on optimization and search algorithms that are not common in artificial vision research. Nevertheless, our proposals are also based on well-known and amply accepted criteria used within the computer vision community.

The results presented in this work offer three types of contributions. First, from a pragmatic perspective, several operators and algorithms are proposed that compare favorably with state-of-the-art techniques in computer vision.

Second, the proposed solutions combine the results generated by artificial evolution with human analysis, and integrate them with classical mathematical methods.

Third, from a conceptual perspective this thesis proposes a change in approach during the design process of possible solutions to these problems. It is experimentally shown that evolutionary algorithms can generate new and different solutions to wellknown and complex problems in computer vision.

In conclusion, this thesis shows that under certain restrictions, the proposed approach provides a different perspective that complements the type of work that has historically been done in computer vision.

Keywords: Computer vision, evolutionary computation, genetic algorithms, genetic programming, interest points and regions, Hölder exponen, t object and place recognition, Gaussian mixture models.

Dedicatoria

A todos l@s de abajo y a la izquierda, rebeldes y dign@s

Agradecimientos

Agradezco sinceramente:

A mi familia, mis padres, mi novia, mi hermano, mis amigos, y ahora mi sobrina, el apoyo de todos ellos ha sido invaluable y no existen palabras que puedan explicar lo que siento por ellos y la gratitud que les tengo.

A mi asesor, sus consejos y su guía me permitieron desarrollar esta tesis de manera plena, y me ayudaron a comprender lo que implica realizar investigación científica en todos sus niveles.

A todas las personas con las que interactue en CICESE, el grupo EvoVisión, APIS Team-INRIA Francia, el Grupo de Evolución Artificial-Universidad de Extremadura España, y el Instituto Tecnológico de Tijuana, porque la ciencia, como toda actividad humana, se desarrolla de manera más plena en comunidad y en un ámbito de cooperación, discusión, crítica y apoyo.

A La Otra Ensenada, La Otra Campaña, y los pueblos rebeldes de México y el mundo, por darme un marco de referencia y un propósito a mi trabajo.

Finalmente, a todo el pueblo mexicano por darme el apoyo económico que me permite ejercer un trabajo creativo y personalmente satisfactorio, un privilegio que desafortunadamente, hasta el momento, solo se le ofrece a una pequeña minoría de nuestra gente.

Ensenada, México 21 de Noviembre del 2008. Leonardo Trujillo Reyes

Contenido

Secc	ión			Págin	a
\mathbf{Res}	umer	1			i
\mathbf{Abs}	tract	5		i	ii
Ded	licato	oria			v
Agr	adec	imientos	3	v	/i
Tab	la de	Conten	nido	\mathbf{v}	ii
\mathbf{List}	a de	Figuras		2	ci
\mathbf{List}	a de	Tablas		xi	x
Ι	Inti	roducció	on and the second se		1
	I.1	Enfo	ques modernos a la visión por computadora	• •	3
	I.2	Meta	s, motivación y objetivos	• •	7
		I.2.1	Motivación		8
		I.2.2	Objetivos específicos	• •	9
	I.3	Soluc	iones propuestas	1	.0
		I.3.1	Contribuciones	1	1
	I.4	Orgai	nizacin del documento	1	3
Π	Ant	ecedent	es teóricos y trabajo previo	1	6
	II.1	La ev	volución como un paradigma computacional	1	6
		II.1.1	Características de los algoritmos evolutivos	1	.8
		II.1.2	Algoritmos genéticos	2	!1
		II.1.3	Programación genética	2	2
	II.2	Evolu	ción con objetivos múltiples	2	:4
		II.2.1	Optimización multiobjetivo	2	24
		II.2.2	Algoritmos evolutivos multiobjetivo	2	27
	II.3	Cómp	puto evolutivo aplicado a visión	2	28
III	Pur	ntos de i	interés	3	6
	III.1	Rasgo	os visualmente interesantes	3	6
	III.2	2 Detec	ción de puntos de interés	3	8
	III.3	8 Propi	iedades de los puntos de interés	4	1
	111.4	l Medie	das de evaluación para un detector de puntos de interés .	4	±6
		111.4.1	Tasa de repetibilidad	4	±6
		111.4.2	Dispersión de puntos	4	±7
	F	111.4.3	Contenido de información	4	±8
IV	Des	criptore	es locales	5	1
	IV.1	Descr	iptor incluyente	· · · · ·	4
	1V.2	2 Descr	riptor discriminante	č	16 16
		IV.2.1	Taxonomia	••• 5	,9 ,9
		1V.2.2	Metodo de evaluación		52

	IV	.2.3	El descriptor SIFT y sus derivados	66
\mathbf{V}	Diseño	auto	mático de operadores que detectan puntos de interés	71
	V.1	La evo	olución de operadores con Programación Genética	71
	V.	1.1	Espacio de búsqueda	73
	V.	1.2	Evaluación de aptitud	75
	V.2	Result	ados experimentales	78
	V.	2.1	Implementación	78
	V.	2.2	Operadores sintetizados	80
	V.	2.3	Análisis y discusión	86
	V.	2.4	Resultados competitivos con diseños hechos por humanos .	96
	V.3	Búsqu	eda distribuida y paralela de operadores	98
	V.	3.1	Configuración experimental	101
	V.	3.2	Resultados preliminares	101
	V.	3.3	Rendimiento del esquema BOINC/VMware	104
	V.4	Resun	nen y conclusiones	105
VI	Un enf	ioque	multiobjetivo para la detección de puntos de interés	106
	VI.1	Objeti	ivos de optimización	106
	VI	.1.1	Propiedades básicas para evaluar los detectores de puntos	
	1 7 1		interés	107
	VI VI	1.1.2	Medidas de evaluación para un detector de puntos de interés	107
	V1.2	Medid	as para el contenido de información	108
	VI VI 2	.2.1 CD	Contenido de información con la entropia de Shannon	109
	V1.3	GP m	ultiobjetivo para la sintesis de operadores que detectan puntos	111
	17T	ae mu 191	Errosia da búzarrada	111
	VI VI	1.0.1 [2 0	Espació de busqueda	112
	VI A	L.J.Z Dogult	Evaluación de los objetivos	110
	VI.4 VI	1.esun [/ 1	Estabilidad Dispersión	115
	VI	[]]]	Estabilidad Contonido de información	194
	VI	[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]	Dispersión-Contenido de información	124
	VI	[4 4	Estabilidad-Dispersión-Contenido de información	141
	VI 5	Discus	sión y trabajo futuro	143
VII	Detecc	ión ir	avariante a escala con operadores evolucionados	147
, 11	VII 1	Detec	ción invariante a escala	147
	VII.2	Espac	jo escala	148
	VII.3	Como	elegir una escala característica	151
	VII.4	Estade	o-del-arte en detectores invariantes a escala	153
	VII.5	Como	evaluar un detector invariante a escala	154
	VII.6	Detect	tores propuestos	155
	VII.7	Evalua	ación experimental	158
	VII.8	Resun	nen y conclusiones	160

VIII	[Descripte	ores locales construidos a partir de una búsqueda evolutiva162
	VIII.1 De	escripción de la propuesta
	VIII.2 El	espacio del descriptor
	VIII.3 Mo	odelos de mezcla de Gaussianas
	VIII.	3.1 El discriminante lineal de Fisher 169
	VIII.4 El	algoritmo genético 169
	VIII.	4.1 Representación de la solución
	VIII.	4.2 Evaluación de aptitud $\ldots \ldots 171$
	VIII.5 Ev	aluación experimental
	VIII.	5.1 Reconocimiento de objetos $\dots \dots \dots$
	VIII.	5.2 Reconocimiento de lugares reales
	VIII.6 Re	sumen y trabajo futuro \ldots \ldots \ldots \ldots \ldots \ldots 192
\mathbf{IX}	Un descr	iptor local basado en la regularidad de Hölder 194
	IX.1 Re	gularidad de Hölder
	IX.1.	1 Estimación del exponente Hölder
	IX.2 De	escripción local utilizando la regularidad puntual de Hölder $\ .\ .\ .\ 201$
	IX.2.	1 Resultados experimentales $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 204$
	IX.2.	2 Observaciones, limitaciones y preguntas abiertas 208
	IX.3 Re	educir las dimensiones del descriptor utilizando un GA 212
	IX.3.	1 Algoritmo genético
	IX.3.	2 Representación de la solución
	IX.3.	3 Función de evaluación $\dots \dots \dots$
	IX.3.	4 Resultados experimentales
	IX.4 Es	timaciónón del exponente Hölder utilizando regresión simbólica
	COI	n GP \ldots \ldots \ldots \ldots 221
	IX.4.	1 Algoritmo GP
	IX.4.	2 Espacio de búsqueda
	IX.4.	3 Función de evaluación $\dots \dots \dots$
	IX.4.	4 Resultados experimentales
	IX.4.	5 Discusión y conclusiones $\dots \dots \dots$
\mathbf{X}	Resumen	a, conclusiones y trabajo futuro 242
	X.1 Co	ontribuciones y limitaciones de la investigación $\dots \dots \dots \dots \dots 243$
	X.1.1	Detección de puntos de interés
	X.1.2	2 Descripción de regiones locales
	X.2 Tr	abajo futuro
Refe	erencias	247
\mathbf{A}	Computa	ación GRID voluntaria con BOINC 265
	A.1 Inf	fraestructura BOINC
	A.1.1	$El Cliente \dots 266$
	A.1.2	$ El Servidor \dots 266 $
	A.1.3	B Como utilizar BOINC
	A.2 Vi	rtualización $\ldots \ldots 268$
	A.2.1	Virtualización y tecnologías GRID

A.3 Extensión de BOINC con una capa de virtual	lización 269
--	--------------

Lista de Figuras

Figura

1	Un modelo simplicado de sistemas de visión basados en el análisis de	
	regiones locales.	5
2	Regiones de interés detectadas en tres imágenes de la base de datos	5
3	Una perspectiva de alte nival de des enfegues para resolver problemas:	5
0	a) El método <i>clásico</i> , donde un proceso analítico es empleado para de-	
	scubrir las propiedades intrínsecas de un problema. Dependiendo del	
	proceso de análisis empleado diferentes soluciones tal vez en conflicto	
	pudieran encontrarse. Por lo tanto, se requieren de medidas de com-	
	paración cuantitativas para poder elegir una solución dados los requer-	
	imientos en algún contexto. b) El método del CE, en donde la síntesis y	
	el análisis se explotan de forma equivalente. El problema y las funciones	
	empleadas para evaluar posibles soluciones son analizados de manera	
	conjunta. Las funciones de evaluación se utilizan como guía para el pro-	
	ceso de búsqueda evolutiva. De tal forma que se sintetizan soluciones y	
	se ordenan en base a su rendimiento	19
4	La estructura básica de un EA, con tres módulos primordiales: evalu-	
	ación, administración de la población, y variación	20
5	La estructura básica de un GP.	23
6	El espacio de decisión, y su correspondiente espacio de objetivos. Una	
	solución \mathbf{x} se mapea a una función vectorial f a un vector en el espacio	
	de las funciones objetivo. Los puntos remarcados en la frontera de Λ son	05
7	elementos del Frente Pareto.	25
1	en un par estáreo de imágenes	97
8	Cada columna muestra al proceso de detección de puntos: columna	57
0	izquierda la imagen de entrada I: en medio la <i>imagen de interés</i> I*:	
	columna derecha los puntos finales detectados después de la supresión	
	de no-máximos y el umbralizado. Los puntos de interés en estos ejemplos	
	se detectaron con el operador K_{IPGP2}	41
9	Un punto 3D X se proyecta en x_1 y x_2 en las imágenes I_1 y I_2 respec-	
	tivamente. El punto x_1 se repite en x_i si se detecta un punto en un	
	vecindario de x_i del tamaño ϵ . Para escenas planas x_1 y x_i se relacionan	
	por la homografía $H_{1,i}$	46
10	Dispersión de puntos: Imagen original, puntos altamente dispersos, pun-	
	tos con poca dispersión espacial	48
11	Una descripción gráfica del concepto de contenido de información	49

12	Se muestra el diagrama básico empleado en la descripción de regiones locales. Primero, se detectan las regiones de interés. Segundo, se nor- maliza cada región de la imagen, por ejemplo en base a la intensidad o la orientación principal del gradiente. Tercero, un proceso o algoritmo con-	
	struye un vector de rasgos numéricos, normalmente concatenados, que	
	constituyen al descriptor final	52
13	Diagrama conceptual para los descriptores incluyentes. En la figura se observa como los descriptores que se obtienen de cada objeto se mapean	
14	a una misma región del espacio del descriptor empleado Diagrama conceptual que describe la manera en la cual se obtienen la	54
	correspondencia entre dos regiones de interes utilizando el descriptor io-	57
15	Aplicación bésica de un descriptor discriminante, en este caso es el SIFT	57
10	(Lowe, 1999), para el problema de reconocimiento de objetos. $(\mathbf{a} - \mathbf{c})$ Imágenes originales de una escena y dos objetos. $(\mathbf{d} - \mathbf{f})$ Begiones y	
	descriptores SIFT calculados en cada imagen. (g. h) Correspondencias	
	calculadas utilizando los descriptores SIFT.	58
16	Dos curvas típicas Recall vs 1-Precisión.	64
17	Una comparación entre el análisis ROC (TPV/TPF) y Recall/Precisión.	
	En donde: $PV = positivo verdadero; PF = positivo falso; NF = negativo$	
	falso; $NV = negativo verdadero; TPV = tasa de positivos verdaderos;$	
	TPF = tasa de positivos falsos.	66
18	El descriptor SIFT (Mikolajczyk y Schmid, 2005). (a) Región detectada.	
	(b) Imagen del gradiente y cuadricula de posición. (c) Las tres dimen-	
	siones del histograma. (d) Cuatro de los ocho planos de orientación. (e)	
	Cuadricula cartesiana y rejilla en coordenadas polares; la segunda es la	
	implementada por el descriptor GLOH (Mikolajczyk y Schmid, 2005).	68
19	El espacio de operadores construïdos con primitivas tomadas de $\{F \cup T\}$.	
	Se muestran tres subespacios: a) Ω_{δ} contiene derivadas de la imagen	
	tomadas de I; b) Ω_A contiene operadores que utilizan algun elemento de la matriz A; y c) Ω_A que contiene operadores que con funcionalmento	
	de la matriz A, y c) Ω_{β} que contiene operadores que son funcionalmente equivalentes, en este esse operadores que obtienen una medida de interés	
	relacionada con la curvatura do una suporficio	74
20	Como se comporta el término ϕ^{α} de la función de aptitud $f(K)$ Muestra	14
20	el valor que obtiene tres operadores: de izquierda a derecha: primero, un	
	operador que detecta puntos muy amontonados: segundo. K_{IPCP2} con	
	una dispersión buena de puntos; y finalmente, un operador que detecta	
	puntos muy dispersos.	77
21	Ejemplos de la secuencia de entrenamiento Van Gogh: I_1 , I_3 , y I_7	79
22	Estadísticas durante la evolución de los operadores K_{IPGP1} y K_{IPGP2} ,	
	los operadores corresponden a la primera y segunda columna respectiva-	
	mente. Las columnas de izquierda a derecha: 1) Imagen de interé s $I^{\ast},2)$	
	Gráficas de aptitud, y 3) Diversidad de la población.	81

23	La tasa de repetibilidad que obtiene K_{IPGP1} y K_{IPGP2} sobre con la sequencia completa de Van Coghi comparados con los operadores de	
	(Beaudet 1978: Kitchen v Rosenfeld 1982: Harris v Stephens 1988:	
	Wang v Bradv 1991: Förstner v Gülch 1987)	82
24	El genotino de los operadores K_{LPGP2} , K_{LPGP2} v K_{uarris} En el caso	02
21	del operador de Harris este sería el genotipo más sencillo que pudiera	
	tener si se construve con el algoritmo GP	84
25	Comparación cualitativa entre los operadores K_{IPCP1} , K_{IPCP2} , v $K_{Harrier}$	01
	Renglones: Van Gogh, Graph v Monet,	85
26	Estadísticas para la evolución de los operadores en la Tabla V. Las	
	columnas: 1) aptitud (mediana, promedio, v mejor), 2) diversidad, 3)	
	complejidad (<i>Profundidad máxima dinámica</i> , la profundidad del mejor	
	individuo, y el número de nodos del mejor individuo). Renglones:	
	$IPGP1^*$, $IPGP3$, $IPGP4$, $IPGP5$, y $IPGP6$	88
27	Estadísticas para la evolución de los operadores en la Tabla V, por	
	renglón: IPGP7, IPGP8, IPGP9, IPGP10, y IPGP11	89
28	Estadísticas para la evolución de los operadores en la Tabla V, por	
	renglón: $IPGP12$, $IPGP13$, $IPGP14$, $IPGP15$, y $IPGP16$	90
29	La frecuencia con que aparecen cada una de las primitivas de $\{F \cup T\}$	
	en la estructura del mejor individuo que se obtiene en cada corrida del	
	algoritmo.	93
30	La imagen de referencia (derecha) y una imagen transformada (izquierda)	
	para cada secuencia de prueba. a) Mars ($N = 18$, rotacion), b) New	
	York $(N = 35, rotacion), c)$ Graph $(N = 12, cambio de iluminacion), d)$ Maggia $(N = 10, cambio de iluminación)$	05
21	Mosaic $(N = 10, \text{ cambio de numinación})$	95
51	entrenamiento y las cuatro secuencias de prueba	96
32	Una comparación cualitativa Primer renglón imagen de New York con	50
02	400 puntos Segundo renglón imagen Mosaic con 200 puntos	97
33	Imágenes de interés que producen cada uno de los operadores incluidos	0.
	en la comparación.	98
34	Estadísticas referentes a la mejor aptitud y la aptitud promedio de la	
	población; se muestran promedios y desviación estándar con respecto al	
	número de soluciones que se obtuvieron en cada experimento.	103
35	El operador K_{IPGP17} aplicado a la secuencia Van Gogh	104

36	Frentes Pareto encontrados por el GP-MO para los criterios de Estabil- idad y Dispersión. La gráfica contiene los resultados de cuatro ejecu-	
	ciones del algoritmo, cada una con una profundidad máxima diferente.	
	También, se presenta la posición que ocupan, con respecto al frente, cua-	
	tro operadores de la literatura (Beaudet, 1978; Kitchen y Rosenfeld, 1982;	
	Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987)	
	y dos operadores evolucionadas con el enfoque mono-objetivo K_{IPGP1} y	
	K_{IPGP2} . Además, también se identifican tres operadores colocados cerca	
	de los puntos de inflexión: (a), (b) y (c). \ldots \ldots \ldots \ldots	116
37	Primer renglón: Imagen de interés extraída con los operadores (a), (b) y	
	(c) del frente Pareto mostrado en la Figura 36. Segundo renglón: Puntos	
	de interés detectados.	117
38	Comportamiento de las funciones de costo asociadas a la estabilidad y	
	dispersión de puntos para el operador K_{MO} con respecto al factor de peso	
	W, aplicado a la secuencia de Van Gogh. (a) El comportamiento de la	
	estabilidad del descriptor; el mejor rendimiento se obtiene en el rango $(0, 1]$ ha ha se la statistica (1). El se transforma de la statistica (1) estatistica (1) estatistic	
	(0, 1], donde es bastante estable. (b) El punto minimo en la granca	
	corresponde al operador (b) de la Figura 50. Es importante notar que se	
	ostabilidad geométrica del operador	110
39	Comportamiento de las funciones de costo asociadas a la estabilidad y	119
00	dispersión de puntos para el operador K_{MO} aplicado a dos secuencuas	
	de prueba: New York v Mars.	121
40	Comportamiento de las funciones de costo asociadas a la estabilidad	
	y dispersión de puntos para el operador de Harris cuando se varia el	
	parámetro k ; utilizando la secuencia de Van Gogh. Para valores de	
	k > 0.3el rendimiento del operador se degrada considerablemente y	
	los puntos en la gráfica son mucho mayores al rango mostrado. \ldots .	122
41	Puntos de interés detectados con el operador K_{MO} y cuatro valores de	
	peso W : 0, 0.05, 05, y 1. El primer renglón contiene la imagen de	
	entrenamiento Va Gogh, el segundo una imagen de la secuencia New	
	York, el tercero la imagen de Casita, en el cuarto la imagen Puerta, y en	100
49	el quinto la imagen Bip.	123
4Z	Comportamiento de las funciones de costo asociadas a la establidad y dispersión de puntos para el operador $K_{\rm est}$ simplificado, dondo se elimina	
	al toreor término de la Ecuación 21	195
43	Frentes Pareto encontrados con el GP-MO utilizando los objetivos de	120
10	Estabilidad y Contenido de información: el segundo se calcula utilizando	
	el descriptor SIFT. La gráfica contiene los resultados de cuatro ejecu-	
	ciones del algoritmo, cada una con una profundidad máxima diferente.	
	y se identifican tres operadores nuevos sobre el frente: (d), (e) v (f)	127

44	Primer renglón: Imagen de interés extraída con los operadores (d), (e) y (f) del frente Pareto mostrado en la Figura 43. Segundo renglón: Puntos	
	de interés detectados	128
45	La entropía promedio alcanzada en cada dimensión del descriptor SIFT	
	por cada operador identificado en la Figura 43: (d). (e) v (f)	129
46	Frentes Pareto encontrados con el GP-MO utilizando los objetivos de	
	Estabilidad y Contenido de información; el segundo se calcula utilizando	
	el descriptor Hölder. La gráfica contiene los resultados de cuatro ejecu-	
	ciones del algoritmo, cada una con una profundidad máxima diferente.	
	(a) Comparacion con operadores previos, donde se identifica un caso	
	extremo en la detección de puntos, denominado como el operador (z).	
	(b) El frente Pareto de soluciones donde se identifican tres operadores	
	nuevos: (g), (h) e (i). \ldots \ldots \ldots \ldots \ldots \ldots	131
47	(a) Imagen de interés extraída con el operador (z). (b) Puntos de in-	
	terés detectados con el operador (z) sobre la imagen de entrenamiento.	
	El operador (z) se encuentra en uno de los extremos del frente Pareto	100
10	mostrado en la Figura 40a	132
40	(i) del fronte Parete mostrado en la Figura 46b. Segundo renglén:	
	Puntos de interés detectados	133
49	Frentes Pareto encontrados con el GP-MO utilizando los objetivos de	100
10	Dispersión v Contenido de información: el segundo se calcula utilizando el	
	descriptor SIFT. La gráfica contiene los resultados de cuatro ejecuciones	
	del algoritmo, cada una con una profundidad máxima diferente. (a)	
	Comparacion con operadores previos. (b) El frente Pareto de soluciones	
	donde se identifican tres operadores nuevos: (j), (k) y (l)	135
50	Primer renglón: Imagen de interés extraída con los operadores (j), (k) y	
	(l) del frente Pareto mostrado en la Figura 49. Segundo renglón: Puntos	100
F 1	de interés detectados	136
51	La entropia promedio alcanzada en cada dimension del descriptor SIF I	190
52	Frontos Paroto oncontrados con ol CP MO utilizando los obiotivos do	190
02	Dispersión y Contenido de información: el segundo se calcula utilizando	
	el descriptor Hölder. La gráfica contiene los resultados de cuatro ejecu-	
	ciones del algoritmo, cada una con una profundidad máxima diferente.	
	(a) Comparacion con operadores previos. (b) El frente Pareto de solu-	
	ciones, donde se identifican tres operadores nuevos: (m), (n) y (o)	139
53	Primer renglón: Imagen de interés extraída con los operadores (m), (n) y	
	(o) del frente Pareto mostrado en la Figura 52. Segundo renglón: Puntos	
	de interés detectados	140

54	Frente Pareto generado con el GP-MO utilizando los tres criterios de	
	mismo fronto. El contonido de información se colcula utilizando el de	
	scriptor SIFT	149
55	Fronte Pareto generado con el CP-MO utilizando los tres criterios de	142
55	optimización simultáneamente: se presentan cuatro vistas diferentes del	
	mismo fronto. El contonido de información se calcula utilizando al de	
	scriptor Höldor	1/2
56	Un punto de interés se puede convertir en una región de mayor tamaño	140
50	si el cambio de oscala os lo suficientemento grando	1/18
57	Si el cambio de escala es lo suncientemente grande	140
57	se intestra el electo que tiene el suavizado sobre la intagen, fasgos promi-	
	de combie de casele	150
EO	de campio de escara.	190
99	se muestra un caso noneo en el que la misma región obtiene un maximo	
	a la respuesta de un operador en escalas diferentes. Es importante notar	
	como un operador ideal produce un solo maximo bien definido para cada	1 - 1
50		191
59	La misma region se detecta aun cuando se observa a diferentes escalas; el	150
co	tamano de la region detectada es proporcional a la escala característica t_n	.152
60	Respuesta de los operadores propuestos para la detección invariante a	150
01		150
61	La detección invariante a escala con K_{IPGP1*} : Izquierda , la imagen	
	de entrada I; Medio, imagen de interes I^* calculada en cada escala;	
	Derecha , regiones detectadas despues de elegir una escala característica.	150
<u> </u>	Las regiones se dibujan con un circulo de tamano $3 \cdot t_n$	157
62	La imagen de referencia (izquierda) y una imagen transformada (derecha)	
	para cada secuencia de prueba. a) Boat ($N = 5$, rotacion y escala), b)	
	Asterix $(N = 16, \text{ escala}), \mathbf{c})$ Bip $(N = 8, \text{ escala}), \mathbf{d})$ Laptop $(N = 20, \mathbb{C})$	150
60	escala), e) Van Gogh ($N = 16$, escala)	158
63	Regiones de interes detectadas. Cada region se representa con un circulo	150
0.4	de radio $r = 3 \cdot t_n$ pixeles, con t_n la escala característica	159
64	La tasa de repetibilidad calculada para cada secuencia de prueba.	159
65	Diagrama de flujo del esquema propuesto, ilustrado para el problema de	101
0.0	reconocimiento de objetos.	164
66	Un GMM de dos dimensiones compuesto de tres componentes Gaussianos	.167
67	Un problema de clasificación con tres clases, donde dos de ellas, la Clase	
	1 (con tres componentes) y la Clase 3 (con dos componentes), son mul-	
	timodales. En este caso, los parámetros del GMM para cada clase se	
	estimaron con el algoritmo FJ	168
68	Ejemplos de objetos en la base de imagenes COIL-100. El primer renglon	
	muestra varios puntos de vista para un mismo objeto. El segundo renglón	. – -
	muestra cuatro objetos diferentes	176

69	Los primero 40 objetos de la base de datos COIL-100, estos son los objetos utilizados durante la experimentación. Las imágenes utilizadas en	
	los primeros dos experimentados están marcados mientras que todas las	
	imágenes se utilizan en el tercero. Los objetos más prominentes elegidos	
	por el algoritmo se muestran encerrados por un círculo. El obieto 32 es	
	el único con el cual fallo el sistema.	178
70	Gráficas de convergencia, para el logaritmo de la aptitud del mejor indi-	
	viduo de cada generación.	179
71	Regiones de interés detectadas en imágenes de cuatro lugares reales. (1)	
-	El primer renglón muestra las imágenes originales: (2) El segundo renglón	
	muestra las regiones detectadas en cada imagen.	182
72	El problema de reconocer lugares en el mundo real. (1) El primer renglón	
	muestra varias vistas de un mismo lugar (Laboratorio EvoVisión); (2) El	
	segundo renglón muestra cuatro lugares diferentes (Laboratorio de Estu-	
	diantes, Laboratorio de Computación, Laboratorio EvoVisión, y Oficina).	183
73	Ejemplos de imágenes representativas para cada uno de los lugares uti-	
	lizados en el experimento uno. Por columna: (1) Cuarto, (2) WC, (3)	
	Comedor, y (4) Sala. \ldots	187
74	Graficas de convergencia para cada experimento; se muestran los resul-	
	tados para cada etapa del GA	189
75	Ejemplos de imágenes de los lugares utilizados en el segundo experi-	
	mento: (1) Laboratorio de Estudiantes, (2) Laboratorio de Computación,	
	(3) Laboratorio de EvoVisión, (4) Armarios, (5) Primer Piso, (6) Segundo	
	Piso, (7) Oficina, y (8) Correo. \ldots	190
76	La envoltura de Hölder para la señal unidimensional f alrededor del	
	punto x_0	196
77	El exponente Hölder calculado para una imagen de prueba	197
78	Estimación del exponente Hölder con un análisis de oscilaciones. Izquierda	1:
	la región de interés λ , y tres de los siete vecindarios alrededor del punto	
	t, cuando $r = 1, 2, \dots, 7$. Centro: el vecindario de radio $\tau_5 = 32$ pixeles,	
	con $base = 2$. Derecha: cálculo del supremo de las diferencias dentro	100
-	del radio τ_5 , done <i>d</i> representa la distancia Euclidiana	199
79	El proceso para construir el descriptor Holder. Primero, un detector ex-	
	trae un conjunto A de regiones de interes. Despues, $\forall \mathbf{A} \in A$ se calcula un	
	descriptor $D_{\mathbf{A}}$. El descriptor contiene el exponente Holder puntual para	
	el puto central de la region (x_A, y_A) , y para 52 puntos en el perimetro de sustre síngulas consistences codo una con un radio de ¹ con ¹	
	de cuatro circulos concentricos, cada uno con un radio de $\frac{1}{4} \cdot s_{\mathbf{A}} = \frac{1}{2} \cdot s_{\mathbf{A}}$,	<u> </u>
80	$\frac{1}{4}$, $S_{\mathbf{A}}$ y $S_{\mathbf{A}}$ respectivalmente	202
81	Resultados experimentales para la secuencia New Tork	207
82	Resultados experimentales para la secuencia Mars	200
83	Resultados experimentales para la secuencia Monet	205
84	Resultados experimentales para la secuencia Graph.	211

85	Resultados experimentales para la secuencia Mosaic	212
86	Resultados experimentales para la secuencia UBC	213
87	Resultados experimentales para la secuencia Laptop.	214
88	Resultados experimentales para la secuencia Bip.	215
89	Errores con cambios de escala muy grandes.	216
90	Pares de imágenes de entrenamiento.	217
91	Las gráficas de convergencia de tres ejecuciones típicas.	218
92	Posiciones muestreadas por el mejor individuo de cada experimento, y el	
	número de puntos que contienen.	219
93	Comparación del descriptor H-GA con el descriptor Hölder para la se-	
	cuencia de New York	220
94	Comparación del descriptor H-GA con el descriptor Hölder para la se-	
	cuencia de Van Gogh	221
95	Comparación del descriptor H-GA con el descriptor Hölder para la se-	
	cuencia de Monet	222
96	Comparación del descriptor H - GA para la secuencia de Graph	223
97	Comparación del descriptor H - GA para la secuencia de Mosaic	224
98	Estadísticas del proceso evolutivo que generó a los operadores H - GP . Las	
	columnas: (1) aptitud del mejor individuo; (2) diversidad de la población;	
	y (3) complejidad (profundidad máxima dinámica, la profundidad del	
	mejor individuo, y el número de nodos del mejor individuo)	229
99	Imágenes utilizadas para probar la estimación del exponente Hölder re-	
	alizado por los operadores H - GP	230
100	Comparación cualitativa entre la estimación del exponente Hölder por	
	oscilaciones con la estimación de los operadores evolucionados para la	
101	imagen New York.	231
101	Comparación cualitativa entre la estimación del exponente Holder por	
	oscilaciones con la estimación de los operadores evolucionados para la	000
100	imagen Van Gogn.	232
102	Comparación cualitativa para la imagen Casita.	233
103	Comparación cualitativa para la imagen Puerta.	235
104	Operadores <i>H-GP</i> aplicados a la secuencia New York	230
100 106	Operadores H - GP aplicados a la secuencia Van Gogn	237
$100 \\ 107$	Operadores $H CP$ aplicados a la secuencia Mars	200 220
107	Operadores $H \subseteq P$ aplicados a la secuencia Monet	239
100	Operadores H_{-GP} aplicados a la secuencia Orapii	239 240
110	Operadores H_{-GP} aplicados a la secuencia UBC	240
111	Operadores H - GP aplicados a la secuencia Lapton	241
112	Operadores H - GP aplicados a la secuencia Bip	$\frac{2}{241}$
113	Arquitectura Servidor/Cliente.	266
114	El wrapper v el starter.	268
115	Clientes con BOINC y VMware trabajando con el Servidor BOINC	270
115	<i>Clientes</i> con BOINC y VMware trabajando con el <i>Servidor</i> BOINC	270

Lista de Tablas

Tabla

Ι	Tipo de transformaciones que se le pueden aplicar a una imagen;	
	donde I y x representan a la imagen original y un punto en esta	
	imagen respectivamente, mientras que \mathbf{I} y \mathbf{x} son las versiones trans-	20
	formadas.	38
	Areas de aplicación para los descriptores locales discriminantes.	59
	Descriptores derivados del SIFT.	69
IV	Parámetros del GP utilizado en la síntesis de operadores que detectan	
	puntos de interés.	80
V	Operadores sintetizados con GP.	86
VI	Criterios para determinar si una solución diseñada automáticamente	
	por un proceso de aprendizaje de máquina es competitiva con solu-	
	ciones propuestas por humanos	99
VII	Criterios de competitividad con humanos con los que cumplen las	
	soluciones generadas por el GP	100
VIII	Configuraciones experimentales en el ambiente BOINC/VMware.	103
IX	Tiempo de ejcución sequencial y BOINC/VMware/GP	105
Х	Parámetros básicos del algoritmo GP-MO	112
XI	Expresión simbólica de los operadores (a), (b) y (c) identificados en	
	el frente de la Figura 36 para los objetivos de Estabilidad y Dispersión	.117
XII	Expresión simbólica de los operadores (d), (e) y (f) identificados en el	
	frente de la Figura 43 para los objetivos de Estabilidad y Contenido	
	de información.	129
XIII	Expresión simbólica de los operadores (g), (h) y (i) identificados en el	
	frente de la Figura 46 para los objetivos de Estabilidad y Contenido	
	de información.	134
XIV	Expresión simbólica de los operadores (j), (k) y (l) identificados sobre	
	el frente de la Figura 49 para los objetivos de Estabilidad y Contenido	
	de información.	137
XV	Expresión simbólica de los operadores (m), (n) y (o) identificados en	
	el frente de la Figura 52 para los objetivos de Estabilidad y Contenido	
	de información.	140
XVI	Los canales de información que se utilizan para construir el espacio	
	de búsqueda Φ . Para cada canal de información se incluve una breve	
	descripción, y entre paréntesis se presenta un símbolo que correspon-	
	diente para cada cada uno	165
	1	

XVII	Estadísticas calculadas sobre cada canal de información de la Tabla	
	XVI. Son seis rasgos estadísticos calculados para 18 canales de infor-	
	mación, esto produce un espacio de búsqueda Φ con 108 dimensiones.	166
XVIII	Parámetros del GA de búsqueda y aprendizaje.	170
XIX	Resumen de los tres experimentos realizados para el reconocimiento	
	de objetos.	176
XX	Rendimiento del algoritmo despues de la fase de entrenamiento con	
	los objetos en N	177
XXI	Rendimiento del sistema en la detección de objetos nuevos	178
XXII	El GA de dos etapas	185
XXIII	Descripción del Experimento 1	188
XXIV	Rendimiento en cada experimento y características del descriptor	
	que se obtiene. Se muestra el valor de aptitud, el tamaño de $\Phi^*,$ la	
	cardinalidad de F , los rasgos estadísticos que fueron elegidos, y la	
	exactitud durante validación.	188
XXV	Descripción del Experimento 2	189
XXVI	Matriz de confusión para el Experimento 2	191
XXVII	Secuencias utilizadas en las pruebas experimentales para el descriptor	
	local construido con el exponente Hölder puntual	205
XXVIII	Parámetros del GA.	214
XXIX	Parámetros del GP para la evolución de operadores que estiman el	
	exponente Hölder puntual	227
XXX	Operadores sintetizados con GP para estimar el exponente Hölder	
	puntual	228
XXXI	Comparación cuantitativa entre la estimación del exponente Hölder	
	por oscilaciones, con la estimación de los operadores evolucionados;	
	oscuras indican el mejor	234

Capítulo I Introducción

La visión por computadora intenta dotar a un sistema artificial con la capacidad de construir una descripción explícita y significativa del mundo real a partir de imágenes. La definición dada, aunque bastante general, engloba el reto fundamental que cualquier investigador del área debe de enfrentar: dotar a un sistema artificial con la capacidad de percepción visual.

El sentido de la visión es fundamental para la mayoría de los seres vivos que exhiben capacidades cognitivas muy desarrolladas; por ejemplo, esta afirmación es evidente en el ser humano. Por lo tanto, el estudio de la visión ha atraído la atención de innumerables filósofos y científicos, esta lista incluye a Aristóteles, Platón, Al-Kindi, Euclides, Ptolomeo, Galen, Alhazen, Da Vinci, Bacon, Kepler, Descartes, y Berkley, por nombrar solo algunos ejemplos históricos (Lindberg, 1976; Wolf-Devine, 1993; Schwartz, 2006). Los métodos propuestos para el estudio de la visión han incluido enfoques espirituales, filosóficos, epistemológicos, físicos, anatómicos, matemáticos, mecánicos, psicológicos y cognitivos. En la última parte del siglo pasado, se desarrolló el enfoque computacional propuesto por Marr (1982), entre otros, donde el análisis de la visión se centra en tres aspectos fundamentales: (1) definir cuales son los problemas en la percepción visual; (2) proponer soluciones a estos probleas; y (3) entender como es que estos procesos se ejecutan en el sustrato subvacente (biológico/neuronal o artificial/silicio). En otras palabras, el enfoque computacional define a la visión como una serie de tareas o problemas que se deben resolver, propone soluciones algorítmicas a estos problemas, e intentan explicar como es que estos procesos se llevan acabo en sistemas biológicos reales.

Sin embargo, a lo largo de décadas de investigación, se ha reconocido que reproducir las habilidades de percepción visual que poseen los seres vivos no es una tarea trivial. Es más, este problema, como parte de la inteligencia artificial, representa uno de los grandes retos para las ciencias computacionales y cognitivas. En cierto sentido, la dificultad de este reto es producto de nuestras preconcepciones e intuiciones innatas que como seres humanos tenemos con respecto al proceso de ver. Esta idea fue expresada de manera concisa por Wittgenstein (1953) cuando dice que, *para nosotros* [los seres humanos], *ciertos aspectos de la "acción de ver" son confusos, porque todo lo relacionado con la "acción de ver" no se nos hace lo suficientemente confuso*¹.

Por lo tanto, aunque investigadores en el área han intentado diferentes enfoques, modelados y técnicas computacionales para construir sistemas capaces de reproducir la "acción de ver", que incluye tareas como reconocer objetos, escenas y personas, todos estos problemas siguen estando abiertos y representan líneas de investigación muy activas.

Por otro lado, aún considerando la amplia diversidad de propuestas que hasta hoy se tienen, es posible enumerar cuatro preguntas fundamentales que deberán considerarse cuando se diseña un sistema de visión artificial (Faugeras, 1993):

- 1. ¿Que información debe ser extraída a partir de la salida que proporcionan los sensores visuales?
- 2. ¿Cómo se extrae esta información?
- 3. ¿Cómo se se representa la información una vez que se extrae?
- 4. ¿Cómo se utilizará esta información para resolver tareas de un nivel más alto?

¹Esta cita fue tomada de: N. R. Hanson, (1969) *Perception and Discovery: An Introduction to Scientific Inquiry.* Editado por W.C. Humphreys, Freeman, Cooper & Company, p. 63.

Dar respuestas generales para estas preguntas aún no es posible, y hacer una revisión general de todas las propuestas que se han dado queda fuera del alcance de esta tesis. Sin embargo, en este documento se presenta un recuento de aquellos enfoques que tienen la mayor relevancia con el trabajo de investigación que aquí se expone.

I.1 Enfoques modernos a la visión por computadora

Históricamente, la mayoría de los sistemas de visión incluían un proceso de segmentación en una etapa de bajo o mediano-nivel (Marr, 1982; Faugeras, 1993; Ylä-Jääski y Ade, 1996). La segmentación es un proceso mediante el cual se intenta agrupar píxeles, u otros rasgos de bajo-nivel, de una imagen para formar regiones conectadas y homogéneas, utilizando una medida de similitud y un análisis global de la imagen. En base a este proceso, se espera que cada región cubra objetos reales o conceptos claros; por ejemplo, un carro, una casa, o el cielo. De esta forma, el análisis de alto-nivel se puediéra simplificar, debido a que cada imagen estaría divida en regiones semánticamente coherentes, espacialmente compactas y visualmente informativas.

Sin embargo, segmentar una imagen representa un problema mal-planteado para el caso general (Kumar *et al.*, 2005; Unnikrishnan *et al.*, 2007), razón por la cual esta tarea ha resultado ser muy difícil y compleja. Por lo tanto, desde la década de los 90s muchos de los sistemas de visión se han diseñado utilizando enfoques que no requieran de una previa segmentación de la imagen para resolver tareas como: la detección/reconocimiento de objetos, recuperación de imágenes basada en contenido, indexado de imágenes, entre otras.

Por ejemplo, los métodos holísticos se basan en análisis que englobe a toda la información contenida dentro de una imagen; esto lo logran, por ejemplo, utilizando estadísticas generales de la imagen, o su respuesta a cierto tipo de operadores y filtros (Oliva y Torralba, 2001; Lazebnik *et al.*, 2006; Mutch y Lowe, 2006; Trujillo *et al.*, 2005; Olague *et al.*, 2007; Hernández *et al.*, 2007).

Por otro lado, el enfoque que se estudia en esta tesis se basa en un análisis local de la imagen, utilizando cantidades relativamente pequeñas de información. Este enfoque lo introdujo Schmid y Mohr (1997), y posteriormente se extendió con el trabajo de Lowe (1999), entre otros. Estos trabajos han comprobado que cuando se utilizan regiones locales para analizar e interpretar imágenes, es más fácil construir sistemas robustos con respecto a oclusiones o distorsiones parciales de la información visual.

En la literatura existe una amplia variedad de propuestas que siguen este enfoque, cada una con sus particularidades. Ejemplos recientes incluyen a los métodos de constelaciones de puntos (Fergus *et al.*, 2003; Fei-Fei *et al.*, 2004; Bouchard y Triggs, 2005) y *bolsas* de puntos de interés (Sivic y Zisserman, 2003; Willamowski *et al.*, 2004; Sivic *et al.*, 2005; Perronnin, 2008). Sin embargo, es posible identificar ciertas características que todos suelen exhibir, ver Figura 1.

La metodología introducida por Schmid y Mohr (1997) se caracteriza por tener dos etapas que son comunes para muchas aplicaciones de aprendizaje máquina: entrenamiento y prueba.

Durante el entrenamiento, primero se extraen regiones prominentes o *interesantes* utilizando detectores estables y robustos (Harris y Stephens, 1988; Lowe, 1999; Schmid *et al.*, 2000; Tuytelaars y Mikolajczyk, 2008), ver Figura 2. Después, se calcula un descriptor para cada una de las regiones, estos descriptores sirven para representar la estructura local de la señal en cada región de interés (Lowe, 1999; Ke y Sukthankar, 2004; Mikolajczyk y Schmid, 2005). Finalmente, las regiones detectadas y sus descriptores correspondientes se utilizan para construir modelos generativos (Fergus *et al.*, 2003; Fei-Fei *et al.*, 2004) o discriminantes (Schmid y Mohr, 1997; Lowe, 1999) para la escena u objeto observado en la imagen.



Figura 1: Un modelo simplicado de sistemas de visión basados en el análisis de regiones locales.



Figura 2: Regiones de interés detectadas en tres imágenes de la base de datos COIL-100.

Posteriormente, durante la etapa de prueba el mismo proceso se repite, con la diferencia de que ahora se desea buscar una correspondencia entre la información extraída de la imagen de entrada y los modelos previamente guardados. Por lo tanto, cuando los criterios de correspondencia se cumplen, es posible reconocer objetos o escenas conocidas. Resumiendo, este enfoque responde a las cuatro preguntas de Faugeras (1993) de la siguiente manera:

- ¿Que información debe ser extraída a partir de la salida que proporcionan los sensores visuales? Información local de la imagen.
- ¿Cómo se extrae esta información? Con algoritmos llamados detectores de regiones de interés; en su mayoría, estos estan basados en operadores locales que extraen una medida de cuan interesante es cada porción de la imagen.
- ¿Cómo se se representa la información una vez que esta se extrae? Utilizando descriptores locales que caracterizan la información contenida en cada región.
- ¿Cómo se utilizará esta información para resolver tareas de más alto nivel? Construyendo modelos representativos que permiten identificar escenas u objetos previamente vistos.

Dada la explicación previa, es evidente que la metodología descrita depende fuertemente de los procesos utilizados para la detección y descripción de rasgos locales. Este tipo de rasgos son comúnmente llamados regiones de interés dado a que se asume que estos contienen información que pudiera ser de *interés* para un procesamiento posterior. Ahora bien, existe una gran cantidad de técnicas para la detección y descripción de rasgos locales (Schmid *et al.*, 2000; Mikolajczyk y Schmid, 2005; Mikolajczyk *et al.*, 2005; Moreels y Perona, 2007; Tuytelaars y Mikolajczyk, 2008). Esto deja al diseñador de una aplicación de alto-nivel con la tarea de elegir la mejor combinación para la *detección-descripción* que requiere su sistema (Schmid *et al.*, 2000; Moreels y Perona, 2007), o tal vez llegue a la conclusión que lo mejor será desarrollar una metodología propia (Lepetit y Fua, 2006; Rebai *et al.*, 2007; Li *et al.*, 2008).

Por otro lado, una característica común de cada uno de los métodos, utilizados durante el proceso de *detección-descripción* de regiones de interés, es la manera en

que estos fueron diseñadas. Esencialmente, cada uno de los algoritmos, operadores o metodologías son el producto de un análisis detallado llevado acabo por un experto humano.

Sin embargo, otro planteamiento es posible, uno donde el diseño es automatizado, impulsado por la síntesis de soluciones por medio de un proceso estocástico, y guiado por criterios de evaluación apropiados. Esta perspectiva para la solución de problemas computacionales representa lo esencial del paradigma conocido como cómputo evolutivo (Holland, 1975; Koza, 1992; Schwefel, 1993; Langdon y Poli, 2002). Una de las ventajas del enfoque evolutivo, posiblemente la más importante, es que permite una búsqueda amplia dentro del espacio de posibles soluciones, promoviendo la generación de soluciones aptas que posiblemente sean poco ortodoxas o difíciles de imaginar para un experto humano. De tal forma que se encuentren soluciones nuevas y de un rendimiento mayor a lo obtenido previamente (Poli *et al.*, 2008). Además, puede ayudar a establecer una marco algorítmico, y conceptual, que permita generar soluciones de manera automática, y así fomente un estudio más detallado de las propiedades del problema mismo (Olague y Mohr, 2002).

Estas consideraciones, además de otras más especificas que son presentadas en los capítulos y las secciones adecuadas, establecen las bases para las metas y objetivos generales de esta tesis.

I.2 Metas, motivación y objetivos

La meta principal de esta tesis es proponer un enfoque basado en el paradigma del cómputo evolutivo que nos permitan estudiar y sintetizar operadores y técnicas computacionales que faciliten la detección y descripción de regiones interesantes en imágenes digitales. Por ende, se realiza un planteamiento novedoso para el problema de *detección-descripción* de regiones locales, utilizando procesos de búsqueda y optimización evolutiva en base a criterios de evaluación bien establecidos y ampliamente aceptados por la comunidad de visión artificial (Schmid *et al.*, 2000; Mikolajczyk y Schmid, 2005; Moreels y Perona, 2007).

I.2.1 Motivación

En forma de resumen, esta tesis aborda problemás de mucha relevancia actual para la comunidad de visión, y realiza un cambio de paradigma con respecto a las estrategias que se utilizan para dar solución a estos problemas. La problemática se reduce a la detección y descripción de puntos interesantes en imágenes, y el paradigma elegido para generar soluciones es el cómputo evolutivo. La motivación del primero es más evidente, si consideramos la atención que estos problemas reciben a nivel internacional en conjunto con el rendimiento de los sistemas de visión que este enfoque ha producido. Con respecto al cambio de paradigma, es más una cuestión conceptual relacionada con los principios y objetivos básicos de la inteligencia artificial. Si consideramos que la visión es una de las habilidades más prominentes en los seres cognitivos, entonces podemos apreciar el interés que hay en proponer soluciones a estos problemas utilizando un enfoque cuyo objetivo primordial es reproducir ciertas características de la inteligencia humana a través de procesos computacionales. Más aun, la evolución artificial agrega un aspecto adicional, no solo se busca construir sistemas artificiales con capacidades biológicas, como la visión, sino que el mecanismo que se utiliza para sintetizar estos sistemas, ahora de manera automática, es a su vez inspirado en el proceso natural, la evolución, que ha jugado un papel central en la generación, una y otra vez, de todas las capacidades visuales que ahora queremos reproducir en una computadora. Por otro lado, los métodos evolutivos han demostrado una habilidad incuestionable para generar

soluciones nuevas, diferentes y poco ortodoxas para problemas muy complejos, que compiten con el estado-del-arte de los métodos clásicos en muchas áreas de la ciencia y la tecnología. Por estas características, la evolución artificial puede ofrecer, en algunas circunstancias, una perspectiva diferente que completa y/o complementa el trabajo que históricamente se ha realizado en el área de visión.

Dada la motivación de esta investigación, así como la meta principal que se propone, esta tesis se desenvuelve en torno a los siguientes objetivos específicos.

I.2.2 Objetivos específicos

- Diseñar un sistema para la síntesis automática de operadores capaces de detectar puntos de interés. Las soluciones obtenidas deben ser competitivas con lo disponible en el estado-del-arte.
- 2. Hacer un estudio multiobjetivo del problema de detección de puntos de interés, un enfoque que no ha sido considerado por trabajos previos en el área.
- 3. Construir un detector invariante a escala en base a operadores sintetizados por un proceso evolutivo.
- Proponer un descriptor local discriminante, apoyándose en un proceso de diseño evolutivo. Dicho descriptor deberá capturar las características particulares de cada región en una imagen.
- 5. Proponer un descriptor incluyente, capaz de capturar las características generales de una clase y a la vez diferenciar entre clases diferentes. Dicho descriptor será de utilidad para problemas en donde se requiere hacer un modelo generativo de los datos.

I.3 Soluciones propuestas

Para lograr estos objetivos se proponen las siguientes soluciones.

- 1. Se utiliza programación genética para la síntesis de operadores capaces de detectar puntos de interés. El espacio de búsqueda contiene primitivas que han sido ampliamente utilizadas por operadores propuestos en literatura previa, y algunas otras que se consideran apropiadas para el problema. Además, el proceso evolutivo optimiza en base a criterios que promueven la estabilidad del detector y la dispersión de los puntos detectados. Finalmente, debido al costo computacional que involucra la búsqueda genética, se emplean técnicas de cómputo distribuido para facilitar y optimizar el proceso (Lombraña *et al.*, 2007).
- 2. Se consideran tres criterios generales para la evaluación de detectores de puntos de interés, que miden: 1) la estabilidad del detector; 2) la dispersión de los puntos detectados; y 3) el contenido de información presente en el conjunto de puntos extraídos. Posteriormente, se utiliza un enfoque multiobjetivo de programación genética, donde los operadores se evaluan en base al criterio de optimalidad Pareto. Este enfoque nos permite estudiar la relación que existe entre cada uno de los criterios de evaluación de manera directa. Es importante notar que el enfoque multiobjetivo no puede aplicarse de forma trivial utilizando técnicas clásicas; sin embargo, los métodos evolutivos son más flexibles y ayudan a simplificar este estudio.
- 3. En base a los operadores sintetizados con programación genética para la detección de puntos de interés, se construyen detectores invariantes a escala tomando un operador y embebiéndolo en el espacio escala lineal. Además, se propone una implementación algorítmica que es más sencilla que la utilizada por los detectores del estado-del-arte.

- 4. Se propone un nuevo descriptor en base a un análisis de regularidad local expresado por el exponente Hölder. Debido a que una buena estimación de la regularidad local en una señal es computacionalmente costosa, se construyen operadores con programación genética que sean capaces de aproximar el cálculo del exponente Hölder y que al mismo tiempo sean más eficientes.
- 5. Para construir un nuevo descriptor incluyente, su utiliza un proceso de selección de rasgos a través de un algoritmo genético. Se define un espacio de búsqueda con seis descriptores estadísticos obtenidos de 18 canales de información, un total de 108 dimensiones. Posteriormente, un algoritmo genético selecciona los descriptores que mejor caractericen a las clases utilizadas durante entrenamiento, por lo que la metodología propuesta construye soluciones que dependen de la instancia del problema que se esta abordando. La función objetivo promueve una exactitud alta durante el proceso de reconocimiento, favorece la construcción de descriptores compactos, y una separación alta entre los modelos de clases diferentes.

I.3.1 Contribuciones

A partir de las propuestas y la experimentación realizada en esta tesis es posible identificar las siguientes contribuciones científicas:

- Este trabajo describe una metodología para sintetizar operadores que son de uso general; i.e., pueden utilizarse con diferentes tipos de sistemas de visión que dependen de la detección robusta y dispersa de puntos o regiones de interés (Trujillo y Olague, 2006a,b, 2008). Esto se puede asegurar debido a que el proceso de optimización promueve propiedades bien establecidas y ampliamente aceptadas como deseables en la detección de dichos rasgos.
- 2. Un total de 20 operadores de interés nuevos son presentados, todos competitivos

con el estado-del-arte. De esta forma, queda experimentalmente comprobado la validez del enfoque evolutivo propuesto en esta investigación.

- 3. La síntesis evolutiva de operadores permitió estudiar, de manera experimental, la estructura del espacio de búsqueda empleado (Trujillo y Olague, 2007, 2008). Se identificaron tendencias de convergencia y se analizaron el tipo de operadores que la evolución promueve. Estos resultados proveen un entendimiento más profundo del problema de detección de puntos de interés.
- 4. El estudio multiobjetivo de la detección de puntos de interés es único, y representa una de las pocas contribuciones que utilizan este paradigma en el área de visión (Trujillo *et al.*, 2008b). Los resultados experimentales sugieren que el problema es intrínsecamente de carácter multiobjetivo, y por lo tanto se requieren de estrategias adecuadas para evaluar detectores y elegirlos para su uso en problemas específicos.
- 5. Se proponen dos detectores invariantes a escala basados en operadores evolucionados. Los resultados experimentales sugieren que la metodoloía propuesta obtiene un rendimiento alto cuando se compara con los métodos del estado-del-arte (Trujillo y Olague, 2007).
- 6. A partir del trabajo realizado para desarrollar un nuevo descriptor local se desprenden dos contribuciones significativas. Primero, se presenta un descriptor que es altamente competitivo y algorítmicamente sencillo de calcular (Trujillo *et al.*, 2007b). Además, este descriptor rompe con la tendencia prevaleciente en la literatura actual de diseñar descriptores locales que comparten la misma estructura básica. Segundo, la síntesis de operadores para estimar el exponente Hölder a través de GP, proporciona un enfoque que puede aumentar el campo de aplicación para el análisis de señales basada en la regularidad local.

7. Finalmente, se establecen las bases para el diseño de un sistema de visión capaz de reconocer objetos o lugares; esto se logra utilizando un proceso evolutivo para la selección de características locales (Trujillo *et al.*, 2007a, 2008a).

I.4 Organizacin del documento

El resto de este documento esta organizado de la siguiente forma. En el Capítulo II se describen las bases conceptuales del cómputo evolutivo, haciendo énfasis en los paradigmas de programación genética y algoritmos genéticos. También, se presentan las metodologías multiobjetivo desarrolladas bajo la perspectiva de la evolución artificial. Finalmente, se hace un recuento general de como el cómputo evolutivo ha sido empleado para resolver problemas de visión por computadora (Cagnoni *et al.*, 2007).

El problema de detección de puntos de interés se presenta en el Capítulo III. Se describe el estado-del-arte y se definen los criterios de evaluación. Después, en el Capítulo IV se introduce el concepto de un descriptor local, se describen los diferentes enfoques que existen sobre el tema, se discuten las áreas de aplicación que tienen, y se presenta una revisión del estado-del-arte.

En el Capítulo V se detalla la implementación de programación genética propuesta para la síntesis automática de operadores utilizados durante el proceso de detección de puntos. Se incluyen secciones en donde se presentan nuevos detectores, se analizan las tendencias del algoritmo propuesto, y se realizan evaluaciones cualitativas y cuantitativas. Finalmente, para realizar un estudio estadístico se presenta una metodología que permite una ejecución distribuida y paralela de muchas instancias del algoritmo propuesto.

En el Capítulo VI se presenta un análisis multiobjetivo del problema de detección de puntos de interés. Con el motivo de estudiar la relación que existe entre cada uno de los criterios de evaluación, se describe un algoritmo de programación genética multiobjetivo. Se presentan los resultados experimentales y se analizan las implicaciones que tienen sobre el tema de detección de puntos. El enfoque propuesto no es común en el área de visión, por ende se proponen varias lineas de investigación a futuro.

En el Capítulo VII se describe el problema de detección de rasgos locales invariante a escala. Se incluye una discusión sobre los conceptos de espacio escala y escala característica, y se describe como su aplican estos conceptos al problema de detección. Además, se presentan dos detectores nuevos, que se basan en operadores diseñados a través de una búsqueda evolutiva. El rendimiento de los detectores se compara con el estado-del-arte, y los resultados sugieren que la implementación propuesta es altamente competitiva.

En el Capítulo VIII se propone un enfoque para describir el contenido de una imagen en base a datos estadísticos, untilizando modelos generativos. Se define el espacio de búsqueda empleado y las herramientas pertinentes, con atención especial a los modelos de mezclas de Gaussianas que se utilizan para caracterizar la descripción de los rasgos visuales y como herramientas de clasificación. La estrategia propuesta, basada en una búsqueda genética, se aplica a dos problemas: 1) reconocimiento de objetos; y 2) reconocimiento de lugares en el mundo real. El capítulo incluye una descripción amplia de los experimentos realizados, el rendimiento de las soluciones obtenidas y las limitaciones del enfoque propuesto.

El Capítulo IX presenta un descriptor local basado en análisis de regularidad. Se presentan definiciones para los conceptos de regularidad y el exponente Hölder, y se describe el método para estimar dicho exponente utilizando oscilaciones locales de la señal 2D. También, se presenta una metodología nueva que permite construir un descriptor local discriminante en base a la regularidad de la imagen caracterizada con el exponente de Hölder. Dicho descriptor se compara experimentalmente con SIFT
para analizar el rendimiento de la propuesta. Posteriormente, esta misma estrategia se optimiza a través de un algoritmo genético, buscando minimizar las dimensiones del descriptor. Finalmente, se describe como la programación genética se puede utilizar para construir operadores que estimen al exponente de Hölder para una señal 2D de una forma más eficiente que la técnica basada en oscilaciones. Este nuevo operador también se utiliza para construir un descriptor local, y experimentalmente se compara con el descriptor original.

Finalmente, en el Capítulo X se realiza un breve resumen, se exponen conclusiones, se discuten las limitaciones de las propuestas hechas, y se proponen posibles lineas de investigación para el futuro.

Capítulo II

Antecedentes teóricos y trabajo previo

Este Capítulo presenta una descripción general de las bases conceptuales del cómputo evolutivo. Se describe como se utiliza para diseñar soluciones a problemas computacionales de manera automática. La discusión se centra en dos de los paradigmas más importantes en el área: la programaciónón genética y los algoritmos genéticos. También, se presentan las técnicas de cómputo evolutivo que consideran, de manera explícita, el análisis multiobjetivo de problemas complejos. Finalmente, se introducen algunos de los trabajos más relevantes que emplean el paradigma de la evolución artificial para resolver problemas de visión por computadora (Olague *et al.*, 2006a).

II.1 La evolución como un paradigma computacional

Charles Darwin contemplaba a todas las especies de seres vivos que actualmente habitan al planeta como descendientes directos de las criaturas que vivieron mucho antes de que se depositara la primera capa del sistema Cámbrico, y no como creaciones aisladas o independientes. Para Darwin, solo desde este punto de vista la vida obtiene un carácter único y ennoblecedor (Darwin, 1872). Esta perspectiva le permitió a Darwin apreciar, entender y exponer, el poder de búsqueda, adaptación y diseño que exhiben los sistemas biológicos. Más aun, sus ideas han impactado a todas las ramas de la filosofía y la ciencia, incluyendo las áreas de diseño y desarrollo tecnológico. En particular, las ciencias de la computación se han visto fuertemente influenciadas por metodologías algorítmicas que buscan abstraer y reproducir las capacidades de diseño que exhibe el proceso de evolución natural.

Esta fue la inspiración conceptual que les permitió a Rechenberg (1965), Fogel *et al.* (1966), Holland (1975) y Koza (1992) desarrollar sus propuestas seminales en el área. Sin embargo, es válido recordar que la idea de utilizar un proceso basado en la evolución natural para construir máquinas inteligentes primero fue propuesto por Alan Turing en 1948, tal como lo notan Koza *et al.* (2000). Esta familia de algoritmos conforman las bases de lo que hoy en día se conoce con el nombre de cómputo evolutivo (CE) (Goldberg, 1989; Schwefel, 1993; Langdon y Poli, 2002; Price *et al.*, 2005; Poli *et al.*, 2008), que a su vez forman parte del grupo de métodos computacionales inspirados en procesos biológicos (Dorigo, 1992; Kennedy y Eberhart, 2001; de Castro y Timmis, 2002; Olague *et al.*, 2006b; Olague y Puente, 2006).

Si consideramos al CE como un paradigma general que nos permite resolver problemas complejos de diversas áreas del conocimiento, es posible identificar una característica que lo distingue de los métodos clásicos. La diferencia es una de enfoque, mientras que métodos clásicos enfatizan un análisis profundo del problema, el CE incluye a la síntesis como parte integral del proceso requerido para obtener una posible solución. Por ejemplo, históricamente en visión por computadora los problemas son abordados casi exclusivamente con técnicas analíticas. Sin embargo, el CE ha impulsado una perspectiva diferente en donde la síntesis es explotada de manera explícita por los procesos algorítmicos propuestos, modelando de esta forma la descendencia genética empleada dentro de los sistemas biológicos.

Los métodos analíticos buscan obtener un entendimiento profundo del problema

estudiado, y de esta forma deducir las propiedades intrínsecas del mismo. Por otro lado, las soluciones que se encuentran para problemas del mundo real van a depender del proceso analítico empleado. Por ende, es común que se propongan diferentes soluciones para la misma instancia de un problema. Es por esto que en las ciencias o ingenierías se requieren de medidas de rendimiento efectivas, normalmente basadas en evaluaciones experimentales, que permitan categorizar a las soluciones en base a criterios cuantitativos.

El CE permite a los investigadores hacer un planteamiento diferente de los problemas estudiados, empleando un enfoque de búsqueda y optimización (Goldberg, 1989; Gen y Cheng, 1997). A través del CE un investigador no requiere enfocarse en resolver el problema de manera directa, lo que debe lograr es definir un espacio de soluciones, y una función de evaluación que proporcione la suficiente estructura para poder guiar al proceso de búsqueda. Cuando se definen objetivos de evaluación y un espacio de búsqueda apropiados entonces un algoritmo evolutivo (en inglés, Evolutionary Algorithm: EA) puede sintetizar soluciones utilizando combinaciones poco ortodoxas de elementos básicos. De esta forma, es posible generar soluciones que un humano tal vez no pudiera imaginarse; un diagrama ilustrativo de este proceso se muestra en la Figura 3.

II.1.1 Características de los algoritmos evolutivos

El CE engloba una familia de algoritmos basados en heurísticas estocásticas que modelan y abstraen las propiedades básicas de la evolución biológica. Aún considerando que existen muchas variantes dentro del área, es posible identificar cinco mecanismos comunes que todos estos métodos comparten:

1. Un método de codificación que permite al EA representar un conjunto de soluciones o *población*, donde cada solucion representa un *individuo*.



Figura 3: Una perspectiva de alto-nivel de dos enfoques para resolver problemas: **a**) El método *clásico*, donde un proceso analítico es empleado para descubrir las propiedades intrínsecas de un problema. Dependiendo del proceso de análisis empleado, diferentes soluciones, tal vez en conflicto, pudieran encontrarse. Por lo tanto, se requieren de medidas de comparación cuantitativas para poder elegir una solución dados los requerimientos en algún contexto. **b**) El método del CE, en donde la síntesis y el análisis se explotan de forma equivalente. El problema y las funciones empleadas para evaluar posibles soluciones son analizados de manera conjunta. Las funciones de evaluación se utilizan como guía para el proceso de búsqueda evolutiva. De tal forma que se sintetizan soluciones y se ordenan en base a su rendimiento.

- Una función de evaluación f que cuantifica el rendimiento de cada individuo cuando este se aplica al problema dado; dicha función asigna un valor de *aptitud* para cada individuo de la población.
- 3. Una estrategia pseudo-aleatoria para *seleccionar* el mejor subconjunto de individuos que serán utilizados para generar una nueva población.
- 4. Un mecanismo de variación que produce soluciones nuevas a partir de aquellas que fueron elegidas por el proceso anterior. Esto se puede lograr recombinando individuos, o a través de modificaciones estocásticas conocidas como mutación.



Figura 4: La estructura básica de un EA, con tres módulos primordiales: *evaluación*, *administración de la población*, y *variación*.

5. Una estrategia estocástica de *supervivencia* que decide cuales individuos van a estar presentes durante la siguiente iteración (*generación*) del algoritmo.

La Figura 4 muestra un diagrama conceptual de la forma en que opera un EA; es posible identificar tres módulos principales: 1) *evaluación*; 2) *administración de la población*; y 3) *variación*. En estos módulos se realizan cada uno de los mecanismos enlistados arriba.

Entonces, los EA son procesos iterativos que operan sobre un conjunto de soluciones parametrizadas empleando meta-heurísticas de poblaciones. Dependiendo de como se define la función de aptitud un EA puede contemplar restricciones y decisiones de diseño de manera explícita, evitar mínimos locales, y realizar la búsqueda de una forma intrínsecamente paralela (Holland, 1975). A continuación, se presentan dos de los paradigmas más importantes en el CE: los algoritmos genéticos y la programación genética.

II.1.2 Algoritmos genéticos

Los algoritmos genéticos (en inglés, *Genetic Algorithms*: GA) fueron propuestos por Holland (1975), mientras que el trabajo de Goldberg (1989) ayudó a introducirlos de manera más amplia como herramientas robustas y flexibles para problemas de *búsqueda*, *optimización y aprendizaje de máquina*. Esta familia de algoritmos no solo se basa en el marco conceptual de supervivencia del más apto de Darwin, sino que también incorpora las teorías de herencia de rasgos a través de los genes descritas por Mendel. En los GAs, un individuo se especifica utilizando uno o más cromosomas, en donde cada cromosoma esta compuesto por una serie de genes que determinan sus rasgos o propiedades particulares. Cuando se habla de esta forma de representación, es necesario identificar tres espacios complementarios dentro de los cuales se desarrolla un GA.

- El espacio del genotipo: En este espacio están los cromosomas, y sus genes correspondientes, que codifican las características de un individuo. También, los procesos de variación, u operadores genéticos trabajan en este espacio, como el cruze (recombinación) y la mutación.
- 2. El espacio del **fenotipo**: Este es el espacio del individuo; i.e., la solución una vez que es decodificada a partir de su cromosoma.
- El espacio de aptitud: Este es el espacio de la función de aptitud, donde suelen operar los procesos de selección y supervivencia, un GA busca máximos en este espacio ¹.

En el GA canónico los cromosomas son cadenas de bits de longitud fija. Los GAs emplean dos operadores básicos para generar individuos nuevos utilizando los individuos que existían dentro de la población durante la generación anterior.

 $^{^{1}}$ Si se plantea un problema de minimización entonces en lugar de hablar de una función de aptitud lo correcto seria hablar de una función de costo. Sin embargo, la distinción no es necesaria en la mayoría de los casos.

El cruce o recombinación, toma los cromosomas de dos individuos llamados *padres*, y a partir de estos contruye un individuo nuevo denominado *hijo*. Lo que se espera es que los padres que tienen aptitud alta transmitan los alelos al hijo que le permitan obtener una aptitud aún mayor al de ellos. De esta forma, el GA busca descubrir bloques de alelos que puedan ser optimizadas independientemente y recombinados para construir soluciones cada vez más aptos; estas son las bases de la llamada teoría de Schema o patrones (Holland, 1975; Goldberg, 1989).

La mutación se aplica como un proceso aleatorio que modifica el cromosoma de algunos de los individuos en la población, normalmente un porcentaje muy pequeño. Mientras que la operación de cruce busca promover alelos en la población que han demostrado ser útiles, el propósito de la mutación es introducir información nueva, alelos que posiblemente no estén en ningún individuo de la población. Por ende, la mutación funciona como un mecanismo que ayuda al GA escapar de óptimos locales.

II.1.3 Programación genética

El concepto moderno de la programación genética (en inglés, Genetic Programming: GP) 2 fue desarrollado por Koza (1992), con aportaciones importantes por otros investigadores (Langdon y Poli, 2002; Poli *et al.*, 2008). Como su nombre sugiere, el GP es muy similar a los AGs, la diferencia esta en la manera en que se representan las soluciones. En un GP, los individuos no son cadenas de datos, son arboles, una representación similar a la utilizada por el lenguaje de programación LISP. De tal forma que lo que se optimiza no son valores de ciertos genes (parámetros de una solución), sino que son pequeños programas o funciones computacionales. Por esta razón, se suele referir al GP como una metodología que permite generar programas de manera automática.

 $^{^{2}}$ La abreviación GP se suele referir a dos conceptos: 1) el paradigma de programación genética; y 2) una instancia de un algoritmo que emplea el paradigma.



Figura 5: La estructura básica de un GP.

Debido a la representación que utiliza en forma árbol, las operaciones genéticas son especializadas y bastante más complejas que sus homólogos en AGs, ver (Langdon y Poli, 2002). También, existen otras representaciones menos comunes para un GP, como la representación lineal (Perkis, 1994; Poli, 2000) o una basada en grafos (Poli, 1999; Teller y Veloso, 1996; Miller y Smith, 2006).

El GP canónico, con representación de árbol, construye funciones colocando en los nodos elementos de un conjunto finito de funciones elementales F, mientras que las hojas del árbol contienen variables de entrada tomadas de un conjunto finito de terminales T. Los elementos del conjunto $F \cup T$ también se conocen como las *primitivas* del algoritmo; donde la difinición de estas depende principalmente del dominio del problema. En la Figura 5 se puede ver una representación a manera de bloques del GP básico.

II.2 Evolución con objetivos múltiples

Establecer un criterio para definir la optimalidad es trivial cuando solo existe un objetivo en el proceso de optimización. Sin embargo, existen problemas que deben ser evaluados utilizando objetivos múltiples, y estos objetivos pueden estar en conflicto en ciertas instancias. Cuando se fija una solución para un problema multiobjetivo, es necesario imponer una estructura de preferencias entre los criterios considerados. La integración de preferencias, que determina las propiedades de la solución deseada, se puede hacer previo al proceso de optimización o posterior a el.

Una integración previa implica especificar la manera en la cual interactúan los criterios de evaluación. De tal forma que el proceso de optimización esta dirigida a una sola solución que satisface la estructura explícita que se define entre los criterios. Ejemplos de este enfoque son la agregación aditiva o multiplicativa de objetivos, combinaciones no lineales de objetivos, o el ordenamiento lexicográfico, por nombrar solo algunos.

Ahora bien, el enfoque contrario es la integración posterior de preferencias. En este caso, el proceso de optimización no se ve limitado por supuestos previos en cuanto a la interacción de criterios. Por ende, si los criterios están en conflicto el proceso de búsqueda es dirigida hacia un conjunto de soluciones a partir de las cuales el usuario final podrá elegir en base a una estructura preferida.

Esta sección presenta el tema de la optimización multiobjetivo (MO) con una integración posterior de preferencias, y describe como se utilizan las técnicas de CE para dar solución a problemas de esta índole.

II.2.1 Optimización multiobjetivo

El tratado de problemas de optimización que consideran varios objetivos comenzó con el trabajo seminal de Pareto (1896). Sin embargo, trabajo en el área estuvo limitada



Figura 6: El espacio de decisión, y su correspondiente espacio de objetivos. Una solución \mathbf{x} se mapea a una función vectorial \vec{f} a un vector en el espacio de las funciones objetivo. Los puntos remarcados en la frontera de Λ son elementos del Frente Pareto.

a las áreas de teoría económica e investigación de operaciones. Fueron los trabajos de Kuhn y Tucker (1951) y Hurwicz (1958) los que sentaron las bases matemáticas para el área de optimización MO que se conoce actualmente (Ehrgott y Gandibleux, 2002).

La optimización MO es considerablemente más compleja cuando se compara con problemas mono-objetivos; la diferencia principal radica en cómo se define la optimalidad. Para problemas con un solo objetivo esto es trivial, pero para problemas MO se requiere utilizar los conceptos de dominancia en un espacio multidimensional.

Por otro lado, cuando se estudia un problema MO es necesario tener presente dos espacios diferentes y complementarios: uno para las variables de decisión (espacio del genotipo); y el otro es el espacio de los objetivos (espacio de aptitud multidimensional), ver Figura 6. En el caso de funciones reales, estos dos espacios están relacionados por el mapeo $\vec{f}: \mathcal{R}^n \to \mathcal{R}^k$. El conjunto de restricciones sobre $\vec{f}(\mathbf{x}) = [f_1(x), ..., f_k(x)]$ define una región $\Omega \subset \mathcal{R}^n$ dentro del espacio de decisión, con su homólogo en el espacio de la función objetivo $\Lambda \subset \mathcal{R}^n$, ver Figure 6. Por lo tanto, los óptimos se encuentran en la frontera del espacio de los objetivos, donde esta frontera se conoce como el Frente Pareto. Además, los elementos correspondientes en Ω conforman el conjunto óptimo de Pareto. En términos concretos, las funciones óptimas satisfacen las siguientes relaciones de dominancia.

Definición 1. Relación de dominancia Pareto: Dados k objetivos y un conjunto ordenado $N = \{1, ..., k\}$, un vector de objetivos $\vec{f^u}$ domina a otro vector $\vec{f^v}$ (escrito como $\vec{f^u} \preceq \vec{f^v}$) $\Leftrightarrow \forall i \in N, f_i^u \leq f_i^v \land \exists j \in N \mid f_j^u < f_j^v$.

Definición 2. Optimalidad Pareto: Un vector de solución $\mathbf{x}^* \in \Omega$ es óptimo si $\forall \mathbf{x} \in \Omega$ es verdad que $\forall i \in N, f_i(\mathbf{x}^*) = f_i(\mathbf{x}) \lor \exists i \in N | f_i(\mathbf{x}^*) < f_i(\mathbf{x}).$

Definición 3. Conjunto óptimo Pareto: Para un problema multiobjetivo $\vec{f}(\mathbf{x})$, el conjunto de soluciones óptimas en el sentido Pareto esta dado por $\mathcal{P}^* = \left\{ \mathbf{x} \in \Omega \mid \nexists \ \mathbf{x}' \in \Omega \ tal \ que \ \vec{f}(\mathbf{x}') \preceq \vec{f}(\mathbf{x}) \right\}.$

Definición 4. Frente Pareto: Para un problema multiobjetivo con un vector de objetivos $\vec{f}(\mathbf{x})$ y un Conjunto óptimo Pareto \mathcal{P}^* , el Frente Pareto es dado por $\mathcal{PF}^* = \{\mathbf{u} = (f_1(\mathbf{x}), ..., f_k(\mathbf{x})) | \mathbf{x} \in \mathcal{P}^*\}.$

Lo que dejan observar estas definiciones es que cuando los objetivos de un problema MO están en conflicto entonces no puede existir una solución óptima única; sino que, existen múltiples soluciones y todas son óptimas en el sentido Pareto.

Cuando un problema MO carece de una solución cerrada, entonces es necesario utilizar algoritmos de búsqueda para obtener una aproximación del conjunto de soluciones óptimas Pareto. En general, se espera que este tipo de algoritmos satisfagan las siguientes condiciones:

1. Deberá converger hacia el verdadero Frente Pareto. Este conjunto corresponde al óptimo global en problemas de un solo objetivo; sin embargo, lograr esto es difícil en espacios discontinuos e irregulares. 2. Debe de muestrear de manera representativa al verdadero Frente Pareto. Para lograr esto es necesario tener mecanismos que generan soluciones no solo óptimas, sino que estas también deben ser diversas. Sin embargo, es necesario entender que dependiendo de la estructura del espacio se podrá dar el caso en el que ciertas partes del Frente Pareto no puedan alcanzarse por el proceso de búsqueda.

Entre las metodologías algorítmicas propuestas, aquellas basadas en CE han recibido mucho énfasis y atención en los útimos 15 años; por lo tanto, estos métodos se discuten a continuación.

II.2.2 Algoritmos evolutivos multiobjetivo

Los algoritmos evolutivos multiobjetivo (en inglés, Multiobjective Evolutionary Algortihm: MOEA) mantienen la mayoría de las características de un EA convencional, ver Figura 4. Las diferencias primordiales se encuentran en el módulo encargado de administrar a la población Coello et al. (2002); Zitzler et al. (2004). Primero, la asignación de aptitud debe de considerar la naturaleza MO del problema para buscar que el algoritmo converja al Frente Pareto. Para esto, los MOEAs suelen considerar relaciones de dominancia Pareto. Segundo, como lo ideal sería un muestreo uniforme del Frente Pareto, un MOEA debe promover la diversidad de la población con respecto al espacio de objetivos. Esto se logra utilizando métodos basados en kernel (Deb et al., 2002), métodos de clustering (Zitzler et al., 2002), métodos de histogramas (Corne et al., 2000), o utilizando el concepto de dominancia- ϵ (Deb et al., 2003). Finalmente, los algoritmos del estado-del-arte en MOEAs han reconocido los beneficios de la supervivencia evolutiva basada en elitismo, y esto suele implementarse con mecanismos de archivado que llevan un control fino de todos los individuos que genera el algoritmo (Coello et al., 2002; Zitzler et al., 2004). Los ejemplos más comunes de MOEAs son el Non-Domintated Sorting Genetic Algorithm NSGA-II de Deb et al. (2002), el Pareto Envelope-Based Selection PESA de Corne et al. (2000) y el Improved Strength Pareto Evolutionary Algorithm SPEA2 Zitzler et al. (2002). El campo de aplicación para estas técnicas ha demostrado ser bastante amplio, que incluye desde problemas de visión (Dunn y Olague, 2004) hasta problemas de robótica móvil (Castillo et al., 2006).

II.3 Cómputo evolutivo aplicado a visión

Actualmente, el uso de CE para resolver problemas de visión comienza a tener relevancia para ambas áreas de investigación. Para el área de visión, el CE ofrece la habilidad de explorar espacios enormes para encontrar soluciones que pueden estar lejos del espacio que normalmente se analiza con técnicas tradicionales (Olague *et al.*, 2006a; Cagnoni *et al.*, 2007). Por otro lado, en el área del CE los problemas de visión representan retos computacionales extremadamente difíciles que además son de gran interés en las áreas de inteligencia artificial y aprendizaje de máquina (Faugeras, 1993; Forsyth y Ponce, 2002; Hartley y Zisserman, 2003).

La clave del trabajo realizado en esta línea es definir funciones de evaluación apropiadas para el problema, y acotar el espacio de búsqueda de manera razonable, con el fin de obtener la habilidad visual deseada. A continuación se hace una revisión de la literatura relevante para así establecer el contexto apropiado de las contribuciónes que se exponen en esta tesis.

Un tema que ha recibido mucha atención es la selección de rasgos o características que ayuden resolver tareas de clasificación o reconocimiento (Siedlecki y Sklansky, 1989; Raymer *et al.*, 2000). Por ejemplo, Bala *et al.* (1996) desarrollaron un sistema híbrido, donde un GA se utiliza para seleccionar rasgos y un árbol de decisión funciona como clasificador. Los autores prueban su metodología utilizando imágenes capturadas por satélite e imágenes comunes de rostros humanos.

Yu *et al.* (2002) emplean un GA para elegir el menor número de rasgos necesarios para alcanzar una exactitud de clasificación predefinida. La implementación que proponen se prueba en un problema de clasificación de imágenes multiespectrales de satélite. Junto con la búsqueda genética, los autores emplean un algoritmo de clasificación difuso que mejora los resultados con respecto a la exactitud de la clasificación.

Sun *et al.* (2004) emplean análisis de componentes principales (PCA) para dos clases de objetos, y utilizan un GA para elegir el mejor subconjunto de eigenvectores utilizados durante el reconocimiento. Un resultado interesante de este trabajo es que el GA no solo se enfoca en los eigenvectores con un eigenvalor asociado alto, algo inesperado según el análisis PCA clásico. Posteriormente Zheng *et al.* (2005) extendieron los resultados de (Sun *et al.*, 2004) introduciendo un teorema que afirma que no solamente los eigenvectores con un eigenvalor alto son útiles para la clasificación. En base a estos avances, los autores presentan una GA que incorpora este conocimiento como parte de la función objetivo, aplicado a un problema de reconocimiento de rostros.

Lee (2004) propone un GA híbrido para seleccionar el mejor subconjunto de rasgos utilizados durante la clasificación de datos; sistemas como este que combinan GAs y métodos de optimización local también se conocen como algoritmos meméticos (Moscato, 1989). Las pruebas experimentales que realizan los autores comprueban que su propuesta alcanza un rendimiento mayor que el GA canónico y un algoritmo de búsqueda secuencial.

Más recientemente, Hernández *et al.* (2007) estudian el problema de reconocimiento de expresiones faciales en imágenes térmicas, y proponen un GA para resolver dos tareas de forma acoplada. El cromosoma empleado esta compuesto de dos partes, en una se seleccionan las regiones faciales y en la otra los rasgos de textura para cada región, lo que se conoce como un GA jerárquico. Trabajo por el mismo grupo utilizó esta técnica para resolver el problema de clasificación de imágenes en base a su contenido de textura (Olague *et al.*, 2007); los dos trabajos utilizan máquinas de soporte vectorial (SVM) durante la etapa de clasificación. De manera similar, Gökberk *et al.* (2007) también emplearon un GA para elegir las regiones y los descriptores, en este caso rasgos tipo Gabor, para resolver el problema de reconocimiento de rostros. En otra área de aplicación, Coillie *et al.* (2007) seleccionan el mejor subconjunto de rasgos de textura, color y forma para clasificar imágenes aéreas de zonas forestales, y utilizan una red neuronal para clasificar. Finalmente, Lu *et al.* (2008) seleccionan entre rasgos MPEG-7 para clasificar imágenes de 20 clases diferentes, además de la búsqueda genética utilizan un proceso clasificación sencillo del tipo vecino más cercano.

También existen trabajos en donde el CE se utiliza para construir operadores capaces de detectar objetos sin la necesidad de utilizar un clasificador como un proceso adicional (Krawiec *et al.*, 2007). Uno de los primeros ejemplos es el trabajo de Tackett (1993), donde los autores utilizan GP para construir un clasificador de imágenes en el espectro infrarrojo. Otro ejemplo, es el trabajo de Andre (1994), donde se construyen máscaras empleadas para reconocer dígitos de baja resolución utilizando un GP.

Aplicado a visión por computadora, la primer contribución substancial que se puede mencionar es el sistema PADO (*Parallel Algorithm Discovery and Orchestration*) de Teller y Veloso (1996). Este sistema en muchos sentidos funcionó como una prueba de concepto, demostrando experimentalmente que el CE, en especial el GP, es capaz de construir un sistema completo de reconocimiento visual.

Daniel Howard (Howard *et al.*, 1999, 2006) presenta ejemplos más elaborados de GPs utilizados para detectar objetos en imágenes tipo SAR (en inglés, Synthetic Aperture Radar). El GP se utiliza para construir clasificadores binarios, la evolución se adapta para generar funciones que producen un valor de salida, y dependiendo del signo de dicho valor se decide si *se detecta* o *no se detecta* al objeto buscado (Sherrah *et al.*, 1997). La dificultad del problema es bien conocido, debido a la resolución baja e información difusa que tienen este tipo de imágenes. Sin embargo, los resultados muestran que el GP es capaz de diseñar operadores locales capaces de detectar objetos bajo estas condiciones. Más aún, las funciones y terminales empleadas en esos trabajos permiten interpretar los operadores que el GP construye, algo que es más difícil con técnicas de caja-negra como las redes neuronales.

En esta misma línea, Zhang *et al.* (2003) emplean clasificadores de árbol construidos con GP para la detección de otro tipo de objetos, como son rostros, monedas y figuras sencillas, enfatizando la robustez con respecto a escala y punto de vista. En ese trabajo los autores emplean rangos estáticos en la salida del árbol sintetizado, donde cada rango de valores corresponde a una de las clases. En trabajo posterior, Zhang y Smart (2004) emplean rangos dinámicos asignados durante la evolución. Finalmente, Zhang y Smart (2006) extienden este enfoque utilizando distribuciones Gaussianas para determinar el rango de salida que le corresponde a cada clase. Además, para asignar membresía de clase para una imagen de prueba los autores emplean un comité de clasificadores tomados de la población en lugar de un solo individuo. Este tipo de técnicas coevolutivas (Potter, 1997), o de evolución simbiótica (Watson, 2002; Dunn *et al.*, 2006), permiten al algoritmo explotar, aún más, el aprendizaje a nivel de población que se genera en los procesos evolutivos.

Los ejemplos que se discuten arriba están en dos extremos. El primer grupo de trabajos utilizan al CE solo para seleccionar entre rasgos predefinidos y emplean un sistema de clasificación externo (red neuronal o SVM, por ejemplo) para resolver una instancia de un problema de reconocimiento. En el segundo grupo, se presentaron trabajos, basados en GP, que se emplean para clasificar la información de una imagen directamente. Mientras que el primero enfoque resuelve problemas de una forma altamente competitiva, deja la impresión de que no se explotan las capacidades de diseño que puede ofrecer el CE, ya que solo se emplea como un proceso de selección. En el segundo, la carga de análisis e interpretación queda exclusivamente sobre el proceso evolutivo, y no se aprovecha el conocimiento previo que se tiene sobre el dominio del problema, ni las herramientas bien establecidas que pudieran facilitar el diseño de una solución para el problema que se estudia.

Un punto medio entre estos dos extremos esta en la construccion o síntesis de rasgos descriptivos utilizando CE (Bhanu *et al.*, 2005). En este enfoque, lo que se desea es desarrollar un proceso evolutivo que sea capaz de sintetizar rasgos nuevos que dependan de las características del problema que se esta abordando; uno de los primeros trabajos en esta línea es el de Raymer *et al.* (1996), mientras que Krawiec (2002) estableció un marco conceptual más general; otros ejemplos incluyen a los siguientes.

Lin y Bhanu (2005) emplean GP en un marco de co-evolución cooperativa para construir rasgos aplicados al problema de detección de objetos en imágenes SAR. El enfoque cooperativo se utiliza para construir varios rasgos, no solo uno. Para esto, se utilizan varias poblaciones evolucionando en paralelo, donde cada población se encarga de construir un rasgo descriptivo; los individuos compuestos son la concatenación de cada uno de estos rasgos. El individuo compuesto representa la entrada a un clasificador Bayesiano en la etapa final. Las terminales empleadas en este trabajo son rasgos sencillos relacionados con el dominio del problema. Con dichas terminales, al proceso evolutivo se le proporciona información relevante al problema y se acota el espacio de búsqueda de forma razonable.

Krawiec y Bhanu (2005) utilizan un enfoque similar al anterior con las siguientes diferencias: emplean una representación de GP lineal, y clasifican con dos algoritmos para comparar, el algoritmo C4.5 y SVM. La representación lineal emplea funciones de procesamiento de imágenes básicas, que pueden ser definidas en base al dominio del problema abordado. Adicionalmente, para mejorar la robustez del sistema los autores emplean un comité de clasificadores, básicamente un sistema modular. Los resultados que reportan son mejores que en el trabajo anterior; los autores atribuyen esto a la representación lineal que permite construir operadores más detallados, y al comité de clasificadores que tiende a ser más robusto.

Un trabajo similar fue desarrollado por Tan *et al.* (2005), quienes abordan el problema de clasificación de huellas digitales; utilizan GP lineal para sintetizar rasgos discriminantes con resultados altamente competitivos. Yu y Bhanu (2006) utilizan un GP canónico para sintetizar lo que llaman rasgos compuestos a partir de rasgos extraídos con filtros Gabor; el problema que abordan es el reconocimiento de expresiones faciales. Los autores emplean un sistema multi-agente similar al que utiliza (Krawiec y Bhanu, 2005).

Krawiec y Bhanu (2007) extienden el trabajo en (Krawiec y Bhanu, 2005) modificando el enfoque conceptual de la co-evolución. En lugar de construir los individuos agregados al nivel de rasgos o clasificadores modulares, la cooperación que proponen se da al nivel del genotipo, concatenando los cromosomas de los individuos que se eligen de cada población.

Hasta el momento, una mayoría de los trabajos mencionados se enfocan en problemas de alto-nivel. Esto se puede apreciar en el tipo de aptitud que proponen, donde la mayoría lo relacionan con el desempeño de la tarea de reconocimiento o clasificación que cada trabajo aborda. Sin embargo, existen otros trabajos que se enfocan en obtener rasgos o realizar un preprocesamientos de bajo-nivel (Cagnoni *et al.*, 2007). Por otro lado, todos los trabajos descritos arriba tienen la peculiaridad de que deben ser ejecutados cada vez que se tenga una instancia nueva de un problema, algo que puede ser computacionalmente costoso (Lombraña *et al.*, 2007); característica que no es exclusiva de las aplicaciones de alto-nivel (Yao *et al.*, 2005; Olague y Puente, 2006). También, es importante notar que aunque si existen propuestas para diseñar rasgos especiales a través del CE, estos trabajos se diferencian de una manera substancial a las propuestas hechas en esta tesis (Krawiec, 2002; Bhanu *et al.*, 2005; Krawiec y Bhanu, 2005; Yu y Bhanu, 2006; Krawiec y Bhanu, 2007). En aquellos trabajos, los autores evolucionan rasgos nuevos y especializados, que dependen fuertemente de una tarea de alto nivel, y que además no forman parte, de manera directa, con ningún enfoque general utilizado por la comunidad de visión. En esta tesis, lo que se propone es diseñar detectores y descriptores de rasgos visuales bajo un marco conceptual bien establecido y maduro, de tal forma que las soluciones que se obtengan puedan ser de utilidad en un dominio más amplio de problemas.

Por lo tanto, después de una revisión de la literatura, se afirma que existen pocos trabajos que utilicen al CE para construir operadores de bajo-nivel que puedan ser utilizados por diferentes tipos de sistemas de visión sin que se requiera correr un proceso evolutivo cada vez que se aborda una instancia nueva de la aplicación. A continuación, se describen algunos ejemplos relevantes al trabajo de esta tesis.

Por ejemplo, Ebner (1998) intenta construir un operador, basándose en GP, que sea capaz de reproducir la funcionalidad del detector de puntos de interés propuesto por Moravec (1977). Los resultados experimentales muestran que el mejor operador sintetizado tiene un error del 15% con respecto al detector de Moravec. En un trabajo posterior, Ebner y Zell (1999) utilizan el rendimiento de una aplicación real para definir el criterio de evaluación, en este caso la estimación del flujo óptico. Para la secuencia de imágenes que emplean, el detector evolucionado logra un rendimiento bastante bueno. Sin embargo, en los dos trabajos (Ebner, 1998; Ebner y Zell, 1999) se presentan limitaciones considerables para el planteamiento que proponen. En el primero, se asume que identificar puntos de interés tal como lo hace el detector de Moravec es una meta razonable, sin embargo la literatura dice lo contrario (Schmid *et al.*, 2000). En el segundo, la función de aptitud incorpora sesgos indeseables en la evolución, por enfocarse en un problema tan particular y por utilizar una secuencia de imágenes que no puede garantizar la síntesis de un operador para uso general.

Recientemente, Krawiec (2007) propuso un GP para construir operadores capaces de identificar figuras sencillas. Otro aspecto interesante de ese trabajo es el uso de la optimización MO, algo poco común en trabajos de visión por computadora. Sin embargo, aún no es claro si el enfoque propuesto se puede extender a dominios de problemas con una complejidad mayor.

Para resumir, a partir de la revisión bibliográfica que se realizó en este capítulo, se derivan tres conclusiones básicas que sustentan el enfoque elegido en esta tesis.

Primero, el paradigma del cómputo evolutivo, desde sus inicios a mediados del siglo pasado, ha madurado con el tiempo y hoy día ofrece a la comunidad científica herramientas muy poderosas para problemas de búsqueda, optimización y diseño automático de soluciones para problemas de diferentes dominios. Segundo, el área de visión por computadora presenta muchos problemas interesantes que debido a su complejidad siguen sin resolverse. Además, existe una comunidad, cada vez mayor, de investigadores que han decidido abordar estos retos con enfoques inspirados en sistemas biológicos, especialmente el proceso de la evolución natural. Tercero, el enfoque de visión basado en la *detección-descripción* de regiones locales en imágenes aún no ha sido abordado empleando técnicas evolutivas.

Posiblemente, si se emplea el paradigma evolutivo será posible descubrir soluciones nuevas a problemas viejos, que impacten por su sencillez y efectividad. Tal vez, en un ambiente artificial se podrán observar algunas de las características ennoblecedoras que Darwin identificó en los sistemas naturales que se dedicó a estudiar.

Capítulo III

Puntos de interés

En este Capítulo se presenta el problema de detección de puntos de interés. Primero, se define lo que constituye una región visualmente interesante, enfocándose en los rasgos más sencillos que son los puntos. Después, se revisa el estado-del-arte y se estudian los criterios empleados para evaluar el proceso empleado para la detección de estos *puntos de interés*.

III.1 Rasgos visualmente interesantes

En el Capítulo I, se describe el enfoque de visión por computadora que se basa en la detección estable y la descripción representativa de regiones locales en imágenes (Schmid y Mohr, 1997; Lowe, 1999). Este tipo de rasgos son relativamente pequeños comparados con toda la imagen, y se les llama *regiones de interés* porque transmiten información que se considera *visualmente interesante*. Una medida de cuan interesante es un píxel, o región, de una imagen se puede extraer empleando un mapeo u operador $K : \mathbb{R}^+ \to \mathbb{R}$. Aquí cabe hacer la distinción entre un operador y un detector. El primero, solo se utiliza para calcular una medida de lo *interesante* que es cada píxel; el segundo, es el proceso algorítmico empleado para identificar todos los pixeles o regiones interesantes presentes en una imagen. De tal forma que diferentes detectores emplean diferentes operadores K durante el proceso de detección. Cuando se aplica K a una imagen I se obtiene lo que se llama como una imagen de interés I^{*}. Después, la mayoría de los



Figura 7: Puntos de interés utilizados para resolver el problema de correspondencia en un par estéreo de imágenes.

detectores emplean un proceso similar que consiste en: 1) supresión de no-máximos; y 2) un umbralizado para identificar los picos en la respuesta al operador K.

Hasta el momento, la definición que se ha presentado para el concepto de regiones de interés es aún muy general. Para poder taxonomizar este tipo de rasgos es necesario entender cuales son las propiedades primordiales que se esperan de ellos. Como ya se mencionó, un uso común para las regiones de interés es en problemas de reconocimiento, por lo que normalmente se requiere que la detección de las mismas sea estable y robusta con respecto a las condiciones bajo las cuales se captura la imagen. Regiones estables y fáciles de detectar son, en las palabras de Shi y Tomasi (1994), *rasgos fáciles de seguir*. La Figura 7 muestra un ejemplo de aplicación, en donde puntos prominentes se utilizan para resolver el problema de correspondencia en un par estéreo de imágenes (Olague *et al.*, 2006b).

Por lo tanto, las regiones de interés contienen pixeles que exhiben una propiedad distintiva que las hacen adecuadas para aplicaciones en donde es necesario identificar las mismas regiones en una secuencia de imágenes, o en imágenes tomadas bajo diferentes condiciones. Ahora bien, la estabilidad geométrica de un detector se puede medir en base a diferentes tipos de transformaciones, ver Tabla I. Entonces, la forma de las Tabla I: Tipo de transformaciones que se le pueden aplicar a una imagen; donde $\mathbf{I} \ge \mathbf{x}$ representan a la imagen original y un punto en esta imagen respectivamente, mientras que $\mathbf{I}' \ge \mathbf{x}'$ son las versiones transformadas.

1) Iluminación:
$$I'(\mathbf{x})' = a \cdot I(\mathbf{x}) + b$$

2) Rotación: $\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$
3) Escala: $\mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$
4) Proyectiva: $\mathbf{x}' = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{x}$

regiones de interés dependen del tipo de invarianza que se desea. Por ejemplo, cuando un detector solo detecta pixeles, o puntos de interés, entonces solo se espera que este sea invariante con respecto a trasformaciones de iluminación y rotación (Schmid *et al.*, 2000; Trujillo y Olague, 2006a,b, 2008). Cuando un detector también es invariante a cambios de escala, entonces es capaz de detectar regiones de interés isotrópicas de diferentes tamaños, o bien, es un detector invariante a escala (Mikolajczyk y Schmid, 2004; Trujillo y Olague, 2007). Finalmente, cuando un detector es invariante a las cuatro transformaciones de la Tabla I se dice que este detecta regiones afinmente-covariantes (Mikolajczyk *et al.*, 2005). En esta tesis, solo se estudia los primeros dos tipos de regiones, y en este capítulo la discusión se centra en la detección de puntos de interés.

III.2 Detección de puntos de interés

Las técnicas para la detección de puntos de interés se desarrollaron como resultado del trabajo realizado para resolver el problema de detección de esquinas. La taxonomía para detectores de esquinas esta compuesta por tres clases principales: métodos basados en contornos (Asada y Brady, 1986; Mokhtarian y Suomela, 1998; Gao *et al.*, 2007), métodos basados en modelos paramétricos (Rohr, 1992; Deriche y Giraudon, 1993; Olague y Hernández, 2005; Sinzinger, 2008), y métodos basados en intensidad (Moravec, 1977; Beaudet, 1978; Kitchen y Rosenfeld, 1982; Förstner y Gülch, 1987; Harris y Stephens, 1988; Wang y Brady, 1991; Shi y Tomasi, 1994; Schmid *et al.*, 2000). La clase de detectores de esquinas que operan directamente sobre los valores de intensidad en una imagen son más comúnmente conocidos como detectores de puntos de interés, la diferencia es sutil pero importante conceptualmente. Por un lado, las esquinas son rasgos que se encuentran en uniones entre lineas y superficies; por ejemplo, uniones tipo L, T, Y o X, o bien, cualquier tipo de vértice. Por otro lado, el concepto de punto de interés puede incluir estos rasgos además de otros que carecen de una interpretación semántica estricta o clara. A continuación, se presenta ejemplos importantes de detectores de puntos de interés.

Por ejemplo, algunos detectores basan la medida de interés que arroja el operador K, en la matriz de autocorrelación local A que caracteriza la distribución del gradiente dentro del vecindario de cada pixel, dado por

$$A(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 \cdot G_{\sigma_I} * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x(\mathbf{x}, \sigma_D) L_y(\mathbf{x}, \sigma_D) \\ L_x(\mathbf{x}, \sigma_D) L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix},$$

donde σ_D y σ_I son, respectivamente, las escalas de derivación e integración, $L_u(\mathbf{x}, \sigma_D)$ es la derivada Gaussiana en dirección u de la imagen I en el punto \mathbf{x} dado por

$$L_u(\mathbf{x}, \sigma_D) = \frac{\delta}{\delta u} G_{\sigma_D} * I(\mathbf{x}) ,$$

 G_{σ} es una función Gaussiana de suavizado con desviación estándar σ ; en esta tesis $\sigma_D = 1$ al menos que se especifique lo contrario. Los detectores que utilizan A incluye a los propuestos por Förstner y Gülch (1987), Harris y Stephens (1988), y Shi y Tomasi (1994), con los operadores de cada uno definidos de la siguiente forma:

$$K_{Forstner}(\mathbf{x}) = \frac{det(A)}{Tr(A)} ,$$

$$K_{Harris\&Stephens}(\mathbf{x}) = det(A) - k \cdot Tr(A)^2$$

$$K_{Shi\&Tomasi}(\mathbf{x}) = \min\{\lambda_1, \lambda_2\}$$
,

donde λ_1, λ_2 son los eigenvalores de A; la definición de A se tomó de (Schmid *et al.*, 2000), en donde se emplea con el detector de *Harris Mejorado* descrito en aquél trabajo.

Beaudet (1978) propone el determinante de la matriz Hessiana como un operador, una medida que es proporcional a la curvatura de Gauss,

$$K_{Beaudet}(\mathbf{x}) = I_{xx}(\mathbf{x}) \cdot I_{yy}(\mathbf{x}) - I_{xy}^2(\mathbf{x})$$

donde $I_u(\mathbf{x})$ es la derivada de la imagen en la dirección u. Wang y Brady (1991) caracterizan la curvatura local empleando el Laplaciano y la magnitud del gradiente,

$$K_{Wang\&Brady}(\mathbf{x}) = (\nabla^2 I(\mathbf{x}))^2 - s |\nabla I(\mathbf{x})|^2$$

Kitchen y Rosenfeld (1982) proponen una medida de interés que busca detectar esquinas utilizando el producto entre la magnitud del gradiente y la magnitud del cambio de dirección del gradiente,

$$K_{Kitchen\&Rosenfeld}(\mathbf{x}) = \frac{I_{xx}(\mathbf{x})I_y^2(\mathbf{x}) + I_{yy}(\mathbf{x})I_x^2(\mathbf{x}) - 2I_{xy}(\mathbf{x})I_y(\mathbf{x})I_x(\mathbf{x})}{I_x^2(\mathbf{x}) + I_y^2(\mathbf{x})}$$

A pesar de los diferentes operadores propuestos, todos los detectores siguen un proceso similar básico. Primero, se aplica el operador K a la imagen I para obtener una imagen de interés I^* . Después, un pixel \mathbf{x} se etiqueta como un punto de interés si se cumple la siguiente condición,

$$K(\mathbf{x}) > \sup \{ K(\mathbf{x}_{\mathbf{W}}) | \forall \mathbf{x}_{\mathbf{W}} \in \mathbf{W}, \mathbf{x}_{\mathbf{W}} \neq \mathbf{x} \} \land K(\mathbf{x}) > h , \qquad (1)$$

donde **W** es un vecindario cuadrado de tamaño $n \times n$ alrededor de **x**, y h es un umbral que se define de forma experimental. La primera condición representa un proceso de supresión de no-máximos y la segunda es un umbralizado para detectar el conjunto final de puntos; este proceso se ilustra en la Figura 8. Los experimentos en esta tesis utilizan n = 5, mientras que h depende del operador.



Figura 8: Cada columna muestra el proceso de detección de puntos: columna izquierda, la imagen de entrada I; en medio, la *imagen de interés* I^* ; columna derecha, los puntos finales detectados después de la supresión de no-máximos y el umbralizado. Los puntos de interés en estos ejemplos se detectaron con el operador K_{IPGP2} .

Cabe mencionar que se han propuesto varias extensiones al proceso descrito arriba. Por ejemplo, Laptev y Lindeberg (2003) incluyen un análisis espacio-tiempo, y van de Weijer *et al.* (2006) incorporan información de color. Sin embargo, estos temas no se contemplan en el contenido de esta tesis. Además, existen otros enfoques para la detección de puntos de interś, por ejemplo: utilizando un análisis geométrico (Smith y Brady, 1997), teoría de catástrofe (Platel *et al.*, 2006), modelos de la ley de potencias (Caron *et al.*, 2007), análisis basado en wavelets Sebe y Lew (2003) y un análisis de variaciones de intensidad local (Lepetit y Fua, 2006).

III.3 Propiedades de los puntos de interés

En la Sección anterior se describe la manera en que se detectan puntos de interés; ahora se discuten las características cualitativas y cuantitativas que se esperan del conjunto de puntos que identifica un detector.

Por ejemplo, el *Manual de fotogrametría* (McGlone *et al.*, 2004) describe cuatro propiedades cualitativas que uno debería esperar de los puntos de interés: **a**) *distintivos*: los puntos deben ser notablemente diferentes a otros puntos en su vecindario; **b**) *inusuales*: los puntos de interés no deben de ser comunes en una imagen; i.e., representan una minoría del total de puntos; **c**) *invariantes*: la detección de un punto no debe ser afectada por distorsiones radiométricas o geométricas; y **d**) *estabilidad*: el proceso de detección debe ser estable con respecto a ruido o perturbaciones desconocidas. Esta lista es razonable, sin embargo existen varias preguntas obvias: ¿Existe alguna diferencia práctica entre los puntos **a** y **b**, o entre **c** y **d**? ¿Cómo se pueden medir estas propiedades de manera cuantitativa? Este tipo de cuestiones no se abordan en (McGlone *et al.*, 2004).

Otra forma de evaluar el proceso de detección es definir una lista de propiedades y después determinar que operadores satisfacen dicha lista. Este es el enfoque tomado por Kenney *et al.* (2005), donde los autores definen cuatro axiomas matemáticos que pueden ser verificados analíticamente. Dichos axiomas se basan en la relación que guarda la matriz de autocorrelación con la imagen de una esquina ideal. Kenney *et al.* (2005) determinan que el detector de Shi y Tomasi (1994) es el único que satisface todos los axiomas que proponen. Sin embargo, dos observaciones son relevantes para ese trabajo. Primero, los axiomas suponen que solo las esquinas son *puntos interesantes*, algo que no se puede asumir como verdadero en muchas situaciónes del mundo-real. Segundo, se asume que la matriz de autocorrelación es esencial para detectar puntos de interés. Sin embargo, esto solo se argumenta para una aplicación donde se calcula el flujo óptico, no es evidente que este supuesto sea verdadero en otros dominios.

Por otro lado, un enfoque experimental se propone por Schmid *et al.* (2000), definiendo dos propiedades cuantitativas: **1**) la repetibilidad; y **2**) el contenido de información en el conjunto de puntos detectados. El primero cuantifica la invarianza de un detector dada un transformación geométrica o radiómetrica, ver Tabla I. El segundo mide que tan informativo es el vecindario local de cada punto; i.e., que tan dispersas están las regiones locales de cada punto una vez que estas se mapean a un espacio de descriptores. La evaluación experimental de Schmid et al. (2000) sugiere que su versión mejorada del detector de Harris y Stephens (1988), K_{Harris} , tiene el rendimiento más alto. Cabe mencionar, que debido a esos resultados, por lo menos en parte, el detector de Harris en la actualidad es el más utilizado por la comunidad de visión por computadora. Sin embargo, la comparación experimental de (Schmid et al., 2000) exhibe varias carencias notables. Primero, varios detectores no se incluyen en la comparación, por ejemplo los de (Beaudet, 1978; Kitchen y Rosenfeld, 1982). Segundo, en (Schmid et al., 2000) el contenido de información se estima utilizando una descripción basada en el jet local (Koenderink y van Doom, 1987). Pero resultados más recientes de Mikolajczyk y Schmid (2005) sugieren que el descriptor SIFT (Lowe, 1999) es mejor para caracterizar de manera única cada región local. Tercero, aunque los criterios de repetibilidad y contenido de información son razonables, existen otros que se pudieran tomar en cuenta. Por ejemplo, la cantidad de separación global de los puntos; i.e., la dispersión en el plano de la imagen (Rebai et al., 2007). Finalmente, la evaluación hecha en (Schmid et al., 2000) no estudió la relación que existe entre cada uno de los criterios, este tema se analiza en el Capítulo VI de esta tesis utilizando un enfoque multiobjetivo.

Tissainayagam y Suter (2004) presentan otro ejemplo de como se puede evaluar la detección de puntos de manera experimental. Los autores definen dos medidas de rendimiento: *localización* y *estabilidad*. El segundo es similar al criterio de repetibilidad de Schmid *et al.* (2000), mientras que el primero se enfoca más en la precisión de la detección, similar al análisis hecho por Olague y Hernández (2005). En realidad, los criterios propuestos son similares y algo redundantes. Más aún, los autores hacen supuestos basados en una aplicación especifica, seguir rasgos en una secuencia de video, lo cual hace que su propuesta no se extienda fácilmente a una evaluación general. Sin embargo, sus resultados son congruentes con los de (Schmid *et al.*, 2000; Kenney *et al.*, 2005) ya que sugieren que los mejores detectores son el de Harris y Stephens (1988) y Shi y Tomasi (1994).

Por otro lado, elegir un detector para una aplicación en particular también se ve influenciado por los requerimientos específicos del dominio de la tarea que se quiere resolver. El trabajo de Rebai *et al.* (2007) sirve como ejemplo, porque en dicho trabajo lo que los autores deseaban era detectar puntos centrados en conceptos semánticos de alto-nivel para así simplificar un procesamiento de reconocimiento posterior. Li et al. (2008) desarrollan un detector basado en un proceso de selección especializado, para mejorar la detección de puntos en imágenes que carecen de textura; su esquema lo prueban con una tarea de reconocimiento de rostros. Lepetit y Fua (2006) desarrollan un detector optimizado para aplicaciones de tiempo real, sacrifican un cierto grado de rendimiento para reducir la complejidad computacional. Dos ejemplos finales son los trabajos de Davison y Molton (2007) y Yang et al. (2007). En esos dos ejemplos, se busca que la dispersión de los puntos detectados sea alta, en la primera para un problema de SLAM monocular y en la segunda para resolver el problema de registrar imágenes. En los dos casos, los autores deciden utilizar un detector previamente diseñado; sin embargo, es razonable asumir que un detector optimizado para los requerimientos de su aplicación hubiera ayudado a mejorar el rendimiento final del sistema.

Resumiendo, debido a la cantidad, tal vez abrumadora, de detectores disponibles en la literatura, elegir uno para una aplicación concreta no es una tarea trivial. Además, si se requiere que un detector exhiba ciertas características a veces es necesario diseñar uno empezando desde cero. Se puede observar que si existiera un sistema capaz de automatizar este proceso, el peso de diseño sería menos para los desarrolladores de sistemas de visión.

Por otro lado, la literatura muestra que evaluar detectores puede hacerse utilizando diferentes enfoques: cualitativo (McGlone *et al.*, 2004), analítico (Kenney *et al.*, 2005), o experimental (Schmid *et al.*, 2000; Tissainayagam y Suter, 2004). De estos, los que más se apegan al espíritu del paradigma del CE son aquellos que utilizan medidas experimentales.

Tomando en cuenta estas consideraciones, además de las observaciones hechas en (Davison y Molton, 2007; Yang *et al.*, 2007; Rebai *et al.*, 2007), se eligen tres propiedades básicas para evaluar los detectores de puntos interés.

- Estabilidad con respecto a cierto tipo de transformaciones: traslación, rotación y cambios de iluminación. Para medir la estabilidad se utiliza el criterio de *repetibilidad* definido en (Schmid *et al.*, 2000).
- 2. Dispersión de los puntos detectados. Lo que se espera es que en la mayoría de las escenas los puntos de interés estarán dispersos en el plano de la imagen. Este criterio depende de las características particulares de cada imagen, y para poder evaluarlo objetivamente se tendrá que definir a priori cual es una buena dispersión de puntos.
- 3. La información contenida en el conjunto de puntos detectados. Cuando el contenido de información es alto se supone que los puntos detectados son distintivos y prominentes. Esta propiedad ayuda a simplificar el problema de correspondencia entre dos imágenes.



Figura 9: Un punto 3D X se proyecta en x_1 y x_2 en las imágenes I_1 y I_2 respectivamente. El punto x_1 se repite en x_i si se detecta un punto en un vecindario de x_i del tamaño ϵ . Para escenas planas x_1 y x_i se relacionan por la homografía $H_{1,i}$.

III.4 Medidas de evaluación para un detector de puntos de interés

A continuación se definen medidas explicitas que permiten cuantificar cada una de las propiedades enlistadas al final de la Sección III.3.

III.4.1 Tasa de repetibilidad

La tasa de repetibilidad mide el grado de invarianza que un detector tiene con respecto a cambios producidos cuando se adquiere una imagen, transformaciones como la rotación y el cambio de iluminación (Schmid *et al.*, 2000). Un punto de interés x_1 detectado en una imagen I_1 se repite en una imagen I_i si un punto correspondiente x_i se detecta en I_i ; por simplicidad, se asume que las dos imágenes son del mismo tamaño. En el caso de escenas planas, la relación entre los puntos x_1 y x_i se establece por la homografía $H_{1,i}$, donde

$$x_i = H_{1,i} x_1 \ . \tag{2}$$

La tasa de repetibilidad mide el número de puntos repetidos entre las dos imágenes con respecto al total de puntos detectados. Sin embargo, partes de la imagen I_1 tal vez no aparecen en la imagen transformada I_i . Para contemplar este fenómeno, los puntos repetidos y detectados solamente se cuantifican si caen dentro de la región que comparten las dos imágenes I_1 y I_i . Más aún, se debe de permitir una cantidad pequeña de error en la detección de puntos repetidos porque normalmente no se requiere localización de alta precisión . Por lo tanto, se dice que existe un punto repetido en x_i si se detecta un punto en un vecindario de x_i de tamaño ϵ , con $\epsilon = 2.5$ pixels en los experimentos de esta tesis, ver Figure 9. El conjunto de pares de puntos (x_i^c, x_i^c) que se encuentran en la parte común de las dos imágenes y corresponden dentro un margen de error ϵ esta definido por

$$R_{I_i}(\epsilon) = \{ (x_1^c, x_i^c) | dist (H_{1,i} x_1^c, x_i^c) < \epsilon \} .$$
(3)

Tal que la tasa de repetibilidad $r_i(\epsilon)$ de los puntos detectados en I_i con respecto a los puntos de la imagen I_1 esta definido por

$$r_{I_i}(\epsilon) = \frac{|R_{I_i}(\epsilon)|}{\min(\gamma_1, \gamma_i)}, \qquad (4)$$

donde $\gamma_1 = |\{x_1^c\}| \ge \gamma_i = |\{x_i^c\}|$ son el número total de puntos detectados en $I_1 \ge I_i$ respectivamente.

III.4.2 Dispersión de puntos

Este criterio sugiere que en la "mayoría de las imágenes" los puntos de interés no estarán amontonados en regiones asiladas del plano de la imagen. Además, si los puntos están dispersos en la imagen entonces es más probable que los puntos detectados sean diferentes y capturen diferente tipo de información. Para evaluar esta propiedad se requiere un conocimiento aproximado del número de puntos que se espera encontrar



Figura 10: Dispersión de puntos: Imagen original, puntos altamente dispersos, puntos con poca dispersión espacial.

en una imagen y su posición dentro de la imagen. Sin embargo, si es una medida útil para determinar que tan bueno es un operador o detector para muestrear la información contenida dentro de la imagen. La dispersión de puntos puede cuantificarse utilizando un criterio basado en la entropía de la posición de los puntos. De tal forma que la entropía se calcule de la partición $\{\mathcal{I}_j\}$ del plano de la imagen *I*. Por lo tanto , \mathcal{D} será la entropía de la distribución en el espacio de los puntos detectados, dado por

$$\mathcal{D}(I,X) = -\sum P_j \cdot \log_2(P_j) , \qquad (5)$$

donde P_j se aproxima por el histograma 2D de la posición de los puntos detectados en *I*. De tal forma que las probabilidades se calculan con $P_j = \frac{\Gamma(x,y)}{\gamma}$, donde $\Gamma(x,y)$ el número de puntos de interés contenidos dentro de la casilla (x, y) y γ el total de puntos detectados. En la figura 10 se muestran dos extremos en la detección de puntos, en uno con puntos muy dispersos y el caso contrario en el otro.

III.4.3 Contenido de información

Schmid *et al.* (2000) definen esta medida en base a la probabilidad de calcular un descriptor en el vecindario de un punto de interés. Para cada punto de interés \mathbf{x} que se detecta, se calcula un descriptor local γ . De tal forma que si se considera que un detector identifica un conjunto X de n puntos de interés, entonces existe un conjunto



Figura 11: Una descripción gráfica del concepto de contenido de información.

correspondiente de descriptores Γ , donde $\forall \mathbf{x} \in X \exists \gamma \in \Gamma$. Más aún, si Υ representa el espacio de descriptores, cuando los descriptores en Γ estén amontonados en regiones aisladas de Υ entonces se dice que el conjunto X transmite una cantidad pequeña de información \mathcal{I} ; lo opuesto es verdad en el caso contrario. Entonces, cuando los descriptores se utilizan para resolver problemas de correspondencia el conjunto Γ que maximiza la probabilidad $p(I|\Gamma)$ es el conjunto de descriptores que mejor caracterizan a I. Basándose en la teoría de la información, \mathcal{I} se puede obtener de la entropía en el conjunto Γ . Por lo tanto, si se considera una partición $\Upsilon = {\Upsilon_j}$, y la probabilidad p_j se aproxima con un histograma en el espacio de los descriptores $\gamma \in \Upsilon_j$ del conjunto Γ , entonces la información contenida en el conjunto X de puntos detectados en I esta dada por

$$\mathcal{I}(\Gamma) = -\sum p_j \cdot \log_2(p_j) .$$
(6)

Dada esta definición, será necesario elegir un descriptor local para calcular \mathcal{I} ; la Figura 11 ilustra el concepto de una manera gráfica. Schmid *et al.* (2000) utilizaron el *jet local*, sin embargo en esta tesis se utiliza el descriptor SIFT y un descriptor nuevo basado en un análisis de regularidad local; detalles de la implementación se discuten en la Sección VI.

Dado como concluido, para los propósitos de esta tesis, la introducción sobre la detección de rasgos interesante el siguiente capítulo aborda el área de los descriptores locales, describe de forma general sus dominios de aplicación, y presenta una revisión del estado-del-arte.
Capítulo IV

Descriptores locales

Retomando el diagrama que se muestra en la Figura 1, en este capítulo nos enfocamos en el proceso que tiene como tarea describir el vecindario local alrededor de cada rasgo que fue identificado, por ejemplo, por un detector de puntos de interés. La Figura 12 ilustra el proceso básico utilizado para obtener el descriptor para una región local; donde dicho proceso se conforma de tres pasos básicos:

- 1. Detectar regiones. Este es el proceso que se describe en el Capítulo III.
- 2. Normalización. Con el propósito de mantener uniformidad en procesos posteriores y facilitar el cálculo de correspondencias, y además para obtener invarianza con respecto a cambios de iluminación, rotación y escala, cada región se normaliza en base a tres criterios: a) la intensidad de la región (invarianza con respecto a iluminación); b) la orientación principal del gradiente (rotación); y c) todas las regiones se ajustan a un tamaño común (escala).
- 3. *Construir del descriptor*. Este es el proceso mediante el cual se obtienen valores numéricos representativos a partir de la información contenida en cada región; este es el proceso que se estudia con mayor detalle en el resto del capítulo.

El funcionamiento básico de este proceso se puede entender mejor si se utiliza un ejemplo ilustrativo. Asume que se detecta una región de interés \mathbf{A} centrada en el pixel \mathbf{x} . Si se utiliza un detector de puntos de interés (ver Capítulo III) el radio t, o escala, de \mathbf{A} será igual para todas las regiones detectadas en la imagen, mientras que si se utiliza



Figura 12: Se muestra el diagrama básico empleado en la descripción de regiones locales. Primero, se detectan las regiones de interés. Segundo, se normaliza cada región de la imagen, por ejemplo en base a la intensidad o la orientación principal del gradiente. Tercero, un proceso o algoritmo construye un vector de rasgos numéricos, normalmente concatenados, que constituyen al descriptor final.

un detector invariante a escala diferentes regiones podrán tener diferentes tamaños (ver Capítulo VII). Para calcular el descriptor, se utiliza una porción de la imagen conocida como la región de soporte \mathbf{A}_S del descriptor, de un tamño proporcional al de \mathbf{A} , por ejemplo de radio 3t, y también centrado en \mathbf{x} . Después, \mathbf{A}_S se recorta de la imagen y se normaliza a un tamaño común para todas las regiones, de 41×41 pixels por ejemplo; la porción normalizada la denominamos con $\overline{\mathbf{A}_S}$. Esta región local de la imagen, $\overline{\mathbf{A}_S}$, se rota con respecto a la dirección principal del gradiente; de tal forma que todas las regiones $\overline{\mathbf{A}_s^r}$ están orientadas con respecto al gradiente con el fin de obtener un descriptor invariante con respecto a rotaciones. Finalmente, se aplica algún algoritmo que calcula un vector numérico $D_{\mathbf{A}}$ característico utilizando la informacion contenida en $\overline{\mathbf{A}_s^r}$; de tal forma que $D_{\mathbf{A}}$ describe a la región original \mathbf{A} .

Existen dos enfoques básicos que se pueden tomar cuando se intenta describir cualquier tipo de información, en este caso pequeñas porciones de una imagen digital. En términos generales se habla de modelar a la información de forma generativa o discriminatoria; en el primer caso lo que se desea es utilizar un *descriptor incluyente*, mientras que en el segundo se desea tener un *descriptor discriminante*.

- Descriptor incluyente. Este término lo utilizamos para hacer referencia a los descriptores locales cuyo propósito es capturar las similitudes que tienen regiones locales detectadas sobre un mismo objeto o imagen. En otras palabras, se considera que cada región local proviene de una misma *clase* o categoría, y por lo tanto el descriptor debe de ser capaz de capturar la similitud de rasgos entre regiones de una misma clase, y al mismo tiempo diferenciarse de las características generales que exhiben las regiones extraídas de otra clase u objeto.
- Descriptor discriminante. Este tipo de descriptores se utilizan para generar una descripción única para cada región local detectada en una imagen. Por lo tanto, el propósito de estos descriptores es capturar los rasgos visuales de cada región, eliminar información superflua o redundante, y caracterizar de manera única los patrones visuales presentes dentro de cada región.

Cabe mencionar que esta taxonomía no esta libre de ambigüedades, particularmente porque un descriptor incluyente, tal como se define arriba, puede construirse a partir de un conjunto de descriptores altamente discriminantes. Esto se puede observar en los métodos de *bolsas de rasgos* (Sivic y Zisserman, 2003; Willamowski *et al.*, 2004; Sivic *et al.*, 2005; Perronnin, 2008), en donde se emplean técnicas de *clustering* para encontrar descriptores característicos relacionados a un rasgo que se puede identificar como útil durante el proceso de reconocimiento. Sin embargo, para los propósitos de la discusión hecha en esta tesis, la taxonomía anterior será suficiente para diferenciar entre los dos enfoques tomados para resolver la tarea de descripción local.

A continuación, se presenta una discusión más detallada de cada uno de los enfoques de descripción que se mencionan arriba, con un énfasis en el segundo enfoque debido al alto rendimiento que produce en una variedad amplia de problemáticas.



Figura 13: Diagrama conceptual para los descriptores incluyentes. En la figura se observa como los descriptores que se obtienen de cada objeto se mapean a una misma región del espacio del descriptor empleado.

IV.1 Descriptor incluyente

Este tipo de descriptores se construyen cuando se desea capturar los rasgos compartidos entre regiones que fueron detectados en una misma clase, en donde el concepto de clase se puede referir a un objeto, imagen, o escena en general. Normalmente, este tipo de descriptores se diseñan con un problema, o un dominio de problemas, en mente, o bien, las soluciones propuestas suelen estar fuertemente asociadas a una problemática muy específica. La Figura 13 muestra un diagrama conceptual de este tipo de descriptores, en donde los descriptores que se obtienen de cada objeto se mapean a una misma región del espacio del descriptor empleado.

Por otro lado, los rasgos que se utilizan para construir este tipo de descriptores suelen ser estadísticas o histogramas relacionados con el contenido de color o textura en cada región. Por ejemplo, Szummer y Picard (1998) abordan el problema de identificar cuando una imagen muestra una escena de interior o una escena de exterior. Para esto, los autores describen cada porción de la imagen utilizando dos tipos de información: color y textura. Para capturar la información de color utilizan histogramas calculados en el espacio Otha (Ohta *et al.*, 1980), un espacio de color que tiene como ejes los tres eigenvectores principales del espacio RGB encontrados utilizando PCA. Además, para describir la información de textura emplean modelos autoregresivos simultáneos y de resolución multiple (MSAR).

Vogel y Schiele (2007) construyen modelos semánticos de escenas naturales con el fin de recuperar imágenes en base a su contenido. Su propuesta consiste en asignar cada porción local de la imagen a una clase semántica predefinida; e.g., agua, piedras o follaje. En base a la frecuencia con que aparece cada clase semántica determinan el tipo de escena natural presente en la imagen. Como rasgos descriptivos utilizan histogramas de 82 casillas en el espacio HSI con lo cual describen la información de color. Para capturar la información de textura los autores utilizan dos tipos de rasgos, un histograma de 72 casillas que representa la distribución de bordes en cada región y 32 valores descriptivos calculados a partir de la la matriz de coocurrencia. Es preciso mencionar que en este trabajo, al igual que el anterior, las regiones locales no son elegidas utilizando un detector, sino que se divide a la imagen con una cuadrícula regular, y se utilizan todas las casillas como regiones locales.

Olague *et al.* (2007) también emplean una descripción basada en la matriz de coocurrencia; sin embargo, en este caso los autores utilizan un algoritmo evolutivo para elegir las regiones de interés, un enfoque similar al propuesto originalmente en (Hernández *et al.*, 2007). Esta metodología permite a los autores diseñar un sistema que es capaz de clasificar imágenes de diferentes tipos de objetos, tales como caras, carros, casas, arboles y vacas.

Carneiro *et al.* (2007) utilizan un enfoque similar al descrito en (Vogel y Schiele, 2007), con algunas diferencias notables, para etiquetar imágenes y recuperarlas en base

al contenido. Por ejemplo, en lugar de utilizar una cuadricula para identificar las regiones los autores emplean una ventana que se desplaza a lo largo y ancho de la imagen de tal forma que existe una superposición entre regiones adyacentes. El vector descriptivo se construye utilizando la transformada coseno directa en el espacio de color YBR. Cada vector local se concatena para obtener una representación global de toda la imagen, posteriormente las clases se representan utilizando modelos de mezcla de Gaussianas.

Markou y Singh (2006) abordan el problema de identificar objetos novedosos en una secuencia de imágenes, en otras palabras, se asigna cada región de un *frame* de video a una clase conocida o se etiqueta como una instancia de una clase nueva previamente desconocida. A diferencia de los métodos descritos arriba, los autores utilizan un algoritmo de segmentación, Fuzzy C-Means, para identificar cuales son las regiones locales que serán analizadas posteriormente. Los autores primero definen un espacio general de rasgos que se pueden utilizar para describir cada región, un total de 506 rasgos. Después, emplean el *sequential floating forward selection algorithm* (SFFS) para seleccionar un subconjunto de rasgos; el algoritmo selecciona 10 rasgos que son concatenados para construir el vector descriptivo de cada región, los rasgos incluyen estadísticas de color, máscaras de Law, momentos y coeficientes de la transformada *wavelet*.

IV.2 Descriptor discriminante

Como se dice arriba, los descriptores de esta clase normalmente se utilizan para generar una descripción única para cada región detectada en una imagen. Esto facilita la tarea de calcular correspondencias locales entre dos imágenes y así poder reconocer partes de una imagen u objeto conocido; el proceso básico se ilustra en la Figura 14.

Un ejemplo má completo de este proceso se presenta en la Figura 15, donde se emplea



Figura 14: Diagrama conceptual que describe la manera en la cual se obtienen la correspondencia entre dos regiones de interés utilizando el descriptor local que se calcula para cada una.

el descriptor SIFT (Lowe, 1999) para reconocer dos objetos dentro de una escena. Las Figuras 15a-c muestran las imágenes originales de una escena compleja con muchos objetos, y dos imágenes de objetos por si solos que también se encuentran en la escena. Después, las Figuras 15d-f muestran las regiones detectadas en cada una de las imágenes en forma de flechas, el tamaño y orientación de cada flecha es proporcional al tamaño y dirección principal del gradiente en cada región correspondiente. Finalmente, las Figuras 15g-h muestran las correspondencias calculadas entre la imagen de la escena y la imagen de cada objeto. Este proceso es una de las aplicaciones más comunes que tienen los descriptores discriminantes, pero no es la única.

Descriptores locales de este tipo han demostrado ser de gran utilidad en muchas áreas de la visión artificial, la Tabla II resume varios ejemplos importantes. Como se puede apreciar, estas técnicas han tenido un impacto significativo en el campo de la



(g) Correspondencia: escena-libro

(h) Correspondencia: escena-caja

Figura 15: Aplicación básica de un descriptor discriminante, en este caso es el SIFT (Lowe, 1999), para el problema de reconocimiento de objetos. $(\mathbf{a} - \mathbf{c})$ Imágenes originales de una escena y dos objetos. $(\mathbf{d} - \mathbf{f})$ Regiones y descriptores SIFT calculados en cada imagen. (\mathbf{g}, \mathbf{h}) Correspondencias calculadas utilizando los descriptores SIFT.

visión artificial, y es posible encontrar un gran número de propuestas en la literatura correspondiente. Por ende, a continuación se realiza una revisión del estado-del-arte en el área.

Área de aplicación	Referencias
Correspondencia en vistas	(Schaffalitzky y Zisserman, 2002),
muy separadas.	(Tuytelaars y Van Gool, 2004)
	(Tola <i>et al.</i> , 2008)
Reconocimiento de objetos.	(Lowe, 2004; Ferrari <i>et al.</i> , 2006)
Reconocimiento de texturas.	(Lazebnik <i>et al.</i> , 2005)
Reconocimiento de rostros.	(Mian <i>et al.</i> , 2008)
Recuperación de imágenes.	(Schmid y Mohr, 1997; Mikolajczyk y Schmid, 2001)
Localización de robots.	(Se <i>et al.</i> , 2002)
Minería de datos en video.	(Sivic y Zisserman, 2003)
Construccion de panoramas.	(Brown y Lowe, 2003; Brown <i>et al.</i> , 2005)
Reconocimiento de categorías	(Dorkó y Schmid, 2003; Fergus <i>et al.</i> , 2003),
de imágenes.	(Amores et al., 2007; Leibe et al., 2008)
SLAM monocular.	(Davison y Molton, 2007)
Deteccion de peatones.	(Dalal y Triggs, 2005)

Tabla II: Áreas de aplicación para los descriptores locales discriminantes.

IV.2.1 Taxonomía

Para presentar una revisión organizada de las técnicas propuestas en la literatura, nos basamos en la taxonomía hecha en (Mikolajczyk y Schmid, 2005), que consta de cuatro clases: 1) descriptores diferenciales; 2) descriptores basados en análisis de frecuencia;
3) descriptores basados en distribuciones; y 4) otras técnicas.

Descriptores diferenciales

Un conjunto de derivadas de la imagen, calculadas hasta algún orden dado, pueden representar el vecindario alrededor de un punto. Las propiedades de las derivadas locales, conocidas como el *local jet*, fueron investigadas por Koenderink y van Doom (1987). Florack *et al.* (1991) emplearon componentes del *local jet* para construir invariantes diferenciales. Freeman y Adelson (1991) desarrollaron filtros dirigibles, que orientan a las derivadas en direcciones estimadas a partir del *local jet*. Cuando las derivadas se orientan en la dirección del gradiente se hacen invariantes a la rotación. Otros descriptores diferenciales fueron propuestos en (Baumberg, 2000; Schaffalitzky y Zisserman, 2002).

Descriptores basados en análisis de frecuencia

Existe un grupo de técnicas que describen el contenido de frecuencia en cada región. La transformada de Fourier descompone el contenido de una imagen utilizando un conjunto de funciones base. Sin embargo, en esta representación la relación espacial entre los puntos no se especifica de manera explícita y las funciones base son infinitas, por lo tanto no es trivial adaptar esta representación a un método local. La transformada Gabor (Gabor, 1946) permite superar algunas de estas limitantes, pero se requiere de un número muy grande de filtros Gabor para capturar cambios pequeños de frecuencia y orientación. El análisis local de textura también se realiza utilizando una descomposición en *wavelets* (Vetterli y Kovačevic, 1995).

Descriptores basados en distribuciones

Estos descriptores utilizan histogramas para representar los rasgos característicos de apariencia y forma dentro de una región. El descriptor más sencillo sería la distribución de los valores de intensidad expresada como un histograma. Una mejor representación fue propuesta por Johnson y Hebert (1997) para reconocer objetos en 3D con un sensor de distancia. La representación hecha en su propuesta la denominan *spin image*, y corresponde a un histograma de la posición de puntos dentro del vecindario de un punto de interés en 3D. Lazebnik *et al.* (2005) adaptaron el *spin image* para poder aplicarlo en imágenes 2D, en donde las dos dimensiones del histograma son la distancia al punto central y el valor de intensidad correspondiente.

Zabih y Woodfill (1994) desarrollan un enfoque que es robusto a cambios de iluminación. Se basa en histogramas de ordenamiento y relaciones recíprocas entre intensidades de pixeles. Las relaciones binarias entre las intensidades de varios pixeles en el mismo vecindario se codifican con cadenas binarias y la distribución de todas las posibles combinaciones se expresa como un histograma. El descriptor es útil para representar textura pero requiere un número muy grande de dimensiones para que funcione de manera confiable (Ojala *et al.*, 2002).

Lowe (1999) propuso la transformada de rasgos invariante a escala (en inglés, Scale Invariant Feature Transform: SIFT), que combina un detector invariante a escala con un descriptor basado en la distribución del gradiente dentro de cada región. El descriptor se representa como un histograma 3D de la posición y orientación del gradiente.

El histograma geométrico (Ashbrook *et al.*, 1995) y el *shape context* (contexto de forma) (Belongie *et al.*, 2002) utilizan un enfoque similar y además son muy parecidos al SIFT. Los dos descriptores utilizan un histograma que describe la distribución de bordes dentro de cada región, y han demostrado ser útiles en la descripción de letras y dibujos, imágenes en donde los bordes son rasgos prominentes.

Recientemente, Chen *et al.* (2008) proponen un descriptor local basado en teoría de psicología experimental llamada la *Ley de Weber*.

Otras técnicas

Van Gool *et al.* (1996) introducen momentos invariantes generalizados para describir la naturaleza multiespectral de las imágenes. Estos invariantes combinan momentos centrales, y caracterizan la forma y distribución de la intensidad dentro de una región, además son independientes y pueden calcularse para cualquier orden y grado. Sin embargo, mientras mayor sea el grado y el orden estos momentos se hacen más sensibles a distorsiones fotométricas. Estos descriptores son más útiles para imágenes de color, en donde los momentos se pueden calcular para cada canal de manera independiente, o entre canales.

IV.2.2 Método de evaluación

Los descriptores discriminantes construyen una firma única para cada región local, por lo tanto el rendimiento de dichos algoritmos se debe de evaluar con respecto al cálculo de correspondencias locales, como se muestra en la Figuras 14 y 15. Con el propósito de analizar el grado de invarianza que tiene un descriptor, se calculan correspondencias en secuencias de imágenes que muestran algún tipo de transformaciones geométricas o fotométricas; e.g., rotación, cambio de escala, cambio de iluminación, o compresión JPEG. El proceso de evaluación propuesto por Mikolajczyk y Schmid (2005), y de igual manera utilizado en esta tesis, se aplica a escenas que se pueden considerar planas. De tal forma que la imagen de referencia I_1 y las imágenes de prueba I_i en una secuencia $J = \{I_1, I_2, ..., I_i, ..., I_n\}$ de n imágenes están relacionadas por una homografía $H_{1,i}$.

El criterio de evaluación, propuesto en (Ke y Sukthankar, 2004) y además utilizando en (Mikolajczyk y Schmid, 2005), se basa en el número de correspondencias correctas y el número de correspondencias falsas que se obtienen entre dos imágenes. El término *correspondencias correctas* hace referencia al termino en inglés *match*, o bien a las correspondencias calculadas en base a un clasificador. Mientras que el número de *correspondencias reales*, que se conocen a priori porque se conoce la transformación que existe entre las dos imágenes, se refiere el limite máximo de posibles correspondencias correctas que se pueden calcular.

Se dice que se obtiene una correspondencia entre dos regiones \mathbf{A} y \mathbf{B} cuando la distancia entre sus descriptores $D_{\mathbf{A}}$ y $D_{\mathbf{B}}$ esta por debajo de un umbral h. Cada descriptor de la imagen de referencia se compara con cada descriptor de la imagen de prueba transformada, y se calcula el número de correspondencias correctas y el número

de correspondencias falsas que se generan. El comportamiento del descriptor se puede analizar variando el valor de h y así generando una curva *recall* vs 1-precisión. Recall es el número de correspondencias correctas con respecto al número de correspondencias reales que existen entre las dos imágenes, dado por:

$$recall = \frac{\# \ correspondencias \ correctas}{\# \ correspondencias \ reales} \ . \tag{7}$$

El número de correspondencias reales se calcula utilizando el criterio de repetibilidad presentado en las Secciones V y VII (ver ecuaciones 3 y 27). De tal forma que cuando existe una región repetida entre la imagen de referencia y la de prueba entonces se considera como una correspondencia real, peremitiendo un error de traslape de $\epsilon_S <$ 50%, donde el traslape esta dado por,

$$\epsilon_S = 1 - \frac{\mathbf{A} \cap \mathbf{H}_{1,i}{}^T \mathbf{B} \mathbf{H}_{1,i}}{\mathbf{A} \cup \mathbf{H}_{1,i}{}^T \mathbf{B} \mathbf{H}_{1,i}} \,. \tag{8}$$

donde **A** es una región de la imagen I_1 y **B** es de la imagen I_i .

Además, cuando se calcula una correspondencia entre descriptores se puede verificar si esta es verdadera o falsa utilizando el mismo análisis.

El número de correspondencias falsas con respecto al número total de correspondencias que se calcularon se expresa con 1-precisión:

$$1 - precision = \frac{\# \ correspondencias \ falsas}{\# \ correspondencias \ correctas + \# \ correspondencias \ falsas} \ . \tag{9}$$

La Figura 16 muestra dos ejemplos del tipo de gráficas que se generan con el método descrito arriba. Un descriptor ideal obtuviera un recall igual a uno para cualquier precisión. Sin embargo, en la practica el recall aumenta y la precisión disminuye cuando se incrementa h, debido al ruido que introducen las transformaciones entre descriptores de regiones correspondientes. Una curva que en su mayoría presenta una forma horizontal indica que el recall es estable después de un cierto umbral. Por otro lado, una curva que aumenta de manera monótona sugiere que el descriptor es muy susceptible al ruido



Figura 16: Dos curvas típicas Recall vs 1-Precisión.

que genera la transformación de la imagen. En general, una curva que esta por encima de otra sugiere que el descriptor que lo generó es mejor que el otro, tal es el caso de las dos curvas de la Figura 16.

Criterios de correspondencias

Para calcular una correspondencia entre dos descriptores $D_{\mathbf{A}}$ y $D_{\mathbf{B}}$ se necesita definir un criterio que la determine, aquí se consideran tres diferentes:

1. Correspondencia basada en umbral. En este caso, se considera que existe una correspondencia entre dos regiones $\mathbf{A} \neq \mathbf{B}$ si la distancia euclidiana entre ellas es menor que un umbral, $||D_{\mathbf{A}} - D_{\mathbf{B}}|| < t$. Cuando se aplica este criterio, se pueden obtener más de una correspondencia para cada descriptor, y varias de ellas pueden estar correctas debido al error de traslape que se permite. Además, este criterio es el apropiado para analizar la distribución de los descriptores en el

espacio, y posiblemente es el análisis más informativo aunque no sea el óptimo para aplicaciones reales.

- 2. Vecino más cercano. En este caso, existe una correspondencia entre dos regiones **A** y **B** si el descriptor $D_{\mathbf{B}}$ es el vecino más cercano de $D_{\mathbf{A}}$, y además si la distancia entre ellos esta por debajo de un umbral $||D_{\mathbf{A}} - D_{\mathbf{B}}|| < t$. Con este criterio se asegura que para cada descriptor solo puede existir una correspondencia, a lo más, en la imagen de prueba; este criterio es el más practico para aplicaciones reales.
- 3. Razón del vecino más cercano. Este criterio es similar al de vecino más cercano, con la diferencia de que el umbral se aplica a la razón entre la distancias con el vecino más cercano y el segundo vecino más cercano, $||D_{\mathbf{A}} - D_{\mathbf{B}}||/||D_{\mathbf{A}} - D_{\mathbf{C}}|| < t$; donde, $D_{\mathbf{B}}$ es el vecino más cercano a $D_{\mathbf{A}}$ y $D_{\mathbf{C}}$ es el segundo vecino más cercano; con este criterio se mejora el poder discriminatorio del descriptor.

Recall/Precisión - ROC

El análisis *recall* vs *1-precisión* está estrechamente relacionado con el análisis basado en curvas ROC (en inglés, Receiver Operating Characteristic), ver Figura 17 (Davis y Goadrich, 2006).

El término de recall es igual al TPV, pero el término de precisión difiere con respecto a TPF. Específicamente, mientras que la tasa de positivos falsos dependen del número de negativos verdaderos el término de precisión solo depende del número total de positivos calculados.

Ahora bien, debido al problema de correspondencia que aquí se describe se puede ver que el termino NV no es relevante para el análisis, debido a que no se verifica si dos descriptores son ejemplos de correspondencias falsas. Dicho de otra forma, para cada descriptor el valor de NV siempre será muy alto con respecto a PF y por lo tanto un

			$Recall = \frac{PV}{PV + NF}$
			$Precision = \frac{PV}{PV + PF}$
	Positivo real	Negativo real	$TPV = \frac{PV}{PV + NF}$
positivo predicho	PV	PF	$TPF = \frac{PF}{PF}$
positvo predicho	NF	NV	PF + NV

(a) Matriz de confusión.

(b) Medidas de rendimiento.

Figura 17: Una comparación entre el análisis ROC (TPV/TPF) y Recall/Precisión. En donde: PV= positivo verdadero; PF= positivo falso; NF= negativo falso; NV= negativo verdadero; TPV= tasa de positivos verdaderos; TPF= tasa de positivos falsos.

análisis ROC con TPF no será tan informativo.

IV.2.3 El descriptor SIFT y sus derivados

Mikolajczyk y Schmid (2005) además de proponer una metodología para evaluar a los descriptores también realizan una comparación experimental entre once descriptores diferentes. Los resultados de dicha comparación mostraron que el descriptor SIFT, o sus variantes, suelen obtener el rendimiento más alto y estable, confirmando la intuición general dentro de la comunidad de visión. Más aún, ha sido tanto el éxito de SIFT que en los últimos tres o cuatro años las nuevas propuestas en el área suelen ser versiones modificadas, y mejoradas en algún sentido, del descriptor SIFT. A continuación se realiza una descripción básica del descriptor SIFT, y se presentan algunos de los descriptores derivados del mismo.

Scale Invariant Feature Transform: SIFT

SIFT, como su nombre lo indica, combina un detector invariante a escala y un descriptor local basado en distribución (Lowe, 1999). Sin embargo, normalmente cuando se hace referencia al SIFT lo que es de interés es el descriptor, y es este el que ha alcanzado un uso amplio dentro de la comunidad de visión, mientras que el detector normalmente se sustituye por uno con mejor rendimiento.

El descriptor SIFT construye un histograma que caracteriza a la distribución y orientación del gradiente dentro de cada región local, ver Figura 18a-d; el proceso básico es el siguiente. El histograma de SIFT tiene tres dimensiones, dos para la posición (x, y) y uno para la orientación del gradiente, donde la contribución a las casillas de posición y orientación están ponderadas con la magnitud del gradiente en cada pixel. Para construir el histograma, la posición dentro de la región de soporte normalizada y rotada $\overline{\mathbf{A}_s^r}$ se divide con una cuadrícula de tamaño 4×4 , y el ángulo de orientación del gradiente se divide en ocho intervalos $\{0, \frac{\pi}{4}, \frac{\pi}{2}, ..., \frac{7 \cdot \pi}{4}\}$. De tal manera que el descriptor SIFT tiene un total de 128 dimensiones. Finalmente, el descriptor se normaliza para hacerlo invariante con respecto a cambios uniformes de iluminación.

Descriptores derivados de SIFT

Ke y Sukthankar (2004) introdujeron el PCA-SIFT, un descriptor que se construye a partir de los siguientes pasos: (1) calcular un eigen-espacio para representar el gradiente de las regiones locales; (2) dada una región nueva, $\overline{\mathbf{A}_s^r}$, calcular el gradiente local; (3) proyectar el gradiente de la región utilizando el eigen-espacio pre-calculado para derivar el descriptor $D_{\mathbf{A}}$. El gradiente de la región se muestrea en 39 × 39 posiciones dentro de cada región, lo cual produce un descriptor de 3,042 dimensiones. Sin embargo, con PCA estas dimensiones se reducen a solo 36, una descripción mucho más compacta que la de SIFT.



Figura 18: El descriptor SIFT (Mikolajczyk y Schmid, 2005). (a) Región detectada. (b) Imagen del gradiente y cuadricula de posición. (c) Las tres dimensiones del histograma. (d) Cuatro de los ocho planos de orientación. (e) Cuadricula cartesiana y rejilla en coordenadas polares; la segunda es la implementada por el descriptor GLOH (Mikolajczyk y Schmid, 2005).

Mikolajczyk y Schmid (2005) proponen el descriptor histograma posición-orientación del gradiente (en inglés, Gradient Location-Orientation Histogram: GLOH). Esta es una extensión de SIFT diseñado para aumentar la robustez y distintividad del descriptor. En el GLOH se calcula el descriptor SIFT utilizando una división de la región local basada en coordenadas polares, ver Figura 18e. El GLOH utiliza tres casillas en la dirección del radio y ocho en la dirección angular, un total de 17 casillas. La orientación del gradiente utiliza 16 casillas, esto produce un histograma de 272 casillas. Para reducir el tamaño del descriptor se emplea PCA, similar a lo hecho en (Ke y Sukthankar, 2004), y se obtiene un descriptor de 128 dimensiones.

Bay *et al.* (2008) proponen el descriptor *SURF* (en inglés, Speeded-Up Robust Features: SURF), una modificación de SIFT diseñado para aplicaciones en tiempo real. Básicamente, los autores eliminaron cualquier etapa redundante y simplificaron los procesos para solo dejar lo esencial, siempre pensando en velocidad y eficiencia de cálculos; por ejemplo, utilizan imágenes integrales para calcular el gradiente y rasgos tipo Haar para construir el histograma.

Descriptor	Características			
PCA-SIFT (Ke y Sukthankar, 2004)	Análisis PCA sift, descriptor más compacto.			
GLOH (Mikolajczyk y Schmid, 2005)	Coordenadas polares y análisis PCA.			
SURF (Bay <i>et al.</i> , 2008)	Cómputos simplificados y rápidos, tiempo-real.			
HOG (Dalal y Triggs, 2005),	Calculo en celdas de una rejilla densa,			
(Bosch <i>et al.</i> , 2007)	análisis holístico para la detección de peatones.			
C-SIFT (Abdel-Hakim y Farag, 2006)	Utilizan información de color.			
DAISY (Tola et al., 2008)	Veloz, para correspondencias densas			
	en un par estéreo.			
Deformable (Cheng <i>et al.</i> , 2008)	Utiliza múltiples regiones de soporte,			
	y concatena varios descriptores SIFT.			
RDGP (Pérez y Olague, 2008)	Operador para ponderar el histograma,			
	optimizado con GP.			

Tabla III: Descriptores derivados del SIFT.

Dalal y Triggs (2005) proponen el uso de rejillas de histogramas de gradientes orientados como descriptor (en inglés, Histograms of Oriented Gradients: HOG) para la detección de peatones. Estos descriptores se calculan en una rejilla densa de celdas distribuidas uniformemente, utilizan una normalización de contraste, y permiten traslape entre celdas adyacentes de la rejilla para mejorar el rendimiento en la aplicación de alto nivel. Bosch *et al.* (2007) extienden al descriptor HOG, utilizando una representación en pirámide similar a la propuesta en (Lazebnik *et al.*, 2006). En esencia, los dos descriptores están diseñados para aplicar el tipo de información que extrae SIFT a un análisis holístico de la imagen que capture la forma global que presenta la escena.

Abdel-Hakim y Farag (2006) introducen información de color para construir el descriptor SIFT, de tal forma que el descriptor es más robusto con respecto a cambios en el contenido del color dentro de la imagen.

Tola *et al.* (2008) introducen una versión de SIFT diseñado para que se calcule rápidamente, y así poder aplicarlo al problema de correspondencias densas en visión estéreo con vistas muy separadas. El descriptor lo llaman *DAISY*, y la clave para mejorar la velocidad del descriptor esta en que realiza cómputos densos y convoluciones encadenadas para calcular el histograma.

Cheng *et al.* (2008) introducen un descriptor que es más robusto a deformaciones generales de la imagen. La propuesta de los autores se basa en utilizar varias regiones de soporte, de tal forma que para cada región se calcula un descriptor SIFT y la concatenación de todos caracterizan al punto de interés central.

Pérez y Olague (2008) utilizan un GP para modificar una parte del proceso en la construcción del descriptor SIFT. Específicamente, lo que hacen es evolucionar un operador que se utiliza para ponderar el histograma, sustituyendo la magnitud del gradiente que utiliza SIFT. Pruebas experimentales como las descritas en (Mikolajczyk y Schmid, 2005) muestran que el GP sintetiza operadores que permiten un rendimiento mucho mayor para el descriptor híbrido RDGP (en inglés, Region Descriptors with Genetic Programming). La clave de la metodología propuesta en (Pérez y Olague, 2008) fue la definición de un criterio de optimización adecuado que captura numéricamente lo que en (Mikolajczyk y Schmid, 2005) solo se expresaba gráficamente. Este paso le permitió a los autores plantear un problema de optimización y resolverlo con un proceso evolutivo. La Tabla III resume a los descriptores propuestos basados en la metodología establecida en SIFT.

Capítulo V

Diseño automático de operadores que detectan puntos de interés

En este Capítulo se describe el algoritmo GP propuesto para sintetizar, de forma automática, operadores empleados durante el proceso de detección de puntos de interés. El problema se plantea en términos de optimización monoobjetivo utilizando una combinación multiplicativa de objetivos. Se realiza un estudio experimental de los operadores que se obtienen, y se compara su rendimiento con los detectores del estado-del-arte. Además, se analiza las tendencias del algoritmo para determinar en que partes del espacio de soluciones se enfoca la búsqueda genética. Finalmente, para poder realizar un estudio más completo del rendimiento del algoritmo, se utiliza un esquema computacional que permite una ejecución distribuida y paralela de muchas instancias del algoritmo GP.

V.1 La evolución de operadores con Programación Genética

La estrategia de diseño que se toma en esta tesis se enfoca en dos aspectos primordiales: 1) la estructura de los operadores utilizados para la detección de puntos de interés; y 2) la medida de aptitud empleada. La estructura de todos los operadores presentados en la Sección III.2 es el producto de la manera en que cada investigador modeló y abstrajo las características principales que indentificó en el problema. Cada uno es el resultado de un análisis detallado de las propiedades observables del rasgo que se desea detectar; e.g., esquina, mancha, o borde. Por otro lado, también es posible construir a todos esos operadores empleando un conjunto finito de operaciones básicas. Por ejemplo, todos los operadores utilizan por lo menos una operación aritmética, y también utilizan derivadas de la imagen o algun suavizado de tipo Gaussiano. Con respecto a la manera en que se evalúa el rendimiento de un operador, tres criterios ya se discutieron en la Sección III.4. Por lo tanto, una suposición razonable, por ejemplo, es afirmar que un operador *óptimo* deberá ser capaz de obtener una tasa de repetibilidad máxima para cualquier secuencia de imágenes dada. De estas obervaciones generales se propone la siguiente pregunta de investigación:

 ¿ Es posible sintetizar un operador para la detección de puntos de interés que se optimiza para maximizar su rendimiento y se construye a partir de un conjunto de operaciones básicas?

La hipótesis central de esta tesis es que la respuesta a esta pregunta es afirmativa, y el resto de este Capítulo describe un algoritmo GP que lo demuestra experimentalmente.

En este Capítulo, el problema se plantea en términos de optimización monoobjetivo, y solo se consideran dos criterios de evaluación: la repetibilidad y la dispersión espacial de los puntos detectados. Es necesario entender que estos dos criterios no están ligados a ningún concepto semántico en particular. Entonces, el GP tiene la libertad de producir operadores que detecten rasgos que pudieran no corresponder a lo que normalmente se entiende como un punto de interés; i.e., no necesariamente se debe de enfocar en detectar esquinas o vértices. Sin embargo, esto no se debe de tomar como una limitante del enfoque propuesto. Todo lo contrario, se debe de ver como una posibilidad de investigar lo que verdaderamente se debe esperar de un detector de puntos de interés dados los criterios de evaluación que se utilizan.

V.1.1 Espacio de búsqueda

Después de analizar los operadores descritos en la Sección III.2 se puede definir un conjunto de primitivas básicas; esto nos lleva a los siguientes conjuntos de Funciones y Terminales,

$$F = \left\{ +, |+|, -, |-|, |I_{out}|, *, \div, I^2_{out}, \sqrt{I_{out}}, log_2(I_{out}), EQ(I_{out}), k \cdot I_{out} \right\}$$
$$\bigcup \left\{ \frac{\delta}{\delta x} G_{\sigma_D}, \frac{\delta}{\delta y} G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2} \right\} ,$$
$$T = \left\{ I, L_x, L_{xx}, L_{xy}, L_{yy}, L_y \right\} ,$$
(10)

donde I es la imagen de entrada, y I_{out} puede ser cualquier terminal en T, o la salida de una de las funciones en F; EQ(I) es una normalización de histograma; L_u son derivadas Gaussianas en la dirección u de la imagen; G_{σ} son filtros de suavizado Gaussiano; $\frac{\delta}{\delta u}G_{\sigma_D}$ son derivadas de la función de Gauss¹; y la constante k = 0.05. Para la función log_2 se tomo la convención de $log_2(0) = 0$ para evitar operaciones con cantidades infinitas. Por razones similares, cuando se da el caso en que se aplica la raíz a un valor negativo $\sqrt{-x}$, solo se toma la parte real del resultado para evitar operaciones posteriores con números complejos.

Los autores de (Kitchen y Rosenfeld, 1982; Wang y Brady, 1991; Beaudet, 1978) no emplean derivadas Gaussianas; sin embargo, T se define de esta forma porque este tipo de derivadas son menos susceptibles al ruido y tienen propiedades isotrópicas. También es necesario entender que todos los operadores construidos con elementos de $\{F \cup T\}$ en realidad solo tendrán una entrada, la imagen I; la diferencia entre F y T solo es para ayudar al proceso de búsqueda evolutiva incluyendo información que se asume que pudiera ser útil a priori (Lin y Bhanu, 2005). Además, no se supone que el conjunto de primitivas $\{F \cup T\}$ es necesario o suficiente para el problema dado. Sin embargo, los resultados experimentales si ayudan a confirmar la validez de las funciones y terminales que se proponen.

¹Todos los filtros Gaussianos se aplican utilizando la convolución.



Figura 19: El espacio de operadores construidos con primitivas tomadas de $\{F \cup T\}$. Se muestran tres subespacios: a) Ω_{δ} contiene derivadas de la imagen tomadas de T; b) Ω_A contiene operadores que utilizan algún elemento de la matriz A; y c) Ω_{β} que contiene operadores que son funcionalmente equivalentes, en este caso operadores que obtienen una medida de interés relacionada con la curvatura de una superficie.

La Figura 19 es una abstracción del espacio de búsqueda Ω que se utiliza, contiene todos los operadores que se pueden construir con las primitivas en $\{F \cup T\}$. Adicionalmente, se puede conceptualizar un subespacio $\Omega_{\delta} \subset \Omega$ que contiene a todos los operadores que incluyen derivadas L_u tomadas de T. La Figura 19 también muestra un subespacio que contiene operadores que emplean a algún componente de la matriz de autocorrelación Ω_A . Dada la definición en el Capítulo III para A, se puede afirmar que $\Omega_A \subset \Omega_{\delta}$. Tanto Ω_A como Ω_{δ} contienen operadores que se asemejan con respecto a su contenido genético, sin embargo esto no garantiza ninguna semejanza funcional (fenotípica). Otro posible subespacio puede ser $\Omega_{\beta} \subset \Omega$ que contiene a operadores que extraen una medida relacionada con la curvatura de la superficie en la imagen. En este caso, Ω_{β} es una agrupación de operadores que comparten una funcionalidad similar. Estos ejemplos se describen porque una correspondencia uno-a-uno entre estructura y funcionalidad (genotipo y fenotipo) no se puede esperar; esto se conoce como el fenómeno de *convenciones en competencia* bien conocido en la literatura de redes neuronales (Montana y Davis, 1989). El esquema evolutivo propuesto se enfrenta con este fenómeno explícitamente cuando se consideran las aproximaciones de funciones. Por ejemplo, un filtro LoG (Laplaciano-de-Gausianas) se puede aproximar con una operación DoG (Diferencia-de-Gaussianas); estos operadores son muy diferentes genéticamente dentro del esquema propuesto, pero funcionalmente pueden tomarse como equivalentes para algúnas aplicaciones.

V.1.2 Evaluación de aptitud

Dados los criterios que se quieren optimizar, una función de aptitud apropiada debe depender de la tasa de repetibilidad $r_{K,J}(\epsilon)$, medido con un operador K para una secuencia de imágenes J, tal que

$$f(K) \propto r_{K,J}(\epsilon)$$
 . (11)

La medida $r_{K,J}(\epsilon)$ representa el rendimiento que K exhibe cuando se aplica a la secuencia J de imágenes transformadas secuencialmente I_i , con i = 1...N, comenzando de una imagen de referencia I_1 y calculando un valor de repetibilidad $r_{K,I_i}(\epsilon) \forall I_i \in J$ con $i \neq 1$. De esta forma, $r_{K,J}(\epsilon)$ puede ser el mínimo o el valor máximo de $\{r_{K,I_i}\}$. En este trabajo se propone el promedio de repetibilidad para la secuencia J,

$$r_{K,J}(\epsilon) = \frac{1}{N-1} \sum_{2}^{N} r_{K,I_i}(\epsilon) .$$
 (12)

Sin embargo, si $r_{K,J}(\epsilon)$ es el único objetivo que se contempla en f(K) la búsqueda evolutiva se puede perder en máximos no deseados. Por ejemplo, un caso degenerado sería cuando el GP se concentra en explotar individuos que extraen puntos inútiles, todos localizados sobre una porción reducida de la imagen, en este caso la repetibilidad sería alta pero los rasgos probablemente no serian de gran utilidad, ver Figura 20(a). Además, debido a que la imagen de entrenamiento que se emplea contiene textura sobre todo el plano de la imagen, es prudente asumir que un operador *bueno* sería capaz de detectar puntos que estén distribuidos de manera dispersa sobre el plano de la imagen. Obviamente este es un problema multiobjetivo, un objetivo es obtener una tasa de repetibilidad alta, y el otro es maximizar la dispersión de los puntos detectados. Como se analizó en la Sección II.2.2, la interacción entre los criterios debe de definirse antes o después de la optimización. En esta Sección, se elige el primer enfoque, en el Capítulo VI se estudia el segundo.

Por lo tanto, se combinan los dos objetivos de forma multiplicativa en la función de aptitud, tal que

$$f(K) = r_{K,J}(\epsilon) \cdot \phi_x^{\alpha} \cdot \phi_y^{\beta} \cdot N_{\%}^{\gamma} , \qquad (13)$$

donde ϕ_u es un término que promueve una dispersión alta de los puntos detectados. El término ϕ_u se comporta como una función sigmoidal dentro de un rango determinado,

$$\phi_u = \begin{cases} \frac{1}{1 + e^{-a(\mathcal{D}_u - c)}}, & cuando & \mathcal{D}_u < \mathcal{D}_u^{max}, \\ 0 & en \ caso \ contrario \ . \end{cases}$$
(14)

Donde \mathcal{D}_u es la dispersión de los puntos calculado para la dirección u de la imagen de referencia I_1 de la secuencia de entrenamiento J.

Los valores para D_x^{max} y D_y^{max} se definen empíricamente utilizando la imagen de referencia. Para explicar la función ϕ_u un ejemplo ilustrativo es útil. La Figura 20 muestra la curva generada con la parte sigmoidal de ϕ_x^{α} , con los siguientes parámetros: $\alpha = 20, a = 7$ y c = 5. En esta gráfica se identifica la posición de tres operadores sobre la curva, cada uno con una dispersión diferente para la misma cantidad de puntos, o para ser más precisos, diferentes valores de entropía en la distribución de los puntos sobre la dirección x. Por ejemplo, los puntos de interés detectados por el operador de la Figura 20(a) están sobre la misma porción de la imagen, por lo tanto el valor



Figura 20: Como se comporta el término ϕ_x^{α} de la función de aptitud f(K). Muestra el valor que obtiene tres operadores; de izquierda a derecha: primero, un operador que detecta puntos muy amontonados; segundo, K_{IPGP2} con una dispersión buena de puntos; y finalmente, un operador que detecta puntos muy dispersos.

de dispersión es baja $D_x = 5.6$. El siguiente individuo, el operador K_{IPGP2} (Trujillo y Olague, 2006a,b), Figura 20(b), extrae puntos con una repetibilidad alta y también mantiene una dispersión buena de los puntos detectados $D_x = 5.74$. El ultimo individuo, Figura 20(c), es un operador que extrae puntos muy dispersos, pero que tiene un valor de repetibilidad bastante bajo; el individuo se enfoca en extraer puntos que estén muy dispersos. Los resultados experimentales sugieren que operadores estables con una buena repetibilidad tienden a obtener valores de ϕ_x^{α} dentro del recuadro de la Figura 20. Por lo tanto, valores arriba de $D_x^{max} = 5.9$ se umbralizan para no permitir que la evolución se concentre en individuos que explotan el criterio de dispersión; utilizando un análisis similar, la dispersión en la dirección y se umbraliza en $D_y^{max} = 5$.

El ultimo término,

$$N_{\%} = \left(\frac{puntos \ extraidos}{puntos \ deseados}\right)^{\gamma},\tag{15}$$

representa un factor de penalización que reduce la aptitud de detectores que regresan

menos puntos de los deseados. Como el GP es capaz de producir operadores muy diversos, no es posible establecer un valor constante para h en la Ecuación 1 que fuera útil para cualquier operador K. Por ende, se elige detectar un número constante de puntos, *puntos deseados=500*, para todos los operadores, de la siguiente manera. Después de obtener la imagen de interés I_i^* todos los pixeles en I_i^* se ordenan en forma descendiente y h se fija al valor del pixel número 500. Porque la condición en la Ecuación 1 es $K(\mathbf{x}) > h$ y no $K(\mathbf{x}) \ge h$ cabe la posibilidad de que *puntos extraídos < puntos deseados*. Si la diferencia es pequeña entonces la penalización es despreciable. Pero si un K produce una I_i^* en donde todos los pixeles reciben un valor similar, tal como $K(\mathbf{x}) = 0 \ \forall \mathbf{x} \in I_i$, entonces la penalización para K será alta ya que el operador no es capaz de diferenciar entre los pixeles interesantes y los no-interesantes.

V.2 Resultados experimentales

En esta Sección se describe como se implementó el algoritmo GP, las imagenes de entrenamiento y prueba, y los resultados obtenidos.

V.2.1 Implementación

El algoritmo propuesto en este Capítulo se programó utilizando Matlab, con el *toolbox* de programación genética (GPLAB)². Para entrenar al GP se utilizó la secuencia de imágenes Van Gogh, imágenes planas que tienen transformaciones progresivas de rotación, ver Figura 21.

La secuencia Van Gogh tiene una imagen de referencia y 16 imágenes transformadas, N = 17 en Ecuación 12, cada una con una rotación de 11.25° en el sentido de las manecillas del reloj con respecto a la imagen anterior. Sin embargo, debido

²http://gplab.sourceforge.net/index.html, GPLAB A Genetic Programming Toolbox for MATLAB.



Figura 21: Ejemplos de la secuencia de entrenamiento Van Gogh: I_1 , I_3 , y I_7 .

al costo computacional que implica detectar puntos de interés y después calcular la repetibilidad para cada imagen, solo se utiliza la mitad de las imágenes en la secuencia de entrenamiento. Por ende, N = 9 con una rotación de 22.5°, un compromiso entre un entrenamiento riguroso y el costo computacional. La secuencia de entrenamiento, junto con las que se utilizan durante la etapa de prueba, se obtuvieron del repositorio del Visual Geometry Group ³ junto con código de Matlab para calcular la tasa de repetibilidad, y binarios para probar el detector *Harris mejorado* utilizado para comparar. Todos los experimentos se hicieron en una PC con procesador Intel Dual-Core y 2 Gb de RAM con sistema Linux Suse. En la Tabla IV se especifican los parámetros principales del algoritmo GP propuesto.

Los primeros cuatro parámetros, tamaño de la población, generaciones, inicialización, y probabilidades de operadores genéticos, tienen valores canónicos, y fueron ligeramente ajustados experimentalmente. Los siguientes tres ayudan a controlar el tamaño de los individuos, problema conocido como code bloat en la literatura de GP. La profundidad de los árboles se determina de forma dinámica, para esto se cuenta con dos valores máximos de profundidad que un árbol puede tener. La profundidad máxima dinámica es un límite que los individuos solo pueden sobrepasar si su aptitud es igual o mayor a la aptitud del mejor individuo obtenido hasta el momento. Cuando esto

³Visual Geometry Group: http://www.robots.ox.ac.uk/ vgg/research/ .

Tabla IV: Parámetros del GP utilizado en la síntesis de operadores que detectan puntos de interés.

Parámetro	Descripción y valor			
Tamaño de la población	50 individuos.			
Generaciones	50.			
Inicialización	Ramped Half-and-Half.			
Probabilidades de operadores genéticos.	Cruce $p_c = 0.85$; Mutación $p_\mu = 0.15$.			
Profundidad de los árboles	Profundidad Dinámica.			
Profundidad máxima dinámica	5 niveles.			
Profundidad máxima real	7 niveles.			
Selección	Torneo con presión lexicográfica			
Supervivencia	Elitismo.			
Parámetros de la función objetivo (Ecuación 13)	$a_x = 7, c_x = 5.05, a_y = 6, c_y = 4.3,$			
	$\alpha = 20, \beta = 20, \gamma = 2.$			

sucede la *profundidad máxima dinámica* se aumenta al valor de la profundidad de dicho individuo. El siguiente parámetro, *profundidad máxima real* es un límite rígido que no puede ser sobrepasado en ninguna circunstancia. Finalmente, el manejo de la población se hace con una selección de individuos utilizando *torneo con presión lexicográfica*, y una estrategia *elitista* para la supervivencia de individuos.

V.2.2 Operadores sintetizados

Los primeros resultados de alto rendimiento que arrojó este enfoque fueron los siguientes:

$$K_{IPGP1}(\mathbf{x}) = G_{\sigma=2} * (G_{\sigma=1} * I(\mathbf{x}) - I(\mathbf{x})) , \qquad (16)$$

$$K_{IPGP2}(\mathbf{x}) = G_{\sigma=1} * \left[L_{xx}(\mathbf{x}) \cdot L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x}) \right], \qquad (17)$$

donde el acrónimo IPGP significa Interest Point detection with Genetic Programming (Detección de Puntos de Interés con Programción Genética). En la Figura 22 se muestra la imagen de interés que se obtiene con cada operador cuando se aplica a la imagen de referencia en la secuencia de Van Gogh. También se muestran las estadísticas de



Figura 22: Estadísticas durante la evolución de los operadores K_{IPGP1} y K_{IPGP2} , los operadores corresponden a la primera y segunda columna respectivamente. Las columnas de izquierda a derecha: 1) Imagen de interés I^* , 2) Gráficas de aptitud, y 3) Diversidad de la población.

la corrida que generó a cada individuo: la primer gráfica tiene la aptitud del mejor individuo en cada generación, la aptitud promedio y la mediana; la segunda es la diversidad de la población calculada como el porcentaje de individuos que tienen un genotipo único dentro de la población.

En el caso de K_{IPGP1} , las gráficas muestran que fue descubierto en la generación 9, mientras que K_{IPGP2} apareció en la generación 18. Para K_{IPGP1} la aptitud de la población tiende a ser mayor, posiblemente una consecuencia de la diversidad tan baja, o porque este operador esta en una región amplia dentro del espacio genético donde los operadores tienen una aptitud alta. Posteriormente se analizará esta región, denominada $\Omega_1 \subset \Omega$, ya que contiene individuos que son genéticamente similares y que tienen un rendimiento bastante alto. Para el caso de K_{IPGP2} , la aptitud de la población es más inestable a través de las generaciones, una posible razón de esto puede ser que la región en el espacio Ω en donde se encuentra contiene menos óptimos locales. Además, en todas las corridas adicionales que se hicieron del GP, en ninguna se encontró un



Figura 23: La tasa de repetibilidad que obtiene K_{IPGP1} y K_{IPGP2} sobre con la secuencia completa de Van Gogh; comparados con los operadores de (Beaudet, 1978; Kitchen y Rosenfeld, 1982; Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987).

operador que fuera genéticamente similar a K_{IPGP2} .

La expresión matemática de cada operador y la imagen de interés que generan, revelan propiedades interesantes de cada uno. K_{IPGP1} se enfoca en bordes, la operación es básicamente una operación DoG. Mientras que K_{IPGP2} realiza un suavizado del determinante de la matriz Hessiana, una modificación sencilla sobre el operador propuesto por Beaudet (1978). Aunque la mejora descubierta en K_{IPGP2} es sutil, es similar a la propuesta por Schmid *et al.* (2000) para el detector de Harris y Stephens (1988) que también produce una mejora significativa en la tasa de repetibilidad.

La Figura 23 muestra la tasa de repetibilidad que obtiene K_{IPGP1} y K_{IPGP2} sobre la secuencia completa de Van Gogh, junto con la repetibilidad de los operadores propuestos en (Beaudet, 1978; Kitchen y Rosenfeld, 1982; Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987). Para el detector de Harris y Stephens se utilizo la versión *mejorada* de (Schmid *et al.*, 2000). El resto de los operadores se programaron utilizando derivadas Gaussianas. La gráfica muestra que K_{IPGP1} y K_{IPGP2} obtienen un rendimiento mayor a cualquiera de los diseños humanos, excepto el operador *mejorado de Harris* que obtiene un rendimiento similar. Si se consideran los tres detectores evaluados por Schmid *et al.* (2000) que no se incluyen aquí, entonces los operadores que produce el GP demostraron ser significativamente mejores que un total de siete operadores propuestos en la literatura de visión por computadora.

Adicionalmente, el único operador que obtiene un rendimiento similar, el operador *mejorado de Harris*, tiene una estructura más compleja que puede ser un criterio importante cuando se requiere detectar puntos en aplicaciones de tiempo-real, ver Figura 24. Finalmente, en la Figura 25 se muestra una comparación cualitativa entre los operadores K_{IPGP1} , K_{IPGP2} , y K_{Harris} .

Después de obtener los resultados descritos arriba, se realizaron 30 corridas nuevas del algoritmo y se obtuvieron los siguientes resultados. La Tabla V presenta una lista de las 15 soluciones encontradas. De las 30 corridas que se hicieron, solo 18 de ellas generaron una solución que era capaz de obtener una aptitud alta cuando se aplicaba a la secuencia completa de Van Gogh. Además, de esas 18 soluciones tres de ellas eran repetidas; i.e., se obtuvieron en dos corridas diferentes; por ende, la Tabla V solo contiene 15 operadores. Las 12 corridas que no fueron productivas tal vez se pudieran haber evitado si se utilizará la secuencia completa de Van Gogh durante la evaluación de aptitud. La Tabla V muestra el nombre de cada operador, su expresión matemática, la corrida en la cual se obtuvo, el sub-espacio en el que se encuentra, su valor de aptitud, la repetibilidad promedio que alcanza con la secuencia de Van Gogh completa, y si la expresión matemática fue simplificada algebraicamente. La tabla se organiza en tres conjuntos de operadores, cada uno contiene operadores de tres subespacios conceptuales $\Omega_1, \Omega_{\sigma} \setminus \Omega_1 \ge \Omega_{\nabla}$. El subespacio $\Omega_{\sigma} \subset \Omega$ contiene operadores que no utilizan derivadas Gaussianas L_u como parte de su material genético; i.e., solo emplean filtros Gaussianos



Figura 24: El genotipo de los operadores K_{IPGP1} , K_{IPGP2} , y K_{Harris} . En el caso del operador de Harris, este sería el genotipo más sencillo que pudiera tener si se construye con el algoritmo GP.



Figura 25: Comparación cualitativa entre los operadores K_{IPGP1} , K_{IPGP2} , y K_{Harris} . Renglones: Van Gogh, Graph y Monet.

y operaciones aritméticas. El subespacio $\Omega_1 \subset \Omega_{\sigma}$, como se especificó arriba, contiene operadores que son genéticamente similares a K_{IPGP1} . Finalmente, $\Omega_{\nabla} \subset \Omega_{\delta}$ contiene operadores que utilizan de manera explícita el Laplaciano de la imagen, por lo tanto es un subespacio de Ω_{δ} . Se debe de notar que Ω_{∇} y Ω_1 se encuentran en diferentes regiones de Ω , pero al mismo tiempo convergen a operaciones similares en cuanto a su funcionalidad o fenotipo. Adicionalmente, el GP también encuentra varios operadores poco comunes y poco intuitivos.

Nombre	Operador	Corrida	Sub.	Apt.	r_J *	Sim.
$IPGP1^*$	$G_2 * I - G_2 * I ^2$	3, 28	Ω_1	77.91	94.78	no
IPGP3	$G_1 * G_1 * G_1 * G_2 * G_2 * (G_1 * I - I)$	5	Ω_1	83.98	95.9	no
IPGP4	$G_2 * G_2 * G_2 * (G_2 * I - I)$	7	Ω_1	85.98	96.35	no
IPGP5	$G_1 * G_2 * I - G_1 * I ^2$	19	Ω_1	83.86	93.22	no
IPGP6	$G_2 * G_2 * G_1 * \left(\frac{I}{G_2 * I}\right)$	15	Ω_1	78.13	94.84	no
IPGP7	$G_2 * 2 \cdot L_{yy} + 2 \cdot L_{xx} $	4, 9	Ω_{∇}	78.33	94.92	yes
IPGP8	$G_2 * L_{xx} + 2 \cdot G_2 (L_{xx} + L_{yy})^2 $	11	Ω_{∇}	73.86	90.54	yes
IPGP9	$G_2 * G_2 * 2 \cdot L_{yy} + 2 \cdot L_{xx} + L_{xy} $	16	Ω_{∇}	80.3	93.44	no
IPGP10	$G_2 * L_{yy} + L_{xx} $	18, 20	Ω_{∇}	77	92.81	no
IPGP11	$G_1 * \left(\frac{G_1 * I}{(G_1 * G_1 * I)^3} \right)$	21	$\Omega_{\sigma}\setminus\Omega_{1}$	78.23	92.44	yes
IPGP12	$\frac{G_2 * I^{\frac{3}{2}}}{(G_1 * I)^{\frac{9}{4}}}$	22	$\Omega_{\sigma} \setminus \Omega_1$	72.67	91.91	yes
IPGP13	$G_2 * G_2 * [(G_2 * I) \cdot (G_2 * G_1 * I - I)]$	6	$\Omega_{\sigma} \setminus \Omega_1$	85.72	96.37	no
IPGP14	$G_2 * G_2 * [(G_2 * G_2 * G_2 * I) \cdot (G_2 * G_2 * I - I)]$	23	$\Omega_{\sigma} \setminus \Omega_1$	85.94	96.5	no
IPGP15	$\frac{G_2 * [G_2 * G_2 * I - G_1 * I - G_2 * G_2 * I]}{G_2 * G_2 * I}$	24	$\Omega_{\sigma}\setminus\Omega_{1}$	85.81	95.15	no
IPGP16	$\frac{G_2 * G_2 * \left[G_1 * G_2 * I^2 - I^2\right]}{G_2 * G_2 * G_1 * I}$	30	$\Omega_{\sigma} \setminus \Omega_1$	84.63	96.8	no
	\star Para obtener r_J el tamaño del vecindario ${\bf W}$					
	para la supresión de no-máximos se fijo a $n = 3$.					

Tabla V: Operadores sintetizados con GP.

V.2.3 Análisis y discusión

El análisis de los operadores presentados arriba comienza con aquellos que pertenecen al espacio Ω_1 . Primero, se observa que el operador K_{IPGP1^*} es extremadamente sencillo y muy similar a K_{IPGP1} . K_{IPGP3} y K_{IPGP4} son proporcionales a K_{IPGP1} porque aplican el mismo tipo de filtrado, donde $G_{\sigma_1} * I - G_{\sigma_2} * I$ con $\sigma_1 > \sigma_2$. La misma analogía se puede hacer con K_{IPGP1^*} y K_{IPGP5} , porque la única diferencia entre estos operadores es la cantidad de difuminado que utilizan. Las operaciones adicionales de suavizado o difuminado están relacionadas con el concepto de espacio escala (ver Capítulo VII), donde estructuras prominentes más grandes se detectan en escalas de difuminado mayores (Lindeberg, 1998). Por ende, es evidente que el GP se enfoca en detectar las estructuras más grandes dentro de las imágenes, rasgos que tienden a ser más estables.
Los operadores que son proporcionales a K_{IPGP1} detectan rasgos con una intensidad relativa alta; mientras que el inverso aditivo de K_{IPGP1} detecta rasgos en regiones de intensidad baja. Es interesante que la estructura básica de K_{IPGP1} se descubrió varias veces por el GP, mientras que su inverso aditivo, denominado K_{-IPGP1} , nunca apareció en alguna de las corridas. Esto probablemente esta relacionado con las características particulares de la imagen, y no se debe a un sesgo significativo por parte del algoritmo propuesto. Por otro lado, K_{IPGP1*} , con el uso del valor absoluto, obtiene máximos relacionados con K_{IPGP1} y con su inverso aditivo.

Una propiedad interesante de la mayoría de los operadores en Ω_1 es que utilizan un filtro DoG.

Proposición 1: Los operadores K_{IPGP1} y K_{IPGP1*} son proporcionales a un filtro DoG (Difference-off-Gaussian), si se asume que la imagen I proviene de una imagen desconocida \hat{I} que fue suavizada por un filtro Gaussiano con desviación estándar desconocido $\hat{\sigma}$ tal que $I = G_{\hat{\sigma}} * \hat{I}$, y

$$G_{\sigma} * I - I = G_{\sigma} * G_{\hat{\sigma}} * \hat{I} - G_{\hat{\sigma}} * \hat{I} \propto G_{\sigma+\hat{\sigma}} * \hat{I} - G_{\hat{\sigma}} * \hat{I} = (G_{\sigma+\hat{\sigma}} - G_{\hat{\sigma}}) * \hat{I} .$$
(18)

Dada la Proposición 1, K_{IPGP1^*} , K_{IPGP3} , K_{IPGP4} , y K_{IPGP5} , están relacionados porque usan un filtro DoG, no solo por su semejanza genética con K_{IPGP1} . El último operador en Ω_1 es K_{IPGP6} , la diferencia principal con K_{IPGP1} es que en lugar de utilizar una diferencia utiliza la razón entre dos valores. Si se considera que la convolución Gaussiana es un filtro que genera un promedio ponderado alrededor de cada punto, entonces este operador resalta aquellos puntos que son más intensos que el resto de los pixeles en su vecindario local. Por lo tanto, este operador se asemeja al método de detección propuesto en (Lepetit y Fua, 2006). De tal forma que K_{IPGP6} es un vecino muy cercano a K_{IPGP1} con respecto a su contenido genético, pero funcionalmente es diferente al resto de los operadores en el subespacio Ω_1 .



Figura 26: Estadísticas para la evolución de los operadores en la Tabla V. Las columnas: 1) aptitud (mediana, promedio, y mejor), 2) diversidad, 3) complejidad (*Profundidad máxima dinámica*, la profundidad del mejor individuo, y el número de nodos del mejor individuo). Renglones: *IPGP*1^{*}, *IPGP*3, *IPGP*4, *IPGP*5, y *IPGP*6.



Figura 27: Estadísticas para la evolución de los operadores en la Tabla V, por renglón: IPGP7, IPGP8, IPGP9, IPGP10, y IPGP11.



Figura 28: Estadísticas para la evolución de los operadores en la Tabla V, por renglón: IPGP12, IPGP13, IPGP14, IPGP15, y IPGP16.

El siguiente grupo de operadores en la Tabla V corresponden con el espacio Ω_{∇} . Los operadores K_{IPGP7} y K_{IPGP10} realizan una operación similar, solo hay una diferencia de escala, por lo que se puede decir que el GP convergió a esta estructura un total de cuatro veces. Al parecer, dadas las primitivas que se definieron, al GP se le facilita construir operadores sencillos utilizando derivadas Gaussianas, y estos exhiben un rendimiento bastante alto. Más aún, K_{IPGP8} y K_{IPGP9} realizan operaciones más complejas, pero siguen utilizando el Laplaciano como parte fundamental de sus respectivas medidas de interés. Por ejemplo, K_{IPGP9} es la suma de todas las derivadas de segundo orden, y K_{IPGP8} emplea el cuadrado del Laplaciano. Ahora bien, si el filtro DoG se considera como una aproximación del Laplaciano, entonces se puede argumentar que Ω_{∇} y Ω_1 contienen operadores funcionalmente similares, o tal vez equivalentes.

Finalmente, la Tabla V presenta seis operadores del subespacio $\Omega_{\sigma} \setminus \Omega_1$; estos operadores utilizan variaciones interesantes del filtro DoG; por ejemplo, K_{IPGP12} , K_{IPGP14} y K_{IPGP16} aplican este filtro al cuadrado de la imagen. La operación de la división también la utilizan tres de los operadores en este conjunto, mientras que de los 18 individuos sintetizados solo 4 utilizaron esta primitiva. La razón entre dos medidas parece ser más útil cuando no se involucran derivadas en la operación.

Las Figuras 26, 27 y 28 muestran las estadísticas relevantes del GP durante el proceso de evolución que generó a cada uno de los operadores en la Tabla V, ordenados de la misma forma renglón por renglón. Las columnas, de izquierda a derecha, muestran las estadísticas de aptitud, diversidad de la población y complejidad de la mejor solución encontrada. Las gráficas de aptitud incluyen la mediana de la población, el promedio, y la aptitud del mejor individuo. La diversidad se presenta con respecto a cada generación, y la complejidad muestra el valor del parámetro de *profundidad máxima dinámica*, junto con la profundidad y el número de nodos del mejor individuo (los primeros dos valores se multiplican por 10 para facilitar la lectura de las gráficas). Para los operadores que

aparecieron en más de una corrida, solo se presentan las estadísticas de una de ellas.

La mayoría de los operadores se descubrieron en las primeras generaciones, antes de la generación 25, con K_{IPGP5} , K_{IPGP7} , K_{IPGP12} K_{IPGP14} , K_{IPGP15} , y K_{IPGP16} las excepciones. Los operadores de $\Omega_{\sigma} \setminus \Omega_1$ suelen aparecer más tarde que otros, probablemente porque las medidas complejas que utilizan requieren de una mayor exploración para poder converger. Más aún, las gráficas denotan que las corridas que convergieron a $\Omega_{\sigma} \setminus \Omega_1$ tienen una aptitud mediana y promedio menor. Lo opuesto se ve en las poblaciones que convergieron a Ω_{∇} , porque las derivadas permiten a los individuos identificar, de forma directa, las propiedades de variación alta que se espera de un punto de interés. Las gráficas de diversidad en todos los casos están relacionados con la complejidad de la solución que se produjo. Mientras más pequeño el el mejor individuo de la población, en cuanto a número de nodos, entonces la población tiende a ser menos diversa; mientras que en el caso opuesto la población es más diversa. Las gráficas de complejidad también exponen los beneficios de la *profundidad dinámica* empleada. En las primeras generaciones, de todas las corridas, primero se exploran estructuras sencillas para los operadores, la complejidad de las soluciones aumenta de forma paulatina una vez que la aptitud del mejor individuo aumenta.

De todas las corridas que se realizaron, también se puede analizar cuales fueron las primitivas más comúnmente utilizadas por las soluciones que se obtuvieron. Para este análisis se consideraron las 30 corridas del algoritmo, y a los operadores K_{IPGP1} y K_{IPGP2} ; la Figura 29 expone la frecuencia de cada primitiva.

De las 21 primitivas en $\{F \cup T\}$, varias nunca se utilizaron, como $|+|, k \cdot I_{out}, EQ(I)$ y $\frac{\delta}{\delta u}G_{\sigma_D}$ en el mejor individuo. Un resultado notable es la nula utilidad que tuvieron las derivadas de primer orden L_u , ya que estas las utilizan todos los operadores que se basan en la matriz de autocorrelación. Más aún, en ningún caso se reprodujo una



Figura 29: La frecuencia con que aparecen cada una de las primitivas de $\{F \cup T\}$ en la estructura del mejor individuo que se obtiene en cada corrida del algoritmo.

operación que presentara elementos de la matriz de autocorrelación, un resultado inesperado debido a que existen autores que la consideran una de las herramientas más útiles, o incluso necesarias, para la detección de puntos de interés (Schmid *et al.*, 2000; Kenney *et al.*, 2005).

De la Tabla V, se seleccionaron siete operadores para realizar más pruebas, junto con los operadores K_{IPGP1} , K_{IPGP2} , y el Harris mejorado. Los operadores elegidos fueron: K_{IPGP1^*} , por su simplicidad y estructura complementaria a la de K_{IPGP1} ; K_{IPGP6} porque realiza una operación única entre aquellos que pertenecen a Ω_1 ; K_{IPGP7} representa a los cuatro operadores que emplean el Laplaciano de forma explícita en Ω_{∇} ; K_{IPGP9} , porque contiene la suma de las derivadas de segundo orden, asemejándose así a los componentes del local jet; y K_{IPGP14} , K_{IPGP15} , y K_{IPGP16} porque realizan operaciones que poco ortodoxas.

En la Figura 30 se presentan cuatro secuencias de imágenes para probar el rendimiento de los operadores. Las secuencias Mars y New York tienen transformaciones de rotación progresiva, y las secuencias Graph Y Mosaic tienen cambios de iluminación uniforme. Si consideramos que el GP solo evalúa la aptitud en base a una secuencia de imágenes rotadas, no hay ningún motivo a priori para esperar que los operadores evolucionados mantengan un rendimiento alto cuando se presenta otro tipo de transformación. Además, la secuencia de Van Gogh presenta imágenes con mucha textura, mientras que New York y Graph tienen una estructura más geométrica.

La Figura 31 muestra las gráficas de rendimiento que los operadores arrojan en estas pruebas. Primero, para la secuencia de Van Gogh, como se espera, todos los operadores obtienen un rendimiento alto (es necesario observar la escala de estas gráficas). Segundo, para la secuencia Mars los resultados son similares, algo esperado ya que es similar a la secuencia de entrenamiento. Solamente el operador K_{IPGP15} es notablemente peor que los demás, mientras que K_{IPGP1*} y K_{IPGP9} están un poco más abajo del promedio. Para la secuencia de New York se obtienen resultados positivos, todos los operadores alcanzan un rendimiento alto, con K_{IPGP1*} y K_{IPGP9} de nuevo un poco peor que el promedio.

Para las secuencias que presentan cambios de iluminación, el proceso de detección tuvo que modificarse un poco. Específicamente, un número fijo de puntos no se puede extraer debido a que el cambio de intensidad elimina máximos de la respuesta al operador. Por lo tanto, el número de puntos se fija a la misma cantidad que extrae el detector *Harris mejorado*. Utilizando este criterio, todos los operadores exhiben un rendimiento alto para las imágenes con cambio de iluminación, y el detector K_{IPGP2} es notablemente mejor en las dos secuencias. Esto sugiere que el GP fue capaz de producir operadores que no estaban *sobre-entrenados* a una secuencia en particular, o a un solo



Figura 30: La imagen de referencia (derecha) y una imagen transformada (izquierda) para cada secuencia de prueba. a) Mars (N = 18, rotación), b) New York (N = 35, rotación), c) Graph (N = 12, cambio de iluminación), d) Mosaic (N = 10, cambio de iluminación).

tipo de transformación.

Una posible crítica que se puede hacer a los resultados presentados hasta el momento es que tal vez el rendimiento de los operadores evolucionados solo se debe a que resaltan el mismo tipo de información que detecta el operador de Harris. En otras palabras, la información que identifican es la misma, algo que limitaría el interés en utilizar los detectores evolucionados. Sin embargo, una comparación cualitativa comprueba que los operadores son capaces de detectar rasgos diferentes a los que detecta Harris, y aún así obtener un rendimiento comparable, ver Figura 32.

Finalmente, la Figura 33 presenta el tipo de imágenes de interés que se obtienen cuando se aplica cada uno de los operadores a una imagen de prueba.



Figura 31: La tasa de repetibilidad calculado para cada imagen en la secuencia de entrenamiento y las cuatro secuencias de prueba.

V.2.4 Resultados competitivos con diseños hechos por humanos

La meta principal en el área de aprendizaje de máquina, y el campo de inteligencia artificial en general, es desarrollar sistemas autónomos que sean capaces de aprender y adaptarse al medio, o a una problemática, de manera semejante a como lo hace un humano. En otras palabras *la meta es hacer que las máquinas exhiban comportamientos, que si fueran realizados por humanos, donde se asume que involucran el uso de la inteligencia* (Samuel, 1983).

¿Pero, qué significa que una computadora genere un comportamiento, o una solución a un problema, que es comparable, o competitivo, con lo que hizo o pude hacer un humano? En base a esta pregunta, Koza *et al.* (2000) definen ocho criterios que pudieran



Figura 32: Una comparación cualitativa. Primer renglón, imagen de New York con 400 puntos. Segundo renglón, imagen Mosaic con 200 puntos.

servir de guía; estos se enumeran en la Tabla VI.

Dados los criterios de la Tabla VI, se puede argumentar que los resultados presentados en la sección anterior cumplen con varios de los criterios (Poli *et al.*, 2008).



Figura 33: Imágenes de interés que producen cada uno de los operadores incluidos en la comparación.

Argumentos para cada uno de los criterios con los que cumplen los resultados se muestran en la Tabla VII.

V.3 Búsqueda distribuida y paralela de operadores

En esta sección se presenta un estudio a mayor escala del algoritmo GP propuesto en este Capítulo, con el propósito de analizar las tendencias de convergencia que este

Tabla VI: Criterios para determinar si una solución diseñada automáticamente por un proceso de aprendizaje de máquina es competitiva con soluciones propuestas por humanos.

Criterio	Descripción
(A)	El resultado fue patentado como un invento en el pasado,
	mejora una patente previa, o califica como un
	invento que se puede patentar en el presente.
(B)	El resultado es igual o mejor que un resultado que fue
	aceptado como un descubrimiento científico en algún
	momento, y fue publicado en una revista arbitrada.
(C)	El resultado es igual o mejor que un resultado que fue
	colocado en una base de datos o archivo de resultados
	que un grupo internacional de expertos mantiene.
(D)	El resultado se puede publicar por si solo como un
	resultado científico nuevo, independientemente de que
	fue creado de forma automática por una computadora.
(E)	El resultado es igual o mejor que la última solución
	creada por un humano para un problema bien conocido,
	que ha sido abordado durante mucho tiempo por
	investigadores en el área.
(F)	El resultado es igual o mejor a una solución que fue
	considerada un logro dentro de su área en el momento
	en que se introdujo.
(G)	El resultado resuelve un problema de una dificultad
	indiscutible en su área.
(H)	El resultado exhibe un buen rendimiento, o inclusive gana,
	una competencia en donde participan humanos o soluciones
	diseñadas por humanos.

exhibe. Se utiliza un enfoque de computación GRID distribuida, donde una red local de computadoras ejecutan versiones aisladas del algoritmo de manera paralela y coordinada. Esto se logra utilizando un sistema basado en la infraestructura BOINC e imágenes virtuales VMWare del sistema de software en el cual se ejecuta el GP. Estas tecnologías se conocen como Computación GRID Voluntaria (en inglés, Volunteer GRID Computing: VGC), y tienen como meta establecer una infraestructura que permita explotar al máximo los recursos computacionales de PCs de escritorio en un

Tabla VII: Criterios de competitividad con humanos con los que cumplen las soluciones generadas por el GP.

Criterio	Argumentos
(B)	Los resultados son mejores o iguales que lo publicado
	por (Kitchen y Rosenfeld, 1982; Förstner, 1986),
	(Wang y Brady, 1995; Schmid <i>et al.</i> , 2000).
(C)	Las pruebas para evaluar los operadores, y las secuencias
	de imágenes están disponibles en el sitio del Visual
	Geometry Group: http://www.robots.ox.ac.uk/ vgg/research/.
	Investigadores de las siguientes instituciones participan:
	INRIA, Oxford, Universiteit Leuven, y Czech Technical University.
(D)	Resultados preliminares de este trabajo se publicaron en dos
	de los congresos más importantes en visión por computadora:
	International Conference on Pattern Recognition
	(Trujillo y Olague, 2006b).
(E)	Los resultados son iguales o mejores que el detector más reciente
	y más común en aplicaciones de visión (Schmid <i>et al.</i> , 2000).
(F)	Los resultados son mejores que al menos siete operadores
	propuestos en la literatura (ver Sección V.2.2).
(G)	Los argumentos para este punto se presentan en el Capítulo III.

entorno parcialmente controlado. El enfoque permite un ahorro significativo de tiempo al ejecutar versiones paralelas del algoritmo, esto es especialmente importante para sistemas con una complejidad de tiempo alta. Además, el uso de máquinas virtuales elimina la necesidad de portar el código del algoritmo para poder explotar los beneficios del cómputo distribuido. Una explicación detallada del método se describe en el Apéndice A; mientras que el resto de esta sección se centra en detallar los experimentos realizados. Finalmente, los resultados se analizan en torno a dos perspectivas: primero, referente al desempeño del esquema BOINV/VMWare; y el segundo, referente a las tendencias que exhibe el algoritmo GP.

V.3.1 Configuración experimental

El algoritmo GP es un candidato ideal para el sistema BOINC/VMware porque:

- Una parte del código esta escrito en Matlab, y el trabajo de portar todo el código no es trivial. Además, el algoritmo contiene dependencias con librerías de Matlab.
- El algoritmo es lento, aproximadamente un día de ejecución para generar una solución, por lo que hacer experimentos en un ambiente paralelo y distribuido obviamente es mejor que realizarlos de manera secuencial.

Dos laboratorios de la Universidad de Extremadura, España, en el Centro Universitario de Mérida, fueron utilizados para probar el sistema distribuido. El sistema BOINC/VMware/GP se instaló en diez máquinas MS Windows en uno de los laboratorios, y en diez máquinas GNU/Linux con la distribución Ubuntu en el otro laboratorio.

Cada PC de escritorio contiene una instalación de VMware Player, además del cliente BOINC (ver Figura 115 del Apéndice A). En el servidor BOINC se dió de alta un proyecto nuevo, en este caso el algoritmo GP para puntos de interés. Para crear la imagen VMware se utilizó el *VMware Server*, dicha imagen contiene el sistema operativo Ubuntu, una versión de Debian, y cada imagen estaba configurada para correr el algoritmo GP un total de cinco veces, o bien, para generar 5 soluciones.

Por otro lado, el tamaño de la imagen VM es algo que también se debe de tener en cuenta. Por esta razón, Matlab y Ubuntu se redujeron al tamaño mínimo requerido para correr la aplicación.

V.3.2 Resultados preliminares

Los experimentos que se describen en esta sección tiene como meta exponer y estudiar dos aspectos del algoritmo GP propuesto este capítulo.

- 1. Analizar el efecto que se produce cuando se utilizan todas las imágenes de la secuencia Van Gogh, un total de 17, durante el proceso de entrenamiento. En los resultados experimentales de la Sección V.2 se observa que existen ocasiones en que el GP no es capaz de encontrar operadores robustos; i.e., no son capaces de obtener un rendimiento alto cuando se prueban con toda la secuencia de entrenamiento. Hacer esto es más factible porque se utiliza el ambiente distribuido BOINC/VMWare, donde se obtiene una ejecución paralela de muchas instancias del algoritmo.
- 2. Analizar el efecto que se produce cuando se reduce el espacio de búsqueda que se le presenta al GP. En particular, se busca eliminar algunas de las primitivas que fueron menos utilizadas por el GP en las soluciones presentadas hasta el momento, esto se muestra en la Figura 29.

Para estudiar los puntos mencionados arriba, se utilizan la configuración experimental descrita en la Tabla VIII. Cabe mencionar que debido al nivel relativamente bajo de control laxo que se puede tener con el BOINC/VMWare, existieron instancias en las que los clientes no pudieron terminar la ejecución de la tarea asignada y por lo tanto no generaron una solución. Debido a esto, el número de soluciones obtenidas no son igual al número que en un principio se deseaba obtener. Sin embargo, consideramos que los resultados son ilustrativos de la tendencia general que el GP exhibe.

La Figura 34a contiene las estadísticas con respecto a la aptitud del mejor individuo, y la aptitud promedio de cada población; estos resultados se presentan como promedios con desviación estándar calculados en base al número de soluciones generadas. La Figura 34b presenta la frecuencia con la que aparecen cada una de las primitivas en el mejor individuo de cada ejecución del algoritmo. Como se puede ver, el GP prefiere la operación del Laplaciano; por lo tanto, este operador se puede considerar como un

	Configuración experimental	
Población	75	
Generaciones	75	
Funciones	Un conjunto reducido	
	de funciones.	
	$F_A = \left\{+, -, I_{out} , *, \div, I_{out}^2, \sqrt{log_2}\right\}$	
	$\bigcup \left\{ \frac{\delta}{\delta x} G_{\sigma_D}, \frac{\delta}{\delta y} G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2} \right\}$	
Terminales	Solo utiliza derivadas de	
	primero y segundo orden.	
	$T_A = \{L_x, L_{xx}, L_{xy}, L_{yy}, L_y\}$	
Soluciones	18	
Características	El GP siempre converge al	
	Laplaciano, con una excepción.	

Tabla VIII: Configuraciones experimentales en el ambiente BOINC/VMware.



Figura 34: Estadísticas referentes a la mejor aptitud y la aptitud promedio de la población; se muestran promedios y desviación estándar con respecto al número de soluciones que se obtuvieron en cada experimento.

óptimo local para el espacio de búsqueda reducido.



Figura 35: El operador K_{IPGP17} aplicado a la secuencia Van Gogh.

De las 18 soluciones encontradas, solo un operador fue diferente, el cual denominamos K_{IPGP17} , dado por,

$$K_{IPGP17}(\mathbf{x}) = G_{\sigma=1} * \sqrt{G_{\sigma=2} * \frac{1}{I(\mathbf{x})}}$$
 (19)

En la Figura 35a-b se muestran puntos detectados con K_{IPGP17} en dos imágenes de la secuencia de entrenamiento Van Gogh. Además, la Figura 35c muestra el comportamiento de la repetibilidad de este operador con respecto a toda la secuencia Van Gogh. Obviamente este operador es tan efectivo como cualquiera de los presentados en la Tabla V.

V.3.3 Rendimiento del esquema BOINC/VMware

Es necesario determinar si en efecto el VGC con BOINC/VMware facilitó la ejecución de los experimentos que se describen arriba. Para esto, analizamos un periodo experimental de 48 hrs (un fin de semana), en donde diez computadoras MS Windows generaron 12 soluciones, con un promedio de 18 hrs por cada solución. Se debe aclarar, que de las diez máquinas que estuvieron activas, no todas terminaron la ejecución de sus tareas,

	T_{seq}	$T_{B/VM}$	Speed-up.	Poder de Cómputo
75 Gen, 75 Ind.	215h	48h	4.48	25.67 GFLOPS

Tabla IX: Tiempo de ejcución sequencial y BOINC/VMware/GP

algunas fueron apagadas o modificadas por usuarios del laboratorio, acontecimientos comunes en un sistema VGC.

El tiempo completo que se hubiera requerido de manera secuencial son 215 hrs, un factor de *speed-up* de 4.48. El poder de cómputo total que se alcanzo fue de 25.76 GFLOPS; estos resultados estan resumidos en la Tabla IX.

V.4 Resumen y conclusiones

En este capítulo se introduce un algoritmo de GP que sintetiza, de manera automática, operadores de imágenes que detectan puntos de interés. El problema de optimización monoobjetivo que se plantea, promueve la síntesis de operadores que son estables y que detecten putos dispersos sobre el plano de la imagen. En general, se puede decir que la búsqueda GP encuentra operadores basados en filtrado DoG, o que emplean el Laplaciano de forma explícita. La tendencia fue más obvia cuando se redujo el espacio de búsqueda y se realizaron varias corridas del algoritmo en paralelo utilizando un ambiente VGC. Sin embargo, también se encontraron operadores menos intuitivos, que detectan rasgos visuales diferentes a los que detectan los operadores que son comúnmente utilizados en visión por computadora. La propuesta descrita en este capítulo, además, genera soluciones que exhiben características que las hacen competitivas con las soluciones diseñadas a mano por un humano, una cualidad importante para cualquier sistema de inteligencia artificial.

Capítulo VI

Un enfoque multiobjetivo para la detección de puntos de interés

En este capítulo se presenta un análisis multiobjetivo del problema de detección de puntos de interés. Con el motivo de estudiar la relación que existe entre cada uno de los criterios de evaluación, se introduce un algoritmo de programación genética multiobjetivo que incorpora varios objetivos de manera explícita durante el proceso de síntesis de nuevos operadores. Además, el capítulo incluye muchos resultados experimentales, entre ellos nuevos operadores que dominan en el sentido Pareto a todos los operadores propuestos previamente en la literatura de visión, y se discuten las implicaciones que estos resultados tienen sobre el tema. El enfoque propuesto no es común en el área de visión, por ende se proponen varias líneas de investigación como parte del trabajo futuro relacionado con esta tesis.

VI.1 Objetivos de optimización

En las Secciones III.3 y III.4 se describen, respectivamente, las propiedades que se desean observar en un detector de puntos de interés y tres medidas numéricas para cuantificarlas. Los detalles de estos criterios se pueden revisar en dichas secciones de esta tesis; sin embargo, aquí vamos a recordar solo las ideas básicas que sirven de fundamento para la propuesta que se hace en este capítulo.

VI.1.1 Propiedades básicas para evaluar los detectores de puntos interés

- Estabilidad con respecto a cierto tipo de transformaciones: traslación, rotación y cambios de iluminación. Para medir la estabilidad se utiliza el criterio de *repetibilidad* definido en (Schmid *et al.*, 2000).
- Dispersión de los puntos detectados. Lo que se espera es que en la mayoría de las escenas los puntos de interés estarán dispersos sobre el plano de la imagen.
- La información contenida en el conjunto de puntos detectados. Cuando el contenido de información es alto se supone que los puntos detectados son distintivos y prominentes.

VI.1.2 Medidas de evaluación para un detector de puntos de interés

Dadas las propiedades mencionadas arriba, se definen tres medidas de rendimiento correspondientes:

- 1. Tasa de repetibilidad.
- 2. Dispersión de puntos detectados.
- 3. Contenido de información.

Las primeras dos medidas se definen claramente en las Secciones III.4.1 y III.4.2, y en la Seccion V.1 se emplean para sintetizar operadores que detectan puntos de interés utilizando una planteamiento mono-objetivo del problema.

Sin embargo, la medida del contenido de información solo es parcialmente definida en la Sección III.4.3. Esta medida depende de la manera en la cual se extrae información relevante del vecindario local alrededor de cada punto; o bien, la manera en que el vecindario de cada punto es descrito. Además, también es necesario definir cómo se estima la entropía de la información que se extrae.

Por lo tanto, antes de proponer un problema de optimización multiobjetivo, primero se presentan medidas prácticas para cuantificar el contenido de información que provee un conjunto de puntos de interés.

VI.2 Medidas para el contenido de información

Para cada punto de interés \mathbf{x} que se detecta, se calcula un descriptor local γ . De tal forma que si se considera que un detector identifica un conjunto X de n puntos de interés, entonces existe un conjunto correspondiente de descriptores Γ , donde $\forall \mathbf{x} \in$ $X \exists \gamma \in \Gamma$. Si Υ representa el espacio de descriptores, cuando los descriptores en Γ estén amontonados en regiones aisladas de Υ entonces se dice que el conjunto Xtransmite poca información \mathcal{I} , mientras que lo opuesto es verdad en el caso contrario; una medida para la dispersión de los descriptores se obtiene utilizando el concepto de entropía.

Dada esta definición, será necesario elegir un descriptor local para calcular \mathcal{I} , en esta tesis se realizan pruebas con dos descriptores para analizar la información que extraen los detectores: el descriptor SIFT y el descriptor Hölder (ver (Lowe, 1999) y los Capítulos IV y 76 de esta tesis).

Ahora bien, para estimar el valor de entropía es posible utilizar un análisis discreto del histograma con la entropía de Shannon, o un modelado del espacio continuo y la entropía diferencial; aquí proponemos utilizar el primero.

VI.2.1 Contenido de información con la entropía de Shannon

El contenido de información se puede estimar utilizando una partición discreta del espacio, en tal caso la dispersión de los descriptores extraídos se puede medir utilizando la entropía de Shannon (Shannon, 1950).

Para esto, retomamos la definición dada dada en la Sección III.4.3 para el concepto de contenido de información. Si se considera una partición $\Upsilon = {\Upsilon_j}$, y la probabilidad p_j se aproxima con un histograma en el espacio de los descriptores $\gamma \in \Upsilon_j$ del conjunto Γ , entonces la información contenida por el conjunto X de puntos detectados en I esta dada por

$$\mathcal{I}(\Gamma) = -\sum p_j \cdot log_2(p_j) \;.$$

Ahora bien, es necesario elegir un descriptor local que se utilizará para caracterizar el vecindario alrededor de cada punto. Para esto, se propone utilizar dos descriptores: (1) El descriptor SIFT, el más ampliamente utilizado por la comunidad de visión; y (2) Un descriptor basado en un análisis de regularidad local, descrito con más detalle en el Capítulo IX de esta tesis.

Descriptor SIFT

Es necesario contemplar las características particulares del descriptor SIFT para poder calcular el contenido de información como se define arriba¹. Por ejemplo, dadas las dimensiones del descriptor SIFT, se requiere calcular un histograma en un espacio de 128 dimensiones. Para evitar esto, y simplificar así los cálculos, cada dimensión del descriptor SIFT se considera independientemente y se calcula la media de la entropía \mathcal{I}_{μ} que se obtiene en cada una de las 128 dimensiones. Para esto, los valores de cada componente del descriptor se normalizan en un rango de [0, 1], y cada dimensión se

 $^{^1}$ una descripción completa de este algoritmo se presenta en la Sección IV.2.3

divide en 40 casillas para calcular el valore de entropía correspondiente. Entonces, el contenido de información que se calcula lo denotamos con $\mathcal{I}_{SIFT_{\mu}}$.

Descriptor Hölder: Basado en la regularidad local de la imagen

Además de utilizar el descriptor SIFT, también se calcula el contenido de información utilizando un descriptor basado en la regularidad de la sñal alrededor de cada punto de interés. Este descriptor se introduce de manera completa en el Capítulo IX y en (Trujillo *et al.*, 2007b); por lo tanto, aquí solo hacemos una descripción rápida del mismo 2 .

El descriptor se construye utilizando el exponente Hölder puntual, una medida que caracteriza la regularidad de la señal 2D en cada pixel de una imagen. El método utilizado para estimar dicho exponente se basa en un análisis de las oscilaciones que contieen la señal dentro de un vecindario local alrededor de cada punto. Una manera de entender el exponente Hölder, es como una medida de la diferenciabilidad de una señal alrededor de cada punto. En otras palabras, mientras más regular sea una señal también sera más diferenciable, mientras que si la señal es altamente irregular en un punto, con oscilaciones discontinuas, entonces la señal también es menos diferenciable en dicho punto.

La idea básica es muestrear el valor del exponente Hölder en 32 puntos concéntricos, distribuidos de manera uniforme, alrededor de cada punto de interés, a cuatro distancias diferentes. Esto nos da un total de 129 valores cuando se incluye el exponente Hölder que le corresponde al mismo punto de interés; la concatenación de estos valores conforman lo que denominamos como el descriptor local de Hölder.

Entonces, cada dimensión del descriptor Hölder se considera independientemente,

 $^{^{2}}$ En el Capítulo IX se define la teoría básica sobre este concepto, y se describen los métodos tradicionales empleados para estimar la regularidad. Además, la construcción de este descriptor local se describe con mayor detalle y se discuten los resultados experimentales que validan su calidad.

igual que para el descriptor SIFT, y se calcula la media de la entropía \mathcal{I}_{μ} que se obtiene en cada una de las 129 dimensiones. El contenido de información lo denotamos con $\mathcal{I}_{HOLDER_{\mu}}$.

VI.3 GP multiobjetivo para la síntesis de operadores que detectan puntos de interés

En el Capítulo V se presenta un enfoque que permite sintetizar, de manera automática, operadores que detectan puntos de interés, combinando dos criterios de evaluación en una función objetivo. Aunque los resultados que se obtuvieron exhiben un rendimiento bastante alto, no es claro que esta sea la forma en que el proceso de evaluación se debe de llevar acabo. Tal vez, existen interdependencias entre los criterios de evaluación, con posibles conflictos entre ellos. De ser así, un proceso que maneja estas situaciones de manera explícita es el apropiado; por ejemplo, el enfoque de optimización multiobjetivo (ver Secciones II.2 y II.2.1).

Por lo tanto, en esta sección presentamos un algoritmo basado en programación genética con objetivos múltiples (GP-MO) para sintetizar operadores. El algoritmo que proponemos utiliza el MOEA SPEA-2 de Zitzler *et al.* (2002) (ver Sección II.2.2) para llevar acabo la administración y manejo de la población de operadores que se están evolucionando; ver Figuras 5 y 4.

El enfoque que se propone permite ir agregando objetivos, o criterios de evaluación, al proceso de síntesis sin la necesidad de modificar substancialmente al algoritmo evolutivo. Esto no se puede lograr trivialmente con la evaluación mono-objetivo del capítulo anterior. Con la optimización multiobjetivo, los posibles conflictos entre los criterios se manejan de una forma natural considerando relaciones de dominancia, y generando así un conjunto de soluciones óptimas en el sentido Pareto. Además, y posiblemente el

Parámetro	Descripción y valores
Población	200 individuos.
Generaciones	50 iteraciones.
Método de inicialización	Ramped Half-and-Half.
Probabilidades de operadores genéticos.	Cruce $p_c = 0.85;$
	Mutación $p_{\mu} = 0.15$.
Profundidad máxima	3,5,7 y 9 niveles.
Tamaño del archivo	El tamaño del archivo de SPEA2: 100.
Tamaño de la selección	Número de individuos elegidos
	para reproducción por SPEA2: 100.

Tabla X: Parámetros básicos del algoritmo GP-MO.

punto más importante, el enfoque permite estudiar al problema de detección de rasgos locales desde una perspectiva novedosa, que anteriormente no se había considerado en la literatura de visión.

Los parámetros generales del GP-MO se muestran en la Tabla X. Los primeros cuatro parámetros tienen un valor canónico, mientras que los últimos dos son específicos al algoritmo SPEA-2. El único que se varia para obtener diferentes resultados es el limite máximo de profundidad que pueden tener los árboles que representan a los operadores. Por lo tanto, para cada configuración experimental se presentan cuatro resultados, uno para cada nivel de profundidad que se permite.

VI.3.1 Espacio de búsqueda

El espacio de búsqueda es similar al utilizado en la evolución mono-objetivo descrita en el Capítulo V. Entonces, los conjuntos de funciones y terminales son,

$$F = \left\{ +, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out} \right\}$$
$$\bigcup \left\{ \frac{\delta}{\delta x} G_{\sigma_D}, \frac{\delta}{\delta y} G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2} \right\} ,$$
$$T = \left\{ I, L_x, L_{xx}, L_{xy}, L_{yy}, L_y \right\} ,$$

donde I es la imagen de entrada, y I_{out} puede ser cualquier terminal en T, o la salida de una de las funciones en F; L_u son derivadas Gaussianas en la dirección u de la imagen; G_{σ} son filtros de suavizado Gaussiano; $\frac{\delta}{\delta u}G_{\sigma_D}$ son derivadas de la función de Gauss; y la constante k = 0.05. Para la función log_2 se tomo la convención de $log_2(0) = 0$, para evitar operaciones con cantidades infinitas. Por razones similares, cuando se da el caso en que se aplica la raíz a un valor negativo $\sqrt{-x}$, solo se toma la parte real del resultado para evitar operaciones posteriores con números complejos.

VI.3.2 Evaluación de los objetivos

Arriba se definen tres medidas numéricas que nos permiten cuantificar las propiedades de estabilidad, dispersión, y contenido de información. Entonces, definimos las siguientes funciones de costo que se utilizaran para guiar al proceso evolutivo:

• Estabilidad: $f_1(K) = \frac{1}{r_{K,J}(\epsilon) + c_1}$.

• Dispersión:
$$f_2(K) = \frac{1}{\exp(\mathcal{D}(I, X) - c_2)}$$

• Contenido de información:
$$f_{3a}(K) = \frac{1}{\exp(\mathcal{I}_{SIFT_{\mu}}(\Gamma) - c_{3a})}$$
.
 $f_{3b}(K) = \frac{1}{\exp(\mathcal{I}_{HOLDER_{\mu}}(\Gamma) - c_{3b})}$.

Donde, $r_{K,J}$ es la tasa de repetibilidad promedio calculada para el operador K en la secuencia J (Ecuación 12), \mathcal{D} e la dispersión espacial de los puntos, y \mathcal{I}_{μ} es el contenido de información calculado con la entropía de Shannon; además, las constantes fueron fijadas experimentalmente para obtener el mejor rendimiento a $c_1 = 0.01$, $c_2 = 10$, $c_{3a} = 2.8$, y $c_{3b} = 3.8$.

La secuencia de entrenamiento es la de Van Gogh, ver la Figura 21; sin embargo, para las medidas de dispersión y contenido de información solo se utiliza la primera imagen de la secuencia. También, es necesario enfatizar que las medidas de evaluación se definen como funciones de costo que se desean minimizar, o bien, valores menores de las funciones representan un incremento en el criterio correspondiente. Por lo tanto, en las gráficas de esta sección los valores menores siempre representan un mejor rendimiento.

VI.4 Resultados experimentales

La serie de resultados experimentales que se presentan en este capítulo tienen tres propósitos básicos:

- Estudiar las relaciones que existen entre los tres criterios de evaluación. Por ejemplo, si encontramos un frente de soluciones no dominadas, esto nos sugiere que los criterios utilizados efectivamente se encuentran en conflicto. Para esto, se ejecuta el GP-MO utilizando diferentes combinaciones de dos objetivos a la vez, y finalmente utilizando los tres objetivos simultáneamente.
- 2. En los casos que se encuentren frentes Pareto, se desea observar que tipo de relación guardan con dicho frente los detectores propuestas en la literatura (Beaudet, 1978; Kitchen y Rosenfeld, 1982; Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987), así como los dectores evolucionados utilizando el enfoque mono-objetivo, K_{IPGP1} y K_{IPGP2}.
- 3. Mostrar que la propuesta GP-MO es capaz de generar no solo un operador por ejecución, sino un conjunto de operadores óptimos en el sentido Pareto. Dicho resultado sería de gran utilidad, ya que el diseñador de un sistema de visión tendría a su disposición varios operadores de los cuales elegir para resolver la tarea en particular que desea abordar.

VI.4.1 Estabilidad-Dispersión

En estos experimentos se ejecuta el GP-MO utilizando dos objetivos de optimizaciónón: Estabilidad y Dispersión de puntos. Para esto, se parte de la siguiente hipótesis inicial. Se asume que existe un conflicto práctico entre los objetivos de estabilidad y dispersión para un detector de puntos de interés; esta hipótesis se formula a partir de la siguiente premisa básica. Cuando todos los puntos de interés están muy aglomerados sobre el plano de la imagen, o bien con poca dispersión, entonces estos puntos pueden ser detectados de manera estable por un operador isotrópico. O bien, la repetibilidad de los puntos no depende de la dispersión de los mismos. Los resultados del proceso evolutivo que a continuación se presentan, se utilizan para argumentar a favor de esta hipótesis de manera experimental.

La Figura 36 muestra los frentes Pareto que se generaron con cada profundidad máxima que se permite, ver Tabla X. En el frente también se incluye el rendimiento de otros seis operadores, cuatro diseños humanos propuestos en la literatura (Beaudet, 1978; Kitchen y Rosenfeld, 1982; Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987) y dos operadores evolucionados con el enfoque mono-objetivo, K_{IPGP1} y K_{IPGP2} . En dicha figura se puede apreciar que, en general, los operadores previos representan puntos en el espacio de aptitud, o de decisión, dominados por las soluciones del frente Pareto. Además, estos operadores tienden a estar más cercanos al frente en la dimensión de estabilidad. Es evidente que estos diseños se enfocan en mantener una estabilidad alta, cuantificado por la repetibilidad, mientras que el criterio de dispersión no es satisfactoriamente considerado.

En el caso de los operadores evolucionados con el enfoque mono-objetivo, se puede concluir que los términos sigmoidales ϕ en la función de evaluación (ver, Ecuación 13) limitan el impacto que el criterio de dispersión tiene sobre la aptitud de cada individuo.

Por otro lado, en la Figura 36 también se identifican tres operadores colocados



Figura 36: Frentes Pareto encontrados por el GP-MO para los criterios de Estabilidad y Dispersión. La gráfica contiene los resultados de cuatro ejecuciones del algoritmo, cada una con una profundidad máxima diferente. También, se presenta la posición que ocupan, con respecto al frente, cuatro operadores de la literatura (Beaudet, 1978; Kitchen y Rosenfeld, 1982; Harris y Stephens, 1988; Wang y Brady, 1991; Förstner y Gülch, 1987) y dos operadores evolucionadas con el enfoque mono-objetivo K_{IPGP1} y K_{IPGP2} . Además, también se identifican tres operadores colocados cerca de los puntos de inflexión: (a), (b) y (c).

cerca de los puntos de inflexión del frente Pareto. En la Figura 37 se presenta la imagen de interés generada con dichos operadores en la imagen Van Gogh y el conjunto correspondiente de puntos de interés detectados. El conflicto entre los criterios de dispersión y estabilidad es evidente. En un extremo, el operador (a) detecta puntos muy aglomerados que producen una repetibilidad alta, y por otro lado el operador (c) detecta puntos muy dispersos con una detección poco estable. El punto intermedio se ilustra con el operador (b) que domina, en el sentido Pareto, a todos los operadores

(a) (b) (c)

Figura 37: Primer renglón: Imagen de interés extraída con los operadores (a), (b) y (c) del frente Pareto mostrado en la Figura 36. Segundo renglón: Puntos de interés detectados.

Tabla XI: Expresión simbólica de los operadores (a), (b) y (c) identificados en el frente de la Figura 36 para los objetivos de Estabilidad y Dispersión.

Operador	Expresión simbólica
Operador (a):	$G_{2} * \left G_{1} * log \left(G_{1} * I^{2} \right) + G_{2} * \left(G_{1} * I - I \right) + \frac{G_{1} * I}{I} \right ^{2}$
Operador (b):	$G_{2} * \left G_{1} * log(G_{1} * I^{2}) + k \cdot G_{2} * G_{1} * I - I + \frac{G_{1} * I}{I} \right ^{2}$
Operador (c):	$G_2 * \left(\frac{L_y}{L_{yy}}\right)$

propuestos previamente. Finalmente, la Tabla XI presenta la expresión simbólica de los operadores (a), (b) y (c) identificados sobre el frente de la Figura 36.

Comenzamos la discusión con el operador (c) de la Tabla XI, que es inversamente

proporcional a la curvatura de la señal sobre la dirección y. Los puntos que este operador detecta carecen de estabilidad geométrica porque solo consideran una dimensión principal, pero están altamente dispersos ya que se encuentran en pequeñas zonas de intensidad homogénea. Experimentalmente se probó el mismo operador pero utilizando derivadas sobre la dirección x y el rendimiento alcanzado fue prácticamente el mismo.

Por otro lado, se puede apreciar que los primeros dos operadores, (a) y (b), son casi idénticos. Los dos operadores se representan a través de una suma absoluta de tres términos. El primero, es proporcional, de manera no-lineal, a la intensidad en cada punto. El tercer término, es una razón entre una versión suavizada de la imagen y la versión original de la misma, la operación inversa a la que realiza el operador K_{IPGP6} . Este término obtiene valores máximos sobre puntos de la imagen que son de una intensidad menor con respecto al promedio dentro de un vecindario Gaussiano. El primer y el tercer término son el mismo en los dos operadores, la diferencia entre (a) y (b) se presenta en el segundo término, una operación DoG, similar a la realizada por el operador K_{IPGP1^*} , que resalta los bordes de la imagen. Se puede apreciar que la magnitud y el signo del peso que recibe este término modula la dispersión de los putos detectados por cada operador, una propiedad que pudiera aprovecharse para ajustar el operador a las características particulares de una aplicación en concreto.

Por lo tanto, se reescribe la expresión básica de los operadores (a) y (b) como,

$$K_{MO} = G_2 * \left| G_1 * log \left(G_1 * I^2 \right) + W \cdot G_2 * \left| G_1 * I - I \right| + \frac{G_1 * I}{I} \right|^2 , \qquad (20)$$

donde W es el factor de peso que controla la dispersión de los puntos detectados. Con el propósito de facilitar el análisis de este operador se utiliza la siguiente notación:

$$K_{MO} = G_2 * \left| K_{MO}^1 + W \cdot K_{MO}^2 + K_{MO}^3 \right|^2 , \qquad (21)$$

con $K_{MO}^1 = G_1 * log (G_1 * I^2), \ K_{MO}^2 = G_2 * |G_1 * I - I| \ y \ K_{MO}^3 = \frac{G_1 * I}{I}.$



Figura 38: Comportamiento de las funciones de costo asociadas a la estabilidad y dispersión de puntos para el operador K_{MO} con respecto al factor de peso W, aplicado a la secuencia de Van Gogh. (a) El comportamiento de la estabilidad del descriptor; el mejor rendimiento se obtiene en el rango (0, 1], donde es bastante estable. (b) El punto mínimo en la gráfica corresponde al operador (b) de la Figura 36. Es importante notar que se puede manipular la dispersión de los puntos detectados sin modificar la estabilidad geométrica del operador.

El término K_{MO}^2 , es un filtro DoG que tiende a resaltar los bordes presentes en la imagen. Ahora, si se supone que en la mayoría de las imágenes los bordes son rasgos que solo cubren una pequeñ porción de la imagen, es razonable el efecto que este peso tiene sobre la dispersión de los puntos, dando más o menor importancia a los puntos colocados sobre bordes. Para analizar el efecto que W tiene sobre el rendimiento del operador aplicado a la imagen de entrenamiento, con respecto a las funciones de costo para la estabilidad y dispersión, W se varía dentro de un rango de [-1, 1] con incrementos de 0.05; los resultados se presentan en las gráficas de la Figura de 38.

En la Figura 38a, se presenta el comportamiento del criterio de repetibilidad. Se aprecia claramente que existe una discontinuidad cuando W > 0; i.e., para valores de $W \leq 0$ el operador no es geométricamente estable, mientras que lo contrario es verdad cuando W > 0. En la Figura 38b se muestra el comportamiento del criterio de dispersión con respecto a W. Se presentan los valores obtenidos en cada intervalo, y además se ajusta una curva para ilustrar el comportamiento general del operador; la curva es una suma ponderada de funciones sinodales de ocho componentes. En este caso, la mejor dispersión se alcanza en el punto donde W = 0.05 que corresponde al operador (b) de la Figura 36.

Para confirmar el funcionamiento del operador K_{MO} se repitieron estas pruebas con dos secuencias diferentes, New York y Mars; la Figura 39 muestra como varía la estabilidad y la dispersión en estas dos secuencias. Primero, en los dos casos la estabilidad del operador muestra un comportamiento similar al que se obtiene en la secuencia de entrenamiento. En los dos casos, la estabilidad no se ve afectada de manera significativa cuando el peso W es mayor a cero. Segundo, para la secuencia NY la dispersión de los puntos detectados se puede controlar igual que en la secuencia de entrenamiento, mientras mayor sea W menor es la dispersión de los puntos detectados. Por otro lado, para la secuencia de Mars la relación es inversa, mientras mayor sea W mayor también es la dispersión de los puntos. En este caso se refleja como la dispersión de los puntos depende de las características en cada imagen, y para el operador K_{MO} esto depende de la dispersión de los bordes dentro de la escena. Sin embargo, lo importante en los dos casos es que *se puede* modificar la dispersión sin afectar adversamente la estabilidad del detector.

El operador K_{MO} efectivamente provee un parámetro que permite controlar directamente, o inversamente, la dispersión de los puntos de interés que se detectan en una imagen. En la literatura de visión, el operador de Harris también tiene un parámetro similar (Harris y Stephens, 1988),

$$K_{Harris\&Stephens}(\mathbf{x}) = det(A) - k \cdot Tr(A)^2$$

donde A es la matriz de autocorrelación. Por lo tanto, con el fin de comparar los dos



Figura 39: Comportamiento de las funciones de costo asociadas a la estabilidad y dispersión de puntos para el operador K_{MO} aplicado a dos secuencuas de prueba: New York y Mars.

operadores, en la Figura 40 se muestran los resultados cuando se varía el parámetro kdel operador de Harris utilizando la secuencia de Van Gogh. En el caso de la dispersión, se puede observar que la función de costo nunca baja al mismo rango de valores que se obtienen con el operador K_{MO} . Además, en las dos gráficas de la Figura 40 tanto la dispersión de puntos como la estabilidad del detector se ve seriamente afectada y se vuelve casi nula una vez que el parámetro k esta por arriba de 0.3. Por ende, se puede afirmar que no se puede obtener el mismo control de la dispersión de puntos a través del parámetro k del operador de Harris que se obtiene a través del parámetro W del



Figura 40: Comportamiento de las funciones de costo asociadas a la estabilidad y dispersión de puntos para el operador de Harris cuando se varia el parámetro k; utilizando la secuencia de Van Gogh. Para valores de k > 0.3 el rendimiento del operador se degrada considerablemente y los puntos en la gráfica son mucho mayores al rango mostrado.

operador K_{MO} .

Retomando los resultados mostrados en las Figuras 38 y 39, se puede suponer que el rango de valores que generan el mejor rendimiento para el operador K_{MO} sen encuentra en [0, 1]. Para visualizar este control, en la Figura 41 se presenta la detección de puntos de interés con el operador K_{MO} utilizando cuatro valores para W: 0, 0.05, 0.5 y 1. En todos los casos, se puede observar como la dispersión de los puntos detectados se ve directamente influenciado por el valor que toma W. Además, este comportamiento se observa tanto en la imagen de entrenamiento como en las imágenes de prueba que contienen escenas con estructuras diferentes.

Con el propósito de simplificar la operación que realiza K_{MO} , se analizan con mayor detalle cada uno de los términos que lo conforman: K_{MO}^1 , K_{MO}^2 y K_{MO}^3 . En particular, lo primero que es evidente, y se confirma experimentalmente, es que el tercer término tiende a ser considerablemente menor que los otros dos. En la practica, este valor es


Figura 41: Puntos de interés detectados con el operador K_{MO} y cuatro valores de peso W: 0, 0.05, 05, y 1. El primer renglón contiene la imagen de entrenamiento Va Gogh, el segundo una imagen de la secuencia New York, el tercero la imagen de Casita, en el cuarto la imagen Puerta, y en el quinto la imagen Bip.

de uno o dos ordenes de magnitud menor que los valores que arrojan los primeros dos términos. Por lo tanto, se puede considerar como innecesario en el cálculo del valor de interés para cada punto, de tal forma que el operador ahora se puede expresar como,

$$K_{MO} = G_2 * \left| K_{MO}^1 + W \cdot K_{MO}^2 \right|^2 .$$
(22)

Para verificar que este cambio no afecta el rendimiento del operador con respecto a ninguno de los dos criterios, se repiten las pruebas experimentales mostrados arriba para esta versión simplificada del operador. La Figura 42 grafica el rendimiento en las tres secuencias de prueba utilizadas arriba, Van Gogh, New York y Mars. Es claro que el cambio de las funciones de costo es prácticamente nulo, por lo que se puede concluir que esta versión mas compacta detecta los mismos puntos con un costo computacional menor.

Como conclusión, se quiere hacer énfasis en el hecho de que a través de la búsqueda GP-MO se diseñó un nuevo operador, K_{MO} , para la detección de puntos de interés. Esta metodología de diseño no es común, ya que difiere notablemente del enfoque ortodoxo basado en un análisis profundo del problema. Sin embargo, en este caso este proceso de diseño ha demostrado ser de gran utilidad, ya que se sintetizo un operador que proporciona, de manera explícita, un parámetro que controla la dispersión global de los puntos detectados y mantiene una estabilidad geométrica alta; características que lo hacen único en la literatura actual.

VI.4.2 Estabilidad-Contenido de información

En estos experimentos se ejecuta el GP-MO utilizando dos objetivos de optimización: Estabilidad y Contenido de información. Como hipótesis inicial, se supone que existe un conflicto práctico entre estos dos objetivos; esta hipótesis se basa en dos premisas.



Figura 42: Comportamiento de las funciones de costo asociadas a la estabilidad y dispersión de puntos para el operador K_{MO} simplificado, donde se elimina el tercer término de la Ecuación 21.

Primero, puntos que estén muy cercanos en el plano de la imagen tendrán vecindarios locales muy similares, y por lo tanto sus descriptores locales también serán muy parecidos. Además, puntos muy aglomerados también pueden ser detectados de manera estable, como se observó en los resultados de la sección anterior. Segundo, existen operadores que detectan puntos que son muy estables pero que suelen encontrarse colocados sobre estructuras visuales muy homogéneas, tales como bordes; hecho que limita la diversidad de los descriptores que de ellos se obtienen.

Por otro lado, debido a que se proponen dos formas para calcular el contenido de información, con dos descriptores diferentes, se presentan dos conjuntos de resultados experimentales.

Contenido de información: SIFT

En este caso, el contenido de información se calcula utilizando el descriptor SIFT. La Figura 43 muestra los frentes Pareto que se generaron con cada profundidad máxima que se permite. Se puede apreciar que, en general, los operadores diseñados previamente representan puntos en el espacio de aptitud dominados por las soluciones que estab sobre el frente Pareto. En la Figura 43 también se identifican tres operadores colocados cerca de los puntos de inflexión del frente. La Figura 44 presenta la imagen de interés generada con dichos operadores y el conjunto correspondiente de puntos de interés detectados; la expresión simbólica de cada operador se presenta en la Tabla XII.

Ahora bien, los resultados que se obtienen en estos experimentos no se interpretan tan claramente como en el caso anterior; la dificultad esta en la dimensión que corresponde al contenido de información. Por un lado, la medida propuesta si genera un ordenamiento razonable de los detectores propuestos previamente en la literatura. Por ejemplo, el operador de Beaudet y el operador K_{IPGP2} obtienen un rendimiento similar en este eje, ya que los dos utilizan el determinante de la matriz Hessiana. También, el



Figura 43: Frentes Pareto encontrados con el GP-MO utilizando los objetivos de Estabilidad y Contenido de información; el segundo se calcula utilizando el descriptor SIFT. La gráfica contiene los resultados de cuatro ejecuciones del algoritmo, cada una con una profundidad máxima diferente, y se identifican tres operadores nuevos sobre el frente: (d), (e) y (f)

mejor rendimiento se obtiene con el operador de Harris, un resultado consistente con el estudio realizado en (Schmid *et al.*, 2000). Además, el rendimiento más bajo en este eje, de los operadores previos, lo obtiene el operador K_{IPGP1} , algo que se puede esperar porque los puntos que detecta tienden a ser similares ya que se encuentran sobre bordes de la imagen.

Sin embargo, referente a los operadores (d), (e) y (f) del frente en la Figura 43 el análisis de los resultados resulta ser menos intuitivo. Esto se debe a que en el eje del contenido de información, los operadores (e) y (f) obtienen un rendimiento mejor que aquel obtenido por el operador (d). Esto no es intuitivo ya que los puntos detectados

(d) (e) (f)

Figura 44: Primer renglón: Imagen de interés extraída con los operadores (d), (e) y (f) del frente Pareto mostrado en la Figura 43. Segundo renglón: Puntos de interés detectados.

con los operadores (e) y (f) están aglomerados en porciones reducidas de la imagen, mientras que los puntos detectados por el operador (d) están más dispersos.

Una explicación plausible a este fenómeno se atribuye a las características básicas del descriptor SIFT y las propiedades de la imagen de entrenamiento. Recordando, SIFT describe el comportamiento de la orientación del gradiente en la imagen. Por otro lado, en la parte superior izquierda de la imagen Van Gogh se encuentra un dibujo de un objeto circular que representa al Sol. Por ende, se asume que aunque los puntos están amontonados en esta parte de la imagen sus descriptores muestrean el espacio SIFT de una forma altamente dispersa debido a la estructura circular de la escena.

Este fenómeno se ilustra en la Figura 45, donde se grafíca la entropía promedio calculada para cada dimensión del descriptor SIFT, para los tres operadores. Se puede observar que el operador (d) alcanza picos de entropía más altos, debido a la variedad de puntos que detecta. Sin embargo, también existen muchas dimensiones en la cual

Operador	Expresión simbólica
Operador (d):	$\frac{G_1 * G_1 * G_2 * G_1 * I }{G_2 * G_2 * G_2 * I }$
Operador (e):	$ L_{yy} + G_1 * G_1 * G_1 * I + G_1 * L_{xx} $
Operador (f):	$\frac{G_2 * G_1 * I}{(L_{yy} \cdot G_2 * L_y) \cdot (G_2 * L_{xx}) \cdot (k \cdot G_1 * L_{xx})}$

Tabla XII: Expresión simbólica de los operadores (d), (e) y (f) identificados en el frente de la Figura 43 para los objetivos de Estabilidad y Contenido de información.



Figura 45: La entropía promedio alcanzada en cada dimensión del descriptor SIFT por cada operador identificado en la Figura 43: (d), (e) y (f).

la entropía es muy baja, esto debido a que en general el tipo de puntos de interés que detecta tienden a tener una estructura local similar, algo común para los operadores que se enfocan en bordes o esquinas, por ejemplo. Por otro lado, los operadores (e) y (f) no exhiben estas variaciones altas de entropía, debido a los cambios casi continuos presentes en la estructura circular en que se encuentran la mayoría de los puntos detectados.

Por lo tanto, se puede concluir que la evolución es capaz de explotar características intrínsecas en los datos de entrenamiento, un problema que no se puede evitar de manera trivial.

Contenido de información: Regularidad Hölder

En este caso, el contenido de información se calcula utilizando el descriptor basado en la regularidad local, cuantificada a través del exponente Hölder. La Figura 46 presenta los frentes Pareto que se generaron con cada profundidad máxima que se permite. En el caso de la Figura 46a, se pueden identificar dos aspectos significativos. Primero, todos los operadores diseñados previamente, a mano o a través de la evolución mono-objetivo, representan puntos dominados por las soluciones que se encuentran sobre el frente, esto se ilustra con mayor detalle en la Figura 46b. Estos operadores son deficientes con respecto al contenido de información que proveen, comparados con los operadores evolucionados con el GP-MO que se encuentran sobre el frente. Este hecho no se pudo prever de antemano, pero se puede entender porque el contenido de información de los puntos detectados no se consideró de manera explícita durante el proceso de diseño de cada uno de esos operadores.

Segundo, en la Figura 46a se señalan tres operadores que obtienen un rendimiento que pudieramos calificar como extremo. Por un lado, obtienen un nivel de estabilidad muy bueno, pero por otro, el contenido de información que proporcionan es bastante bajo. Para entender esto mejor, en la Figura 47 se muestran los puntos de interés que detecta uno de estos operadores, aquí lo llamaremos operador (z), y la repuesta que genera cuando se aplica a la imagen de entrenamiento. Es posible observar que todos



Figura 46: Frentes Pareto encontrados con el GP-MO utilizando los objetivos de Estabilidad y Contenido de información; el segundo se calcula utilizando el descriptor Hölder. La gráfica contiene los resultados de cuatro ejecuciones del algoritmo, cada una con una profundidad máxima diferente. (a) Comparacion con operadores previos, donde se identifica un caso extremo en la detección de puntos, denominado como el operador (z). (b) El frente Pareto de soluciones donde se identifican tres operadores nuevos: (g), (h) e (i).

los puntos están aglomerados en pequeñas porciones de la imagen; por lo tanto, aunque la repetibilidad de estos puntos es alta, el contenido de información que proporcionan es muy baja porque sus descriptores tienden a ser muy similares. Este resultado sugiere que la dispersión de los puntos en el plano de la imagen esta ligada directamente con la dispersión de los puntos en el espacio del descriptor, una suposición razonable que no se pudo comprobar cuando se utilizó el descriptor SIFT en los experimentos anteriores. Con este ejemplo, se puede apreciar que el descriptor basado en el exponente Hölder no produce las mismas contradicciones intuitivas que se generan cuando se emplea el descriptor SIFT. El descriptor Hölder es capaz de capturar las similitudes entre los puntos que están cercas dentro del plano de la imagen; i.e., puntos que son vecinos en el plano de la imagen también son vecinos en el espacio del descriptor Hölder.



Figura 47: (a) Imagen de interés extraída con el operador (z). (b) Puntos de interés detectados con el operador (z) sobre la imagen de entrenamiento. El operador (z) se encuentra en uno de los extremos del frente Pareto mostrado en la Figura 46a.

También, en la Figura 46b se identifican otros tres operadores colocados cerca de los puntos de inflexión del frente, los operadores (g), (h) e (i). En la Figura 48 se presenta la imagen de interés que se genera con cada uno de estos operadores, junto con los puntos que se detectan sobre la imagen de entrenamiento. Además, la expresión simbólica de cada operador esta en la Tabla XIII.

El operador (g) se encuentra en un extremo del frente Pareto que se muestra en la Figura 46b. Como se puede apreciar, el operador obtienen un buen rendimiento con respecto al criterio de estabilidad, pero es deficiente según la función de costo utilizada para medir el contenido de información. El rendimiento de este operador es muy similar al que obtienen muchos de los operadores propuestos previamente, tanto aquellos que fueron diseñados a mano, como los que se obtuvieron a través de la evolución mono-objetivo. La expresión simbólica del operador (g), presentada en la Tabla XIII, prácticamente representa el inverso aditivo del operador K_{IPGP1} , una sencilla operación DoG. Como se puede observar en la Figura 48, los puntos que este operador detecta se encuentran mayoritariamente sobre los bordes de la imagen; por lo tanto, los puntos tienden a tener vecindarios locales muy similares, hecho que explica el bajo contenido de

(i) (g) (h) (i)

Figura 48: Primer renglón: Imagen de interés extraída con los operadores (g), (h) e (i) del frente Pareto mostrado en la Figura 46b. Segundo renglón: Puntos de interés detectados.

información que estos puntos transmiten. Este argumento se puede extender a un operador como el de Harris, por ejemplo, que principalmente detecta esquinas, estructuras que también tienden a exhibir pocas variaciones en su estructura básica.

El siguiente operador que se identifica en la la Figura 46b es el operador (h); este obtiene un buen compromiso entre las dos criterios de evaluación. La expresión simbólica del operador (h) es difícil de interpretar, y sin embargo es capaz de extraer puntos que son altamente repetibles e informativos.

Finalmente, el operador (i) se encuentra en otro de los extremos de la Figura 46b. Prácticamente, este operador se puede entender como el caso contrario del operador (g), detecta puntos que son muy distintivos, pero no es estable con respecto a transformaciones de la imagen.

En conclusión, a trvés de la búsqueda evolutiva MO se mostró que si existe un conflicto práctico cuando se intenta detectar puntos que sean repetibles y que también

л.

Expresión simbólica

 $G_2 * G_1 * (I - G_2 * I)$

 $\frac{\sqrt{G_1 * |G_2 * I|}}{|G_2 * ((G_2 * I)^2 - ||L_{xy} + I| + k \cdot L_{xx}|)|}$

 $G_1 * (k \cdot L_{yy}) - \left| \left| L_{yy} - L_{xy} \right| - \frac{\log(I)}{L_y} \right|$

de

Operador

Operador (g):

Operador (h):

Operador (i):

Tabla XIII: Expresión simbólica de los operadores (g), (h) y (i) identificados en el frente

sean únicos y distintivos con respecto a la estructura de su vecindario local. Est
se pudo comprobar experimentalmente describiendo el veindario local de cada punt
utilizando un análisi de la regularidad local de la imagen, algo que no se logró utilizand
al descriptor SIFT.

VI.4.3 Dispersión-Contenido de información

En estos experimentos se ejecuta el GP-MO utilizando dos objetivos de optimización: Dispersión y Contenido de información. En este caso, una suposición razonable pudiera ser que estos dos criterios están altamente correlacionados. Es decir, si la dispersión espacial de los puntos es alta, entonces los descriptores de todos los puntos serán más diversos, y viceversa. Sin embargo, como se vio en los experimentos presentados en la sección anterior, este hecho depende de la manera en la cual se describe el vecindario local de cada punto, además de la estructura particular de la escena en cada imagen. Por lo tanto, igual que en el caso anterior, se presentan dos conjuntos de resultados experimentales, uno para cada descriptor local que se utiliza.



Figura 49: Frentes Pareto encontrados con el GP-MO utilizando los objetivos de Dispersión y Contenido de información; el segundo se calcula utilizando el descriptor SIFT. La gráfica contiene los resultados de cuatro ejecuciones del algoritmo, cada una con una profundidad máxima diferente. (a) Comparacion con operadores previos. (b) El frente Pareto de soluciones donde se identifican tres operadores nuevos: (j), (k) y (l).

Contenido de información: SIFT

En este caso, el contenido de información se calcula utilizando el descriptor SIFT. La Figura 49 muestra los frentes Pareto que se generaron con cada profundidad máxima que se permite. De nuevo, en estos experimentos se aprecia que los operadores propuestos anteriormente representan puntos en el espacio de aptitud dominados por las soluciones del frente Pareto, ver Figura 49a. Inclusive, debido a que los operadores previos tienen un rendimiento muy por debajo de lo alcanzado por las soluciones del conjunto óptimo Pareto, es necesario presentar una gráfica con una escala más pequeña en la Figura 49b.

En la Figura 49b también se identifican tres operadores colocados cerca de los puntos de inflexión del frente Pareto: (j), (k) y (l). En la Figura 50 se muestra la imagen de interés generada con dichos operadores, y los puntos de interés detectados sobre la imagen de entrenamiento; la expresión simbólica de cada operador esta en la Tabla

(j) (k) (l)

Figura 50: Primer renglón: Imagen de interés extraída con los operadores (j), (k) y (l) del frente Pareto mostrado en la Figura 49. Segundo renglón: Puntos de interés detectados.

XIV.

En estos resultados se vuelve a presentar un fenómeno inesperado, que los criterios de estabilidad y contenido de información pueden estar en conflicto. En los resultados anteriores, de la Sección VI.4.2, se mostró que aún cuando los descriptores están aglomerados su descripción puede ser muy variada debido a la estructura subyacente de la escena presente en la imagen, y además debido al descriptor local utilizado. Esto sucede con en el operador (j), como se puede observar con los puntos de interés que se detectan en la imagen de entrenamiento. Sin embargo, en los operadores (k) y (l) se presenta una situación diferente. En los dos casos, el nivel de dispersión espacial es prácticamente la misma, como se puede apreciar cualitativamente en las imágenes de la Figura 50.

Sin embargo, si existe una diferencia con respecto al contenido de información de cada una de las imágenes. Esto también se muestra en la Figura 51, donde se grafíca la

Tabla	XIV:	Expresión	simbólica	ı de los	operadores	(j), (l	k) y (l)	identificados	s sobre el
frente	de la	Figura 49	para los c	bjetivos	de Estabil	idad y	v Conten	ido de infor	mación.

Operador	Expresión simbólica						
Operador (j):	$k^2 \cdot G_2 * G_2 * \left(\frac{L_y}{L_{yy}}\right)$						
Operador (k):	$k \cdot G_2 * G_2 * \sqrt{\frac{\log(G_1 * L)}{G_1 * L_x + L_{yy} }}$						
Operador (l):	$\frac{\frac{k \cdot G_1 * I}{G_2 * L_x \cdot I}}{L_{yy}^2 \cdot L_{yy}L_{xx} + \sqrt{L_{xx}} \cdot G_1 * I + G_1 * L_{xy}^2 - L_{yy} }$						
	$- \left log \left(G_1 * \frac{L_{xy}}{ L_y - L_{xx} - L_x + L_{yy} } \right) - \frac{G_1 * L_{yy}}{ L_{xx} } \right $						

entropía promedio alcanzada en cada dimensión del descriptor SIFT por cada operador, (j), (k) y (l). En dicha gráfica, se observa que los máximos y mínimos de entropía, en la mayoría de las dimensiones, los obtiene el operador (j), mientras que el operador (k) y (l) tienen un comportamiento más estable. Como se discute en la sección anterior, el operador (l) detecta puntos sobre una estructura circular dentro de la imagen, hecho que puede explicar la dispersión alta de los descriptores SIFT para el conjunto de puntos de interés. Sin embargo, este no es el caso para el operador (k), ya que los puntos están claramente dispersos sobre el plano de la imagen.

Se puede asumir que el operador (k) alcanza el mejor compromiso entre dispersión y contenido de información si se cumplen las siguientes condiciones: 1) $f_2(k) < f_2(l)$ y $f_2(k) \approx f_2(j)$; y además 2) $f_{3a}(k) \approx f_{3a}(l)$ y $f_{3a}(k) < f_{3a}(j)$. La primer condición es obvia, y se puede apreciar claramente en la Figura 49. Sin embargo, la segunda condición no se puede apreciar cualitativamente, por ende se se comprueba utilizando



Figura 51: La entropía promedio alcanzada en cada dimensión del descriptor SIFT por cada operador identificado en la Figura 49: (j), (k) y (l).

pruebas Kolmogorov-Smirnov (KS) de dos muestras, significativa a un nivel de p = 5%.

La prueba KS se realiza entre cada dimensión SIFT para el conjunto de descriptores detectados por cada operador: Γ_j , Γ_k y Γ_l . En el caso de los descriptores en Γ_k y Γ_j , la prueba KS determina que la hipótesis nula se rechaza en 106 de las 128 dimensiones, por lo que la diferencia entre $f_{3a}(k)$ y $f_{3a}(j)$ si se considera significativa. Para los descriptores Γ_k y Γ_l , la hipótesis nula solo se rechaza en 46 de las dimensiones SIFT, por esto se asume que $f_{3a}(j) \approx f_{3a}(l)$.

Contenido de información: Regularidad Hölder

En el caso anterior, los experimentos con el descriptor SIFT arrojan resultados inesperados y difíciles de entender. De nuevo, la búsqueda evolutiva es capaz de utilizar rasgos



Figura 52: Frentes Pareto encontrados con el GP-MO utilizando los objetivos de Dispersión y Contenido de información; el segundo se calcula utilizando el descriptor Hölder. La gráfica contiene los resultados de cuatro ejecuciones del algoritmo, cada una con una profundidad máxima diferente. (a) Comparacion con operadores previos. (b) El frente Pareto de soluciones, donde se identifican tres operadores nuevos: (m), (n) y (o).

específicos de la escena y características particulares del descriptor SIFT guiar al proceso de optimización, produciendo así un efecto de sobre entrenamiento. Por lo tanto, en los experimentos de esta sección el contenido de información se calcula utilizando el descriptor Hölder.

La Figura 52 muestra los frentes Pareto que se generaron con cada profundidad máxima que se permite. En estos experimentos, aun más que en los anteriores, es muy notable que todos los operadores propuestos anteriormente están muy retirados del frente Pareto que se encontró con el GP-MO, ver Figura 52a. En los dos ejes de evaluación se aprecia que todas las soluciones previas representan soluciones dominadas por las solucione que estan sobre el frente Pareto.

En la Figura 52b solo se muestran las soluciones que conforman el frente Pareto, y se identifican tres operadores nuevos: (m), (n) y (o). En la Figura 53 se muestra la

(m) (o)

Figura 53: Primer renglón: Imagen de interés extraída con los operadores (m), (n) y (o) del frente Pareto mostrado en la Figura 52. Segundo renglón: Puntos de interés detectados.

Tabla XV: Expresión simbólica de los operadores (m), (n) y (o) identificados en el frente de la Figura 52 para los objetivos de Estabilidad y Contenido de información.

Operador	Expresión simbólica
Operador (m):	$G_2 * G_2 \left(\frac{L_y}{2 \cdot L_{yy} + L_{xy}}\right)$
Operador (n):	$\frac{G_2 * G_2 * \left(\frac{L_{xy}}{2 \cdot L_{yy} + L xy + L_x}\right)}{G_1 * G_2 * G_1 * G_2 * G_1 * I}$
Operador (o):	$\frac{G_2 * \left[G_2 * G_1 * I + G_2 * \left(\frac{L_{xy}}{G_2 * L_y}\right)\right]}{\log(L_y) - \left \left \sqrt{I} - L_{xx} + L_{xy} + G_1 * L_x\right - \left G_1 * G_2 * I - L_y - \left L_y + k \cdot I + \frac{I + L_{xy}}{L_{yy} \cdot L_{xx}}\right \right \right }$

imagen de interés generada con dichos operadores, y los puntos de interés detectados sobre la imagen de entrenamiento; la expresión simbólica de cada operador se presenta en la Tabla XV.

Lo resultados que se muestran en las Figuras 52 y 53 se pueden interpretar de la siguiente manera. Primero, es evidente que si existe un conflicto entre estos dos criterios, aunque aparentemente es menos pronunciado que en los experimentos que utilizan el SIFT como descriptor. Por ejemplo, los dos operadores en los extremos del frente, (m) y (o), detectan puntos sin caer en casos degenerativos de la aglomeración de puntos. Segundo, es interesante observar que aunque los puntos que detecta el operador (m) están más dispersos, estos puntos transmiten menos información que los puntos detectados por el operador (o); sin embargo, este hecho no se da sin precedentes. Schmid *et al.* (2000) reportaron que el detector de Harris alcanzaba una mejor dispersión en el espacio del descriptor que un detector que selecciona puntos de manera aleatoria. En cierta forma, el operador (m) detecta puntos tan dispersos que se acerca a la aleatoriedad. Por otro lado, el operador (o) es capaz de encontrar puntos dispersos en el espacio del descriptor, pero no necesariamente muy dispersos en el plano de la imagen. Finalmente, el operador (n) obtiene un compromiso entre los dos extremos que talvez seria útil en aplicaciones reales.

VI.4.4 Estabilidad-Dispersión-Contenido de información

Finalmente, se ejecutó el algoritmo GP-MO utilizando los tres objetivos de optimización simultáneamente: Estabilidad, Dispersión de puntos, y Contenido de información. Sin embargo, los resultados de los experimentos anteriores muestran que en la mayoría de los casos las ejecuciones con profundidades máximas grandes, siete o nueve niveles, dominan a los frentes generados con profundidades menores, cinco o tres. Por lo tanto, para este experimento solamente se utilizó la profundidad máxima de nueve niveles para los árboles de los operadores. Además, se realizaron dos experimentos, uno para cada una de los descriptores utilizados para calcular el contenido de información. En la la Figura 54 se muestran los resultados que se obtuvieron con el descriptor SIFT, y en



Figura 54: Frente Pareto generado con el GP-MO utilizando los tres criterios de optimización simultáneamente; se presentan cuatro vistas diferentes del mismo frente. El contenido de información se calcula utilizando el descriptor SIFT.

la Figura 55 se muestran los resultados que se obtuvieron con el descritptor de Hölder. Los resultados en las dos figuras corroboran lo que se deduce de los resultados obtenidos con dos objetivos de optimización, los tres criterios pueden considerarse como objetivos que están en conflicto entre ellos.



Figura 55: Frente Pareto generado con el GP-MO utilizando los tres criterios de optimización simultáneamente; se presentan cuatro vistas diferentes del mismo frente. El contenido de información se calcula utilizando el descriptor Hölder.

VI.5 Discusión y trabajo futuro

En este capítulo se lleva acabo un análisis multiobjetivo del problema de detección de puntos de interés. Para esto, el problema se plantea como un proceso de optimización MO y se resuelve utilizando GP. Se contemplan tres criterios de evaluación para el problema de detección de puntos: Estabilidad geométrica, dispersión de los puntos de detectados, y contenido de información en el conjunto correspondiente de descriptores locales. Se proponen cuatro configuraciones experimentales, tres de ellas considerando dos pares de criterios a la vez, y la cuarta utilizando los tres criterios simultáneamente. Los resultados de dichos experimentos conllevan a las siguientes conclusiones.

Primero, se confirma, de manera experimental, que los criterios de estabilidad y dispersión están están en conflicto. Por esta razón, el planteamiento mono-objetivo del Capítulo V es deficiente en el sentido que no contempla la interacción que existe entre estos objetivos. Por otro lado, el GP-MO encontró los términos esenciales que permitieron construir un nuevo operador parametrizado, K_{MO} . Dicho operador proporciona un control directo sobre el nivel de dispersión que se obtiene en el proceso de detección sin degradar la estabilidad geométrica del operador. Este operador fue construido utilizando algunas de las soluciones que conforman al conjunto óptimo Pareto; es decir, la optimización MO propicio el diseño de dicho operador. En nuestro estudio, el operador K_{MO} representa un diseño único en la literatura de puntos de interés, que amerita mayor investigación analítica y pruebas experimentales como parte del trabajo futuro.

Segundo, la medida propuesta para evaluar el contenido de información depende del descriptor local que se utilice; en este trabajo se utilizaron dos, el descriptor SIFT y el descriptor Hölder. En los dos casos, la medida de contenido de información, basada en la entropía de Shannon, obtiene un ordenamiento coherente de los operadores propuestos previamente, similar a los resultados reportados en (Schmid *et al.*, 2000). Sin embargo, los operadores evolucionados, que conforman al conjunto óptimo Pareto, son muy diferentes en un caso y en el otro. Cuando se utiliza el descriptor SIFT, el GP es capaz de explotar características estructurales en la imagen que generan resultados poco intuitivos. Por ejemplo, se observa que puntos que están aglomerados en porciones reducidas de la imagen también pueden tener una representación dispersa en el espacio del descriptor. Este hecho se atribuye a dos razones: 1) las características del descriptor SIFT, que captura la dirección del gradiente en una región local, mientras

que la magnitud es solo un componente secundario en el descriptor; y 2) la capacidad del GP de explotar esta característica para detectar estructuras en la imagen que maximizan la dispersión en el espacio del descriptor. Por otro lado, cuando se utiliza el descriptor Hölder estas ambigüedades son eliminadas. En efecto, los operadores que el GP produce cuando se utiliza este descriptor están más en acorde con las expectativas iniciales que se tenían. Por ejemplo, en un extremo del frente Pareto que se generó, están las soluciones que detectan puntos aglomerados, rasgos muy estables pero que tienen descriptores casi idénticos. Después, están operadores que detectan estructuras dispersas en la imagen pero estructuralmente similares, bordes y esquinas, también en este caso el contenido de información es baja. Finalmente, encontramos que el GP es capaz de encontrar puntos que no solo tienen buena estabilidad pero que también extraen una variedad mayor de descriptores, o bien, que los puntos tienden a ser más distintivos en el espacio del descriptor. Por lo tanto, el GP es capaz de sintetizar un conjunto de operadores que manejan el compromiso entre estos dos objetivos de diferentes formas. Como parte del trabajo futuro, se debe de utilizar estos detectores en tareas de reconocimiento o detección para así confirmar que el aumento en contenido de información es útil en una aplicación real.

Tercero, una hipótesis inicial era que el contenido de información y la dispersión de los puntos son objetivos que no están en conflicto mutuo. Sin embargo, el GP sintetizo operadores que experimentalmente comprueban que esto no es así, particularmente cuando se calcula el contenido de información utilizando el descriptor Hölder. Por un lado, existen operadores que maximizan la dispersión y los puntos aparentan ser casi aleatorios. Por otro, hay operadores que encuentran puntos que tienen una dispersión mayor en el espacio del descriptor que la que tienen en el plano de la imagen. Este segundo resultado esta en acorde con los resultados publicados en (Schmid *et al.*, 2000), donde los autores experimentalmente confirman que puntos aleatorios tienden a ser menos informativos que puntos elegidos en base a las propiedades estructurales de su vecindario. Finalmente, como en todos los casos, el GP aquí también sintetiza operadores que representan un punto medio entre el nivel de dispersión y la cantidad de información que se extrae.

Para concluir, se esbozan otras líneas de trabajo futuro que complementen las contribuciones presentadas en este capítulo. Primero, introducir más criterios de optimización en el algoritmo GP-MO propuesto. Por ejemplo, criterios específicos a un dominio de aplicación en particular, con el propósito de producir operadores cada vez más especializados. Una posible tarea puede ser apoyar la navegación visual de un robot móvil, o la recuperación de imágenes en base al contenido, dos tareas que requieren de medidas de rendimiento especializadas. Segundo, es evidente que las relaciones de dominancia Pareto, especialmente referente a los criterios de dispersión y contenido de información, siempre van a depender de la imagen que se este analizando. Por lo tanto, se propone ejecutar el GP-MO con la visión de diseñar operadores óptimos, en el sentido Pareto, para una clase más reducida de imágenes; por ejemplo, imágenes de interior (e.g., oficinas), o imágenes de exterior (e.g., bosques), por nombrar dos ejemplos ilustrativos.

Capítulo VII

Detección invariante a escala con operadores evolucionados

En este Capítulo VII se describe el problema de detección de rasgos locales invariante a escala. Se incluye una discusión sobre los conceptos de espacio escala y escala característica, y como se aplican a dicho problema. Además, se presentan dos detectores basados en operadores diseñados a través de una búsqueda evolutiva. El rendimiento de los detectores se compara con el estado-del-arte, y los resultados sugieren que la implementación propuesta es altamente competitiva con métodos previos.

VII.1 Detección invariante a escala

La Sección III.1 trata brevemente el problema de detección invariante a escala de regiones de interés. En particular, se menciona que este tipo de detectores además de ser invariantes a transformaciones de iluminación y rotación también deben ser invariantes a cambios de escala (ver Capítulo III y Tabla I). Cambios de escala presentan un problema cuando solo se analizan puntos en una imagen, porque cuando un rasgo se observa como un punto este puede convertirse en una región que contiene muchos puntos cuando la escala de observación se modifica; ver Figura 56.

Una forma de resolver el problema de la detección invariante a escala es utilizando el concepto de espacio escala; a continuación, se introduce este concepto.



Figura 56: Un punto de interés se puede convertir en una región de mayor tamaño si el cambio de escala es lo suficientemente grande.

VII.2 Espacio escala

Lindeberg (1991) da una definición útil, aunque no formal, del concepto de escala en esta frase tomada de su tesis doctoral ¹:

La escala debe interpretarse solamente como un parámetro abstracto, que implica una propiedad que permite un ordenamiento débil de objetos de diferente tamaño, sin ningún mapeo directo entre el tamaño real de la señal y el tamaño en el cual se representa.

El espacio escala es una representación multiescala de la información contenida dentro de una imagen, que se obtiene embebiendo a la imagen dentro de una familia de señales derivadas que dependen de solamente un parámetro: la escala (Lindeberg, 1991).

Definición 5. Dada una señal $f : \Re^D \to \Re$, la representación lineal en el espacio escala

¹Lindeberg también desarrolla el concepto de una forma más formal; sin embargo, para las metas del presente trabajo la descripción dada es más que suficiente para entender el concepto.

 $L: \Re^D \times \Re \to \Re$ de f esta dada por la solución a la función de difusión

$$\delta_t L = \frac{1}{2} \nabla^2 L = \frac{1}{2} \sum_{i=1}^{D} \delta_{x_i x_i} L , \qquad (23)$$

con la condición inicial $L(\cdot; 0) = f(\cdot)$; la solución a esta ecuación es la función de Gauss cuando se utiliza el método de funciones de Green. De manera equivalente, es posible definir el espacio escala como la familia de señales que se obtienen de la convolución de la señal f con filtros Gaussianos a diferentes escalas t (desviación estándar),

$$L(\cdot;t) = G_t * f(\cdot) . \tag{24}$$

Una particularidad de lo que produce el cambio de escala es que cuando una porción de la imagen se observa a escalas cada vez menores entonces es posible apreciar menos rasgos prominentes; ver Figura 56. La propiedad más importante del espacio escala que se define arriba es que puede reproducir este fenómeno de manera artificial, tal como se muestra en la Figura 57. Además, existen estudios neurofisiológicos que sugieren que la representación que se hace en el espacio escala puede funcionar como un modelo para la visión biológica (Young, 1986).

Es necesario establecer una metodología que nos permita determinar cual es la escala adecuada que se debe de utilizar para analizar la información contenida dentro de una imagen. Lindeberg (1998) define un **principio para elegir la escala** de la siguiente forma,

"En la ausencia de otra evidencia, se asume que la escala, en la que una combinación (posiblemente no-lineal) de derivadas normalizadas alcanzan un máximo en el espacio escala, puede ser tratada como si reflejara la longitud característica de la estructura correspondiente en los datos."



Figura 57: Se muestra el efecto que tiene el suavizado sobre la imagen, rasgos prominentes van desapareciendo de forma gradual; así se reproduce el efecto de cambio de escala.

Las derivadas normalizadas con respecto a la escala se mantienen constantes aún cuando la escala cambia (Lindeberg, 1991), son invariantes con respecto a la escala en que se observa un determinado rasgo, como una esquina o borde; una derivada normalizada D de orden n esta dada por,

$$D_{i_1\dots i_n}(\mathbf{x};t) = t^n L_{i_1\dots i_n}(\mathbf{x};t) .$$
⁽²⁵⁾

Además, cuando un operador se construye con este tipo de derivadas se dice que también el operador esta normalizado. Sin embargo, de acuerdo a Lindeberg el principio para elegir escala "... debe ser verificado empíricamente, y con respecto al problema particular que se esta abordando". Por lo tanto, se puede esperar que una búsqueda basada en GP sea un enfoque válido para construir operadores que realizan "una combinación



Figura 58: Se muestra un caso idóneo en el que la misma región obtiene un máximo a la respuesta de un operador en escalas diferentes. Es importante notar como un operador ideal produce un solo máximo bien definido para cada región.

posiblemente no linear de derivadas" (Lindeberg, 1991).

VII.3 Como elegir una escala característica

Desde un punto de vista algorítmico, elegir la escala característica de un rasgo en una imagen es un proceso en el cual máximos locales de la respuesta a una función se buscan a diferentes escalas de análisis (Lindeberg, 1998).

En la Figura 58 se presenta una explicación simplificada de esta idea. Es posible observar como una misma región obtiene un máximo a diferentes escalas, dependiendo de la escala en que se esta observando la escena.

De manera formal, se dice que cuando un operador $K(\mathbf{x}, t_i)$ calcula una medida de interés para cada píxel \mathbf{x} en diferentes escalas t_i , se asume que la escala característica



Figura 59: La misma región se detecta aún cuando se observa a diferentes escalas; el tamaño de la región detectada es proporcional a la escala característica t_n .

en \mathbf{x} es t_n si se cumple que

$$K(\mathbf{x}, t_n) > \sup \left\{ K(\mathbf{x}_{\mathbf{W}}, t_n), K(\mathbf{x}_{\mathbf{W}}, t_{n-1}) | \forall \mathbf{x}_{\mathbf{W}} \in \mathbf{W}, \mathbf{x}_{\mathbf{W}} \neq \mathbf{x} \right\}$$
$$\land K(\mathbf{x}, t_n) > h , \qquad (26)$$

donde h es un umbral de filtrado, y **W** es un vecindario de tamaño $n \times n$ alrededor de **x**. Este proceso, similar a lo que se hace durante la detección de puntos de interés (ver Ecuación 1), regresa un conjunto de regiones que son invariantes a la escala de observación. Cada región tiene su centro en el pixel **x** con una escala característica asociada t_n . La inavarianza con respecto a la escala de observación se logra porque t_n varia proporcionalmente con el tamaño del rasgo que se esta observando (Mikolajczyk y Schmid, 2004). Una interpretación física de esto se puede observar si se define el tamaño de la región de manera proporcional a la escala característica t_n ; ver Figuras 58 y 59.

VII.4 Estado-del-arte en detectores invariantes a escala

A continuación se presenta una revisión básica del estado-del-arte en la detección invariante a escala de rasgos locales. Para comenzar, Lindeberg (1998) propone un operador Laplaciano normalizado con respecto a la escala para detectar rasgos sencillos. Lowe (1999) propone una pirámide en el espacio de escala donde la elección de una escala característica se logra utilizando máximos locales de un filtro DoG aplicado entre escalas adyacentes. El detector DoG forma una parte esencial del algoritmo SIFT tal y como fue propuesto originalmente. Mikolajczyk y Schmid (2004) realizaron una comparación de detectores invariantes a escala, en donde se incluyó el DoG de Lowe y el Laplaciano de Lindeberg, además de dos detectores que ellos propusieron: Harris-Laplace y Hessian-Laplace. Los dos operadores propuestos por Mikolajczyk y Schmid (2004) utilizan dos etapas para la detección de regiones. Primero, encuentran máximos con versiones normalizadas de los operadores de Harris y Beaudet. Después, los máximos en el espacio escala los determina la respuesta al Laplaciano. Mikolajczyk & Schmid mostraron, con resultados experimentales, que sus dos propuestas eran capaces de detectar regiones de manera más estable que los detectores propuestos en (Lindeberg, 1998; Lowe, 1999).

Es necesario entender que todos los detectores mencionados arriba trabajan explícitamente en el espacio escala, y utilizan un el mismo proceso básico. La metodología consiste en detectar máximos en la respuesta de algún operador en tres dimensiones, el 2D de la imagen y la dimensión de la escala. Además, muchos de estos operadores también se utilizan para detectar puntos de interés.

Un ejemplo de un detector que utiliza otro enfoque es el que proponen Kadir y Brady (2001). Este detector mide la entropía local de las intensidades de gris, no emplea el análisis del espacio escala descrito arriba. El detector calcula, para cada pixel, un histograma de intensidades en un vecindario circular de radio s, la escala, que se varia dentro de un rango predefinido. El máximo que se encuentre estará asociado a algún radio, y esto determina la escala característica de la región.

En la siguiente Sección se describe como se extiende la medida de repetibilidad introducida en la Sección III.4.1 para la detección de regiones a diferentes escalas. Esto nos proporciona una medida para evaluar el rendimiento de los detectores.

VII.5 Como evaluar un detector invariante a escala

Para poder hacer comparaciones experimentales se requiere de una medida de rendimiento, para esto se utiliza una versión modificada de la tasa de repetibilidad que se describe en el Capítulo III; en este caso es necesario considerar que la transformación incluye un cambio de escala. Los principios son similares, solo que ahora se debe de tomar en cuenta que las regiones comparadas pueden ser de diferente tamaño, a diferencia de lo que se hace en el caso de los puntos de interés. Por esta razón, la Ecuación 3 se debe de expander antes de asignar un valor de repetibilidad $r_{I_i}(\epsilon)$ entre una imagen I_1 y su versión transformada I_i . Se dice que dos regiones, λ_1 y λ_i corresponden si se cumple la condición en la Ecuación 3, donde el centro de cada región corresponde a x_1 y x_2 , y además el *error de traslape* en el área de la imagen que cubren las dos regiones es menor que un umbral ϵ_i ,

$$\left|1 - s_i^2 \frac{\min(t_1^2, t_i^2)}{\max(t_1^2, t_i^2)}\right| < \epsilon_t , \qquad (27)$$

donde t_1 y t_i son las escalas características asociadas a las regiones λ_1 y λ_i , s es el factor real de escala que se recupera de la homografía $H_{1,i}$ entre las imágenes I_1 y I_i , y $\epsilon_t = 0.4$ Mikolajczyk y Schmid (2004).

Ahora que el concepto de detección invariante a escala esta bien definido y además

se cuenta con una medida de rendimiento apropiada, se pasa a presentar dos detectores nuevos construidos a partir de operadores evolucionados con GP. Los operadores que se eligen son los más compactos, especialmente cuando se comparan con las técnicas más elaboradas mencionadas arriba, y aún así obtienen un rendimiento bastante alto cuando se comparan con los detectores del estado-del-arte.

VII.6 Detectores propuestos

Los operadores que se eligieron para construir los detectores invariantes a escala fueron K_{IPGP1} y K_{IPGP1*} (ver Sección V.2.2), esta decisión se tomó debido a dos razones principales. Primero, los dos tienen las estructuras más sencillas que el GP fue capaz de producir, una característica que pudiera ser de mucha importancia debido a la complejidad computacional que se agrega al proceso de detección cuando se considera la dimensión de la escala Mikolajczyk y Schmid (2004). Los operadores que tienen un costo computacional más alto no son una buena opción para aplicaciones que requieren una ejecución en tiempo real, como un sistema de visión para la navegación de un robot móvil. Por ende, los detectores que aquí se proponen no utilizan derivadas de imágenes que pudieran ser lentas. La segunda razón es que estos operadores obtuvieron los rendimientos más altos en la detección de puntos de interés. Además, dada la Proposición 1 (Sección V.2.3), los dos operadores son similar a un filtro DoG, que también es una aproximación del Laplaciano; estos conceptos los utilizan todos los detectores descritos en el estado-del-arte (Lindeberg, 1998; Lowe, 1999; Mikolajczyk y Schmid, 2004), ver Figura 60. Adicionalmente, el inverso aditivo de K_{IPGP1} también se incluye con propósitos de comparación, al cual denominamos $-K_{IPGP1}$. Esto ayudará a ilustrar como los dos operadores se complementan, y como K_{IPGP1*} se relaciona con los dos.



Figura 60: Respuesta de los operadores propuestos para la detección invariante a escala.

Los operadores elegidos se embeben en el espacio escala de la siguiente forma,

$$K_{IPGP1_{t_j}}(\mathbf{x};t) = G_{t_j} * (G_{t_j} * I(\mathbf{x}) - I(\mathbf{x})) , \qquad (28)$$

$$K_{IPGP1_{t_j}^*}(\mathbf{x};t) = G_{t_j} * |G_{t_j} * I(\mathbf{x}) - I(\mathbf{x})| , \qquad (29)$$

donde j = 0, 1, ..., M, y M representa el número total de escalas que se analizan. Los operadores presentados arriba difieren de las medidas originales descritas en la Sección V.2.2 por las siguientes razones. Primero, para K_{IPGP1} los dos filtros Gaussianos se fijan a la misma escala t_i . No era claro como se puede incrementar el parámetro de la escala para cada filtro de manera independiente; la relación importante que si se mantiene es que $\sigma_1 > \sigma_2$ en la operación de K_{IPGP1} . Segundo, para K_{IPGP1*} el cuadrado del valor absoluto se omitió porque experimentalmente no produjo un incremento en aptitud; por ende, se consideró superfluo para el proceso de detección.

El proceso de detección funciona de la siguiente forma. El número total de escalas que se consideran es M = 20, y el parámetro de escala esta dado por $t_j = 1.2^j$, que implica un factor de 1.2 entre escalas adyacentes. Por lo tanto, la escala más fina y más gruesa están dadas por $t_0 = 1$ y $t_M = 1.2^M$ respectivamente. Sin embargo, las regiones de interés solo se detectan en 18 escalas diferentes, porque el proceso descrito para la elección de una escala característica requiere que cada escala tenga dos escalas adyacentes y esto no se cumple para los extremos j = 1 y j = 20; ver Ecuación 26. Por



Figura 61: La detección invariante a escala con K_{IPGP1^*} : **Izquierda**, la imagen de entrada I; **Medio**, *imagen de interés* I^* calculada en cada escala; **Derecha**, regiones detectadas después de elegir una escala característica. Las regiones se dibujan con un círculo de tamaño $3 \cdot t_n$.

ende, las regiones más grandes que se pueden detectar tendrán una escala asociada de $t_n = 38.33$, mientras que para las más chicas será de $t_n = 1.2$.

Cómo se menciona arriba, Lowe (1999) también emplea filtros DoG, sin embargo existen diferencias entre la forma en que el análisis de espacio escala se lleva acabo. En (Lowe, 1999) los filtros DoG se aplican entre escalas adyacentes, t_j y t_{j+1} , y cada respuesta en el espacio escala se muestrea para construir una representación en pirámide. Los detectores que aquí se proponen aplican el operador utilizando cada escala y la imagen original I con t = 0, que no ha sido suavizada; más aún, no se muestrea la respuesta, o bien, no se utiliza una pirámide (Mikolajczyk y Schmid, 2004) El proceso completo se ilustra gráficamente en la Figura 61.



Figura 62: La imagen de referencia (izquierda) y una imagen transformada (derecha) para cada secuencia de prueba. **a**) Boat (N = 5, rotación y escala), **b**) Asterix (N = 16, escala), **c**) Bip (N = 8, escala), **d**) Laptop (N = 20, escala), **e**) Van Gogh (N = 16, escala).

VII.7 Evaluación experimental

En la Figura 63 se presenta una comparación cualitativa entre los detectores propuestos, y dos detectores del estado-del-arte. Se puede observar que K_{IPGP1} y su inverso aditivo detectan regiones complementarias, mientras que K_{IPGP1*} extrae una combinación de máximos relacionados con los dos. Además, comparados con el Harris-Laplace y el Hessian-Laplace el tipo de rasgos, y la escala de los mismos, es similar. Sin embargo, cuantitativamente el rendimiento de los operadores evolucionados es mejor cuando se aplica a las secuencias de imágenes de prueba mostradas en la Figura 62. La Figura 64 presenta los resultados en la tasa de repetibilidad correspondiente, presenta el rendimiento en las cinco secuencias de prueba.


Figura 63: Regiones de interés detectadas. Cada región se representa con un círculo de radio $r = 3 \cdot t_n$ pixeles, con t_n la escala característica.



Figura 64: La tasa de repetibilidad calculada para cada secuencia de prueba.

Las gráficas muestran la tasa de repetibilidad con respecto a cada imagen en la secuencia, donde cada imagen de la secuencia esta progresivamente vista desde una escala menor. Por ejemplo, para la secuencia BIP el punto de vista de la imagen número cuatro es más cercana al punto de vista de la imagen base que el de la imagen diez.

VII.8 Resumen y conclusiones

En este capítulo se presentaron detectores invariantes a escala que están basados en operadores que fueron sintetizados utilizando GP. Dichos operadores promueven una detección de rasgos robusta e invariante con respecto a cambios durante la adquisición de la imagen, y también detectan rasgos que estén dispersos en el plano de la imagen.

Los operadores se emebebieron en el espacio lineal de escala que se genera con un kernel Gaussiano, y se compararon con detectores del estado-del-arte. Los resultados muestran que los detectores propuestos son, en promedio, mejores que los del estadodel-arte cuando se comparan utilizando la tasa de repetibilidad. Adicionalmente, los operadores que se utilizan tienen una estructura sencilla, algo que puede ser ventajoso en aplicaciones de tiempo real. Esto nos indica que los operadores encontrados por medio de la evolución artificial pueden tener un mejor rendimiento que diseños más complejos hechos por humanos, un resultado prometedor que le da validez al supuesto de que la evolución tiende a encontrar soluciones aptas y simples para problemas reales.

Como trabajo futuro, se observan dos líneas de investigación. Primero, utilizar una búsqueda evolutiva que contemple de manera explícita el análisis en espacio escala durante la evaluación de aptitud. Sin embargo, para lograr esto se necesita diseñar un sistema experimental apropiado para poder garantizar una implementación computacionalmente factible en cuestiones de tiempo. Segundo, extender el uso de los operadores evolucionados a la detección invariante a transformaciones proyectivas, un problema más complejo por su misma naturaleza.

En el siguiente capítulo la discusión de la tesis cambia de enfoque. Los Capítulos V, VI y VII tratan sobre las propuestas hechas en esta tesis referentes al problema de detección de rasgos locales, mientras que en los capítulos restantes se describe el trabajo realizado en torno a la descripción local de imágenes.

Capítulo VIII

Descriptores locales construidos a partir de una búsqueda evolutiva

En este capítulo se propone un enfoque para construir descriptores locales del tipo incluyente. El contenido de cada región de interés se describe utilizando datos estadísticos de color y textura que son seleccionados por un GA. A partir del descriptor local, se construyen modelos de mezclas de Gaussianas para representar a cada clase y también se utilizan como herramientas de clasificación. La búsqueda genética es guiada por una función de aptitud que considera la exactitud de la clasificación, la elección de una cantidad mínima de descriptores, y la separación entre los modelos construidos en el espacio del descriptor. La estrategia propuesta se aplica a dos problemás: 1) reconocimiento de objetos; y 2) reconocimiento de lugares en el mundo real. El capítulo incluye una descripción amplia de los experimentos realizados, el rendimiento de las soluciones obtenidas y las limitaciones del enfoque propuesto.

VIII.1 Descripción de la propuesta

El enfoque que se toma para la descripción local asume que cada región que se detecta en una imagen representa una instancia de la misma clase como se ilustra en la Figura 13. EL vector descriptor contiene rasgos estadísticos tomados de un espacio de búsqueda Φ de información de color y textura. Un GA realiza una búsqueda dentro de Φ para elegir al menor subconjunto de rasgos $F \subseteq \Phi$ que producen el mejor rendimiento basado en la exactitud de un proceso de clasificación probabilística. La búsqueda evolutiva también busca encontrar una separación máxima entre los modelos generados para cada clase. Los modelos de cada clase, ya sea un objeto o escena, se representan utilizando mezclas de Gaussianas, y se propone una extensión heurística del discriminante lineal de Fisher para estimar una medida de separación entre los modelos. Además, esta medida de separación nos permite determinar cual modelo representa a la clase más prominente entre todas. Debido al modelado multimodal que se utiliza, es posible capturar la variabilidad de información que se captura en las regiones locales extraídas de una misma clase, esta ventaja es más evidente para imágenes de objetos o escenas complejas. Una ventaja adicional de utilizar modelos de Gaussianas es que nos permite realizar una clasificación basada en confianza, y así identificar instancias de clases previamente desconocidas cuando la probabilidad de membresía a una clase esta por debajo de un umbral.

La Figura 65 presenta un diagrama a bloques del enfoque que se propone, aplicado específicamente al problema de reconocimiento de objetos. Dada la descripción del enfoque que tomamos, se pueden hacer las siguientes observaciones:

- La propuesta es similar a los trabajos que utilizan bolsas de rasgos visuales (Sivic y Zisserman, 2003; Willamowski *et al.*, 2004; Sivic *et al.*, 2005; Perronnin, 2008), aunque no de una forma explícita.
- El modelado, o descripción, de una imagen u objeto, no impone, y tampoco busca, un ordenamiento topológico o geométrico de los rasgos que se detectan.
- Es evidente que la representación será robusao contra oclusiones parciales, mientras que la invarianza a rotaciones y cambios geométricos o fotométricos estará ligado al detector de regiones que se utilice.



Figura 65: Diagrama de flujo del esquema propuesto, ilustrado para el problema de reconocimiento de objetos.

VIII.2 El espacio del descriptor

Se eligen 18 canales de información para construir el espacio de búsqueda Φ , diez de color y ocho de textura; un resumen se presenta en la Tabla XVI.

- Textura. Representar y analizar la textura en una imagen no es un problema trivial, y continua siendo uno de los problemás más importantes en visión (Zhu et al., 2005). Para este trabajo, utilizamos respuestas generales a operadores sencillos, que identifican bordes, puntos de interés, o regiones que contengan cambios abruptos en en la señal de la imagen.
- 2. *Color*. Para describr la información de color se utilizan los canales de diferentes espacios de color: *RGB*, el modelo de color lineal más conocido; *YIQ*, un modelo

Tabla XVI: Los canales de información que se utilizan para construir el espacio de búsqueda Φ . Para cada canal de información se incluye una breve descripción, y entre paréntesis se presenta un símbolo que correspondiente para cada cada uno.

Información	Descripción
Información del Gradiente	Gradiente, Magnitud y Orientación
	$(abla, \parallel abla \parallel, abla_{\phi}).$
Filtro Gabor	La suma de la respuesta a filtros Gabor con 8
	orientaciones diferentes (gab) .
Operadores de interés †	La respuesta a 3 operadores: Harris, IPGP1
	y $IPGP2$ ($K_{Harris}, K_{IPGP1}, K_{IPGP2}$).
Información de color	Todos los canales de 4 espacios de color:
	RGB, YIQ, Cie Lab, y rg cromaticidad
	(R,G,B,Y,I,Q,L,a,b,r,g).

basado en luminancia y crominancia; *Cie Lab*, un espacio de color que intenta ser perceptualmente lineal; *rg cromaticidad*, espacio de color de dos dimensiones sin información de intensidad (Kuehni, 2003).

Con el propósito de caracterizar cada región local, se calculan seis rasgos estadísticos para cada canal de información: media μ , desviación estándar σ , asimetría estadística γ_1 , curtosis γ_2 , entropía H y energía E; ver Tabla XVII. Cabe recordar que para una variable discreta **x** el momento central k esta dado por

$$\mu_k = \sum_{i=1}^n p_i (x_i - \mu)^k , \qquad (30)$$

donde p_i esta dado por la distribución discreta de **x**, o en este caso la frecuencia del histograma calculado en cada región, y μ es la media aritmética. Mientras que el momento central estandarizado esta dado por

$$\frac{\mu_k}{\sigma^k} , \qquad (31)$$

donde σ es la desviación estándar. Para todos los canales de información, la densidad de probabilidad se aproxima construyendo un histograma de 40 casillas entre el valor

Tabla XVII: Estadísticas calculadas sobre cada canal de información de la Tabla XVI. Son seis rasgos estadísticos calculados para 18 canales de información, esto produce un espacio de búsqueda Φ con 108 dimensiones.

Nombre	Ecuación	Descripción
Media aritmética	$\mu = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$	El promedio de todas las observaciones.
Desviación estándar	$\sigma = \left[\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right]^{\frac{1}{2}}$	Informa de la media de distancias que
		tienen los datos respecto de su media.
Asimetría	$\gamma_1 = \frac{\mu_3}{\sigma^3}$	El tercer momento estándar mide
		la asimetría de la distribución.
Curtosis	$\gamma_2 = \frac{\mu_4}{\sigma^4}$	Es una medida de lo $picudo$ de la
		densidad de probabilidad.
Entropía	$H = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$	Entropía de Shannon, hace referencia
		la cantidad media de información.
Energía	$H = -\sum_{i=1}^{n} \log p(x_i)$	Entropía logarítmica de energía.

máximo y mínimo que cada canal de información puede alcanzar.

VIII.3 Modelos de mezcla de Gaussianas

Este tipo de modelos, (en inglés, Gaussian Mixture Model: GMM) son herramientas útiles cuando se requiere modelar información multimodal, o cuando se requiere aproximar distribuciones más complejas, un ejemplo se presenta en la Figura 66. Un GMM se define como una suma ponderada de funciones de densidad de probabilidad normales,

$$p(\mathbf{x}; \Theta) = \sum_{c=1}^{C} \alpha_c \cdot \mathcal{N}(\mathbf{x}; \mu_c, \Sigma_c) , \qquad (32)$$

donde $\mathcal{N}(\mathbf{x}; \mu_{\mathbf{c}}, \Sigma_c)$ el la *c-ésimo* componente Gaussiano de múltiples variables con media μ_c , matriz de covarianza Σ_c , y un peso asociado α_c . Para estimar los parámetros de un GMM se emplea el algoritmo *expectación-maximización* (EM), esto se hace cuando



Figura 66: Un GMM de dos dimensiones compuesto de tres componentes Gaussianos.

se asume que el GMM contiene un número fijo de componentes Gaussianos (Paalanen et al., 2006). Alternativamente, si no se desea dar un número de componentes a priori, existen extensiones al EM que solo requieren de un número máximo de posibles componentes; e.g., el EM-Voraz (Verbeek et al., 2003) o el algoritmo Figueiredo-Jain (FJ) (Figueiredo y Jain, 2002).

La Figura 67 muestra un ejemplo de un problema de clasificación, donde se tienen tres clases y dos de esas clases son multimodales. El algoritmo FJ, con un limite de cinco componentes por clase, es capaz de aprender los parámetros para el modelo de cada clase en este ejemplo.

Se pueden clasificar datos con GMMs utilizando la regla de Bayes, o empleando una clasificación basada en confianza Paalanen *et al.* (2006). Una valor de confianza $\kappa \in [0, 1]$ y región de confianza $\mathcal{R} \subseteq \Phi$ para una función de densidad de probabilidad son tales que $0 \leq p(\mathbf{x}) < \infty$, $\forall \mathbf{x} \in \Phi$. κ es un valor de confianza relacionado a una región, que no es única, \mathcal{R} tal que

$$\int_{\Phi \setminus \mathcal{R}} p(\mathbf{x}) d\mathbf{x} = \kappa .$$
(33)

Una muestra \mathbf{x} que se encuentre dentro de \mathcal{R} es considerada como miembra de la clase



Figura 67: Un problema de clasificación con tres clases, donde dos de ellas, la Clase 1 (con tres componentes) y la Clase 3 (con dos componentes), son multimodales. En este caso, los parámetros del GMM para cada clase se estimaron con el algoritmo FJ.

modelada por p, o en el caso contrario se considera como un elemento desconocido o *outlier*.

VIII.3.1 El discriminante lineal de Fisher

El discriminante lineal de Fisher define la separación que existe entre dos distribuciones normales \mathcal{N}_i y \mathcal{N}_j como el resultado de la siguiente razón,

$$S_{i,j} = \frac{(\mathbf{w}(\mu_i - \mu_j))^2}{(\mathbf{w}^T(\Sigma_i + \Sigma_j)(\mathbf{w}))} , \qquad (34)$$

donde $\mathbf{w} = (\Sigma_i + \Sigma_j)^{-1} (\mu_i - \mu_j)$ (Fisher, 1936). Se debe notar que *S* esta definida para Gaussianas sencillas de un solo componente. Si se va a aplicar esta medida para dos GMMs, es necesario utilizar una versión ponderada \hat{S} que contemple los pesos α_i y α_j de los componentes Gaussianos, tal que

$$\widehat{S}_{i,j} = \frac{S_{i,j}}{1 + \alpha_i + \alpha_j} \,. \tag{35}$$

La separación entre dos componentes de diferentes modelos \mathcal{N}_i^a y \mathcal{N}_j^b es inversamente proporcional a la suma de sus pesos α_i^a y α_j^b . De esta forma, la separación entre componentes con un peso combinado pequeño, que significa que tienen una influencia menor sobre su modelo correspondiente, va a *parecer* que es mayor con respecto a la separación entre dos componentes que tienen pesos mayores.

Por lo tanto, si decimos que C_a y C_b representan al número de componentes que contienen $p_a(\mathbf{x}; \Theta_a)$ y $p_b(\mathbf{x}; \Theta_b)$ respectivamente, entonces $S^{a,b}$ representa la matriz simétrica de separación *aparente* de tamaño $C_a \times C_b$ que contiene la separación ponderada $\widehat{S}_{i,j}$ entre cada componente de p_a con respecto a cada componente de p_b . Finalmente, la medida de separación aparente \mathcal{S} entre p_a y p_b esta dada por

$$\mathcal{S}^{a,b} = inf(S^{a,b}) \ . \tag{36}$$

VIII.4 El algoritmo genético

Como se menciona arriba, el descriptor que se va a construir se va a utilizar para reconocer objetos o escenas reales. Por lo tanto, se esta planteando un problema de

Parámetro	Descripción y/o valor
$Representaci\'on$	Cromosoma binario.
Selección	Proporcional a la aptitud (Goldberg, 1989).
Cruce	Cruce de máscara o multipunto; $p_c = 0.9$.
Mutación	Mutación binaria de un bit; $p_{\mu} = 0.1$.
Supervivencia	Elitismo del 15%.

Tabla XVIII: Parámetros del GA de búsqueda y aprendizaje.

aprendizaje real, por lo que vamos a asumir que existe un total de l clases diferentes. Entonces, para cada clase L se cuenta con n imágenes diferentes, estas pueden considerarse como vistas representativas del objeto o escena, y son las que se utilizan durante la etapa de entrenamiento. Para cada imagen de entrenamiento, de cada clase, se aplica un detector invariante a escala y se extraen las regiones detectadas. Todas las regiones detectadas en las imágenes de entrenamiento son normalizadas, y se calculan los 108 rasgos estadísticos contenidos en Φ . Después, el GA se encarga de elegir el subconjunto final de rasgos que serán utilizados para construir el descriptor. La evaluación de aptitud considera la exactitud de clasificación que se obtiene con los GMMs de cada clase en base al descriptor que cada individuo especifíca. Los parámetros básicos del GA se presentan en la Tabla XVIII, y a continuación se describe con mayor detalle la representación o cromosoma de las soluciones, y la función de aptitud empleada.

VIII.4.1 Representación de la solución

La representación que se utiliza para cada solución, o individuo en la población, es una cadena binaria $B = (b_1, b_2, ... b_{108})$ de 108 bits. Cada bit en la cadena esta asociada con una de las 108 dimensiones en Φ . Por lo tanto, si el bit b_i es igual a 1, entonces el rasgo correspondiente se selecciona, mientras que si es igual a 0 sucede lo contrario. De tal forma, que cada solución especifica cuales de los rasgos estadísticos que conformaran al descriptor. Entonces, para una solución candidata B, dada una región \mathbf{A} el vector descriptor $D_{\mathbf{A}}$ estará compuesto por la concatenación de todos los rasgos estadísticos elegidos por B.

VIII.4.2 Evaluación de aptitud

Como se observa en el diagrama de la Figura 65, el propósito del descriptor incluyente es facilitar la tarea que tiene que realizar el sistema de reconocimiento que se esta proponiendo. Por lo tanto, la función de aptitud debe de considerar el rendimiento del sistema de reconocimiento para poder establecer un valor de aptitud coherente para cada individuo. Además, con el propósito de eliminar información redundante y así obtener un descriptor que se calcula de forma rápida y sencilla, también es deseable obtener un descriptor compacto.

Criterio de optimización

Considerando los puntos que se mencionan arriba, el diseño del descriptorse plantea en términos de un problema de búsqueda/optimización. Además, es posible establecer que el problema de optimización abordado contempla dos objetivos generales, que son:

- Minimizar las dimensiones, o número de rasgos estadísticos, que utiliza el descriptor; o bien, el individuo óptimo B^o debe de minimizar el número de unos, |B^o|, contenidos en su cromosoma.
- 2. Maximizar la habilidad del descriptor para caracterizar las regiones de clases diferentes; i.e., dado un descriptor B se desea maximizar el rendimiento del sistema de reconocimiento, este término lo dennomiamos C.

En base a esto, se define el siguiente criterio de optimización,

$$(B^{o}, \mathcal{C}^{o}) \leftarrow \underset{B,\mathcal{C}}{\operatorname{arg\,min}} \left\{ \frac{|B|}{\mathcal{C}} \right\} ,$$
 (37)

donde B^o y \mathcal{C}^o son los valores óptimos que se buscan para B y \mathcal{C} respectivamente. El numerador en la expresión de la derecha se refiere al número de unos contenidos enl cromosoma del individuo B, o bien el tamaño del descriptor que se genera. El denominador, por otro lado, depende de la forma en que se modelan los datos, y de la manera en que se evalúa el rendimiento del sistema. Se puede observar que en realidad solo existe una variable independiente que se puede modificar directamente, B. Mientras que \mathcal{C} depende directamente de las características de B. Sin embargo, no es posible derivar \mathcal{C} de forma analítica a partir de B, sino que debe de medirse de manera experimental, y además depende de procesos no determinísticos durante el aprendizaje de los modelos (EM). Por lo tanto, la expresión de arriba proporciona una noción intuitiva de lo que se quiere lograr con la búsqueda evolutiva.

Cada clase que se considera es representada con GMMs, y el rendimiento del sistema depende de la exactitud que alcanza el descriptor con un conjunto de imágenes de entrenamiento. Por lo tanto, C puede definirse como

$$\mathcal{C} \propto \mathcal{A} \cdot \inf \left\{ \mathcal{S}^{p_i, p_j} \right\} \ , \ \forall \ p_i, p_j \in \mathcal{P} \ , i \neq j \ , \tag{38}$$

donde \mathcal{A} es la exactitud de la clasificación de las regiones locales, y \mathcal{P} es el conjunto de todos los GMMs; existe un modelo para cada clase.

Función de evaluación

Durante la evaluación de cada individuo, se construye el descriptor correspondiente, se generan los modelos de cada clase, y se evalúa el rendimiento experimental del sistema de reconocimiento.

Para cada clase L existe un GMM correspondiente $p_L(\mathbf{x}; \Theta_L)$, lo cual genera un conjunto de GMMs $\mathcal{P} = \{p_L(\mathbf{x}; \Theta_L)\}, \text{ con } |\mathcal{P}| = l$, uno para cada clase. Los parámetros de estos modelos se aprenden utilizando las imágenes representativas de cada clase,

las de entrenamiento. De estas imágenes se utiliza el 70% de las regiones para entrenamiento, y el 30% restante se emplea para probar la clasificación de las regiones locales utilizando regla de Bayes; finalmente, este proceso regresa un valor de exactitud \mathcal{A} correspondiente.

De tal forma que la aptitud se asigna en base a la siguiente expresión derivada de la Ecuación 37,

$$f(B) = \begin{cases} \frac{|B|^{\alpha} + 1}{\mathcal{A}^2 \cdot inf(\mathcal{S}^{p_i, p_j})} & \forall \ p_i, p_j \in \mathcal{P} \ , i \neq j \ , \ cuando \quad \forall \ \mathcal{A}_i > 0 \ , \\ \frac{|B| + 1}{\varepsilon} & en \ caso \ contrario \ . \end{cases}$$
(39)

En la Ecuación 39, |B| es el número de unos que contiene el cromosoma del individuo $B, \varepsilon = 0.01, y \alpha$ es un factor de peso. La aptitud se evalúa considerando dos casos principales. El primero caso se aplica cuando el algoritmo EM converge para todas las clases, de tal forma que los individuos más aptos minimizan el tamaño del descriptor que se construye y a su vez maximizan la exactitud promedio \mathcal{A} que se obtiene con modelos para todas las clases. Aún más, el término $inf(S^{p_i,p_j})$ promueve una separación grande entre las clases ya que elige el ínfimo de todas las separaciones aparentes entre las clases modeladas en el conjunto \mathcal{P} . El segundo caso de la Ecuación 39 se aplica cuando el algoritmo de entrenamiento no converge; i.e., cuando no se genera un GMM para alguna de las clases contempladas. Por lo tanto, la aptitud del individo se penaliza fuertemente, ya que es primordial que el descriptor pueda representar la información extraída de todas las clases, no solo de algunas.

VIII.5 Evaluación experimental

Para poder evaluar el rendimiento del sistema propuesto, se abordan dos problemás clásicos en la literatura de visión: (a) reconocimiento de objetos; y (b) reconocimiento

de lugares reales.

VIII.5.1 Reconocimiento de objetos

Se evalúa el rendimiento del enfoque cuando se aplica a un problema de reconocimiento de objetos; el problema se define de la siguiente manera. Se comienza con un conjunto de objetos M, se elige un subconjunto $N \subset M$ de objetos que se utilizan para entrenar al sistema. Por lo tanto, el GA entrega un GMM para cada objeto en M, utilizando un entrenamiento de *uno vs. todos.* En base a esto, el sistema debe ser capaz de realizar tres tareas específicas cuando recibe una imagen de algún objeto tomado del conjunto M:

- 1. Reconocimiento. Debe ser capaz de reconocer objetos conocidos, objetos del conjunto N. Para esto, se extraen las regiones de la imagen de entrada, se construye el descriptor para cada región especificado por B^o , y se clasifican utilizando confianza con los modelos en \mathcal{P} (uno para cada objeto de N), con un umbral de confianza $\kappa = 0.95$. Si la mayoría, arriba del 60%, de las regiones caen dentro de la región de confianza de una de las clases conocidas entonces se dice que se ha detectado el objeto correspondiente.
- Detección de un objeto nuevo. Identificar si un objeto no es conocido, o bien, si el objeto pertenece a *M**N*. Si una cantidad significativa de las regiones se clasifican como desconocidas (outliers), arriba del 60%, entonces el objeto se identifica como novedoso y se entrena un GMM nuevo para este objeto que posteriormente se agrega a *P*.
- 3. **Objeto Prominente.** El sistema es capaz de identificar cual es el objeto más prominente del conjunto de objetos *M*. Para esto se elige el objeto *i* que satisface

la siguiente condición:

$$i^{p} \leftarrow \underset{i}{\operatorname{arg\,max}} \{ \mathcal{S}^{p_{i},p_{j}} \} \ \forall \ p_{j} \in \mathcal{P}^{o} \ con \ i \neq j \ ,$$

$$(40)$$

donde i^p representa el objeto más prominente para el cual se tiene un modelo en \mathcal{P} . Esta medida también se puede utilizar para ordenar a los objetos en base a cuan distintivos son.

Implementación

El código para todos los experimentos fue escrito en su mayoría en MATLAB, por ejemplo se utilizo el Toolbox GMMBAYES¹ para realizar todo lo relacionado con los GMMs, y el Toolbox de Genetic Algorithms for Optimization² se utilizó como parte del código escrito para el GA del sistema. La base de imágenes que se utiliza para probar la aplicación es la COIL-100, la Figura 68 presetna algunos ejemplos del tipo de objetos contenidos en la base de imágenes (Nene et al., 1996). En esta base de datos todos los objetos se observan desde 72 puntos de vista diferentes, rotando al objeto progresivamente cinco grados a la vez. Para agilizar las pruebas, se detectan regiones de interés de cada imagen y se etiquetan de acuerdo al objeto al cual corresponden. El detector que se utiliza es la versión invariante a escala del K_{IPGP1*} , que se presenta en la Sección VII.6 de esta tesis. Los parámetros básicos del algoritmo son los mismos en todos los experimentos, solo modificando el número de objetos utilizados y el tamaño de los conjuntos (M, N). Se realizaron tres experimentos, un resumen de estos se presenta en la Tabla XIX. Los clasificadores GMM se entrenaron utilizando el algoritmo EM con un componente Gaussiano, y si una solución no se encontraba entonces el algoritmo se ejecutaba con dos componentes, y así sucesivamente. Como se menciona arriba, para cada experimento se realizan pruebas para el reconocimiento de objetos y la detección

¹GMMBAYES Matlab Toolbox http://www.it.lut/project/gmmbayes

²Genetic Algorithms for Optimization Toolbox de Andrey Popov http://automatics.hit.bg



Figura 68: Ejemplos de objetos en la base de imagenes COIL-100. El primer renglon muestra varios puntos de vista para un mismo objeto. El segundo renglón muestra cuatro objetos diferentes.

Tabla XIX: Resumen de los tres experimentos realizados para el reconocimiento de objetos.

	Μ	Ν	Objetos del COIL	Generaciones
Experimento 1	10	5	Objetos 1 - 10.	30
Experimento 2	20	10	Objetos 20 - 40.	30
Experimento 3	40	25	Objetos 1 - 40.	40

de objetos nuevos, además de la capacidad que se tiene para identificar el objeto más prominente del conjunto.

La Tabla XX presenta un resumen de los resultados que se obtuvieron para cada experimento. Se presenta la exactitud promedio en la clasificación de las regiones de entrenamiento para el mejor individuo, el valor de aptitud, las dimensiones del descriptor que se construye, los rasgos estadísticos utilizados, los errores producidos durante el reconocimiento de objetos en N, y el objeto más prominente del grupo de entrenamiento.

En la Tabla XX todos los experimentos tienen un valor de exactitud similar, sin embargo la aptitud que se les asigna es diferente, esto se debe a los otros términos incluidos en la función de evaluación. Pero particularmente se atribuye al criterio de separación, porque con más objetos el espacio en cual residen los modelos se vuelve cada vez más aglomerado.

Tabla XX: Rendimiento del algoritmo despues de la fase de entrenamiento con los objetos en N.

Exp.	\mathcal{A}'	$\mathbf{f}(\mathbf{B^o})$	$ \mathbf{B^o} $	Rasgos Estadísticos	Error	$\mathbf{i}^{\mathbf{p}}$
1)	99.6	0.5	27	$\nabla_{(\gamma_2,H)}, \ \nabla \ _{(\sigma,\gamma_2)}, \nabla_{\phi(\gamma_2)}, K_{Harris(E)},$	0	4
				$K_{IPGP1(\sigma)}, R_{(\mu,H)}, G_{(\sigma,\gamma_1)}, B_{(\mu,\sigma,\gamma_1,H)},$		
				$Y_{(\mu,\gamma_2,H,E)}, I_{(\sigma,H)}, L_{(\sigma,E)},$		
				$a_{(\mu,\sigma)}, \ b_{(\sigma,E)}, \ g_{(\mu)}$		
2)	99.2	1.5	43	$ abla_{(\mu,\sigma,\gamma_2,E)}, \parallel abla \parallel_{(\sigma,\gamma_2,H)}, abla_{\phi_{(\mu,\sigma,\gamma_2)}}$	0	25
				$K_{Harris(\gamma_1,H)}, K_{IPGP1(\mu,E)}, K_{IPGP2(\gamma_1,E)},$		
				$gab_{(\gamma_1)}, R_{(\mu,\sigma,E)}, G_{(\mu)}, B_{(\sigma,\gamma_1,\gamma_2,H)}, Y_{(\mu,\gamma_2,H,E)},$		
				$I_{(\sigma,H)}, Q_{(\gamma_2,H)}, L_{(\mu)}, a_{(\mu,\sigma,H)},$		
				$b_{(\mu,\sigma,E)}, \ r_{(\mu,\sigma)}, \ g_{(E)}$		
3)	98.7	6.4	37	$ abla_{(\mu,\sigma,\gamma_2)}, \parallel abla \parallel_{(\mu,\sigma,\gamma_1,\gamma_2,H,E)}, abla_{\phi(\gamma_1,\gamma_2,H)}, abla_{\phi(\gamma_1,\gamma_2,H)}$	0	4
				$K_{Harris(\gamma_2,E)}, K_{IPGP1(H)}, K_{IPGP2(\gamma_2,H)},$		
				$gab_{(\mu)}, R_{(\mu,\gamma_1,E)}, G_{(\mu)}, B_{(\gamma_2,H)}, Y_{(\mu,\sigma,\gamma_2,H,E)},$		
				$I_{(E)}, Q_{(\mu,\gamma_1)}, a_{(\gamma_1)}, b_{(\sigma)}, r_{(\sigma,H)}, g_{(\mu,\sigma)}$		

En la Tabla XXI se presentan los resultados después de probar cada solución con todo el conjunto M, o bien, en la detección de objetos novedosos, probados uno por uno. Los datos que se incluyen son la exactitud con que se clasifican las regiones locales, los objetos que fueron clasificados erróneamente, y los tres objetos más prominentes del conjunto M.

La Figura 69 presenta los objetos que fueron utilizados en cada experimento, identifica el objeto que no fue clasificado de forma correcta, y también muestra aquellos

	$\mathcal{A}_{\mathbf{M}}$	Errores	Objetos prominentes
Exp.1	99.72	ninguno	objetos 4, 7, 3
Exp.2	99.04	objeto 32	objetos $36, 38, 25$
Exp.3	98.68	objeto 32	objetos 36, 28, 4

Tabla XXI: Rendimiento del sistema en la detección de objetos nuevos.



Figura 69: Los primero 40 objetos de la base de datos COIL-100, estos son los objetos utilizados durante la experimentación. Las imágenes utilizadas en los primeros dos experimentados están marcados mientras que todas las imágenes se utilizan en el tercero. Los objetos más prominentes elegidos por el algoritmo se muestran encerrados por un círculo. El objeto 32 es el único con el cual fallo el sistema.

objetos que fueron identificados como prominentes en cada uno de los experimentos.

En general, se puede apreciar que el problema de reconocimiento y detección de objetos nuevos se resolvió de manera casi perfecta. Esto se podía esperar debido a los valores de exactitud tan altos que alcanza el clasificador. Además, los objetos que se identifican como prominentes, o los más sobresalientes, son elecciones razonables. En todos los casos son objetos que son muy uniformes, lo que implica que tendrán un modelo asociado muy compacto. Inclusive, para un observador humano estos objetos



Figura 70: Gráficas de convergencia, para el logaritmo de la aptitud del mejor individuo de cada generación.

son de los primeros que llaman la atención por su forma y color distintivo.

La Figura 70 presenta las gráficas de convergencia utilizando el logaritmo de la aptitud del mejor individuo en cada generación de la evolución. Es necesario observar que las gráficas no son monótonas, aunque el GA utiliza elitismo. Esto se debe a que la aptitud de cada individuo depende del algoritmo de aprendizaje EM, que no es determinístico y que puede generar resultados diferentes cada vez que se ejecuta, aun cuando toma los mismos datos de entrada.

Discusión

Los experimentos realizados para el problema de reconocimiento de objetos generaron resultados prometedores. Para los tres experimentos el algoritmo fue capaz de entrenar clasificadores muy exactos utilizando solo una fracción de los posibles rasgos estadísticos.

Sin embargo, es importante mencionar que en todos los experimentos los modelos que se generaron para cada clase solo contenían un componente Gaussiano, esto se atribuye a dos causas. Primero, los objetos son pequeños y muchos de ellos exhiben patrones repetitivos sobre su superficie; por lo tanto la característica multimodal no era de suma importancia. Segundo, los descriptores que se construyen, aunque utilizan mucho menos de los 108 rasgos posibles siguen siendo grandes. Por lo tanto, las Gaussianas pueden capturar la variabilidad de una clase utilizando un solo componente de manera más fácil que si el espacio fuera menos complejo.

Adicionalmente, las gráficas de convergencia muestran dos patrones significativos. Primero, iniciando con una población aleatoria los resultados que se obtienen son bastante malos, los individuos en estas generaciones suelen ser evaluados con el segundo caso de la función de aptitud (ver Ecuación 39); en estos casos el algoritmo EM no converge para algunas de las clases. Después, se encuentra una buena solución y su información genética se propaga al resto de la población, esto produce individuos que se evalúan con el primer caso de la función de aptitud. Una vez que se están generando clasificadores válidos para cada objeto, entonces el GA puede comenzar a optimizar el tamaño del descriptor.

Con respecto a la detección de objetos nuevos, el enfoque produce resultados que son casi perfectos, con un error, el objeto 32 de la base de datos COIL-100. Sin embargo, se debe de notar que el objeto 32 es prácticamente idéntico al objeto 29, solo con una pequeña diferencia en el color de una parte del objeto.

Finalmente, la identificación de los objetos prominentes también arrojó resultados razonables. Los objetos identificados carecen de textura y exhiben una superficie más homogénea que el resto de los objetos. Además, para muchos de los otros objetos existe una contraparte similar, por ejemplo hay más de un carrito de juguete y varias cajas pequeñas.

VIII.5.2 Reconocimiento de lugares reales

La tarea de reconocer lugares (en inglés, Place Recognition) es esencial en escenarios donde un sistema artificial autónomo se va a desplazar dentro de un medio humano (e.g., oficina o habitaciones) y va a tener interacciones similares a las que tuviera un humano. El reconocimiento se puede realizar de diferentes maneras, por ejemplo utilizando señales artificiales de radio o infrarrojas, pero aquí nos interesa estudiar el problema desde el paradigma de visión artificial. Existe mucho trabajo en esta línea de investigación, pero una revisión detallada de la literatura esta fuera del alcance de esta tesis, razón por la cual solo se mencionan algunos ejemplos ilustrativos.

Torralba *et al.* (2003) presentan un sistema que combina el reconocimiento de objetos con el reconocimiento de lugares, empleando un análisis que depende del contexto. El sistema utiliza rasgos globales de las imágenes para realizar el reconocimiento del lugar. De tal forma que el lugar que se identifica proporciona el contexto bajo el cual se lleva acabo el reconocimiento de objetos dentro de la escena. Para simplificar la tarea de reconocer los lugares, el sistema impone restricciones topológicas deducidas de la relación espacial que existe entre los lugares físicos, representado a través de un modelo oculto de Markov. Esto es posible porque se asume que el sistema forma parte de un sistema móvil que esta limitado en tiempo y en espacio.

A diferencia de ese trabajo, el enfoque que se toma aquí es el de realizar el reconocimiento de lugares utilizando solo la información visual, sin una hipótesis Markoviana con respecto a las relaciones espaciales y temporales que existen entre los lugares, o más específicamente entre las imágenes tomadas de una secuencia continua. Por lo tanto, el enfoque que se toma esta relacionad con el problema de reconocimiento de objetos. Además, en lugar de utilizar una descripción holística de cada imagen, se emplea un enfoque disperso y local utilizando regiones de interés como se muestra en la Figura 71.

Wang *et al.* (2006) resulven el problema de localización para un sistema de robótico móvil por medio de la visión. Los autores utilizan un detector invariante a escala, el detector Harris-Laplace, y describen cada región utilizando el descriptor SIFT. El enfoque



Figura 71: Regiones de interés detectadas en imágenes de cuatro lugares reales. (1) El primer renglón muestra las imágenes originales; (2) El segundo renglón muestra las regiones detectadas en cada imagen.

que proponen se basa en técnicas de indexado de imágenes Schmid y Mohr (1997), una elección que se desprende directamente del descriptor discriminante que utilizan. Sin embargo, el uso de SIFT es una limitante para lugares en donde la información más prominente proviene del color. Más aún, aunque el color puede ser problemático en situaciones donde la iluminación no es constante, este problema se reduce en ambientes de interiores.

Los dos ejemplos anteriores ilustran que se pueden tomar enfoques muy diferentes, siempre en función de la instancia del problema que se quiere resolver.



Figura 72: El problema de reconocer lugares en el mundo real. (1) El primer renglón muestra varias vistas de un mismo lugar (Laboratorio EvoVisión); (2) El segundo renglón muestra cuatro lugares diferentes (Laboratorio de Estudiantes, Laboratorio de Computación, Laboratorio EvoVisión, y Oficina).

Definición del problema

En esta tesis, rl problema de reconocimiento de lugares se define de la siguiente forma. Dado un conjunto L con l lugares del mundo-real, y un conjunto de n imágenes representativas para cada lugar, entrenar un sistema que sea capaz de reconocer que lugar se esta observando en una serie de imágenes de prueba que son diferentes a las imágenes que se utilizaron durante el entrenamiento. La única restricción es que la secuencia de prueba solo contiene imágenes de uno de los l lugares que fueron aprendidos. La ultima restricción se pudiera relajar si se agrega una clase genérica llamada *desconocidas*, pero para un ambiente cerrado es razonable asumir que se conoce el conjunto finito de lugares que se deben de reconocer.

La Figura 72 presenta algunos imágenes que sirven para enfatizar la dificultad del problema que se esta abordando. El primer renglón contiene imágenes tomadas de diferentes vistas de un mismo lugar. Aquí se puede observar que existe mucha variabilidad dentro del mismo lugar; i.e., variabilidad intra-clase. En el segundo renglón, se muestran imágenes de cuatro lugares diferentes; sin embargo, se puede observar que todas estas imágenes comparten rasgos visuales en común. Es evidente que no es trivial encontrar una descripción que permita capturar las similitudes entre imágenes de la misma clase, y al mismo tiempo acentué las diferencias que existen entre clases diferentes.

Un algortimo genético de dos etapas

Lo resultados de la Sección VIII.5.1 sugieren que el método desarrollado en este capítulo es capaz de resolver problemas de reconocimiento complejos. Sin embargo, se identificaron algunas limitantes en el enfoque que se esta tomando. Por ejemplo, el espacio Φ es bastante grande, y aunque el número de rasgos en el descriptor si se minimizó con el GA, aún así se seleccionaba una cantidad bastante grande, entre 27 - 43, de rasgos. Debido a esto, cada clase se podía modelar utilizando solo un componente Gaussiano; esto hacia el uso de GMMs completamente superfluo para aquel problema. Más aún, la aplicabilidad del método en un sistema de tiempo real, por ejemplo, se ve reducido debido a la cantidad de rasgos estadísticos que se tienen que calcular para describir cada región.

Para solucionar este problema, se propone usar un GA de dos etapas, o bien, dos GAs que se ejecutan de manera secuencial, donde la solución producida por el primero sirve para restringir la búsqueda realizado en el segundo. De tal forma que el primer GA se enfoca en *explorar* el espacio de búsqueda Φ , y el segundo busca explotar una región del espacio Φ donde se espera que existan muchas soluciones con aptitud alta. El proceso se implementa de la siguiente forma:

1. Primera etapa: Exploración. El GA se ejecuta de manera normal, con el peso $\alpha = 2$ en la Ecuación 39. De esta forma, el término que promueve la construcción de un descriptor de pocas dimensiones tendrá mayor influencia sobre el valor de

	Característica	Generaciones	Población	Peso α
Etapa 1	Exploración.	30	50	2
Etapa 2	Explotación.	30	50	1

Tabla XXII: El GA de dos etapas.

aptitud; por lo tanto, el GA favorece a soluciones que utilizan menos rasgos. Adicionalmente, debido a que en esta etapa se pueden generar individuos con muchas dimensiones, se utiliza el algoritmo EM básico durante el entrenamiento con un solo componente Gaussiano para cada modelo. Al final de la ejecución, el mejor individuo B^o que se produce sirve para acotar un subespacio $\Phi^* \subset \Phi$ que se utilizará durante la búsqueda evolutiva en la siguiente etapa.

2. Segunda etapa: Explotación. El funcionamiento de la segunda etapa se diferencia en tres aspectos de la etapa anterior. Primero, el espacio de búsqueda ahora esta definido por Φ* en vez de Φ, lo que normalmente será un espacio mucho más compacto. Segundo, con el espacio de búsqueda más pequeño, se espera que las dimensiones de las componentes en el GMM sean menores, lo que va a permitir una representación multimodal; por ende, se utiliza el algoritmo FJ para estimar los parámetros de los GMMs. Tercero, el peso α = 1 en la Ecuación 39 se reduce para que el termino relacionado con la exactitud tenga mayor peso en el cálculo de la aptitud. La salida del GA de dos etapas es un conjunto de rasgos F ⊂ Φ, que se utiliza para construir al descriptor final.

En las dos etapas el entrenamiento se realiza en el estilo *todos vs. todos* porque se asume que solo se van a reconocer imágenes de lugares conocidos. La Tabla XXII resume las diferencias entre las dos etapas y especifica los parámetros que utiliza el GA en cada una.

Criterio de reconocimiento

Dada una imagen nueva I tomada de uno de los lugares conocidos, se detectan regiones de interés y se construye el descriptor utilizando los rasgos especificados por F. Las regiones se clasifican utilizando la regla de Bayes con los modelos en \mathcal{P} . Después, si una mayoría, no absoluta, de las regiones se clasifican como si pertenecieran al lugar L entonces se asume que este es el lugar visto en I. El método de reconocimiento se evalúa con dos experimentos reales.

Experimento 1

El primer experimento para el sistema de reconocimiento contiene cuatro lugares: (1) Cuarto, (2) WC, (3) Comedor, y (4) Sala. Las imágenes de prueba y entrenamiento se tomaron de secuencias representativas para cada lugar; cada imagen a color tiene un tamaño de 1Mb, ejemplos se pueden ver en la Figura 73.

Para este experimento se utiliza el detector invariante a escala de Kadir y Brady (2001). Para simplificar el proceso de entrenamiento, y debido al tamaño de las imágenes, se ajusto el número máximo de regiones por imagen a 3,500. De las imágenes de entrenamiento, el 70% de las regiones detectadas se utilizan para generar los modelos, y el restante para la validación de la exactitud. La Tabla XXIII resume los detalles del experimento: el número de fotos de entrenamiento, el número de componentes en el GMM de cada lugar, el número de imágenes de prueba, y el número de errores hechos durante el reconocimiento.

Adicionalmente, el primer renglón de la Tabla XXIV presenta las características del mejor individuo generado por el GA de dos etapas, con los siguientes datos: valor de aptitud, el tamaño de Φ^* , la cardinalidad de F, los rasgos estadísticos elegidos, y la exactitud durante la validación.

El mejor individuo utiliza seis canales de información, dos de textura y cuatro de



Figura 73: Ejemplos de imágenes representativas para cada uno de los lugares utilizados en el experimento uno. Por columna: (1) Cuarto, (2) WC, (3) Comedor, y (4) Sala.

color. Los dos rasgos de textura pueden considerarse intuitivos, ya que el gradiente siempre se ha utilizado como información de textura básica, y el segundo es un operador DoG que extrae bordes prominentes. Con respecto a la información de color, es interesante ver que dos componentes del espacio CieLab son elegidos, ya que este espacio es una aproximación de la capacidad de visión a color que tenemos los humanos. Los otros canales de información que el GA elige son el canal Q del espacio YIQ y el canal R del espacio RGB.

Finalmente, las gráficas de convergencia del GA se presentan en la Figura 74a,b para cada una de las etapas.

Lugar	Im. Entrenamiento	GMM # Components	Im. Prueba	Error
Room	7	3	2	0
WC	5	6	2	0
Diner	9	4	2	0
Lounge	9	8	2	0

Tabla XXIII: Descripción del Experimento 1.

Tabla XXIV: Rendimiento en cada experimento y caracteristicas del descriptor que se obtiene. Se muestra el valor de aptitud, el tamaño de Φ^* , la cardinalidad de F, los rasgos estadísticos que fueron elegidos, y la exactitud durante validación.

Exp.	Apt.	$ \Phi^* $	F	Rasgos	\mathcal{A}
1)	0.0061	30	7	$\nabla_{\phi(\sigma)}, K_{IPGP1(\gamma_1)}, R_{(\mu,\sigma)}, Q_{(\gamma_2)}, a_{(\sigma)}, b_{(\mu)}$	70.75%
2)	0.0053	36	14	$gab_{(\mu,\gamma_2)}, K_{IPGP2(H)}, G_{(\sigma)}, B_{(\sigma)},$	
				$Y_{(\gamma_1)}, I_{(\mu,\sigma)}, Q_{(\sigma)}, L_{(\sigma)}, a_{(H)}, b_{(\mu,H)}, r_{(\sigma)}$	74.73%

Experimento 2

En este experimento, el problema consta de ocho lugares reales dentro del edificio de Física Aplicada en CICESE, Ensenada, México; estos son: (1) Laboratorio de Estudiantes, (2) Laboratorio de Computación, (3) Laboratorio de EvoVisión, (4) Armarios, (5) Primer Piso, (6) Segundo Piso, (7) Oficina, y (8) Correo. Se capturaron dos secuencias de imagenes en cada lugar, una durante la mañana y la otra en la tarde para tener variedad en el tipo de iluminacion; un ejemplo de cada lugar se muestra en la Figura 75.

Todas las imágenes esta en formato JPEG a color de tamaño 320×340 . Para este experimento se utilizó el detector invariante a escala K_{IPGP1^*} , sin restricciones con respecto a la cantidad de regiones por imagen; en la Tabla XXV se dan más detalles.

En algunos lugares se detectaron más regiones de entrenamiento que en otros, esto se



Figura 74: Graficas de convergencia para cada experimento; se muestran los resultados para cada etapa del GA.

Lugar	Im. Entrenar	Regiones	# Components	Im. Prueba	Error
Estudiantes	17	5469	3	17	4
Computación	14	5477	3	14	0
EvoVisión	27	4941	2	26	12
Armarios	14	507	3	14	14
Primer Piso	13	4534	3	12	11
Segundo Piso	20	2263	2	19	12
Oficina	17	6756	2	16	1
Correo	10	1139	3	10	10

Tabla XXV: Descripción del Experimento 2.



Figura 75: Ejemplos de imágenes de los lugares utilizados en el segundo experimento: (1) Laboratorio de Estudiantes, (2) Laboratorio de Computación, (3) Laboratorio de EvoVisión, (4) Armarios, (5) Primer Piso, (6) Segundo Piso, (7) Oficina, y (8) Correo.

debe a que algunos lugares cuentan con objetos o superficies más ricos en textura, como la Oficina o los Laboratorios, miestras que otros tienen una apariencia más homegenea, como los pasillos en cada piso.

El segundo renglón de la Tabla XXIV describe al mejor individuo que generó la búsqueda genética. En este experimento, el descriptor que se construye emplea once canales de información, dos son de textura y el restante de color. En el caso de textura, uno de los canales de información elegido por el GA son la respuesta a los filtros Gabor, una herramienta muy común en la literatura relacionada con el análisis de textura en imágenes. El segundo tipo de información es el determinante de la matriz Hessiana suavizada $K_I PGP2$, información que normalmente solo se utiliza para la detección de puntos de interés.

En el caso del color, el GA escogió una variedad muy amplia de rasgos, utilizando los modelos lineales y no lineales, inclusive dos de ellos en su totalidad, YIQ y CieLab.

	Estu.	Compu.	\mathbf{EV}	Arma.	1er P.	2do P.	Ofi.	Correo
Estudiantes	13	1	0	0	0	0	3	0
Compu.	0	14	0	0	0	0	0	0
EvoVisión	0	2	14	0	0	0	10	0
Armarios	0	4	0	0	0	0	10	0
1er Piso	1	5	0	0	1	0	5	0
2do Piso	0	0	2	0	0	7	10	0
Oficina	0	1	0	0	0	0	15	0
Correo	0	1	2	0	0	0	7	0

Tabla XXVI: Matriz de confusión para el Experimento 2.

Es interesante observar que de nuevo el GA selecciona los componentes del modelo CieLab, talvez sugiriendo que el algoritmo prefiere información que esta fuertemente correlacionada con la percepción visual humana.

Finalmente, la Figura 74 presenta las gráficas de convergencia en las dos etapas del GA.

Finalmente, la Tabla XXVI presenta la matriz de confusión para este experimento, en donde se puede observar que la mayoría de los errores en el reconocimiento suceden con los lugares más homogéneos, que carecen de rasgos prominentes. Esto sugiere que el algoritmo de aprendizaje construye modelos más representativos para los lugares con un mayor número de regiones de interés, algo que se puede esperar de antemano.

Resumen y discusión

En esta sección se presentó un enfoque de aprendizaje evolutivo para reconocer lugares en un ambiente real, una tarea primordial para sistemas móviles de visión, utilizando un descriptor estadístico construido con un GA de dos etapas. Los resultados experimentales confirmaron la validez del enfoque, ya que se abordaron dos problemas en ambientes reales. Sin embargo, los resultados obtenidos en el Experimento 2 indican que tratar de construir un sistema de reconocimiento que solo utiliza información visual puede implicar un problema de mucha dificultad. Por ejemplo, en un ambiente de oficina muchos de los lugares no solo son homogéneos pero tambien muy similares entre si.

Trabajo futuro relacionado con esta problemática se centrará en integrar el sistema con un robot autónomo, con el propósito de facilitar la tarea de localizar al robot durante eventos de *secuestro*, cuando el robot desconoce su posición dentro del medio en el cual se desenvuelve.

Más aún, los resultados nos sugieren que para obtener un rendimiento mejor en un sistema real, tal vez es recomendable introducir restricciones espaciales durante el proceso de reconocimiento. Esto reduciría el número de clases posibles para cada imagen dado el último lugar en el cual estuvo el robot autónomo.

VIII.6 Resumen y trabajo futuro

En este capítulo se presentó una metodología que permite generar descriptores incluyentes para regiones locales, útilies para diferentes tipos tareas de reconocimiento. La propuesta utiliza regiones locales y construye modelos probabilísticos utilizando mezclas de Gaussianas. Las regiones se describen utilizando rasgos estadísticos relacionados con el contenido de textura y color. Los rasgos finales los elige un GA, a partir de un conunto de 108 rasgos diferentes. El GA busca minimizar el número de rasgos que utiliza el descriptor, y a la vez intenta maximizar la exactitud de la clasificación que se obtiene, y maximizar la distintividad de cada modelo que se construye.

La metodología propuesta se probó en dos problemas: (1) reconocimiento de objetos; y (2) reconocimiento de lugares. Los resultados obtenidos confirman la validez de la propuesta, y sugieren nuevos trabajos a futuro. Por ejemplo, es de interés aumentar el número de rasgos estadísticos, o incluir diferentes canales de información. Otra opción es utilizar un GP para construir rasgos mas complejos a partir de los rasgos básicos utilizados en este estudio. Es posible que el GP sea capaz de generar rasgos especializados que mejoren el rendimiento del descriptor que se genera.

Capítulo IX

Un descriptor local basado en la regularidad de Hölder

En este capítulo se presenta un descriptor local basado en un análisis de regularidad en la imagen. Se presentan definiciones para los conceptos de regularidad y el exponente Hölder, y se describe un método para estimar dicho exponente utilizando oscilaciones locales de la señal 2D.

También, se presenta un nuevo descriptor local discriminante basado en la regularidad de la imagen caracterizada a través del exponente Hölder Dicho descriptor se compara experimentalmente con el descriptor SIFT para analizar el rendimiento de la propuesta. Posteriormente, esta misma estrategia se optimiza a través de un algoritmo genético, buscando minimizar las dimensiones del descriptor, de tal forma que este sea más compacto sin una degradación significativa en el rendimiento.

Finalmente, se describe como la programación genética se puede utilizar para construir operadores que estimen al exponente Hölder para una señal 2D de una manera más eficiente que la técnica basada en oscilaciones. Este nuevo operador también se utiliza para construir un descriptor local, y experimentalmente se compara con el descriptor original de Hölder.
IX.1 Regularidad de Hölder

Cuando se analizan imágenes digitales, o cualquier tipo de señal, gran parte de la información que estas contienen se encuentra en regiones que presentan cambios abruptos en la señal. Dichas regiones se pueden considerar como la expresión discreta de las singularidades en la función continua subyacente.

Una manera de medir la regularidad de una señal, ya sea puntual o en un vecindario local, es a través del uso del espacio Hölder. Por lo tanto, a continuación se presenta una definición de regularidad expresada a través del exponente de Hölder.

Definición 6. El exponente Hölder puntual: Consideramos a una función $f : \Re \to \Re$, $s \in \Re^{+*} \setminus \mathbb{N}$ y un punto $x_0 \in \Re$. También, consideramos que $f \in C^s(x_0) \Leftrightarrow \exists \eta \in \Re^{+*}$, y que existe un polinomio P de grado menor que s, y una constante c tal que

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \le c|x - x_0|^s .$$
(41)

Entonces, el exponente Hölder puntual para f en el punto x_0 esta dado por $\alpha_p = \sup_s \{f \in C^s(x_0)\}$, este concepto se ilustra en la Figura 76.

El polinomio P puede considerarse como la expansión de Taylor de f; por lo tanto, el exponente Hölder¹, también representa una medida de la diferenciabilidad de la función f en el punto x_0 . Ahora bien, si $0 \le \alpha_p \le 1$ entonces la función f no es diferenciable y el exponente cuantifica el tipo de singularidad que se da en x_0 . Por ejemplo, si $\alpha_p = 0$ entonces f es completamente discotinua en x_0 .

Ahora, se presenta el exponente local de Hölder. Consideramos a $\alpha \in (0, 1), \Omega \subset \Re$. Se dice que $f \in C_l^{\alpha}(\Omega)$ si:

$$\exists \ C : \forall x, y \in \Omega : \frac{|f(x) - f(y)|}{|x - y|^{\alpha}} \le C \ .$$

¹También se conoce como el exponente de Lipschitz.



Figura 76: La envoltura de Hölder para la señal unidimensional f alrededor del punto x_0 .

Entonces consideramos: $\alpha_l(f, x_0, \rho) = \sup \{ \alpha : f \in C_l^{\alpha}(B(x_0, \rho)) \}$ y notamos que $\alpha_l(f, x_0, \rho)$ no incrementa como un función de ρ . Con lo que podemos establecer la siguiente definición:

Definición 7. Dada una función continua f. El *exponente local de Hölder* para f en x_0 es el numero real:

$$\alpha_{l}(x_{0}) = \alpha_{l}(f, x_{0}) = \lim_{\rho \to 0} \alpha_{l}(f, x_{0}, \rho)$$

Estas formas de medir la regularidad en una señal han tenido mucho éxito dentro del campo de análisis tipo fractal (Falconer, 1990). Tambi én, en los dominios referentes al análisis de imágenes ha demostrado ser útil porque el exponente Hölder proporciona una gran cantidad de información relacionada con la estructura local alrededor de cada pixel. La Figura 77 muestra una imagen original y una imagen Hölder, en donde el valor de cada pixel se sustituye por el valor del exponente Hölder puntual correspondiente.

El análisis de Hölder se ha utilizado para resolver tareas como la detección de bordes (Lévy-Véhel, 1995), interpolación de imágenes (Legrand y Lévy-Véhel, 2003), y para la eliminación de ruido (Legrand *et al.*, 2006).



Figura 77: El exponente Hölder calculado para una imagen de prueba.

Ahora bien, como ya se menciono antes en esta tesis (ver Capítulo IV), los descriptores locales intentan representar de manera compacta las variaciones locales presentes en una región de la imagen, y en general su meta es capturar la estructura de la señal alrededor de cada punto de interés. Por lo tanto, es razonable esperar que el exponente Hölder sea una herramienta útil en la construcción de descriptores locales.

Sin embargo, antes de pasar a una propuesta de como construir un descriptor en base a esta medida, es necesario repasar cual es la manera en la cual se estima el valor de dicho exponente.

IX.1.1 Estimación del exponente Hölder

Dada la Definición 6, es posible considerar varios métodos para estimar el valor del exponente Hölder, entre ellos están : (1) por oscilaciones; (2) por coeficientes de una descomposición wavelet; (4) y por regresión del limite superior y limite inferior. En este capítulo se repasan las primeras dos técnicas, para mayor información sobre estas, y otras las que no se describen aquí, se sugieren los trabajos de Legrand (2004).

Estimación por oscilaciones

La forma más directa para estimar el exponente Hölder consiste en estudiar las oscilaciones alrededor de cada punto, ya que esto es una aplicación directa de la definición del exponente. A continuación se presenta una descripción resumida de este método, para mayor detalles ver (Tricot, 1995; Legrand, 2004).

Retomando la Definición 6, el exponente Hölder para una función f(t) en el punto t es $\alpha_p \in [0, 1]$, si existe una constante c tal que $\forall t'$ en el vecindario de t,

$$|f(t) - f(t')| \le c|t - t'|^{\alpha_p} .$$
(42)

En términos de las oscilaciones en la señal de la imagen, esta condición se puede escribir de la siguiente forma.

Una función f(t) tiene una regularidad expresada por el exponente Hölder $\alpha_p \in [0, 1]$ en el punto t siempre que $\exists c \ \forall \tau$ tal que $osc_{\tau}(t) \leq c\tau^{\alpha_p}$, con

$$osc_{\tau}(t) = \sup_{|t-t'| \le \tau} f(t') - \inf_{|t-t'| \le \tau} f(t') = \sup_{t',t'' \in [t-\tau,t+\tau]} |f(t') - f(t'')| .$$
(43)

Una estimación de la regularidad se puede construir en cada punto si se calcula la pendiente del resultado de la regresión lineal entre el logaritmo de las oscilaciones y el logaritmo de la dimensión del vecindario en el cual se calcula cada oscilación.

Desde un punto de vista algorítmico, no se consideran a todos los posibles tamaños para los vecindarios que se utilizan durante el cálculo de oscilaciones, tomando un posible rango entre τ_{min} y τ_{max} . Por lo tanto, se calculan las oscilaciones sobre el punto t en intervalos de la forma $[t - \tau_r : t + \tau_r]$, donde $\tau_r = base^r$. Aquí es donde se utiliza regresión lineal de mínimos cuadrados, con base = 2 y r = 1, 2, ..., 7. Para una señal



Figura 78: Estimación del exponente Hölder con un análisis de oscilaciones. Izquierda: la región de interés λ , y tres de los siete vecindarios alrededor del punto t, cuando $r = 1, 2, \dots, 7$. Centro: el vecindario de radio $\tau_5 = 32$ pixeles, con base = 2. Derecha: cálculo del supremo de las diferencias dentro del radio τ_5 , done d representa la distancia Euclidiana.

en dos dimensiones, t define un punto en el espacio 2D y τ_r un radio alrededor de t, tal que las distancias Euclidianas d(t'', t) y d(t'', t) son $\leq \tau_r$; este proceso se ilustra en la Figura 78.

La estimación del exponente Hölder a través de un análisis de oscilaciones es confiable cuando se cumplen las siguientes tres condiciones:

- 1. El exponente $\alpha < 1$.
- 2. El proceso de regresión converge.
- 3. La pendiente de la recta es un resultado valido.

Estimación por coeficientes de wavelets

El exponente Hölder también se pude estimar utilizando un método basado en los coeficientes de una descomposición wavelet, esto esta basado en el teorema de Jaffard (2004). Dicho teorema muestra como se puede estimar la regularidad en un punto t

utilizando los coeficientes de wavelets, siempre y cuando dichos wavelets cumplen con ciertas propiedades relacionadas con la regularidad (Jaffard, 2004).

Si f es una función uniforme de Hölder y α es el exponente Hölder puntual de f en t_0 , entonces existe una constante c tal que los coeficientes de wavelets verifican lo siguiente:

$$|c_{j,k}| \leq c 2^{-j(\alpha + \frac{1}{2})} (1 + |2^j t - k|)^{\alpha} \quad \forall j, k \in \mathbb{Z}^2.$$

De igual manera ;

Si
$$\forall j, k \in \mathbb{Z}^2$$
 uno tiene $|c_{j,k}| \leq c 2^{-j(\alpha + \frac{1}{2})} (1 + |2^j t - k|)^{\alpha'}$,

para una $\alpha' < \alpha$, entonces el exponente Hölder de f en t_0 es α .

De este teorema, una forma de estimar la regularidad se obtiene si solo se consideran los índices j, k, tal que $|k - 2^{j}t| < cste$. Esto es como si se asume que la regularidad local y puntual de Hölder para f en t son iguales (Lvy Vhel y Seuret, 2004).

Trabajando con esta hipótesis, un método de estimación es utilizar la pendiente p de la regresión lineal de $\log_2 |c_{j,k}|$ con respecto a j. Es decir, en cada punto t de una señal descompuesta en n escalas, se estima la regularidad por:

$$\tilde{\alpha}(n,t) = -\frac{1}{2} - K_n \sum_{j=1}^n s_j \log_2 |c_{j,k}| , \qquad (44)$$

con $K_n = \frac{12}{n(n-1)(n+1)}$ y $s_j = j - \frac{n+1}{2}$, y donde $c_{j,k}$ son los coeficientes arriba de t; i.e., $k = \lfloor \frac{t+1}{2^{n-j+1}} \rfloor.$

Comparación

Legrand (2004) realizó una comparación experimental extensa entre estos métodos, utilizando señales con una regularidad prescrita. En general, sus conclusiones son que todos los métodos proveen una estimación confiable, que depende del tipo de señal que se esta analizando. Sin embargo, si existe un sesgo de rendimiento que sugiere que el método de oscilaciones, no solo es el más directo y sencillo, pero también el mejor para muchos tipos de señales.

IX.2 Descripción local utilizando la regularidad puntual de Hölder

En la sección anterior se presenta el concepto de regularidad expresada en términos del exponente Hölder, la cual puede tomar dos formas básicas, puntual y local. Además, se presentan dos métodos que permiten estimar la regularidad; de estos el método de oscilaciones se elige para el trabajo en esta tesis porque exhibe un alto rendimiento en una serie de experimentos numéricos (Legrand, 2004).

A continuación se presenta una metodología para construir un descriptor local en base a la regularidad puntual de los pixeles en una región de interés. Se elige el exponente puntual, porque este permite extraer más información de cada región local que se analice, a diferencia del exponente local que tiende a ser más homogéneo.

Entonces, el proceso para construir el descriptor, ilustrado en la Figura 79, consiste de la siguiente secuencia de pasos.

Primero, se detecta un conjunto de regiones Λ sobre la imagen que se va a analizar. Segundo, se calcula la orientación principal del gradiente $\phi_{\mathbf{A}}$ en la región \mathbf{A} para preservar la invarianza con respecto a la orientación. Finalmente, el vector de rasgos $D_{\mathbf{A}}$ contiene el exponente Hölder del punto central de la región, y de 128 puntos concéntricos distribuidos uniformemente y ordenados con respecto a $\phi_{\mathbf{A}}$.

Los primeros dos pasos se describen en los Capitulos III y IV, por lo que aquí solo se repasan brevemente.



Figura 79: El proceso para construir el descriptor Hölder. Primero, un detector extrae un conjunto Λ de regiones de interés. Después, $\forall \mathbf{A} \in \Lambda$ se calcula un descriptor $D_{\mathbf{A}}$. El descriptor contiene el exponente Hölder puntual para el puto central de la región $(x_{\mathbf{A}}, y_{\mathbf{A}})$, y para 32 puntos en el perímetro de cuatro círculos concéntricos, cada uno con un radio de $\frac{1}{4} \cdot s_{\mathbf{A}} \frac{1}{2} \cdot s_{\mathbf{A}}$, $\frac{3}{4} \cdot s_{\mathbf{A}}$ y $s_{\mathbf{A}}$ respectivamente.

Detección de regiones

El primer paso requiere de la detección de regiones prominentes y estables. El tipo de regiones que se detectan depende de los requerimientos de la aplicación de alto-nivel que las utilizará, con respecto al tipo de invarianza. Por ejemplo, un detector de puntos de interés es suficiente cuando la escala de la escena no cambia de manera significativa entre los diferentes puntos de vistas que se analizan. En estos casos, se utiliza el detector K_{IPGP2} que obtiene un rendimiento alto con respecto a la estabilidad geométrica y una dispersión de puntos. Cuando se utiliza un detector de puntos de interés, a todas las regiones se les asigna la misma escala $w_{\mathbf{A}} = 2.5$ pixeles. Ahora bien, cuando la escala si es relevante, entonces se utiliza el detector Hessian-Laplace (Mikolajczyk y Schmid, 2004). En los dos casos, la escala de la región de soporte para el descriptor, se fija a $s_{\mathbf{A}} = 5 \cdot w_{\mathbf{A}}$, donde $w_{\mathbf{A}}$ es la escala que asigna el detector.

Orientación principal

Con el propósito de preservar la invarianza con respecto a la rotación, se calcula la orientación principal del gradiente dentro de cada región, que sirve como referencia para organizar la información del descriptor. Cuando se utiliza el detector invariante a escala, las regiones primero se normalizan a un tamaño de 41 × 41 pixeles utilizando interpolación bi-cúbica. Después, se construye un histograma utilizando la orientación del gradiente de todos los pixeles de la región analizada. El máximo del histograma indica la orientación principal $\phi_{\mathbf{A}}$ de la región; el proceso es similar al propuesto en Lowe (2004).

Descriptor

Una vez que se detectan las regiones, ahora podemos pasar a construir el descriptor $D_{\mathbf{A}}$, $\forall \mathbf{A} \in \Lambda$. El proceso que se utiliza para muestrear la región es sencilla, ilustrado en la Figura 79; el primer elemento del descriptor es el exponente Hölder calculado en el centro de la región. Después, se calcula el exponente para los puntos que estan sobre el perímetro de cuatro círculos concéntricos, con radios proporcionales a la escala de la región: $\frac{1}{4} \cdot s_{\mathbf{A}}$, $\frac{1}{2} \cdot s_{\mathbf{A}}$, $\frac{3}{4} \cdot s_{\mathbf{A}}$ y $s_{\mathbf{A}}$ respectivamente. Un total de 32 puntos sobre cada círculs, espaciados uniformemente y ordenados en sentido de las manecillas del reloj con respecto a $\phi_{\mathbf{A}}$. Por ende, el vector de rasgos del descriptor es de 129 dimensiones, tales como el tamaño de los círculos y el número de puntos que se muestrean, presentan problemas comunes relacionados con la construccion de un descriptor que se desea

que sea altamente discriminante y a la vez obtenga una representación compacta de la información de la imagen. Los valores finales se eligieron en base a una serie de resultados experimentales con respecto al rendimiento del descriptor; sin embargo, un proceso de búsqueda/optimización tal vez sea capaz de encontrar valores óptimos para estos párametros.

Dada la forma en que se construye el descriptor y el tipo de información que utiliza, se pueden enlistar las siguientes características principales del mismo:

- El descriptor muestrea uniformemente que tan regular, o irregular, es cada región. Esto lo realiza utilizando coordenadas polares, similar a lo propuesto en el descriptor GLOH, por ejemplo.
- 2. En muchos descriptores propuestos anteriormente, la información contenida en una región se divide en subregiones, de tal forma que se considera la aportación de cada punto en conjunto, y no solo se toma un valor representativo. Sin embargo, para el descriptor Hölder esto no es necesario porque el análisis de los vecindarios locales dentro de cada región queda incluido de manera implícita en la estimación del exponente Hölder utilizando el método de oscilaciones.
- 3. El descriptor Hölder no requiere ser normalizado para hacerlo robusto con respecto a cambios uniformes de iluminación, como es el caso para SIFT. Como el método por oscilaciones considera cambios relativos en la señal, entonces los cambios uniformes de intensidad están contemplados en el cálculo del descriptor de manera implícita.

IX.2.1 Resultados experimentales

Con el propósito de analizar el rendimiento del descriptor propuesto, se emplea la metodología descrita en la Sección IV.2.2. Las secuencias de imágenes para pruebas

Nombre	Transformación	# Imágenes
Nueva York	Rotación	35
Van Gogh	Rotación	17
Mars	Rotación	18
Monet	Rotación	18
Graph	Iluminación	12
Mosaic	Iluminación	18
UBC	Compresión Jpeg	6
Laptop	Escala	6 (de 21)
Bip	Escala	5 (de 8)

Tabla XXVII: Secuencias utilizadas en las pruebas experimentales para el descriptor local construido con el exponente Hölder puntual.

benchmark se bajaron del sitio del Visual Geometry Group², donde se puede conseguir más información acerca de cada secuencia.

Para generar las gráficas *Recall vs 1-Precisión* se utiliza el criterio de correspondencia basado en umbral (ver Sección IV.2.2). Además, con el propósito de comprar el descriptor, las gráficas de rendimiento se presentan junto con las gráficas generadas con el descriptor SIFT. Para el descriptor SIFT se utilizó el detector de Harris para detectar puntos de interés, y el detector Harris-Laplace cuando se requiere detección invariante a escala. Los resultados experimentales se organizan en base al tipo de la transformación que se presenta en cada secuencia de prueba. La Tabla XXVII resume cada uno de los experimentos que se llevo acabo, en donde se especifica lo siguiente: el nombre de la secuencia; el tipo de transformación que exhibe, y el número de imágenes que contiene la secuencia.

En todos los experimentos se presentan seis figuras: (a) la imagen de referencia en la secuencia; (b) una imagen de prueba; (c) la imagen construida con el exponente puntal de Hölder para la imagen de referencia; (d) la gráfica *Recall vs 1-Precisión* calculada

²Visual Geometry Group: http://www.robots.ox.ac.uk/ vgg/research/ .

entre la imagen de referencia y la imagen de prueba mostrada; (e-f) el promedio, con la desviación estándar, en *Recall* y *1-Precisión* calculado para toda la secuencia de imágenes para el descriptor Hölder y SIFT respectivamente.

Rotación

Se utilizaron cuatro secuencias de prueba que muestran transformaciones de rotación: la Figura 80 muestra los resultados experimentales para la secuencia New York; la Figura 81 muestra los resultados para Van Gogh; la Figura 82 muestra los resultados para Mars; y la Figura 83 muestra los resultados para Monet.

En general, los resultados para este tipo de transformación son muy buenos, se puede observar que el rendimiento del descriptor Hölder es prácticamente igual a lo que se obtiene con SIFT. Estos resultados sugieren que el poder discriminativo del exponente Hölder es apropiado para este tipo de problemática.

Iluminación

Se utilizaron dos secuencias de prueba que muestran transformaciones de iluminación: la Figura 84 muestra los resultados experimentales para la secuencia Graph; y la Figura 85 muestra los resultados para Mosaic.

En general los resultados para los cambios de iluminación constante también son buenos, comparables con el rendimiento que se alcanza con el descriptor SIFT.

Compresión JPEG

Se utilizo una secuencia para probar los efectos de la compresión JPEG, en la Figura 86 se muestran los resultados para la secuencia UBC.

De antemano se conoce que la compresión afecta adversamente la estimación de la regularidad (Legrand, 2004), por lo tanto el decremento en rendimiento es esperado.



Figura 80: Resultados experimentales para la secuencia New York.

Sin embargo, el descriptor continua siendo comparable con lo que se obtiene con el descriptor SIFT.

Cambio de escala

La ultima trasformación que se probó fue el efecto que produce el cambio de escala. En este caso se probaron dos secuencias, la secuencia Laptop y la secuencia Bip; los resultados se presentan en las Figuras 87 y 88 respectivamente.

En las Figuras 87 y 88 se presenta un rendimiento inferior al que se obtiene con SIFT, aunque todavía el descriptor Hölder es competitivo. Sin embargo, es necesario apreciar que las pruebas para la tolerancia al cambio de escala solo se realizaron con las primeras imágenes de la secuencia. Por ejemplo, la Figura 89 presenta casos en los que



Figura 81: Resultados experimentales para la secuencia Van Gogh.

el cambio de escala es muy grande, y el descriptor de Hölder evidentemente no soporta dichos cambios, un factor mayor de 2x.

IX.2.2 Observaciones, limitaciones y preguntas abiertas

Los resultados experimentales revelan información valiosa respecto al descriptor propuesto. Primero, es evidente que el descriptor es capaz de caracterizar de una forma discriminante las regiones locales, exhibiendo un rendimiento alto que es competitivo con el estado-del-arte en el área, el descriptor SIFT.

Sin embargo, existen dos limitaciones que enfrenta la propuesta: (1) El tiempo de cómputo requerido; y (2) la susceptibilidad al cambio de escala.

Por otro lado, el proceso que se utiliza para construir el descriptor es bastante sencillo, y no requiere de un ajuste de parámetros extenso. De hecho, solo existen



Figura 82: Resultados experimentales para la secuencia Mars.

tres parámetros que fueron ajustados experimentalmente: (1) el número de círculos concéntricos; (2) el número de puntos que se muestrean sobre cada círculo; y (3) la distribuían espacial entre cada círculo y de los puntos entre si. Aunque estos parámetros se ajustaron experimentalmente, aún no es claro si los valores elegidos son los mejores, o si se puede minimizar la cantidad de puntos muestreados, y por ende la dimensión del descriptor, sin incurrir en una perdida significativa de rendimiento.

Tiempo de cómputo

El código que se emplea para estimar el exponente local se basa en la implementación de FracLab: A Fractal Analysis Toolbox³. Dicha implementación, la única disponible

³FracLab página web: http://apis.saclay.inria.fr/FracLab/



Figura 83: Resultados experimentales para la secuencia Monet.

públicamente según nuestra investigación, es muy confiable pero también es demasiado lenta.

Por ejemplo, para una imagen de tamaño 512×512 pixeles se requiere de 924 segundos de tiempo de cómputo por el CPU según las estadísticas que arroja el perfil de ejecución en Matlab. Esto equivale a 0.0035s para estimar el exponente en un punto, y aproximadamente a 0.5s para construir un descriptor de 129 valores. Si consideramos que en una imagen se pueden detectar entre 500 - 1000 regiones locales, entonces se requiere de un total de entre 250s y 500s, comparado con un rango de 2 o 3 segundos para calcular las misma cantidad de datos con el descriptor SIFT.



Figura 84: Resultados experimentales para la secuencia Graph.

Cambios de escala

Los resultados experimentales que se presentan arriba comprueban que el descriptor Hölder obtiene resultados competitivos con el descriptor SIFT en todas las pruebas excepto una, el cambio de escala. Este problema es más complicado que el descrito arriba, debido a que esta limitante se deriva del mismo método de estimación que se utiliza. Cuando se utilizan oscilaciones de la señal alrededor de un vecindario, se debe de elegir un tamaño para cada vecindario. Sin embargo, no es evidente como es que se debe modificar el tamaño de los vecindarios en base a la escala característica de cada región. Más aún, cuando se normalizan todas las regiones detectadas en una imagen, todas a un mismo tamaño, los métodos de interpolación no conservan la regularidad y



Figura 85: Resultados experimentales para la secuencia Mosaic.

por ende el exponente Hölder se ve alterado (Legrand y Lévy-Véhel, 2003).

IX.3 Reducir las dimensiones del descriptor utilizando un GA

En esta sección se presenta un algoritmo de búsqueda genética que intenta minimizar el tamaño del descriptor Hölder que se propone en este capítulo. Se utiliza una codificación binaria, donde cada bit representa uno de los 129 valores utilizados por el descriptor original, y se emplea la evaluación de aptitud propuesta en (Pérez y Olague, 2008) ya que ha demostrado ser capaz de guiar un proceso de búsqueda evolutiva.



Figura 86: Resultados experimentales para la secuencia UBC.

IX.3.1 Algoritmo genético

Al GA se le asigna la tarea de eligir cuales serán los puntos que se utilizan para construir el descriptor local en cada región, en la Figura 79 se ilustra cuales son los 129 puntos originales del descriptor Hölder. Utilizando a estos puntos como un limite máximo, el GA intenta seleccionar un subconjunto óptimo para el descriptor. Los parámetros básicos del GA se presentan en la Tabla XXVIII, y a continuación se describe con mayor detalle que es lo que representa el cromosoma de cada solución, y como se define la función de aptitud para el GA.



Figura 87: Resultados experimentales para la secuencia Laptop.

Parámetro	Descripción y/o valor
$Representaci\'on$	Cromosoma binario.
Tamaño de la población	100.
Número de generaciones	100.
Selección	Proporcional a la aptitud (Goldberg, 1989).
Cruce	Cruce de máscara o multipunto; $p_c = 0.9$.
$Mutaci\'on$	Mutación binaria de un bit; $p_{\mu} = 0.1$.
Supervivencia	Elitismo del 15%.

IX.3.2 Representación de la solución

El cromosoma de cada individuo se expresa como una cadena binaria $B = (b_1, b_2, ... b_{129})$ de 129 bits. Cada bit en la cadena esta asociado con una de las 129 posiciones que se muestrean para construir al descriptor. Por lo tanto, si el bit b_i es igual a 1, entonces el punto correspondiente se muestrea, mientras que si es igual a 0 sucede lo contrario. De



Figura 88: Resultados experimentales para la secuencia Bip.

tal forma que el número de dimensiones del descriptor que especifica cada individuo es igual al número de unos que contiene su cromosoma.

IX.3.3 Función de evaluación

Como ya se ha explicado, el rendimiento de cada descriptor se puede medir utilizando las gráficas *Recall vs 1-Precisión*. Sin embargo, traducir estas gráficas a un valor numérico no es trivial, inclusive este es un defecto que presenta el estudio comparativo de Miko-lajczyk y Schmid (2005).

Más aún, lo que se desea es tener un valor representativo para la aptitud de las soluciones candidatas, de tal forma que la búsqueda converja a óptimos bien definidos. Por ende, la función objetivo que se elige para este problema es la misma propuesta por Pérez y Olague (2008), ya que los autores mostraron que esta medida es una guía



Figura 89: Errores con cambios de escala muy grandes.

útil para un proceso de búsqueda evolutiva, o posiblemente para cualquier algoritmo de optimización que realice la evaluación de soluciones en el espacio *recall vs precisión*.

La base fundamental de la evaluación propuesta por Pérez y Olague (2008) es la llamada *Medida-F* (en inglés, F-Measure) (Van Rijsbergen, 1979). La *Medida-F* fue propuesta por la comunidad de recuperación de información, y es una forma de cuantificar la exactitud con la que se realiza una prueba considerando el recall y la precisión. Un valor perfecto de exactitud generaría un valor de 1 para la *Medida-F*, mientras que un valor de 0 representa el caso opuesto. La fórmula básica de dicha medida esta dada por,

$$F(precision, recall)_{\beta} = \frac{(1+\beta^2) \cdot (\text{precision} \cdot \text{recall})}{\beta^2 \cdot \text{precision} + \text{recall}} , \qquad (45)$$

donde $\beta = 1$ para obtener un balance simétrico entre precisión y recall.



Figura 90: Pares de imágenes de entrenamiento.

Es importante notar que para cada par (recall, precisión) le corresponde un valor de F_{β} , y las gráficas de evaluación que se emplean generan 20 de estos valores. Por lo tanto, se calcula una medida \overline{F}_{β} promedio para caracterizar el comportamiento de la gráfica que le corresponde a un par de imágenes (la imagen base, y la imagen de prueba). Ahora bien, si se utilizan N pares de imágenes de entrenamiento, el valor de la aptitud final se define como una función de costo dada por,

$$f(B) = -\frac{1}{N} \sum_{i=1}^{N} \overline{F}_{\beta}^{i} .$$

$$\tag{46}$$

Para nuestros experimentos se fija N = 3, utilizando los pares de imágenes que se muestran en la Figura 90.



Figura 91: Las gráficas de convergencia de tres ejecuciones típicas.

IX.3.4 Resultados experimentales

Se ejecutaran varias corridas del algoritmo, en la Figura 91 se presentan las gráficas de convergencia para tres experimentos típicos.

En la Figura 92 se muestran las posiciones que utiliza el mejor individuo de cada uno de los tres experimentos.

De los tres resultados que se presentan arriba, a continuación se analiza el que tuvo el mejor rendimiento, el mejor individuo del Experimento 2; denominamos a dicho descriptor con el nombre H-GA (Hölder GA). Es evidente la tendencia de este descriptor, mostrado en la Figura 92d, mustrea al exponente Hölder en posiciones más cercanas al punto central. Esta estrategia es similar a la que se emplea en el SIFT, por ejemplo, donde los puntos más cercanos al centro tienen un peso más alto que los puntos en la periferia durante la construcción del histograma. El descriptor H-GA tiene un total de 40 valores, o dimensiones, una reducción substancial comparado con las 129 dimensiones del descriptor original. Además, se puede observar que en este caso el punto central de la región se encuentra directamente sobre una esquina de la imagen. En este caso, se observa que el descriptor selecciona puntos que principalmente se encuentran afuera de la estructura interna de la esquina.



(a) Punto de interes (b) Descriptor Hölder



Figura 92: Posiciones muestreadas por el mejor individuo de cada experimento, y el número de puntos que contienen.

En las siguientes figuras el descriptor H-GA se compara con el del descriptor Hölder original. Además, se utiliza el criterio de vecino más cercano para analizar su rendimiento con respecto al problema de correspondencias locales (ver Sección IV.2.2). Se elige este criterio porque es el que se utiliza en situaciones reales, a diferencia del criterio de umbral, y porque es más sencillo que el criterio que utiliza la razón del vecino más cercano.

Las Figuras 93,94 y 95 comparan el descriptor H-GA con respecto a ambios de rotación en la imagen. Es posible observar que el rendimiento del H-GA es comparable con lo que se obtiene con el descriptor original.



Figura 93: Comparación del descriptor H-GA con el descriptor Hölder para la secuencia de New York.

Las Figuras 96 y 97 comparan al descriptor H-GA con respecto a cambios de iluminación uniforme en la imagen. Similar a los resultados que se obtuvieron en el caso de rotación, el rendimiento del H-GA es comparable con el rendimiento que exhibe el descriptor original.

En general, los resultados indican que si es posible reducir las dimensiones del descriptor Hölder; i.e., el número de posiciones muestreadas, sin reducir substancialmente el rendimiento del descriptor. Este resultado es importante si se desea minimizar el tiempo de cómputo empleado. Además, es posible apreciar que la propuesta hecha en (Pérez y Olague, 2008) es la apropiada para establecer la estructura necesaria en el espacio de aptitud que guié al proceso de búsqueda evolutiva.



Figura 94: Comparación del descriptor *H*-*GA* con el descriptor Hölder para la secuencia de Van Gogh.

IX.4 Estimaciónón del exponente Hölder utilizando regresión simbólica con GP

En la Sección IX.1.1 se presentaron dos métodos para estimar el exponente Hölder, sin embargo estos no son los únicos. En general, ningún método es mejor que otro y depende mucho de las características de la aplicación y del tipo de señal que se analiza.

En la Sección IX.2.2 se discute cuales son las limitantes más importantes del descriptor Hölder, calculado en base al método de estimación por oscilaciones. Una es el tiempo de cómputo que se requiere para construir cada descriptor, o bien, el tiempo de cómputo involucrado en la estimaciónón del exponente Hölder por oscilaciones. Además, debido al proceso involucrado en dicho método de estimación los descriptores exhiben un rendimiento bajo en pruebas de correspondencia donde existen cambios substanciales



Figura 95: Comparación del descriptor H-GA con el descriptor Hölder para la secuencia de Monet.

de escala.

En esta sección se propone construir un operador nuevo para estimar el exponente Hölder utilizando GP. La búsqueda de un operador capaz de realizar esta tarea se plantea como un problema de regresión simbólica y se resuelve con GP. Los operadores más aptos que se obtienen, llamados H-GP, se utiliza para construir un nuevo descriptor Hölder. La estimación del exponente Hölder utilizando los operadores H-GP se compara con el método de oscilaciones utilizando tres enfoques: cualitativo, cuantitativo y en base a la calidad del descriptor que se construye a partir de ellos. Cualitativamente, los operadores evolucionados se asemejan mucho a la estimación basada en oscilaciones; además, las comparaciones cuantitativas corroboran estos resultados. Finalmente, las pruebas de correspondencia local para los descriptores muestran que la estimación con



Figura 96: Comparación del descriptor H-GA para la secuencia de Graph.

operadores H-GP obtiene un rendimiento similar al descriptor original y al mismo tiempo reduce substancialmente el tiempo de cómputo requerido.

IX.4.1 Algoritmo GP

Como se menciona arriba, el propósito de esta sección es establecer un problema de regresión simbólica para sintetizar un operador que estima el exponente Hölder puntual para una imagen 2D. Para esto, se utilizan los exponentes Hölder de una imagen, estimados con el método por oscilaciones, como la función que se quiere reproducir.



Figura 97: Comparación del descriptor H-GA para la secuencia de Mosaic.

Planteamiento del problema

Comenzamos con una imagen I(x), y decimos que la matriz de datos H(x) contiene el valor del exponente Hölder para cada pixel de I; donde estos valores fueron estimados con el método por oscilaciones. Entonces, el problema es encontrar un operador K tal que

$$K(x) \approx H(x) \ \forall x \in I$$
.

o bien, escrito como un problema de optimización,

$$K^{o} \leftarrow \underset{K}{\operatorname{arg\,min}} \left\{ Err[K(x), H(x)] \right\} ,$$

$$(47)$$

donde Err[A, B] es una medida de error entre A y B; en esencia, el problema que se plantea arriba es uno de regresión simbólica.

Regresión simbólica

La meta en un problema de regresión simbólica es descubrir una expresión matemática o modelo en forma simbólica, que mejor se ajuste a una muestra finita de valores de las variables independientes y los valores de las variables dependientes asociadas (Koza, 1992). Los métodos de regresión lineal, cuadrática o polinomial son más sencillos, ya que en estos solo se deben encontrar coeficientes numéricos para una función que ha sido definida a priori; en la regresión simbólica es la función misma la que se desea construir a través de un proceso de optimización o búsqueda.

La función o modelo que se desea encontrar puede interpretarse como un pequeño programa computacional que toma los valores de las variables independientes como entrada y genera los valores de las variables dependientes como salida ⁴.

IX.4.2 Espacio de búsqueda

Como cualquier otro problema en el que se emplea GP, es necesario definir el espacio de búsqueda; i.e., los conjuntos de terminales y funciones que se utilizarán para construir posibles soluciones al problema planteado.

Para este problema se definen los siguientes conjuntos,

$$F_{puntual} = \{+, |+|, -, |-|, |I_{out}|, *, \div, I^2_{out}, \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out}\},$$

$$F_{vecindario} = \{G_{\sigma=1}, G_{\sigma=2}\},$$

$$F = F_{puntual} \bigcup F_{vecindario}, T = \{I\},$$

$$(48)$$

donde I es la imagen de entrada, y I_{out} puede ser cualquier terminal en T, o la salida de una de las funciones en F; G_{σ} son filtros de suavizado Gaussiano; y la constante k = 0.05.

El conjunto de funciones contiene dos tipos de operaciones: locales y de vecindario. Las funciones locales operan sobre cada pixel en la imagen de manera independiente,

⁴Para leer más sobre como se puede aplicar GP para resolver problemas de regresión simbólica se refiere al lector al trabajo de Koza (1992)

incluyen a las operaciones aritméticas, funciones no lineales y productos con constantes. Las operaciones en vecindarios son filtros Gaussianos sencillos, el propósito de estos es permitir al GP construir funciones que operen sobre un vecindario ya que esta es la manera en que operan los métodos canónicos empleados para estimar al exponente Hölder.

IX.4.3 Función de evaluación

Koza (1992) identifica a los problemas de regresión simbólica para GP como un proceso de evolución guiado por el error, ya que lo que se desea es minimizar la diferencia entre los datos de entrenamiento y la respuesta de una función o modelo que se ajusta a dichos datos.

Por lo tanto, la función de error de la Ecuación 47 se define como,

$$Err[K, B] = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (H(x_i) - K(x_i))^2}, \qquad (49)$$

donde N es el número de pixeles en una imagen I de entrenamiento; o bien, se utiliza la raíz del error cuadrático medio.

Por ende, si se consideran M imágenes de entrenamiento para las cuales se calcula los exponentes Hölder, la función de aptitud para un operador K esta dada por

$$f(K) = \frac{1}{\sqrt{\frac{1}{M} \sum_{j=1}^{M} Err[H(x_{i,j}), \hat{K}(x_{i,j})] + \epsilon}},$$
(50)

donde $x_{i,j}$ representa el pixel *i* de la imagen $j, \epsilon = 0.01$, y $\widehat{K}(x)$ es la versión normalizada de la respuesta al operador K en el punto x dada por

$$\widehat{K}(x) = \frac{K(x)}{\sqrt{\sum_{i=1}^{N} x_i^2}} .$$
(51)

Parámetro	Descripción y valor
Tamaño de la población	200 individuos.
Generaciones	200 generaciones.
Inicialización	Ramped Half-and-Half, con un max de 6 niveles.
Probabilidades de operadores genéticos.	Cruce $p_c = 0.85$; Mutación $p_{\mu} = 0.15$.
Profundidad de los arboles	Profundidad Dinámica.
Profundidad máxima dinámica	11 niveles.
Profundidad máxima real	16 niveles.
Selección	Muestreo estocástico universal
Supervivencia	Elitismo.

Tabla XXIX: Parámetros del GP para la evolución de operadores que estiman el exponente Hölder puntual.

Se utilizan dos modalidades diferentes para calcular la aptitud. En la primera, se toma la imagen base de cuatro secuencias diferentes, New York, Van Gogh, Monet y Mars, por ende M = 4; esta se denomina *Modalidad* A

En la segunda, se toman imágenes de una misma secuencia, en este caso M = 17imágenes de la secuencia New York; esta se denomina *Modalidad B*.

IX.4.4 Resultados experimentales

La Tabla XXIX presenta los parámetros generales con los cuales se ejecutaron los experimentos de esta sección.

Después de una serie de ejecuciones utilizando las dos modalidades, a continuación se analizan los resultados de los mejores resultados obtenidos. La Tabla XXX resume las características de los tres mejores operadores encontrados, especificando su nombre, su expresión simbólica, y la modalidad utilizada en la función objetivo. Es interesante observar los componentes básicos de estos operadores. Por ejemplo, los tres son expresiones no lineales, y dos de ellos (H-GP1 y H-GP3) emplean el logaritmo como parte de la operación que efectúan. Cabe mencionar que para hacer los operadores más legibles estos fueron simplificados algebraicamente, aunque no en su totalidad para ilustrar las



Tabla XXX: Operadores sintetizados con GP para estimar el exponente Hölder puntual.

partes principales de cada operador que sintetizó el GP.

La Figura 98 presenta las estadísticas de la ejecución que generó a cada uno de los operadores descritos en la Tabla XXX.

Estimación del exponente Hölder con los operadores H-GP

En esta sección se compara la estimación del exponente Hölder que realiza cada uno de los operadores evolucionados *H-GP*, tomando como referencia la estimación por el método de oscilaciones. Para esto, se utilizan diez imágenes de prueba: New York, Van Gogh, Casita, Puerta, Bip, Laptop, Monet, Mosaic, Árbol, y UBC, todas se muestran en la Figura 99.

En las Figuras 100, 101, 102 y 103 se presenta una comparación cualitativa entre la estimación llevada acabo con el método de oscilaciones y cada uno de los operadores evolucionados, utilizando cuatro de las imágenes. Estas Figuras ilustran que el GP es capaz de sintetizar operadores que son capaces de estimar el exponete Hölder puntual en cada pixel de una imagen. De los tres operadores evolucionados, el tercero, H-GP3, es cualitativamente más similar a la respuesta que se obtiene cuando se estima el exponente Hölder con las oscilaciones locales de la imagen. La diferencia entre el



Figura 98: Estadísticas del proceso evolutivo que generó a los operadores H-GP. Las columnas: (1) aptitud del mejor individuo; (2) diversidad de la población; y (3) complejidad (*profundidad máxima dinámica*, la profundidad del mejor individuo, y el número de nodos del mejor individuo).

operador H-GP3 y los otros dos es más evidente en la respuesta que se obtiene a las imágenes de Puerta y Casita.

Por otro lado, la Tabla XXXI presenta una comparación numérica entre los operadores, para cuantificar la similitud que tienen con la estimaciónón basada en oscilaciones. En la tabla se presenta el tamaño de cada imagen de prueba, la raíz del error



Figura 99: Imágenes utilizadas para probar la estimación del exponente Hölder realizado por los operadores *H-GP*.

cuadrático medio, el coeficiente de correlación, y el tiempo de cómputo requerido para aplicar el operador a toda la imagen.

Los resultados de dicha tabla se contraponen con los resultados de la evaluación cualitativa. La evaluación numérica sugiere que el operador con el mejor rendimiento es H-GP2. Por otro lado, si se observan las imágenes que generan estos operadores se intuye que H-GP3 obtiene la respuesta más similar al método por oscilaciones. Es particularmente interesante la correlación alta que existe entre la estimación del exponente


Figura 100: Comparación cualitativa entre la estimación del exponente Hölder por oscilaciones con la estimación de los operadores evolucionados para la imagen New York.

Hölder de los operadores H-GP2 y H-GP3, con la estimación que se obtiene a través del método por oscilaciones.

Se asume que el mejor rendimiento cuantitativo que logra el operador H-GP2 esta directamente relacionado con la modalidad utilizada durante el cálculo de la aptitud. En la modalidad (A) el GP utiliza una variedad más amplia de imágenes para optimizar al operador, mientras que en la evolución del operador H-GP3 la función de aptitud utiliza una secuencia de una sola imagen. Aún así, el operador H-GP3 también exhibe un buen rendimiento con respecto al error y la correlación que alcanza. Por otro lado,



Figura 101: Comparación cualitativa entre la estimación del exponente Hölder por oscilaciones con la estimación de los operadores evolucionados para la imagen Van Gogh.

si existen imágenes en las cuales la correlación es bastante baja, en particular la de Monet y Bip. En estos dos casos, ver Figura 98, existen porciones de la imagen que son muy regulares; i.e., que exhiben poca variación. Esto sugiere que los operadores evolucionados son mejores para estimar el exponente Hölder en regiones más irregulares. Estas regiones corresponden a las porciones más interesantes de la imagen, ya que son las que normalmente son las que se identifican por un algoritmo de detección. Por lo tanto, es razonable esperar que los operadores evolucionados sean capaces de calcular una buena estimación de la regularidad Hölder en las regiones de interés de una imagen. Por estas razones, los operadores H-GP2 y H-GP3 son los que a continuación se utilizan para construir descriptores locales, y su rendimiento se compara con el descriptor Hölder original.



Figura 102: Comparación cualitativa para la imagen Casita.

Construcción de descriptores locales utilizando los operadores H-GP

En esta sección se analiza el rendimiento de los operadores H-GP2 y H-GP3 cuando se utilizan para construir descriptores locales. Para mantener uniformidad, se realizan los mismos experimentos descritos en la Sección IX.2.1 y la Tabla XXVII, con las únicas diferencias en el número de imagenes que se utilizan para los experimentos con cambio de escala: Laptop/10 imágenes y Bip/toda la secuencia. Esto se hace para constatar

Tabla XXXI: Comparación cuantitativa entre la estimación del exponente Hölder por oscilaciones, con la estimación de los operadores evolucionados; **oscuras** indican el mejor.

Imagen	Medida	H-GP1	H-GP2	H-GP3
New York (512 × 512)	<i>Error</i> (10^{-4})	7.1094	5.2840	5.750
	Correlación	0.6081	0.8201	0.7525
	$Tiempo \ (seg)$	0.502	0.345	0.429
Van Gogh (348 × 512)	Error	8.42	7.46	9.69
	Correlación	0.6941	0.7776	0.5550
	Tiempo	0.3	0.25	0.29
Casita (484×768)	Error	8.410	8.00	5.266
	Correlación	-0.0037	0.7449	0.7399
	Tiempo	0.646	0.476	0.753
Puerta (256 × 256)	Error	15	9	13
	Correlación	0.542	0.875	0.7045
	Tiempo	0.11	0.08	0.11
Bip (768 × 574)	Error	5.356	5.705	5.642
	Correlación	0.626	0.69	0.5962
	Tiempo	2.38	0.939	1.758
Lap (768 × 574)	Error	6.158	4.743	3.842
	Correlación	0.3185	0.753	0.8194
	Tiempo	1.569	1.049	1.249
Monet (842×842)	Error	7.694	6.935	7.686
	Correlación	0.4138	0.5278	0.3937
	Tiempo	2.957	2.584	2.857
Mosaic (512 × 512)	Error	6.413	5.172	5.479
	Correlación	0.6245	0.7823	0.7404
	Tiempo	1.156	0.65	0.956
Tree (1000 × 700)	Error	9.8959	5.2 3	6.743
	Correlación	-0.0349	0.8576	0.6294
	Tiempo	3.164	1.553	1.945
UBC (800 × 640)	Error	10	4	7
	Correlación	0.013	0.9010	0.5015
	Tiempo	2.077	1.541	1.59

si los nuevos operadores son menos susceptibles al ruido que inducen los cambios de escala en la construcción de los descriptores locales.



Figura 103: Comparación cualitativa para la imagen Puerta.

Las pruebas para cambios de rotación se muestran en las Figuras 104, 105, 106 y 107. Se puede apreciar que el operador H-GP3 obtiene un rendimiento substancialmente mayor que el H-GP2. Además, el rendimiento del H-GP3 es comparable con el descriptor de Hölder que utiliza la estimación basada en oscilaciones.

En todas las pruebas con rotación el rendimiento del operador H-GP3 es substancialmente mayor al que exhibe el operador H-GP2, aunque las pruebas de error y correlación sugieren que el segundo provee una estimación más similar a la que se obtiene con el método por oscilaciones. Esto se atribuye a las secuencias de entrenamiento que se emplearon durante la evolución de cada uno, ya que la modalidad utilizada para H-GP3 explícitamente contempla los cambios de rotación presentes en la secuencia de



Figura 104: Operadores *H-GP* aplicados a la secuencia New York.

prueba para el descriptor.

Las pruebas para cambios de iluminación se muestran en las Figuras 108 y 109. Aquí de nuevo, el operador H-GP3 obtiene un mejor rendimiento que el operador H-GP2, y es comparable con el descriptor basado en oscilaciones. La Figura 110 presenta los resultados que se obtienen para los cambios inducidos por la compresión JPEG.

Finalmente, las Figuras 111 y 112 presentan el rendimiento de los descriptores con cambios de escala en la imagen. En estas pruebas, los operadores evolucionados tienen un rendimiento apreciablemente superior al que se obtiene con el método por oscilaciones. Es evidente que el hecho de que estos operadores no dependen exclusivamente de cálculos en vecindarios predefinidos, y esto les permite ser más robustos frente al ruido que inducen los cambios de escala durante la construcción del descriptor. Los



(d) Hölder sobre la secuencia (e) H-GP2 sobre la secuencia (f) H-GP3 sobre la secuencia

Figura 105: Operadores *H-GP* aplicados a la secuencia Van Gogh.

ejemplos específicos que se muestran en las Figuras 111c y 112c ilustran claramente este efecto. Además, igual que en las otras pruebas, el operador H-GP3 obtiene un rendimiento mayor que el operador H-GP2.

IX.4.5 Discusión y conclusiones

En esta sección se propuso un enfoque de GP para sintetizar operadores capaces de estimar el exponente Hölder en una imagen. Esto con el propósito de utilizar la estimación basada en estos operadores para construir descriptores locales de imágenes. La estimación que se obtiene a partir de los operadores evolucionados se evalúa de forma cualitativa, cuantitativa y en base a la calidad del descriptor que se construye con la información que extraen. Las pruebas cualitativas revelan que la evolución artificial extrae una respuesta similar a la que se obtiene con la estimación del exponente



Figura 106: Operadores *H-GP* aplicados a la secuencia Mars.

Hölder utilizando el método basado en oscilaciones. Las pruebas cuantitativas, basadas en el error y medidas de correlación, corroboran lo que se aprecia de manera visual. Finalmente, las pruebas de correspondencia local para los descriptores muestran que la estimación con los operadores evolucionados exhibe un rendimiento similar al que se obtiene con el descriptor Hölder original. Además, dichos descriptores se pueden calcular en menor tiempo y son más robustos a los cambios de escala en las imágenes probadas.



Figura 107: Operadores H-GP aplicados a la secuencia Monet.



Figura 108: Operadores H-GP aplicados a la secuencia Graph.



Figura 109: Operadores H-GP aplicados a la secuencia Mosaic.



Figura 110: Operadores H-GP aplicados a la secuencia UBC.



Figura 111: Operadores *H-GP* aplicados a la secuencia Laptop.



Figura 112: Operadores H-GP aplicados a la secuencia Bip.

Capítulo X

Resumen, conclusiones y trabajo futuro

En esta tesis se propone un enfoque basado en la evolución artificial para estudiar y diseñar técnicas novedosas para la detección y descripción de regiones de interés en imágenes digitales. La motivación subyacente de esta tesis es encontrar soluciones a los problemas de visión, de relevancia actual, a través de técnicas que intentan reproducir aspectos de la inteligencia humana. En particular, se incorporan a los problemas de visión por computadora que se abordan procesos inspirados en, y similares a, a la evolución natural. Es importante notar que este mecanismo de la naturaleza es el que ha generado, una y otra vez, las mismas capacidades visuales que ahora queremos reproducir en una máquina.

Para cumplir con la la meta general de la tesis, dada la motivación de la misma, fue necesario realizar un planteamiento diferente de los problemas relacionados con la *detección* y *descripción* de regiones locales, utilizando procesos de búsqueda y optimización evolutiva, un enfoque poco común en visión. Sin embargo, para mantener coherencia con los trabajos del área, los procesos evolutivos siempre fueron guiados por criterios de evaluación bien establecidos y aceptados ampliamente.

X.1 Contribuciones y limitaciones de la investigación

La contribución primordial de esta tesis se centra en presentar algoritmos de cómputo evolutivo, especialmente GP, para el diseño de operadores que resuelven las tareas básicas requeridas para realizar un análisis basado en regiones locales. Este tipo de análisis representa el enfoque prevaleciente dentro de la literatura actual en visión por computadora. Por lo tanto, los algoritmos y soluciones propuestas representan una demostración experimental del alcance que el cómputo evolutivo tiene para resolver problemas que representan la base del estado-del-arte en visión artificial.

A continuación, se describen las contribuciones específicas realizadas en este trabajo, y se discuten algunas de las limitaciones correspondientes.

X.1.1 Detección de puntos de interés

Contribuciones

- Este trabajo describe una metodologíaía para sintetizar operadores capaces de detectar puntos interesantes en imágenes. El enfoque desarrollado automatiza un proceso que anteriormente se realizaba a mano, cuando se deseaba obtener operadores sencillos, coherentes, y con aplicabilidad amplia en diferentes problemáticas de visión (Trujillo y Olague, 2006a,b, 2008). Además, se mostró que los operadores sintetizados con GP pueden extenderse sencillamente a la detección de regiones invariantes a cambios de escala (Trujillo y Olague, 2007).
- Se presenta un estudio multiobjetivo del problema de detección de puntos de interés, un enfoque único dentro de la literatura actual (Trujillo *et al.*, 2008b). Los resultados experimentales sugieren que el problema es intrínsecamente de

carácter multiobjetivo. Además, la búsqueda GP-MO nos permitió identificar un nuevo operador parametrizado, que permite un control fino del rendimiento exhibido durante el proceso de detección.

Limitaciones

- Los operadores sintetizados automáticamente demuestran un rendimiento igual o superior a todas las propuestas anteriores, cuando se comparan utilizando pruebas genéricas. Sin embargo, aún debe de comprobarse la efectividad de dichos operadores en aplicaciones reales con requerimientos específicos.
- Los criterios de optimización que cuantifican la dispersión de los puntos detectados y el contenido de información que se extrae con dichos puntos, aún dependen de las imágenes de entrenamiento que se emplean. Por esto, es necesario realizar ejecuciones de los algoritmos GP enfocándose en aplicaciones específicas y problemáticas acotadas.

X.1.2 Descripción de regiones locales

Contribuciones

- Se presenta un descriptor que es altamente competitivo y algorítmicamente sencillo de calcular basado en el análisis de regularidad expresada por el exponente Hölder (Trujillo *et al.*, 2007b). Este nuevo descriptor se diferencia de la línea de soluciones propuestas en los últimos años que siguen las ideas básicas establecidas en el descriptor SIFT.
- Para utilizar el exponente Hölder en la descripción de regiones locales, primero se requiere definir una metodología para estimar su valor. Este tarea se plantea como un problema de regresión simbólica y se resuelve con GP. Esta solución

introduce una nueva herramienta para el análisis general de señales basados en regularidad; este enfoque puede tener una aplicabilidad amplia en otras áreas de investigación.

Se propone un descriptor estadístico que acentúa las similitudes entre imágenes de una misma clase, y ayuda a discriminar entre imágenes de diferentes clases. Los rasgos utilizados por dicho descriptor se definen a través de un proceso de búsqueda y selección genética. Finalmente, el descriptor se utiliza para resolver satisfactoriamente problemas de reconocimiento de objetos y lugares en escenarios reales (Trujillo *et al.*, 2007a, 2008a).

Limitaciones

- El descriptor Hölder propuesto es altamente competitivo con el descriptor SIFT en diferentes tareas de correspondencia local excepto una, situaciones donde se presentan cambios de escala. Por un lado esto se deriva del método de estimación utilizado que explícitamente depende de una análisis a diferentes escala. Se prevé que la estimación a través de operadores evolucionados con GP pueda ser capaz de superar esta limitante, y resultados preliminares apoyan esta hipótesis; sin embargo, más trabajo experimental es necesario.
- El rendimiento de la estimación del exponente Hölder con operadores evolucionados esta intrínsecamente acotada. Esto se debe a que el problema se plantea en términos de regresión, donde la respuesta objetivo esta dada por el método basado en oscilaciones. Es necesario establecer criterios de rendimiento independientes de cualquier otra forma de estimación previa.

X.2 Trabajo futuro

Para finalizar esta tesis, se discuten diferentes líneas de investigación que se desprenden del trabajo aquí descrito.

- Aplicar las técnicas de detección y descripción propuestas para resolver problemas de alto nivel. Los resultados obtenidos en esta tesis establecen herramientas básicas para el desarrollo de sistemas de visón. Dichas herramientas se diseñaron utilizando un enfoque novedoso, y con propuestas únicas en el área. Sin embargo, ahora es posible comenzar a desarrollar sistemas complejos de visión que integren y aprovechen estas herramientas en la solución de problemas a un nivel cognitivo más alto. Por ejemplo, sistemas de reconocimiento y localización en un robot autónomo móvil; o sistemas de recuperación de imágenes basándose en contenido, por nombrar dos ejemplos interesantes.
- Como consecuencia del punto anterior, se contempla que el desarrollo de sistemas de visión basados en las herramientas propuestas continúen con el enfoque evolutivo enfatizado en esta tesis. De tal forma, que a largo plazo se prevé la construcción de un sistema completo de visión artificial, competitivo con el estado-delarte, que sea diseñado y optimizado primordialmente con herramientas inspiradas en la evolución artificial.

Estas metas a mediano y corto plazo se desprenden del interés primordial que inspiro el trabajo descrito en esta tesis, la búsqueda de sistemas artificiales que nos parezcan estar ennoblecidas por su capacidad de resolver problemas complejos tal como lo hacen los sistemas naturales que observamos diariamente.

Referencias

- Abdel-Hakim, A. E. y A. A. Farag 2006. "Csift: A sift descriptor with color invariant characteristics". En: "CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition", Washington, DC, USA. IEEE Computer Society. 1978–1983 p.
- Allen, M. 1999. "Do it yourself climate prediction". Nature, 401(6754):642 p.
- Amores, J., N. Sebe, y P. Radeva 2007. "Context-based object-class recognition and retrieval by generalized correlograms". IEEE Trans. Pattern Anal. Mach. Intell., 29(10):1818–1833 p.
- Anderson, D. 2004. "Boinc: a system for public-resource computing and storage". En: "Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on". 4-10 p.
- Anderson, D. P., J. Cobb, E. Korpela, M. Lebofsky, y D. Werthimer 2002. "Seti@home: an experiment in public-resource computing". Commun. ACM, 45(11):56–61 p.
- Andre, D. 1994. "Automatically defined features: the simultaneous evolution of 2dimensional feature detectors and an algorithm for using them". En: "Advances in genetic programming". MIT Press, Cambridge, MA, USA, 477–494 p.
- Asada, H. y M. Brady 1986. "The curvature primal sketch". IEEE Trans. Pattern Anal. Mach. Intell., 8(1):2–14 p.
- Ashbrook, A. P., N. A. Thacker, P. I. Rockett, y C. I. Brown 1995. "Robust recognition of scaled shapes using pairwise geometric histograms". En: "Proceedings of the 6th British Conference on Machine Vision (BMVC '95)", Surrey, UK, UK, volume 2. BMVA Press. 503–512 p.
- Bala, J., K. D. Jong, J. Huang, H. Vafaie, y H. Wechsler 1996. "Using learning to facilitate the evolution of features for recognizing visual concepts". Evol. Comput., 4(3):297–311 p.
- Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, y A. Warfield 2003. "Xen and the art of virtualization". En: "Proceedings of the nineteenth ACM symposium on Operating systems principles". ACM Press. 164–177 p.
- Baumberg, A. 2000. "Reliable feature matching across widely separated views". En: "Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 13-15, Hilton Head Island, SC", volume 1. IEEE Computer Society. 774–781 p.

- Bay, H., A. Ess, T. Tuytelaars, y L. V. Gool 2008. "Speeded-up robust features (surf)". Comput. Vis. Image Underst., 110(3):346–359 p.
- Beaudet, P. R. 1978. "Rotational invariant image operators". En: "Proceedings of the 4th International Joint Conference on Pattern Recognition (ICPR), Tokyo, Japan". 579–583 p.
- Belongie, S., J. Malik, y J. Puzicha 2002. "Shape matching and object recognition using shape contexts". IEEE Trans. Pattern Anal. Mach. Intell., 24(4):509–522 p.
- Bhanu, B., Y. Lin, y K. Krawiec 2005. "Evolutionary synthesis of pattern recognition systems". Monographs in Computer Science. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 322 pp.
- Bosch, A., A. Zisserman, y X. Munoz 2007. "Representing shape with a spatial pyramid kernel". En: "CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval", New York, NY, USA. ACM. 401–408 p.
- Bouchard, G. y B. Triggs 2005. "Hierarchical part-based visual object categorization.". En: "Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 20-26 June, San Diego, CA", volume 1. IEEE Computer Society. 710-715 p.
- Brown, M. y D. G. Lowe 2003. "Recognising panoramas". En: "Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)", Washington, DC, USA. IEEE Computer Society. 1218 pp.
- Brown, M., R. Szeliski, y S. Winder 2005. "Multi-image matching using multi-scale oriented patches". En: "Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)", Washington, DC, USA, volume 1. IEEE Computer Society. 510–517 p.
- Cagnoni, S., E. Lutton, y G. Olague, editores 2007. "Genetic and evolutionary computation for image processing and analysis", volume 8 of EURASIP Book Series on Signal Processing and Communications. Hindawi Publishing Corporation.
- Carneiro, G., A. B. Chan, y P. J. Moreno 2007. "Supervised learning of semantic classes for image annotation and retrieval". IEEE Trans. Pattern Anal. Mach. Intell., 29(3):394–410 p. Member-Nuno Vasconcelos.
- Caron, Y., P. Makris, y N. Vincent 2007. "Use of power law models in detecting region of interest". Pattern Recogn., 40(9):2521–2529 p.
- Castillo, O., L. Trujillo, y P. Melin 2006. "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots". Soft Computing, 11(3):269–279 p.

- Chen, J., S. Shan, G. Zhao, X. Chen, W. Gao, y M. Pietikäinen 2008. "A robust descriptor based on webers law". En: "Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June, Anchorage, AK". IEEE Computer Society. 1–7 p.
- Cheng, H., Z. Liu, N. Zheng, y J. Yang 2008. "A deformable local image descriptor". En: "Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June, Anchorage, AK". IEEE Computer Society. 1-8 p.
- Coello, C., D. V. Veldhuizen, y G. Lamont 2002. "Evolutionary algorithms for solving multi-objective problems". Kluwer Academic Publishers, New York, New York. 610 pp.
- Coillie, F. M. V., L. P. Verbeke, y R. R. D. Wulf 2007. "Learning the best subset of local features for face recognition". Remote Sensing of Environment, 110(4):476–487 p.
- Corne, D., J. D. Knowles, y M. J. Oates 2000. "The pareto envelope-based selection algorithm for multi-objective optimisation". En: "PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature", London, UK. Springer-Verlag. 839–848 p.
- Dalal, N. y B. Triggs 2005. "Histograms of oriented gradients for human detection". En: "CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1", Washington, DC, USA. IEEE Computer Society. 886–893 p.
- Darwin, C. 1872. "The origin of species by means of natural selection, or the preservation of favoured races in the struggle for life". John Murray, 6th edition. 432 pp.
- Davis, J. y M. Goadrich 2006. "The relationship between precision-recall and roc curves". En: "Proceedings of the 23rd international conference on Machine learning (ICML '06)", New York, NY, USA. ACM. 233–240 p.
- Davison, A. J. y N. D. Molton 2007. "Monoslam: Real-time single camera slam". IEEE Trans. Pattern Anal. Mach. Intell., 29(6):1052–1067 p. Member-Ian D. Reid and Member-Olivier Stasse.
- de Castro, L. y J. Timmis 2002. "Artificial Immune Systems: A New Computational Approach". Springer-Verlag, London, UK. 357 pp.
- Deb, K., S. Agrawal, A. Pratap, y T. Meyarivan 2002. "A fast and elitist multiobjective genetic algorithm: Nsga-ii.". IEEE Trans. Evolutionary Computation, 6(2):182-197 p.
- Deb, K., M. Mohan, y S. Mishra 2003. "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions". KanGAL report2003002, Indian Institute of Technology, Kanpur, India.

- Deriche, R. y G. Giraudon 1993. "A computational approach for corner and vertex detection". Int. J. Comput. Vision, 10(2):101–124 p.
- Dorigo, M. 1992. "Optimization, learning and natural algorithms". Tesis de Doctorado, Politecnico di Milano, Italy.
- Dorkó, G. y C. Schmid 2003. "Selection of scale-invariant parts for object class recognition". En: "Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03)", Washington, DC, USA. IEEE Computer Society. 634 pp.
- Dunn, E. y G. Olague 2004. "Multi-objective sensor planning for efficient and accurate object reconstruction". En: "Applications of Evolutionary Computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoS-TOC, Coimbra, Portugal, April 5-7, 2004, Proceedings", volume 3005 of Lecture Notes in Computer Science. Springer. 312-321 p.
- Dunn, E., G. Olague, y E. Lutton 2006. "Parisian camera placement for vision metrology". Pattern Recogn. Lett., 27(11):1209–1219 p.
- Ebner, M. 1998. "On the evolution of interest operators using genetic programming". En: Poli, R. y others, editores, "Late Breaking Papers at EuroGP'98". The University of Birmingham, UK. 6–10 p.
- Ebner, M. y A. Zell 1999. "Evolving task specific image operator.". En: Poli, R. y others, editores, "First European Workshops, EvoIASP'99 and EuroEcTel'99, Göteborg, Sweden, May 26-27", volume 1596 of LNCS. 74-89 p.
- Ehrgott, M. y X. Gandibleux 2002. "Multiple criteria optimization. state of the art annotated bibliographic surveys". Kluwer Academic, Dordrecht. 520 pp.
- Elnozahy, E., L. Alvisi, Y. Wang, y D. Johnson 2002. "A survey of rollback-recovery protocols in message-passing systems". ACM Computing Surveys (CSUR), 34(3):375– 408 p.
- Falconer, K. 1990. "Fractal geometry: Mathematical foundations and applications". Wiley. 310 pp.
- Faugeras, O. 1993. "Three-dimensional computer vision: A geometric approach". The MIT Press. 662 pp.
- Fedak, G., C. Germain, V. Neri, y F. Cappello 2001. "Xtremweb: A generic global computing system". Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGRID).
- Fei-Fei, L., R. Fergus, y P. Perona 2004. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". En: "Proceedings of the 2004 Conference on Computer Vision and Pattern

Recognition Workshop (CVPRW'04), 27 June - 2 July, Washington, DC", volume 12. IEEE Computer Society. 178 pp.

- Fergus, R., P. Perona, y A. Zisserman 2003. "Object class recognition by unsupervised scale-invariant learning.". En: "Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 16-22 June, Madison WI", volume 2. IEEE Computer Society. 264-271 p.
- Ferrari, V., T. Tuytelaars, y L. Van Gool 2006. "Simultaneous object recognition and segmentation from single or multiple model views". Int. J. Comput. Vision, 67(2):159–188 p.
- Figueiredo, M. A. T. y A. K. Jain 2002. "Unsupervised learning of finite mixture models". IEEE Trans. Pattern Anal. Mach. Intell., 24(3):381–396 p.
- Fisher, R. A. 1936. "The use of multiple measurements in taxonomic problems". Annals of Eugenics, 7:179–188 p.
- Florack, L., B. ter Haar Romeny, J. Koenderink, y M. Viergever 1991. "General intensity transformations and second order invariants". En: "Proceedings of the Seventh Scandinavian Conference on Image Analysis". 338–345 p.
- Fogel, L. J., A. J. Owens, y M. J. Walsh 1966. "Artificial intelligence through simulated evolution". John Wiley, New York, USA. 162 pp.
- Förstner, W. 1986. "A feature based correspondence algorithm for image matching". International Archives of Photogrammetry and Remote Sensing, 26(3):150–166 p.
- Förstner, W. y E. Gülch 1987. "A fast operator for detection and precise location of distinct points, corners and centres of circular features". En: "ISPRS Intercommission Conference on fast processing of photogrammetric data". 149–155 p.
- Forsyth, D. A. y J. Ponce 2002. "Computer vision: A modern approach". Prentice Hall Professional Technical Reference. 693 pp.
- Freeman, W. T. y E. H. Adelson 1991. "The design and use of steerable filters". IEEE Trans. Pattern Anal. Mach. Intell., 13(9):891–906 p.
- Gabor, D. 1946. "Theory of communication". J. IEE, 3(93):429–457 p.
- Gao, X., F. Sattar, A. Quddus, y R. Venkateswarlu 2007. "Multiscale contour corner detection based on local natural scale and wavelet transform". Image Vision Comput., 25(6):890–898 p.
- Gen, M. y R. Cheng 1997. "Genetic algorithms and engineering design". Wiley series in engineering design and automation. Wiley-Interscience, New York, NY, USA. 432 pp.

- Gökberk, B., M. O. İrfanoğlu, L. Akarun, y E. Alpaydin 2007. "Learning the best subset of local features for face recognition". Pattern Recogn., 40(5):1520–1532 p.
- Goldberg, D. E. 1989. "Genetic algorithms in search, optimization, and machine learning". Addison-Wesley Professional. 432 pp.
- Harris, C. y M. Stephens 1988. "A combined corner and edge detector". En: "Proceedings from the Fourth Alvey Vision Conference", volume 15. 147-151 p.
- Hartley, R. y A. Zisserman 2003. "Multiple view geometry in computer vision". Cambridge University Press, New York, NY, USA. 672 pp.
- Hernández, B., G. Olague, R. Hammoud, L. Trujillo, y E. Romero 2007. "Visual learning of texture descriptors for facial expression recognition in thermal imagery". Computer Vision and Image Understanding, Special Issue on Vision Beyond the Visual Spectrum, 106(2-3):258–269 p.
- Holland, J. H. 1975. "Adaptation in natural and artificial systems". University of Michigan Press, Ann Arbor, MI. 228 pp.
- Howard, D., S. C. Roberts, y R. Brankin 1999. "Target detection in sar imagery by genetic programming". Advances in Engineering Software, 30(5):303–311 p.
- Howard, D., S. C. Roberts, y C. Ryan 2006. "Pragmatic genetic programming strategy for the problem of vehicle detection in airborne reconnaissance". Pattern Recogn. Lett., 27(11):1275–1288 p.
- Hurwicz, L. 1958. "Programming in linear spaces". En: Arrow, K. J. y others, editores, "Studies in Linear and Non–linear Programming", volume 2. Stanford University Press.
- Jaffard, S. 2004. "Wavelet techniques in multifractal analysis". En: "Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics", volume 72. 91–151 p.
- Johnson, A. E. y M. Hebert 1997. "Recognizing objects by matching oriented points". En: "Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)", Washington, DC, USA. IEEE Computer Society. 684 pp.
- Kadir, T. y M. Brady 2001. "Saliency, scale and image description.". International Journal of Computer Vision, 45(2):83-105 p.
- Ke, Y. y R. Sukthankar 2004. "Pca-sift: A more distinctive representation for local image descriptors". En: "Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 27 June - 2 July, Washington, DC", volume 2. IEEE Computer Society. 506-513 p.

- Kennedy, J. y R. C. Eberhart 2001. "Swarm intelligence". Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 510 pp.
- Kenney, C. S., M. Zuliani, y B. S. Manjunath 2005. "An axiomatic approach to corner detection". En: "International Conference on Computer Vision and Pattern Recognition (CVPR)".
- Kitchen, L. y A. Rosenfeld 1982. "Gray-level corner detection". Pattern Recognition Letters, 1:95-102 p.
- Koenderink, J. J. y A. J. van Doom 1987. "Representation of local geometry in the visual system". Biol. Cybern., 55(6):367–375 p.
- Koza, J. R. 1992. "Genetic programming: On the programming of computers by means of natural selection". MIT Press, Cambridge, MA, USA. 849 pp.
- Koza, J. R., M. A. Keane, J. Yu, I. Forrest H. Bennett, y W. Mydlowec 2000. "Automatic creation of human-competitive programs and controllers by means of genetic programming". Genetic Programming and Evolvable Machines, 1(1-2):121–164 p.
- Krawiec, K. 2002. "Genetic programming-based construction of features for machine learning and knowledge discovery tasks". Genetic Programming and Evolvable Machines, 3(4):329–343 p.
- Krawiec, K. 2007. "Generative learning of visual concepts using multiobjective genetic programming". Pattern Recognition Letters, 28:2385-2400 p.
- Krawiec, K. y B. Bhanu 2005. "Visual learning by coevolutionary feature synthesis". IEEE Transactions on Systems, Man, and Cybernetics, Part B, 35(3):409-425 p.
- Krawiec, K. y B. Bhanu 2007. "Visual learning by evolutionary and coevolutionary feature synthesis". IEEE Trans. on Evolutionary Computation, 11:635-650 p.
- Krawiec, K., D. Howard, y M. Zhang 2007. "Overview of object detection and image analysis by means of genetic programming techniques". En: "Frontiers in the Convergence of Bioscience and Information Technologies 2007, FBIT 2007, Jeju Island, Korea, October 11-13". IEEE Computer Society. 779-784 p.
- Kuehni, R. G. 2003. "Color space and its divisions: Color order from antiquity to the present". Wiley, New York, NY. 408 pp.
- Kuhn, H. W. y A. W. Tucker 1951. "Nonlinear programming". En: "Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability". Berkeley, California. 481-492 p.

- Kumar, M. P., P. H. S. Torr, y A. Zisserman 2005. "Obj cut". En: "Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 20-26 June, San Diego, CA", volume 1. IEEE Computer Society. 18-25 p.
- Langdon, W. B. y R. Poli 2002. "Foundations of genetic programming". Springer-Verlag, New York, New York. 274 pp.
- Laptev, I. y T. Lindeberg 2003. "Space-time interest points". En: "Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)", Washington, DC. IEEE Computer Society. 432 pp.
- Lazebnik, S., C. Schmid, y J. Ponce 2005. "A sparse texture representation using local affine regions". IEEE Trans. Pattern Anal. Mach. Intell., 27(8):1265–1278 p.
- Lazebnik, S., C. Schmid, y J. Ponce 2006. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories.". En: "Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 17-22 June, New York, NY", volume 2. IEEE Computer Society. 2169-2178 p.
- Lee, J.-S. 2004. "Hybrid genetic algorithms for feature selection". IEEE Trans. Pattern Anal. Mach. Intell., 26(11):1424–1437 p.
- Legrand, P. 2004. "Debruitage et interpolation par analyse de la regularite hölderienne. application a la modelisation du frottement pneumatique-chaussee". Tesis de Doctorado, Université de Nantes, France.
- Legrand, P. y J. Lévy-Véhel 2003. "Local regularity-based interpolation". En: "WAVELET X, Part of SPIE's Symposium on Optical Science and Technology", volume 5207.
- Legrand, P., E. Lutton, y G. Olague 2006. "Evolutionary denoising based on an estimation of hölder exponents with oscillation". En: "Proceedings of EvoIASP2008 Ninth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, incorporated in Evo* 2008, April 10-12, Budapest, Hungary", volume 3907.
- Leibe, B., A. Ettlin, y B. Schiele 2008. "Learning semantic object parts for object categorization". Image Vision Comput., 26(1):15–26 p.
- Lepetit, V. y P. Fua 2006. "Keypoint recognition using randomized trees". IEEE Trans. Pattern Anal. Mach. Intell., 28(9):1465–1479 p.
- Lévy-Véhel, J. 1995. "Fractal approaches in signal processing". Soft Computing, 3:755–775 p.

- Li, Q., J. Ye, y C. Kambhamettu 2008. "Interest point detection using imbalance oriented selection". Pattern Recogn., 41(2):672–688 p.
- Lin, Y. y B. Bhanu 2005. "Evolutionary feature synthesis for object recognition". IEEE Transactions on Systems, Man and Cybernetics, Part C, Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation, 35(2):156–171 p.
- Lindberg, D. C. 1976. "Theories of vision from Al-kindi to Kepler". The University of Chicago Press. 324 pp.
- Lindeberg, T. 1991. "Discrete scale-space theory and the scale-space primal sketch". Tesis de Doctorado, Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden.
- Lindeberg, T. 1998. "Feature detection with automatic scale selection". International Journal of Computer Vision, 30(2):79–116 p.
- Lombraña, D., F. Fernández de Vega, G. Olague, L. Trujillo, y B. Segal 2007. "Customizable execution environments with virtual desktop grid computing". En: "PDCS 2007: Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing Systems, 19-21 November, Cambridge MA, USA". Morgan Kaufmann Publishers Inc. 7-12 p.
- Lowe, D. G. 1999. "Object recognition from local scale-invariant features.". En: "Proceedings of the International Conference on Computer Vision (ICCV), 20-25 September, Kerkyra, Corfu, Greece", volume 2. IEEE Computer Society. 1150-1157 p.
- Lowe, D. G. 2004. "Distinctive image features from scale-invariant keypoints". Int. J. Comput. Vision, 60(2):91–110 p.
- Lu, J., T. Zhao, y Y. Zhang 2008. "Feature selection based-on genetic algorithm for image annotation". Knowledge-Based Systems. to appear.
- Lvy Vhel, J. y S. Seuret 2004. "The 2-microlocal formalism". En: "Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics", volume 72.
- M. Litzkow, T. Tannenbaum, J. B. y M. Livny 1997. "Checkpoint and migration of unix processes in the condor distributed processing system". Reporte tonico, University of Wisconsin.
- Markou, M. y S. Singh 2006. "A neural network-based novelty detector for image sequence analysis". IEEE Trans. Pattern Anal. Mach. Intell., 28(10):1664–1677 p.
- Marr, D. 1982. "Vision: A computational investigation into the human representation and processing of visual information". Henry Holt and Co., Inc., New York, NY. 397 pp.

- McGlone, J. y others, editores 2004. "Manual of photogrammetry". American Society of Photogrammetry and Remote Sensing.
- Mian, A. S., M. Bennamoun, y R. Owens 2008. "Keypoint detection and local feature matching for textured 3d face recognition". Int. J. Comput. Vision, 79(1):1–12 p.
- Mikolajczyk, K. y C. Schmid 2001. "Indexing based on scale invariant interest points". En: "Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada". 525–531 p.
- Mikolajczyk, K. y C. Schmid 2004. "Scale & affine invariant interest point detectors". International Journal of Computer Vision, 60(1):63–86 p.
- Mikolajczyk, K. y C. Schmid 2005. "A performance evaluation of local descriptors". IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10):1615–1630 p.
- Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, y L. Van Gool 2005. "A comparison of affine region detectors". International Journal of Computer Vision, 65(1-2):43–72 p.
- Miller, J. F. y S. L. Smith 2006. "Redundancy and computational efficiency in cartesian genetic programming". IEEE Trans. Evolutionary Computation, 10(2):167-174 p.
- Mokhtarian, F. y R. Suomela 1998. "Robust image corner detection through curvature scale space". IEEE Trans. Pattern Anal. Mach. Intell., 20(12):1376–1381 p.
- Montana, D. J. y L. Davis 1989. "Training feedforward neural networks using genetic algorithms". En: Sridharan, S., editor, "Proceedings of the Eleventh International Joint Conference on Artificial Intelligence", San Francisco, CA. Morgan Kaufman. 762-767 p.
- Moravec, H. P. 1977. "Towards automatic visual obstacle avoidance". En: "IJCAI". 584 pp.
- Moreels, P. y P. Perona 2007. "Evaluation of features detectors and descriptors based on 3d objects". Int. J. Comput. Vision, 73(3):263–284 p.
- Moscato, P. 1989. "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms". Reporte tonico, Caltech Concurrent Computation Program.
- Mutch, J. y D. G. Lowe 2006. "Multiclass object recognition with sparse, localized features". En: "Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)", Washington, DC, USA. IEEE Computer Society. 11–18 p.

- Nene, S., Nayar, y H. S.K. Murase 1996. "Columbia object image library (coil-100)". Reporte tcnico, Department of Comuter Science. Columbia University. http://www.cs.olumbia.edu/CAVE.
- Nieh, J. y O. C. Leonard 2000. "Examining VMware". j-DDJ, 25(8):70, 72–74, 76 p.
- Ohta, Y., T. Kanade, y T. Sakai 1980. "Color information for region segmentation". Computer Graphics and Image Processing, 13(3):222–241 p.
- Ojala, T., M. Pietikäinen, y T. Mäenpää 2002. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". IEEE Trans. Pattern Anal. Mach. Intell., 24(7):971–987 p.
- Olague, G., S. Cagnoni, y E. Lutton 2006a. "Preface: introduction to the special issue on evolutionary computer vision and image understanding". Pattern Recognition Letters, 27(11):1161–1163 p.
- Olague, G., F. Fernández, C. B. Pérez, y E. Lutton 2006b. "The infection algorithm: An artificial epidemic approach for dense stereo correspondence". Artificial Life, 12(4):593–615 p.
- Olague, G. y B. Hernández 2005. "A new accurate and flexible model based multi-corner detector for measurement and recognition". Pattern Recognition Letters, 26(1):27–41 p.
- Olague, G. y R. Mohr 2002. "Optimal camera placement for accurate reconstruction". Pattern Recognition Journal, 35:927–944 p.
- Olague, G. y C. Puente 2006. "The honeybee search algorithm for three-dimensional reconstruction.". En: Rothlauf, F., J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. H. Moore, J. Romero, G. D. Smith, G. Squillero, y H. Takagi, editores, "Proceedings from EvoWorkshops 2006 (EVOIASP), Budapest, Hungary, April 10-12", volume 3907 of LNCS. Springer-Verlag. 427-437 p.
- Olague, G., E. Romero, L. Trujillo, y B. Bhanu 2007. "Multiclass object recognition based on texture linear genetic programming". En: "Proceedings of EvoIASP2007 Ninth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, incorporated in Evo* 2007, 11-13 April, Valencia, Spain", LNCS. Springer-Verlag. 291-300 p.
- Oliva, A. y A. Torralba 2001. "Modeling the shape of the scene: A holistic representation of the spatial envelope". International Journal of Computer Vision, 42(3):145– 175 p.
- Paalanen, P., J.-K. Kamarainen, J. Ilonen, y H. Kälviäinen 2006. "Feature representation and discrimination based on gaussian mixture model probability densitiespractices and algorithms". Pattern Recognition, 39(7):1346–1358 p.

Pareto, V. 1896. "Cours d'economie politique". Rouge, Lausanne.

- Pérez, C. B. y G. Olague 2008. "Learning invariant region descriptor operators with genetic programming and the f-measure". En: "Proceedings of ICPR". aceptado.
- Perkis, T. 1994. "Stack-based genetic programming". En: "Proceedings of the 1994 IEEE World Congress on Computational Intelligence, Orlando, FL, 27-29 June", volume 1. IEEE Press. 148-153 p.
- Perronnin, F. 2008. "Universal and adapted vocabularies for generic visual categorization". IEEE Trans. Pattern Anal. Mach. Intell., 30(7):1243-1256 p.
- Platel, B., E. Balmachnova, L. Florack, y B. M. ter Haar Romeny 2006. "Top-points as interest points for image matching.". En: Leonardis, A. y others, editores, "Proceedings from ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Part I", volume 3951 of Lecture Notes in Computer Science. Springer. 418-429 p.
- Poli, R. 1999. "Parallel distributed genetic programming". En: "New ideas in optimization". McGraw-Hill Ltd., UK, Maidenhead, UK, England, 403–432 p.
- Poli, R. 2000. "Hyperschema theory for gp with one-point crossover, building blocks, and some new results in ga theory". En: "Proceedings of the European Conference on Genetic Programming", London, UK. Springer-Verlag. 163–180 p.
- Poli, R., W. B. Langdon, y N. F. McPhee 2008. "A field guide to genetic programming". Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk. (With contributions by J. R. Koza), 250 pp.
- Potter, M. 1997. "The design and analysis of a computational model of cooperative coevolution". Tesis de Doctorado, George Mason University, Fairfax, VA, USA.
- Price, K., R. M. Storn, y J. A. Lampinen 2005. "Differential evolution: A practical approach to global optimization". Natural Computing Series. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 566 pp.
- Raymer, M., W. Punch, E. Goodman, y L. Kuhn 1996. "Genetic programming for improved data mining – application to the biochemistry of protein interactions". En: "Proceedings of the Second Annual Conference on Genetic Programming", LNCS. Morgan Kaufmann. 375–380 p.
- Raymer, M. L., W. F. P. III, E. D. Goodman, L. A. Kuhn, y A. K. Jain 2000. "Dimensionality reduction using genetic algorithms". IEEE Trans. Evolutionary Computation, 4(2):164-171 p.
- Rebai, A., A. Joly, y N. Boujemaa 2007. "Interpretability based interest points detection". En: "CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval", New York, NY, USA. ACM. 33–40 p.

- Rechenberg, I. 1965. "Cybernetic solution path of an experimental problem". Lybrary Translation 1122, Royal Aircraft Establishement, Farnborough, UK.
- Robert-Démolaize, G. 2006. "Design and performance optimization of the lhc collimation system". Tesis de Maestría, CERN.
- Robin, J., C. Irvine, y N. P. S. M. C. D. O. C. SCIENCE 2000. "Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor". Defense Technical Information Center.
- Rohr, K. 1992. "Recognizing corners by fitting parametric models". Int. J. Comput. Vision, 9(3):213–230 p.
- Samuel, A. 1983. "Ai: Where it has been and where it is going". En: "Proceedings of the Eighth International Joint Conference on Artificial Intelligence", San Francisco, CA. Morgan Kaufman. 1152–1157 p.
- Schaffalitzky, F. y A. Zisserman 2002. "Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?"". En: "ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I", London, UK. Springer-Verlag. 414–431 p.
- Schmid, C. y R. Mohr 1997. "Local grayvalue invariants for image retrieval". IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5):530–534 p.
- Schmid, C., R. Mohr, y C. Bauckhage 2000. "Evaluation of interest point detectors". International Journal of Computer Vision, 37(2):151–172 p.
- Schwartz, R. 2006. "Visual versions". The MIT Press. 265 pp.
- Schwefel, H.-P. P. 1993. "Evolution and optimum seeking: The sixth generation". John Wiley & Sons, Inc., New York, NY, USA. 456 pp.
- Se, S., D. Lowe, y J. Little 2002. "Global localization using distinctive visual features". En: "Proceedings of the International Conferences on Intelligent Robots and Systems (IROS), Lausanne, Switzerland". 226–231 p.
- Sebe, N. y M. S. Lew 2003. "Comparing salient point detectors". Pattern Recogn. Lett., 24(1-3):89–96 p.
- Shannon, C. E. 1950. "Prediction and entropy of printed english". The Bell System Technical Journal, 30:50-64 p.
- Sherrah, J. R., R. E. Bogner, y A. Bouzerdoum 1997. "The evolutionary pre-processor: Automatic feature extraction for supervised classification using genetic programming". En: Koza, J. R., K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, y R. L. Riolo, editores, "Genetic Programming 1997: Proceedings of the Second Annual Conference", Stanford University, CA, USA. Morgan Kaufmann. 304–312 p.

- Shi, J. y C. Tomasi 1994. "Good features to track". En: "Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), June 1994, Seattle, WA, USA". IEEE Computer Society. 593–600 p.
- Siedlecki, W. y J. Sklansky 1989. "A note on genetic algorithms for large-scale feature selection". Pattern Recogn. Lett., 10(5):335–347 p.
- Sinzinger, E. D. 2008. "A model-based approach to junction detection using radial energy". Pattern Recogn., 41(2):494–505 p.
- Sivic, J., B. C. Russell, A. A. Efros, A. Zisserman, y W. T. Freeman 2005. "Discovering objects and their localization in images.". En: "Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV), 17-20 October, Beijing, China", volume 1. IEEE Computer Society. 370-377 p.
- Sivic, J. y A. Zisserman 2003. "Video google: A text retrieval approach to object matching in videos". En: "Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03)", Washington, DC, USA. IEEE Computer Society. 1470 pp.
- Smith, S. M. y J. M. Brady 1997. "Susan-a new approach to low level image processing". Int. J. Comput. Vision, 23(1):45–78 p.
- Sun, Z., G. Bebis, y R. Miller 2004. "Object detection using feature subset selection". Pattern Recognition, 37(11):2165-2176 p.
- Szummer, M. y R. W. Picard 1998. "Indoor-outdoor image classification". En: "CAIVD '98: Proceedings of the 1998 International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98)", Washington, DC, USA. IEEE Computer Society. 42 pp.
- Tackett, W. A. 1993. "Genetic programming for feature discovery and image discrimination". En: Forrest, S., editor, "Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93", University of Illinois at Urbana-Champaign. Morgan Kaufmann. 303–309 p.
- Tan, X., B. Bhanu, y Y. Lin 2005. "Fingerprint classification based on learned features". IEEE Transactions on Systems, Man, and Cybernetics, Part C, 35(3):287-300 p.
- Teller, A. y M. Veloso 1996. "PADO: A new learning architecture for object recognition". En: Ikeuchi, K. y M. Veloso, editores, "Symbolic Visual Learning". Oxford University Press, 81–116 p.
- Tissainayagam, P. y D. Suter 2004. "Assessing the performance of corner detectors for point feature tracking applications.". Image Vision Comput., 22(8):663-679 p.

- Tola, E., V. Lepetit, y P. Fua 2008. "A fast local descriptor for dense matching". En: "Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June, Anchorage, AK". IEEE Computer Society.
- Torralba, A., K. P. Murphy, W. T. Freeman, y M. A. Rubin 2003. "Context-based vision system for place and object recognition". En: "ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision", Washington, DC, USA. IEEE Computer Society. 273 pp.
- Tricot, C. 1995. "Curves and fractal dimension". Springer-Verlag. 323 pp.
- Trujillo, L. y G. Olague 2006a. "Synthesis of interest point detectors through genetic programming". En: Cattolico, M., editor, "Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Seattle, Washington, July 8-12", volume 1. ACM. 887-894 p.
- Trujillo, L. y G. Olague 2006b. "Using evolution to learn how to perform interest point detection". En: "Proceedings of the 18th International Conference on Pattern Recognition (ICPR), 20-24 August, Hong Kong, China", volume 1. IEEE Computer Society. 211-214 p.
- Trujillo, L. y G. Olague 2007. "Scale invariance for evolved interest operators". En: "Proceedings of EvoIASP2007 Ninth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, incorporated in Evo* 2007, 11-13 April, Valencia, Spain", LNCS. Springer-Verlag. 423-430 p.
- Trujillo, L. y G. Olague 2008. "Automated design of image operators that detect interest points". Evolutionary Computation. to appear.
- Trujillo, L., G. Olague, F. Fernández de Vega, y E. Lutton 2007a. "Evolutionary feature selection for probabilistic object recognition, novel object detection and object saliency estimation using gmms". En: "Proceedings from the 18th British Machine Vision Conference, 10-13 September, Warwick, UK". British Machine Vision Association. 630–639 p.
- Trujillo, L., G. Olague, F. Fernández de Vega, y E. Lutton 2008a. "Selecting local region descriptors with a genetic algorithm for real-world place recognition". En: "Proceedings of EvoIASP2008 Ninth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, incorporated in Evo* 2008, 26-28 March, Napoli, Italy", LNCS. 325-334 p.
- Trujillo, L., G. Olague, R. Hammoud, y B. Hernandez 2005. "Automatic feature localization in thermal images for facial expression recognition". En: "Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) - Workshops", Washington, DC, USA. IEEE Computer Society. 14 pp.

- Trujillo, L., G. Olague, P. Legrand, y E. Lutton 2007b. "Regularity based descriptor computed from local image oscillations". Optics Express, 15:6140-6145 p.
- Trujillo, L., G. Olague, E. Lutton, y F. Fernández de Vega 2008b. "Multiobjective design of operators that detect points of interest in images". En: Cattolico, M., editor, "Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Atlanta, GA, July 12-16", New York, NY, USA. ACM. 1299–1306 p.
- Tuytelaars, T. y K. Mikolajczyk 2008. "Local invariant feature detectors: a survey". Found. Trends Comput. Graph. Vis., 3(3):177–280 p.
- Tuytelaars, T. y L. Van Gool 2004. "Matching widely separated views based on affine invariant regions". Int. J. Comput. Vision, 59(1):61–85 p.
- Unnikrishnan, R., C. Pantofaru, y M. Hebert 2007. "Toward objective evaluation of image segmentation algorithms". IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):929-944 p.
- van de Weijer, J., T. Gevers, y A. D. Bagdanov 2006. "Boosting color saliency in image feature detection". IEEE Trans. Pattern Anal. Mach. Intell., 28(1):150–156 p.
- Van Gool, L. J., T. Moons, y D. Ungureanu 1996. "Affine/photometric invariants for planar intensity patterns". En: "Proceedings of the 4th European Conference on Computer Vision (ECCV '96)", London, UK, volume 1. Springer-Verlag. 642–651 p.
- Van Rijsbergen, C. 1979. "Information retrieval". Butterworth-Heinemann. 358 pp.
- Verbeek, J. J., N. Vlassis, y B. Kröse 2003. "Efficient greedy learning of gaussian mixture models". Neural Comput., 15(2):469–485 p.
- Vetterli, M. y J. Kovačevic 1995. "Wavelets and subband coding". Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 448 pp.
- Vogel, J. y B. Schiele 2007. "Semantic modeling of natural scenes for content-based image retrieval". Int. J. Comput. Vision, 72(2):133–157 p.
- Wang, H. y J. Brady 1991. "Corner detection for 3d vision using array processors". En: "Proceedings from BARNAIMAGE 91, Barcelona, Spain", Secaucus, NJ. Springer-Verlag.
- Wang, H. y M. Brady 1995. "Real-time corner detection algorithm for motion estimation". Image Vision Comput., 13(9):695-703 p.
- Wang, J., H. Zha, y R. Cipolla 2006. "Coarse-to-fine vision-based localization by indexing scale-invariant features". IEEE Transactions on Systems, Man, and Cybernetics, Part B, 36(2):413-422 p.

- Watson, R. A. 2002. "Compositional evolution: Interdisciplinary investigations in evolvability, modularity, and symbiosis". Tesis de Doctorado, Brandeis University, Waltham, MA, USA.
- Willamowski, J., D. Arregui, G. Csurka, C. Dance, y L. Fan 2004. "Categorizing nine visual classes using local appearance descriptors". En: "Proceedings of the 17th International Conference on Pattern Recognition, Workshop Learning for Adaptable Visual Systems, 23-26 August, Cambridge, United Kingdom". IEEE Computer Society.
- Wittgenstein, L. 1953. "Philosophical investigations". Macmillan Co., New York, NY. Translated by G. E. M. Anscombe, 212 pp.
- Wolf-Devine, C. 1993. "Descartes on seeing: Epistemology and visual perception". The Journal of the History of Philosophy Monograph. Southern Illinois University Press. 121 pp.
- Yang, G., C. V. Stewart, M. Sofka, y C.-L. Tsai 2007. "Registration of challenging image pairs: Initialization, estimation, and decision". IEEE Trans. Pattern Anal. Mach. Intell., 29(11):1973–1989 p.
- Yao, J., N. Kharma, y P. Grogono 2005. "A multi-population genetic algorithm for robust and fast ellipse detection". Pattern Anal. Appl., 8(1):149–162 p.
- Ylä-Jääski, A. y F. Ade 1996. "Grouping symmetrical structures for object segmentation and description". Comput. Vis. Image Underst., 63(3):399–417 p.
- Young, R. A. 1986. "Simulation of human retinal function with the gaussian derivative model". En: "Proceedings of the 1986 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)". IEEE. 564–569 p.
- Yu, J. y B. Bhanu 2006. "Evolutionary feature synthesis for facial expression recognition". Pattern Recogn. Lett., 27(11):1289–1298 p.
- Yu, S., S. D. Backer, y P. Scheunders 2002. "Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery". Pattern Recogn. Lett., 23(1-3):183–190 p.
- Zabih, R. y J. Woodfill 1994. "Non-parametric local transforms for computing visual correspondence". En: "Proceedings of the third European conference on Computer Vision (ECCV '94)", Secaucus, NJ, USA, volume 2. Springer-Verlag New York, Inc. 151–158 p.
- Zhang, M., V. B. Ciesielski, y P. Andreae 2003. "A domain-independent window approach to multiclass object detection using genetic programming". EURASIP Journal on Applied Signal Processing, Special Issue on Genetic and Evolutionary Computation for Signal Processing and Image Analysis, 2003(8):841–859 p.

- Zhang, M. y W. Smart 2006. "Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification". Pattern Recogn. Lett., 27(11):1266–1274 p.
- Zhang, M. y W. D. Smart 2004. "Multiclass object classification using genetic programming.". En: Raidl, G. R., S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, C. G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, y G. Squillero, editores, "EvoWorkshops", volume 3005 of LNCS. Springer. 369–378 p.
- Zheng, W., J.-H. Lai, y P. C. Yuen 2005. "Ga-fisher: a new lda-based face recognition algorithm with selection of principal components". IEEE Transactions on Systems, Man, and Cybernetics, Part B, 35(5):1065-1078 p.
- Zhu, S. C., C. en Guo, Y. Wang, y Z. Xu 2005. "What are textons?". International Journal of Computer Vision, 62(1-2):121-143 p.
- Zitzler, E., M. Laumanns, y S. Bleuler 2004. "A Tutorial on Evolutionary Multiobjective Optimization". En: Gandibleux, X. y others, editores, "Metaheuristics for Multiobjective Optimisation", Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.
- Zitzler, E., M. Laumanns, y L. Thiele 2002. "Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization". En: "Evolutionary Methods for Design, Optimisation, and Control". 19–26 p.

Apéndice A

Computación GRID voluntaria con BOINC

Hoy en día es común que diferentes tipos de instituciones cuenten con equipo de cómputo de alto rendimiento. Una PC moderna tiene un procesador poderoso, normalmente con varios núcleos, más de 1 GB de memoria RAM, discos duros grandes y veloces, etc. Sin embargo, todas esas cualidades por lo común no son explotadas en su totalidad, primordialmente porque las computadoras normalmente se utilizan para tareas sencillas; e.g., navegar por internet o trabajar con un editor de texto.

En esta apéndice se presenta un esquema computacional que nos permitirá explotar los recursos de cómputo que estan ociosos. La propuesta utiliza el cómputo distribuido voluntario a través del *middleware* BOINC y una capa de virtualización con VMware.

Existen sistemas de software diseñados específicamente para explotar este tipo de recursos cómputo durante periodos de inactividad. Estas tecnologías se conocen como Computación GRID Voluntaria (en inglés, Volunteer GRID Computing: VGC), y tienen como meta establecer una infraestructura que permita explotar al máximo los recursos computacionales de PCs de escritorio, de una forma coordinada y robusta.

Diferentes paquetes o herramientas de software han sido diseñados para lograr esta meta, normalmente en forma de *middleware*, tales como: *Condor* (M. Litzkow y Livny, 1997), *Xtremweb* (Fedak *et al.*, 2001), *BOINC* (Anderson, 2004), entre otros. De los anteriores, BOINC es el más conocido y ampliamente utilizado, con una comunidad muy grande de apoyo y proyectos que incluyen trabajos sobre predicciones climáticas (Allen, 1999) y estudios físicos (Robert-Démolaize, 2006), por nombrar algunos.

A.1 Infraestructura BOINC

BOINC nació del proyecto SETI@HOME (en inglés, Search for Extra Terrestial Intelligence: SETI) (Anderson *et al.*, 2002), en donde los investigadores analizan señales de telescopios de radio para buscar evidencia de inteligencia extraterrestre. Para esto se requieren de cantidades enormes de poder computacional, debido al volumen abrumador de información que se debe analizar. El proyecto SETI invito a gente a participar de manera voluntaria a través del internet, en donde cada usuario presta los recursos computacionales de su PC. De esta forma, el proyecto SETI ha alcanzado un poder computacional de 414.04 TeraFLOPS. Debido al éxito del proyecto SETI, los investigadores liberaron el *middleware* que les permito lograrlo, *The Berkeley Open Infrastracture for Network Computing* (BOINC).



Figura 113: Arquitectura Servidor/Cliente.

BOINC se caracteriza por tener dos rasgos claves: 1) es *multi-plataforma* (MacOSX, MS Windows y GNU/Linux); y 2) es de *código libre*. Emplea una arquitectura de *Servidor/Cliente*, en donde los clientes son las PCs de escritorio y el *servidor* se encarga de distribuir y asignar tareas a los clientes, ver Figura 113.

A.1.1 El Cliente

El cliente es una aplicación multi-plataforma que puede correr en Windows, Linux y MacOSX. El cliente se hace disponible al servidor de un proyecto BOINC, al cual le solicita trabajos para procesar. Una vez que el servidor tiene trabajos disponibles, el cliente puede descargar los archivos necesarios para ejecutar la tarea que se le asigna. Después de que el cliente ejecuta el cómputo necesario para dar por terminada la tarea, este sube los resultados al servidor BOINC y de nuevo se pone disponible para ejecutar tareas nuevas.

A.1.2 El Servidor

El servidor es el lugar donde se establece cuales serán las tareas que se van a realizar por los clientes; se encarga de los siguiente:

- Contiene los proyectos o experimentos científicos. Un proyecto se compone de un binario (el algoritmo), y archivos de entrada o configuración. BOINC construye binarios diferentes para cada arquitectura, hardware y sistema operativo.
- Crear y distribuir trabajos. El servidor utiliza los binarios y los archivos de entrada para crear una Unidad de Trabajo (en inglés, Work Unit: WU). Un WU describe como se ejecutara un experimento en el cliente, especifica cual es el binario, los archivos, los argumentos de entrada, y el nombre de los archivos de salida.
- Asimilación y validación de los resultados. Cuando los clientes terminan su cómputo, los archivos de salida se colocan en el servidor. Después, el servidor
inicia dos procesos. Primero, el programa de *validación* que determina si los resultados son coherentes y confiables. Segundo, el programa de *asimilación* se encarga de agregar los archivos de salida al proyecto para poder: calcular estadísticas, exportar los resultados a una base de datos, etc.

A.1.3 Como utilizar BOINC

Como se puede ver, la estructura de BOINC es sencilla, pero el trabajo requerido para construir un proyecto BOINC puede ser muy complejo debido a que se debe codificar utilizando las librerías BOINC; en este aspecto BOINC es muy rígido. Un proyecto BOINC se puede construir de cuatro formas diferentes:

- 1. Desde el inicio. En este caso, el proyecto BOINC se construye completamente nuevo. No existe código fuente inicial, o binarios, por lo que el desarrollador puede considerar todos los requerimientos de las librerías y estructuras de programación BOINC desde el inicio. Debido a que BOINC esta escrito en C++, el desarrollador programa su experimento también en C++; este se puede ver como el caso ideal para un proyecto BOINC.
- 2. Adaptar un programa C++. En este caso, el programador debe cambiar los métodos de Entrada/Salida (I/O) que la aplicación utiliza, por las rutinas que especifica BOINC. Este cambio es necesario porque la aplicación se va a distribuir dentro de un medio paralelo, y BOINC contempla todos los problemas que se pudieran generar en dicho medio.
- 3. Portar un programa a C++. En este caso, se tiene una aplicación escrita en otro lenguaje de programación. En este escenario, el programa deberá de ser reescrito en C++. Sin embargo, el proceso puede no ser trivial, y el trabajo necesario para realizar esto pudiera prohibir que un investigador utilice BOINC como una infraestructura de cómputo distribuido.
- 4. Utilizando el wrapper/starter. El wrapper es un programa BOINC pequeño que permite correr aplicaciones que no utilizan el API BOINC, por ejemplo cuando solo se cuenta con un binario o cuando portar el código no es factible. El starter, se encarga de activar el experimento escrito en un lenguaje alterno, por ejemplo lenguajes interpretados como MATLAB o R. De tal forma que BOINC solo se comunica con el wrapper, este activa el starter, y finalmente el último se encarga de iniciar el experimento científico; la Figura 114 muestra un diagrama conceptual.

De los métodos anteriores, se utiliza el cuarto durante la experimentación que se describe en esta Sección debido a las características del algoritmo que se introduce en este Capítulo para la síntesis de operadores que detectan puntos de interés.



Figura 114: El wrapper y el starter.

A.2 Virtualización

Una capa de virtualización permite abstraer el *hardware* de computadoras de escritorio; básicamente, a través de una máquina virtual (en inglés, Virtual Machine: VM) se carga un sistema operativo y aplicaciones en una computadora que normalmente no lo permitiera. La VM a su vez es administrada por un software llamado monitor de máquina virtual (en inglés, Virtual Machine Monitor: VMM). El paradigma de la virtualización tiene las siguientes características:

- 1. *Aisla recursos.* La virtualización aisla cada VM dentro de la máquina huésped; esta característica es muy útil porque significa que cualquier falla en el código que corre sobre una VM no afecta a la máquina real.
- 2. Se crea una instancia de un sistema operativo invitado. Esto significa que es posible cargar una imagen de un sistema operativo en cualquier máquina que sea compatible con la VM utilizada.
- 3. Snapshot o serialización de estados. Esto se basa en el uso de banderas o checkpoints (Elnozahy et al., 2002), y le permite a una VM detener la ejecución de todo el sistema y volver a iniciarlo en ese mismo punto.

Esta tecnología se puede implementar de diferentes formas:

1. Virtualización nativa. En este enfoque, el hardware real es igual al virtual en términos de funcionalidad; de esta forma, se evita tener que modificar el sistema operativo que se instala en la VM. Sin embargo, existe un problema serio debido a que la arquitectura x86 no fue diseñada para permitir una virtualización completa (Robin et al., 2000). Esto significa que la VMM tiene que capturar cualquier instrucción x86 que pudiera causar problemas a la VM. Esta solución tiene dos costos, incrementa la complejidad del VMM, y por ende reduce el rendimiento de la VM. El ejemplo más conocido que utiliza este enfoque es VMware (Nieh y Leonard, 2000), el cliente de este software es freeware y existen versiones para muchos sistemas operativos.

2. *Para-virtualización*. En esta técnica lo que se busca es crear una VM que tiene un *hardware* similiar en lugar de tener que ser idéntico al real. Para poder lograr esto, se requiere modificar el kernel del sistema operativo que se ejecuta dentro de la VM. Sin embargo, esta modificación no afecta la interfaz que se le presenta a las aplicaciones, y por lo tanto estas pueden ejecutarse sin ser modificadas. Un ejemplo de una VM que utiliza este enfoque es Xen (Barham *et al.*, 2003), este software es de código libre y solo funciona en GNU/Linux.

A.2.1 Virtualización y tecnologías GRID

Uno de los principales problemas con el que se enfrentan las tecnologías de tipo GRID es la heterogeneidad del *hardware* y *software*. Una tecnología GRID busca unir diferentes tipos de especificaciones de *hardware* y *software* dentro de una sola capa sencilla y extensible. Dada la heterogeneidad de los sistemas, esta capa normalmente se vuelve muy compleja. En muchos casos, ejecutar un experimento científico en un medio GRID no es factible porque el código depende de un conjunto muy especifico de librerías, versiones de *software* y especificaciones de *hardware*. Sin embargo, como se explica arriba, una capa de virtualización puede ayudar a minimizar estos problemas debido a la capa de abstracción inherente en esta tecnología.

A.3 Extensión de BOINC con una capa de virtualización

Debido a que BOINC tiene la habilidad de operar en cualquier plataforma de PC, se utiliza VMware como la capa de virtualización para extender la funcionalidad del VGC. Cuando se agrega una VM al funcionamiento de BOINC, nos permite distribuir de forma paralela el código utilizando BOINC sin tener que modificar una sola linea de código.

Como se menciona arriba, BOINC trabaja con WUs, que contiene: 1) el binario; 2) archivos de entrada o configuración; y 3) los archivos de salida. Por lo tanto, para integrar una VM con BOINC se utiliza el binario de VMware como el binario del WU, y el esquema *wrapper/starter* permite ejecutar los programas dentro de un ambiente distribuido BOINC sin tener que modificar la aplicación original. La Figura 115 muestra un esquema gráfico de este proceso, en donde cada proyecto BOINC se compone de lo siguiente:

- 1. *Imagen VMware*. VMware funciona con imágenes, por lo tanto solo se crea una imagen del sistema mínimo que se requiere para ejecutar el experimento. La imagen es una PC virtual con su *hardware* y sistema operativo, además del experimento que se va a ejecutar.
- 2. *Starter*. Este es un *script* que se utiliza para iniciar el programa VMware Player. Adicionalmente, se utiliza para tomar *snapshots* de la VM que se esta ejecutando.



Figura 115: Clientes con BOINC y VMware trabajando con el Servidor BOINC.

En caso de que la PC huésped se apague, o la ejecución se detenga por otro motivo, el *starter* puede reanudar el sistema en el punto donde se tomo por ultima vez un *snapshot*.

3. Wrapper. Es la interfaz entre BOINC y el starter, ver Figura 114.