TESIS DEFENDIDA POR

Cynthia Beatriz Pérez Castro

Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. Gustavo Olague Caballero

Director del Comité

Dr. Rafael de Jesús Kelly Martínez

Miembro del Comité

Dr. Luis Enrique Sucar Succar

Miembro del Comité

Dr. Andrei Tchernykh Graboskaya

Miembro del Comité

Dr. Jorge Torres Rodríguez

Miembro del Comité

Dr. Hugo Homero Hidalgo Silva

Coordinador del programa de posgrado en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales

Director de Estudios de Posgrado

Julio del 2010

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

PROGRAMA DE POSGRADO EN CIENCIAS EN CIENCIAS DE LA COMPUTACIÓN

CÓMPUTO EVOLUTIVO COMO ENFOQUE EN LA DESCRIPCIÓN DEL CONTENIDO DE LA IMAGEN APLICADO A LA SEGMENTACIÓN Y EL RECONOCIMIENTO DE OBJETOS

TESIS

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

DOCTOR EN CIENCIAS

Presenta:

CYNTHIA BEATRIZ PÉREZ CASTRO

Ensenada, Baja California, México, Julio de 2010

RESUMEN de la tesis de **CYNTHIA BEATRIZ PÉREZ CASTRO**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN . Ensenada, Baja California, Julio de 2010.

CÓMPUTO EVOLUTIVO COMO ENFOQUE EN LA DESCRIPCIÓN DEL CONTENIDO DE LA IMAGEN APLICADO A LA SEGMENTACIÓN Y EL RECONOCIMIENTO DE OBJETOS

Resumen aprobado por:

Dr. Gustavo Olague Caballero

Director de Tesis

La habilidad para analizar y describir la información contenida en las imágenes provenientes de escenarios reales o diseñados por el hombre es una tarea que resulta retadora y atractiva para diferentes áreas de la ciencia. Este trabajo presenta dos algoritmos evolutivos los cuales analizan el contenido de la imagen para dos aplicaciones de la Visión por Computadora, como lo es: la segmentación y el reconocimiento de objetos. Por un lado se realiza un análisis del contenido de imágenes con textura utilizando descriptores estadísticos por medio de la matriz de co-ocurrencia para llevar a cabo la segmentación. Por otro lado, se proponen nuevos descriptores locales que describen el contenido de la imagen para reconocer objetos en escenarios al interior y al exterior. En ambos casos, el análisis del contenido de la imagen es requerido para llevar a cabo la tarea lo más eficientemente posible. En el primer trabajo presentamos nuestro algoritmo de segmentación *EvoSeg* el cual utiliza conocimiento derivado del análisis de textura para identificar el número de regiones homogéneas existentes en la escena sin contar con ninguna información a priori. EvoSeg utiliza descriptores estadísticos en una matriz de co-ocurrencias en escala de grises y optimiza una medida de aptitud basada en el criterio de varianza mínima usando un algoritmo genético canónico. Posteriormente, se incorpora interacción dentro del proceso de optimización del algoritmo *EvoSeg* en donde una persona experta interactúa con el sistema ayudando a mejorar los resultados. La interacción permite al algoritmo adaptarse usando esta nueva información externa basada en el criterio del experto. Finalmente, se presentan resultados experimentales para ambos algoritmos utilizando la base de datos de la USC-SIPI (Signal and Image Processing Institute).

El segundo trabajo describe una metodología basada en la programación genética que sintetiza expresiones matemáticas que son usadas para mejorar el conocido descriptor del estado del arte, SIFT (Scale Invariant Feature Transform). El mecanismo de reconocimiento de este trabajo está inspirado en la idea de cómo se hace el reconocimiento en la corteza cerebral de los primates al hacer uso de características de una complejidad intermedia que son invariantes a cambios de escala, localización e iluminación de manera natural. Estas características locales han sido diseñadas por personas expertas en el área que hacen uso de las representaciones tradicionales a través de definiciones matemáticas bien fundamentadas. Sin embargo, no es muy claro que estas mismas representaciones sean implementadas por el sistema natural de la misma manera. Es por ello, que la posibilidad para diseñar novedosos operadores a través de la programación genética representa una linea de investigación abierta donde la búsqueda combinatoria de los algoritmos evolutivos pueden exceder a la habilidad de los expertos. En ese sentido, este trabajo provee evidencia que la programación genética es capaz de diseñar nuevas características que mejoran el rendimiento general de los mejores

descriptores locales disponibles en la actualidad. Para ello, los resultados experimentales confirman la validez de nuestro enfoque usando un protocolo de evaluación ampliamente aceptado, así como también el uso de nuestro mejor descriptor $RDGP_2$ en el reconocimiento de objetos en escenarios al interior y exterior.

Palabras Clave: Segmentación, Descriptores Estadísticos, Matriz de Co-ocurrencia (GLCM), Descriptores Locales, Programación Genética, Medida F, Reconocimiento de Objetos

ABSTRACT of the thesis presented by **CYNTHIA BEATRIZ PÉREZ CASTRO**, in partial fulfillment of the requirements of DOCTOR IN SCIENCES degree in COMPUTER SCIENCE . Ensenada, Baja California, July 2010.

EVOLUTIONARY COMPUTATION FOR IMAGE CONTENT DESCRIPTION APPLIED TO IMAGE SEGMENTATION AND OBJECT RECOGNITION

The ability to analyse and describe the image content from real or designed scenarios is a challenge and an attractive task for computer vision. This work presents two evolutionary algorithms which analyse the image content for two particular high level tasks such as: image segmentation and object recognition. On the one hand, the image content is analysed using statistical descriptors in a Gray Level Co-ocurrence Matrix (GLCM) in order to achieve good image segmentations. On the other hand, new local descriptor operators are proposed using genetic programming. These operators describe the image content in order to recognize objects localized within indoor and outdoor scenarios presenting different image transformations. First, we present our EvoSeg algorithm, which uses knowledge derived from texture analysis to identify how many homogeneous regions exist in the scene without a priori information. EvoSeg uses texture features derived from the GLMC and optimizes a fitness measure, based on the minimum variance criteria, using a hierarchical GA. Later, we include interaction within the *EvoSeg* optimization process obtaining a new algorithm named $I - EvoSeg$. This algorithm complements the chosen texture information with direct human interaction in the evolutionary optimization process. Interactive evolution helps to improve results by allowing the algorithm to adapt using the new external information based on user evaluation. Finally, we present experimental results using a standard database used for image segmentation from the USC-SIPI (Signal and Image Processing Institute).

Second, we describe a genetic programming methodology that synthesizes mathematical expressions that are used to improve a well known local descriptor algorithm. It follows the idea that object recognition in the cerebral cortex of primates makes use of features of intermediate complexity that are largely invariant to change in scale, location, and illumination. These local features have been previously designed by human experts using traditional representations that have a clear, preferably mathematically, well-founded definition. However, it is not clear that these same representations are implemented by the natural system with the same representation. Hence, the possibility to design novel operators through genetic programming represents an open research avenue where the combinatorial search of evolutionary algorithms can largely exceed the ability of human experts. Hence, we provide evidence that genetic programming is able to design new features that enhance the overall performance of the best available local descriptor. Experimental results confirm the validity of the proposed approach using a widely accept testbed and an object recognition application for indoor and outdoor scenarios using our best descriptor $RDGP_2$.

Keywords: Segmentation, Statistical Descriptors, Co-ocurrence Matrix (GLCM), Local Descriptors, Genetic Programming, F-Measure, Object Recognition.

A Dios.

Agradecimientos

A mis padres Rodolfo Pérez y Ruth Beatriz Castro por el incondicional apoyo y el gran amor que me han dado durante el transcurso de mi vida.

A mi familia, mi esposo Jesús Villavicencio y mi hijo Jesús Rodolfo por ser mi gran motivación y el motor de mi existencia.

A mi suegra Silvia Navarro, por su incondicional apoyo en los momentos más críticos de mi trabajo de investigación.

A mi hermano Humberto Pérez y mi cuñada Damaris Mirleth por los buenos deseos que siempre me han dado.

A mi director de tesis, el Dr. Gustavo Olague porque hemos logrado un muy buen trabajo de investigación.

A mi comité de tesis, el Dr. Rafael Kelly, Dr. Enrique Sucar, Dr. Jorge Torres y el Dr. Andrei Tchernykh por sus buenos comentarios y sugerencias durante la realización de este trabajo.

Al Dr. David Cobarrubias por el apoyo que nos brindó a mi y a mi familia.

A todos mis amigos por el apoyo, buenos deseos y solidaridad que siempre me han otorgado.

Al CONACyT por el apoyo económico brindado durante mis estudios doctorales.

Contenido

Página

Contenido (continuación)

Página

Lista de Figuras

Lista de Figuras (continuación)

Lista de Figuras (continuación)

Lista de Tablas

Capítulo I

Introducción

La habilidad para analizar la información contenida en las imágenes provenientes de escenarios reales o diseñados por el hombre es una tarea que resulta retadora y atractiva para diferentes áreas de la ciencia, como por ejemplo: en visión por computadora para poder realizar tareas de alto de nivel como el reconocimiento y clasificación de objetos, segmentación y recuperación de imágenes, reconstrucción 3D, entre otros. Todas estas tareas están relacionadas a la manera en cómo el ser humano percibe el mundo para transmitirlo a los dispositivos electrónicos con el fin de recrear lo que para él es una tarea sencilla y natural, pero simularlo en las computadoras es altamente complejo. De esta manera, podemos decir que la visión permite a los seres humanos percibir y entender el mundo que les rodea, mientras que la visión por computadora intenta simular el efecto de esta visión electrónicamente percibiendo y entendiendo el contenido de una imagen; es por ello, que dar a las computadoras la habilidad de ver no es una tarea fácil.

Cuando nosotros percibimos las imágenes a través de nuestros ojos, el cerebro humano a través de su corteza visual procesa la información de una manera extraordinaria, ayudándonos a describir e interpretar texturas, objetos, formas, colores, etc. De esta manera, la acción de ver está relacionado con el estímulo que recibe la retina cuando se proyectan los patrones de la luz recibida del mundo que nos rodea mientras que la percepción es mucho mas complejo que eso, ya que es la interpretación de lo que nosotros vemos, es decir, es explicar lo que el procesamiento visual hace para crear lo que actualmente vemos. Es por ello, que para la visión por computadora es de primordial importancia entender el proceso de la percepción para llevar a cabo eficientemente las tareas de alto nivel. Para ello, se han propuesto diversas técnicas, algoritmos y metodologías para procesar y describir la información que se percibe del mundo real. En ese sentido, en este trabajo desarrollaremos dos algoritmos los cuales

analizan el contenido de la imagen para dos aplicaciones de la visión por computadora, como lo es: la segmentación y el reconocimiento de objetos. Por un lado, se realiza un análisis del contenido de imágenes con textura utilizando descriptores estadísticos por medio de la matriz de co-ocurrencia para llevar a cabo la segmentación. Por otro lado, se proponen nuevos descriptores locales que describen el contenido de la imagen para reconocer objetos en escenarios al interior y al exterior. En ambos casos, el análisis del contenido de la imagen es requerido para llevar a cabo la tarea lo más eficientemente posible. De hecho, la segmentación y el reconocimiento de objetos tiene algunos aspectos en común; ambos pueden ser vistos como problemas de clasificación donde se desea etiquetar de alguna manera los pixeles de la imagen de tal forma que indiquen cuáles son las regiones de la imagen o cuáles son las partes del objeto a reconocer. En ambos problemas, la información recae en características locales de la imagen, las relaciones que existen entre estas características, y una consistencia global. En la segmentación las características locales son las texturas o color de los pixeles, y las relaciones entre ellas se dan por la similaridad de los pixeles; mientras que en el reconocimiento de objetos, las características locales son las partes de los objetos que se desean reconocer y las relaciones entre sus características se lleva a cabo por la similitud entre ellas. En ese sentido, existen trabajos que plantean el problema de segmentación como un problema aislado (Deng y Manjunath (2001); Perez y Olague (2007); Bhandarkar y Zhang (1999)), otros trabajos plantean el problema de segmentación como una etapa para el reconocimiento de objetos (Bandlow *et al.* (2000)), otros realizan simultáneamente el reconocimiento y la segmentación (Ferrari *et al.* (2006)) y finalmente, otros optan por abordar el problema de reconocimiento evitando la etapa de segmentación (Lowe (1999); Bay *et al.* (2006b); Moreno *et al.* (2009); Perez y Olague (2008)). En nuestro caso, nosotros abordamos ambos problemas de manera única, tratando el problema de segmentación con imágenes con textura en escala de grises y el problema del reconocimiento con características locales evitando con ello el problema de segmentación, el cual es computacionalmente muy costoso.

I.1 Descripción General de Nuestro Enfoque

I.1.1 Motivación

La motivación general de este trabajo está relacionada con el hecho de aplicar novedosas técnicas evolutivas en el análsis de imágenes, particularmente para el problema de segmentación y reconocimiento de objetos. En ese sentido, proponemos una nueva metodología que nos permite plantear estos dos problemas de una manera distinta con el fin de obtener resultados mejores o comparables con los del estado del arte. Para el primer problema relacionado con la segmentación de imágenes con textura, desarrollamos un algoritmo evolutivo que identifica el número de regiones homogéneas existentes en la imagen sin utilizar información *a priori* como la mayoría de los algoritmos de segmentación. La plataforma de este algoritmo es un algoritmo genético canónico el cual permite optimizar el número de regiones utilizando información de la imagen a través de descriptores estadísticos utilizados en una matriz de co-ocurrencia. Para llevar a cabo esta investigación utilizamos una base de datos de imágenes con textura llamada Brodatz de la USC-SIPI[1]. Posteriormente, incluimos en este trabajo el cómputo evolutivo-interactivo con el fin de mejorar el proceso de optimización en donde el conocimiento de una persona experta en el área es considerada dentro del proceso de evaluación.

Por otro lado, en el segundo trabajo relacionado con el reconocimiento de objetos, desarrollamos un algoritmo novedoso para diseñar nuevos operadores descriptivos utilizando como base el conocido algoritmo patentado SIFT (Scale Invariant Feature Tranform) propuesto por Lowe (1999). En este algoritmo utilizamos como herramienta evolutiva la programación genética, permitiéndonos diseñar operadores para descriptores locales de manera automatizada gracias al proceso de optimización compuesto por una estructura basada en árboles. Cada árbol representa una fórmula matemática que es aplicada a las regiones más representativas de la imagen obteniendo de esta manera, una descripción más robusta. Para llevar

[1]Signal and Image Processing Institute: http://sipi.usc.edu/database. Consultado, Agosto 2010

a cabo la optimización por medio de la programación genética proponemos el uso de la medida F como la medida de evaluación para los descriptores locales ya que se obtiene un valor cuantitativo que representa el rendimiento de cada descriptor, el cual, nunca antes habia sido utilizado para este tipo de evaluaciones. Para este trabajo, utilizamos distintas base de datos; una para llevar a cabo la evaluación de los descriptores locales y otra para la aplicación del reconocimiento de objetos. La primera base de datos esta compuesta por un conjunto de imágenes ampliamente utilizada para este tipo de evaluaciones[2] ya que contiene imágenes con diferentes tipos de transformaciones geométricas y fotométricas llevadas a cabo de una manera muy precisa y en donde se incluye la homografía correspondiente a cada una de ellas. La segunda base de datos es utilizada para el reconocimiento de objetos en interiores y exteriores. Esta base de datos cuenta con imágenes para dos diferentes experimentaciones, una que se llevó a cabo utilizando objetos mexicanos y señales de tránsito y la otra, escenarios turísticos y juguetes variados. En este caso, todas las imágenes fueron adquiridas con una cámara digital SONY Cyber-shot 12.1MP DSC-W230 en la ciudad de Ensenada, Baja California, México.

La idea de usar las técnicas evolutivas en estas dos problemáticas es que estas herramientas además de ser ampliamente utilizadas en los últimos años en diferentes áreas de la ciencia, es que nos permiten explorar nuevos caminos hacia el descubrimiento de resultados novedosos y que en la mayoría de los casos, son mejores a los actualmente propuestos en donde un buen planteamiento del problema es fundamental para tener éxito.

I.1.2 Metas y Objetivos

La meta principal de este trabajo de investigación es proponer un nuevo enfoque basado en la computación evolutiva que nos permita desarrollar una nueva metodología para dos problemas de sumo interés en la visión por computadora como lo es, la segmentación de imágenes y los descriptores locales aplicados al reconocimiento de objetos. La idea es utilizar la evolución para optimizar el número de regiones, en el caso de la segmentación; y diseñar

[2] http://www.robots.oxford.ox.ac.uk/vgg/research/affine. Consultado, Agosto 2010.

nuevos operadores descriptivos para aplicarlos en el reconocimiento de objetos, en el caso de los descriptores locales. La finalidad en ambos problemas es estudiar este mecanismo con el fin de obtener resultados que sean mejores o comparables con los del estado del arte; además de proponer un nuevo enfoque para este tipo de problemas.

Objetivos Específicos

- Proponer un algoritmo basado en el cómputo evolutivo para segmentar imágenes con textura utilizando un algoritmo genético.
- Aplicar computación evolutiva-interactiva en nuestro algoritmo de segmentación para mejorar el proceso de optimización.
- Utilizar descriptores estadísticos en una matriz de co-currencia para analizar la información de la textura de la imagen, con el fin de identificar las regiones que son homogéneas y que cumplen con la propiedad de conectividad.
- Proponer un algoritmo evolutivo para obtener nuevos descriptores locales de manera automática.
- Hacer uso de la programación genética como herramienta de optimización para el diseño de operadores descriptivos utilizando como base el descriptor SIFT.
- Proponer una medida de evaluación cuantitativa para evaluar el rendimiento de los descriptores locales.
- Aplicar los nuevos operadores descriptivos en el reconocimiento de objetos en escenarios al interior y al exterior.

I.2 Resumen de Nuestras Contribuciones

En esta tesis se estudia el problema de segmentación y descriptores locales con un enfoque evolutivo. Por tal motivo, se hacen las siguientes contribuciones mayores:

- Se propone un algoritmo genético para optimizar el número de regiones existentes en una imagen con textura sin tener ninguna información a priori. A dicho algoritmo lo llamamos *EvoSeg* (Evolutionary Segmentation Algorithm).

- Se aplica cómputo evolutivo-interactivo en nuestro algoritmo *EvoSeg* con el fin de mejorar el proceso de optimización reduciendo el tiempo computacional y obteniendo mejores resultados.

- Se proponen 30 operadores descriptivos para el descriptor SIFT a los cuales llamamos RDGP's (Region Descriptor Operators using Genetic Programming).

- Se propone utilizar nuestro mejor operador $RDGP_2$ para evaluar nuestra propuesta. Dicha evaluación se lleva a cabo utilizando tres descriptores del estado del arte donde nuestro descriptor obtiene el mejor rendimiento en cinco pruebas diferentes.

- Se propone un criterio de evaluación cuantitativo para descriptores locales basado en la medida F.

- Nuestro mejor operador $RDGP_2$ es aplicado en el reconocimiento de objetos en escenarios al interior y al exterior obteniendo muy buenos resultados en comparación con el descriptor SIFT.

I.3 Organización de la Tesis

La organización de este documento es dividida en dos grandes capítulos. El primero, explica a detalle el enfoque evolutivo para el problema de segmentación de imágenes. Bajo este enfoque, se utiliza un algoritmo genético para optimizar el número de regiones de la imagen donde al mismo tiempo se va llevando a cabo el proceso de segmentación de la imagen. En este caso, cada individuo de la población representa una posible solución la cual es representada con una imagen segmentada. En este capítulo se plantea el problema de segmentación sin considerar ningún tipo de información a priori para ayudar al algoritmo a identificar las

regiones de la imagen. Primero, en la Sección II.2 se explica cómo analizar la textura de una imagen utilizando descriptores estadísticos en una matriz de co-ocurrencia. Más adelante, en la Sección II.3.1 se detalla el algoritmo propuesto para segmentar imágnes con textura utilizando un algoritmo genético, al que llamamos *EvoSeg* (Evolutionary Segmentation Algorithm). Finalmente, en la Sección II.3.4 incluimos la computación evolutiva-interactiva en nuestro algoritmo *EvoSeg* con el fin de mejorar el proceso de optimización donde una persona experta participa en la etapa de evaluación.

En el segundo Capítulo se plantea el problema de sintetizar operadores descriptivos para el descriptor SIFT utilizando la programación genética como la herramienta de optimización. Primero, en la Sección III.2 se explica el problema dentro del contexto del reconocimiento de objetos usando características locales ya que es la tarea de alto nivel que se eligió para probar nuestros resultados. En la Sección III.3 se hace una reseña de los descriptores del estado del arte donde se explica algunas de las propiedades y funcionalidades más importantes que cada uno tiene. En la Sección III.4 se explica de manera general algunos criterios de evaluación que han sido utilizados para comparar descriptores locales. Posteriormente, en la Sección III.5 se detalla el algoritmo propuesto utilizando la programación genética para automatizar operadores descriptivos para el descriptor SIFT en donde los resultados experimentales se presentan en la Sección III.6. Finalmente, nuestros resultados son probados en el reconocimiento de objetos utilizando escenarios al interior y al exterior; éstos resultados son mostrados en la Sección III.6.3.

Capítulo II

Segmentación de Imágenes Evolutiva usando Descriptores Estadísticos

En este capítulo se propone un algoritmo evolutivo para la segmentación de imágenes con textura. Este algoritmo utiliza un algoritmo genético canónico para optimizar el número de regiones existentes en la imagen. Para ello, el algoritmo analiza el contenido de la imagen usando descriptores estadísticos a través de una matriz de co-ocurrencia sin requerir información a priori para llevar a cabo la tarea. Además, se propone incorporar interacción a nuestro algoritmo de segmentación con el fin de mejorar el proceso de optimización.

II.1 Introducción

En visión por computadora, el complejo proceso cognitivo para identificar colores, formas, texturas y agruparlos automáticamente en objetos separados dentro de una misma escena, es llamado ***segmentación de imágenes***, lo cual continúa siendo un tema de investigación abierto a pesar de los años de estudio y de las diversas contribuciones que se han hecho en el área. Sin embargo, a pesar de ello, aún nos seguimos preguntando, *cómo segmentar eficientemente objetos similares dentro de una misma escena en el menor tiempo posible.*

La segmentación de imágenes se define como un proceso en el cual una imagen de entrada es particionada en regiones que son homogéneas de acuerdo a un grupo de características que tengan en común, por ejemplo, la información de su textura. Formalmente, la segmentación de imágenes podría ser definida como sigue:

Definición. La segmentación de I es una partición P de I en un conjunto de M regiones

R_m, $m = 1, 2, ..., M$, tal que:

$$\begin{aligned} &1)\ \bigcup_{m=1}^{M} R_m = I \quad \text{con} \quad R_m \bigcap R_n = \emptyset, \quad m \neq n \\ &2)\ H(R_m) = true \quad \forall\ m \\ &3)\ H(R_m \bigcup R_n) = false \quad \forall\ R_m \ \text{y}\ R_n \ \text{adyacentes} \end{aligned} \tag{1}$$

donde I es la imagen y H es el predicado de homogeneidad.

De esta manera, cada región en una imagen segmentada necesita satisfacer simultáneamente las propiedades de homogeneidad y conectividad Bhandarkar y Zhang (1999). Una región es homogénea si todos los pixeles satisfacen un predicado de homogeneidad definido sobre uno o mas atributos del pixel, tales como, intensidad, textura y color. Además, una región se dice que esta conectada si existe una ruta que conecta a cualquiera de dos pixeles que pertenecen a la misma región. La bibliografía que aborda técnicas de segmentación es muy extensa, por eso me limito a dar la información más relevante del tema citando los siguientes trabajos Freixenet *et al.* (2002); Haralick y Shapiro (1985); Pal y Pal (1993). De manera general, podemos decir que los métodos de segmentación pueden ser clasificados en aquellos que son basados en histogramas (Lim y Lee (1990)), basados en grafos (Pavan y Pelillo (2003); Duarte *et al.* (2006); Shi y Malik (2000)) o basados en regiones (Deng y Manjunath (2001)), solo por mencionar algunas categorías generales. La primer categoría es muy sencilla ya que se calcula un histograma de la imagen y de esta forma, las regiones son localizadas a través de sus picos y valles. Estas técnicas de umbralización pueden obtener buenos resultados de segmentación donde se incluyen dos diferentes tipos de regiones porque los histogramas no toman en cuentan la relación espacial de la imagen como los métodos de primer orden que veremos en la Sección II.2. Los métodos basados en grafos tienen como objetivo extraer la información global de la imagen definiendo el problema de segmentación en el contexto del problema de partición de grafos. El esquema basado en regiones utiliza un criterio de homogeneidad donde los pixeles adyacentes pertenecen a la misma región si éstos tienen características similares, tales como:

nivel de gris, color o valores de textura. En este enfoque el método de dividir y fusionar (*split and merge* en inglés) es usado ampliamente; éste método usa una técnica de crecimiento de regiones donde un conjunto de semillas iniciales dispersas sobre la imagen son tomadas como información de entrada, con el fin de ir agrupando las regiones similares y obtener con ello, la segmentación de la imagen. Sin embargo, el problema de seleccionar automáticamente el número y posición de cada semilla inicial no es una tarea trivial debido a que no se tiene ningún tipo de información a priori de cuántas regiones existen dentro de la imagen ni tampoco dónde se localizan. Para ello, pudiera ser útil emplear técnicas de optimización para definir el número y posición de las regiones en la imagen que se va a segmentar. Por esta razón, propusimos usar un algoritmo genético para segmentar imágenes con textura usando descriptores estadísticos; este algoritmo es presentado en la Sección II.3.1. Posteriormente, decidimos incorporar evolución interactiva a nuestro algoritmo de segmentación con el fin de mejorar el proceso de optimización, ver Sección II.3.4. La idea de incorporar evolución en una aplicación como la segmentación de imágenes se sustenta en el hecho que los algoritmos evolutivos son actualmente una poderosa técnica de optimización ampliamente utilizados en aplicaciones del área del procesamiento de imágenes y visión por computadora, debido a la robustez del enfoque, Olague *et al.* (2006). Por ejemplo, en la comunidad de visión por computadora evolutiva existen varios trabajos relacionados con segmentación de imágenes cuyas aportaciones han favorecido el enriquecimiento del área (Bhandarkar y Zhang (1999); Cagnoni *et al.* (1999); Bhanu *et al.* (1995); Yoshimura y Oe (1999); Perez *et al.* (2009)) dando origen a nuevas ideas para la resolución de este difícil problema de visión por computadora.

Esta Sección presenta un enfoque nuevo para la segmentación de imágenes con textura donde hacemos uso del Cómputo Evolutivo para su desarrollo. Proponemos el algoritmo *EvoSeg*, el cual usa conocimiento derivado del análisis de textura para identificar el número de regiones homógeneas existentes en la escena sin contar con información a priori. *EvoSeg* usa descriptores estadísticos derivados de una matriz de co-ocurrencia en niveles de gris (GLMC, Gray Level Co-ocurrence Matrix) y optimiza la medida de aptitud, basada en el criterio

de mínima varianza, utilizando un algoritmo genético jerárquico. Además, en la Sección II.3.4 presentamos una extensión del algoritmo *EvoSeg* el cual complementa la información estadística de la textura con la interacción humana en el proceso de optimización. Esta evolución interactiva ayuda a mejorar los resultados permitiendo al algoritmo adaptarse mejor usando información externa de un experto en el área.

II.2 Análisis de Textura

La textura de una imagen es considerada como una característica de bajo nivel que aporta información relevante acerca del contenido de la imagen. Dicha información puede ser utilizada en aplicaciones de alto nivel como la segmentación, clasificación, síntesis y recuperación de imágenes, por mencionar solo algunos ejemplos. Por ello, su estudio es de gran interés en la visión por computadora y otras disciplinas. A pesar de la importancia que sigue teniendo el análisis de textura dentro de la comunidad de visión, no existe hasta el momento una definición precisa y general del término de "textura" ya que su definición permanece ambigüa en la literatura. La principal razón es que las texturas naturales generalmente muestran propiedades contradictorias, es decir, regularidad *versus* aleatoriedad y uniformidad *versus* distorción; lo cual puede ser difícilmente descrito de una manera unificada. Para ello, muchos investigadores han tratado de definir textura desde una cierta perspectiva de acuerdo a su naturaleza. Por ejemplo, Hawkins (1969) propuso la siguiente definición de textura: *"La noción de textura parece depender de tres ingredientes: (i) algún 'orden' local que es repetido sobre una región la cual es mas grande en comparación con el tamaño del orden, (ii) el orden consiste en una estructura no aleatoria de las partes elementales, y (iii) las partes son entidades uniformemente rugosas teniendo aproximadamente las mismas dimensiones en cualquier parte dentro de la región texturizada".*

Realmente, la dificultad de este problema puede estar relacionado al hecho que las texturas del mundo real son complejas para modelarlas y analizarlas. Sin embargo, los investigadores

estan de acuerdo que las imágenes con textura exhiben patrones elementales que son repetidos periódicamente dentro de una región dada. Algunos ejemplos de textura los podemos encontrar fácilmente a nuestro alrededor, por ejemplo: en las puntadas de un suéter, las piedras de un camino empedrado, los granos de arena en la playa, etc. En el caso del suéter, los patrones son periódicos mientras tanto, en las piedras y en los granos son estadísticos debido a la manera en cómo la textura está presente. Además, podemos decir que las características de las texturas pueden ser rugosas, suaves, finas, gruesas, regulares, irregulares, homogéneas, granulares, etc.

Por otro lado, para caracterizar las texturas existen básicamente tres formas de procesar la imagen y extraer su información como: los descriptores de frecuencia, descriptores estructurales y descriptores probabilísticos Gonzalez y Woods (2002); Cocquerez y Philipp (1997). Históricamente, el método más común para describir la información de la textura es el enfoque estadístico, el cual incluye los métodos estadísticos de primer orden, segundo orden, y órdenes más altos. Estos métodos analizan la distribución de propiedades específicas de la imagen usando el valor de sus pixeles. Particularmente, nosotros estamos interesados en el método de segundo orden porque toma en cuenta la distribución de las intensidades de los pixeles y además su posición espacial sin sacrificar mucho tiempo en los cálculos, siendo el caso de los métodos de órdenes más altos. De esta manera, el método de segundo orden representa la densidad de probabilidad conjunta o f.p.d conjunta de los valores de intensidad entre dos pixeles que se encuentran separados por un vector dado $\mathbf{V}$ donde esta información es codificada usando la matriz de co-ocurrencia en niveles de gris, GLCM (Gray Level Co-ocurrence Matrix) denotada por $M_{i,j}$ y que posteriormente es usada para obtener los descriptores estadísticos.

II.2.1 Matriz de Co-ocurrencia en Niveles de Gris, GLCM

Formalmente, la GLCM $M_{i,j}$ (π) define una función de densidad de la probabilidad conjunta $f(i,j|\mathbf{V},\boldsymbol{\pi})$ donde i y j son los niveles de gris de dos pixeles separados por un vector $\mathbf{V}$, y π = $\mathbf{V}$, R es el conjunto de parámetros para $M_{i,j}$ (π). La GLCM identifica qué tan seguido

los pixeles que definen un vector $\mathbf{V}$(d,θ), y que difieren por una cierta cantidad de valor de intensidad $\Delta = i - j$, aparecen en una región R de una imagen $I_{L\times L}$; donde $\mathbf{V}$ define la distancia d y la orientación θ entre los dos pixeles. Al calcular la GLCM puede tomarse en cuenta la orientación espacial de $\mathbf{V}$, lo cual indicaría la dirección a seguir para contabilizar los niveles de gris de los pixeles dentro de la matriz. Las direcciones que se utilizan comúnmente son: $0°, 45°, 90°$ y $135°$; en el caso que no se desee usar ninguna orientación, entonces $0°$ es la que se toma por defecto. La Figura 1 ilustra un ejemplo de GLCM, donde la distancia d es definida como l y la dirección θ es definida como $0°$.

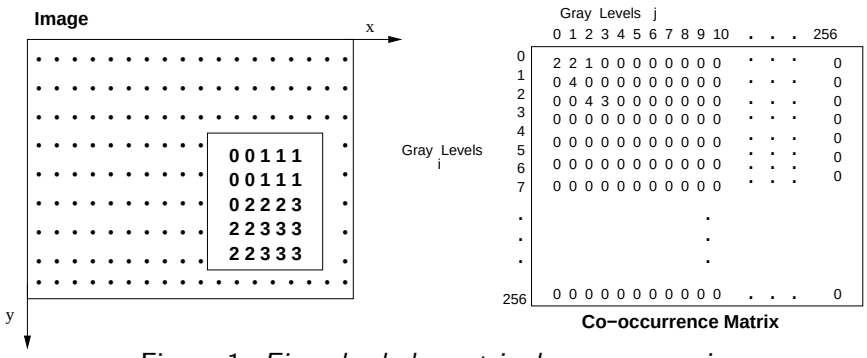


Figura 1. *Ejemplo de la matriz de co-ocurrencia.*

Por otro lado, la GLCM presenta un problema cuando el número de diferentes niveles de gris en una región R incrementa, tornándose difícil manejar la información o usarla directamente, debido a las dimensiones que tendría la matriz. Afortunadamente, la información codificada en la GLCM puede ser expresada por un conjunto de descriptores estadísticos que son relevantes para caracterizar la textura de una imagen. Estos descriptores son extraídos de $M_{i,j}$ dando como resultado diferentes tipos de información acerca de su textura. Algunos de los descriptores estadísticos más conocidos son los siguientes: Entropía, Homogeneidad, Homogeneidad Local, Contraste, Momentos, Momentos Inversos, Uniformidad, Máxima Probabilidad, Correlación y Directividad (Parker (1996); Haralick *et al.* (1973)). Tales descriptores pueden ser definidos en el dominio espacial extrayéndolos directamente de la GLCM, o bien, pueden ser extraídos en otros dominios de frecuencia.

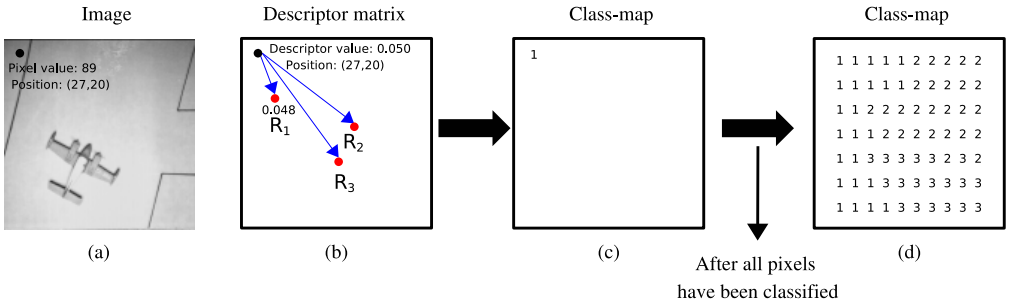


Figura 2. *Ejemplo del uso del descriptor en el proceso de segmentación. Al final se forma un mapa de clase que contiene el número de región a la que pertenece cada pixel.*

II.2.2 Descriptores de Textura Estadísticos

Los descriptores de textura son calculados directamente de la GLCM reduciendo la dimensionalidad de la información que es extraída de la imagen I de tamaño $L \times L$ pixeles. Cada descriptor que se calcula desde la GLMC, mapea los valores de intensidad de cada pixel a una nueva dimensión, ver Figura 2. Los descriptores de textura que nosotros vamos a usar para segmentar imágenes con textura son presentados a continuación donde además, se muestra un ejemplo de la imagen que correspondería a cada uno de ellos en la Figura 3:

- ***Correlación***. La correlación es una mediada de dependencia linear de los nievles de gris entre los pixeles y posiciones específicas relacionadas con cada uno de ellos. Los pixeles más cercanos tienen más correlación que los pixeles más lejanos.

$$S = \frac{1}{Nc.\sigma_x.\sigma_y} \mid \sum_{i}^{L-1} \sum_{j}^{L-1} (i - m_x)(j - m_y) M(i,j) \mid \tag{2}$$

  donde, $m_x = \frac{1}{Nc} \sum_i \sum_j iM(i,j)$

  $m_y = \frac{1}{Nc} \sum_i \sum_j jM(i,j)$

  $\sigma_x^2 = \frac{1}{Nc} \sum_i \sum_j (i - m_x)^2 M(i,j)$

  $\sigma_y^2 = \frac{1}{Nc} \sum_i \sum_j (j - m_y)^2 M(i,j)$

  N_c es el número de ocurrencias en M

- ***Entropía***. Es un término común en termondinámica o mecánica estadística. La en-

tropía es una medida del nivel de desorden en un sistema. Las imágenes que contienen escenas altamente homogéneas tienen una entropía asociada muy baja, mientras que las escenas inhomogéneas tienen una medida de entropía muy alta. La medida de entropía se obtiene de la siguiente manera:

$$H = 1 - \frac{1}{Nc.ln(Nc)} \sum_{i}^{L-1} \sum_{j}^{L-1} M(i,j).ln(M(i,j)).\delta \tag{3}$$

donde $\delta = 1$ si $M(i,j) \neq 0$ y 0 de lo contrario.

- ***Homogeneidad Local***. Esta medida provee la similaridad local de los datos de la imagen usando un factor de peso el cual da pequeños valores para imágenes no-homogéneas cuando $i \neq j$.

$$G = \frac{1}{Nc} \sum_{i}^{L-1} \sum_{j}^{L-1} \frac{M(i,j)}{1 + (i-j)^2} \tag{4}$$

- ***Contraste***. Es lo opuesto a la homogeneidad. El contraste es una medida de la diferencia entre los valores de intensidad de los pixeles vecinos.

$$C = \frac{1}{Nc(L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|i-j|=k} M(i,j) \tag{5}$$

- ***Directividad***. Esta medida provee un valor más grande cuando dos pixeles tienen el mismo nivel de gris.

$$D = \frac{1}{Nc} \sum_{i}^{L-1} M(i,i) \tag{6}$$

- ***Momentos***. Es una medida de la homogeneidad de una imagen. Una escena homogenea contendrá únicamente pocos valores de gris, dando como resultado que la GLMC con-

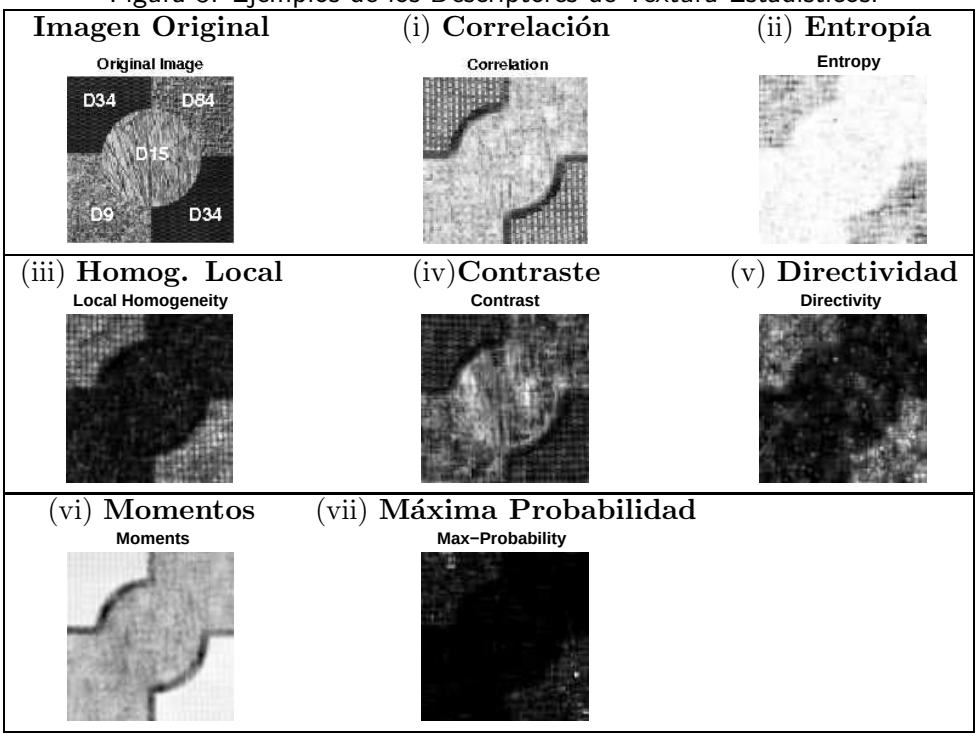
tenga pocos elementos con valores altos. Este descriptor incrementa cuando la mayoría de los valores de $M_{i,j}$ no se encuentran en la diagonal. El segundo momento ($k = 1$) es el que se usa en esta figura.

$$Mom_k = \sum_{i}^{L-1} \sum_{j}^{L-1} (i-j)^k M(i,j) \tag{7}$$

- ***Máxima Probabilidad***. Considerando a la GLCM como una aproximación de la densidad de probabilidad conjunta entre pixeles, este operador extrae la diferencia mas probable entre los pixeles.

$$max(M(i,j)) \tag{8}$$

Figura 3. Ejemplos de los Descriptores de Textura Estadísticos.



II.3 Enfoque Evolutivo para la Segmentación de Imágenes

En los últimos años, los algoritmos evolutivos han tenido gran aceptación en diferentes áreas de la ciencia como algoritmos robustos de búsqueda y optimización. Estos algoritmos han demostrado ser un enfoque que puede adaptarse perfectamente a problemas de optimización donde la solución es difícil de obtenerla de manera directa.

Los algoritmos de evolución utilizan el principio de la evolución natural, donde los más aptos sobreviven y prosperan. Desde hace más de 100 años Charles Darwin, sin conocer nada sobre genética, realizó una extensa investigación sobre la evolución natural e identificó tres principios: (1) *el ciclo reproductivo*, (2) *la selección natural* y (3) *la diversidad por variación*, (Darwin (1859)). El ciclo reproductivo explica la naturaleza iterativa de la evolución mediante el proceso de nacimiento y muerte. A través del proceso de reproducción, los individuos crean hijos que los reemplazan siendo los descendientes que forman la siguiente generación de la especie. La selección natural es el proceso mediante el cual los individuos se adaptan al entorno que los rodea. En este proceso, los mejores individuos tienen mayores probabilidades de sobrevivir, y son ellos los que consiguen reproducirse para generar nuevas poblaciones. De esta manera, en cada generación aparecen nuevos individuos que convierten a los descendientes en mejores o peores individuos que sus padres, en la capacidad de adaptarse a su entorno, consiguiendo con ello, la diversidad por variación.

Actualmente, se han desarrollado un gran número de técnicas evolutivas consideradas como algoritmos evolutivos, entre las que destacan los algoritmos genéticos (GA, Genetic Algorithm), Programación Genética (GP, Genetic Programming), estrategias de evolución (ES), y una gran cantidad de variantes. En lo particular, nosotros vamos a enfocarnos en los algoritmos genéticos ya que es la herramienta que utilizaremos para segmentar imágenes con textura. Estos algoritmos fueron propuestos por John Holland y complementados por Goldberg. Esta técnica evolutiva se basa en la genética de los organismos vivos donde a la

estructura fundamental que contiene los rasgos de un individuo se le conoce como cromosoma. En un algoritmo genético cada solución representa un cromosoma, y un conjunto de cromosomas forman la población de soluciones. Además, un algoritmo genético tiene las siguientes características que comparte con las demás técnicas evolutivas:

- *Una población de individuos*, los cuales representan la solución potencial al problema que se está abordando.
- *Un mecanismo de reproducción* con en el que se producen cambios en las características de los diferentes individuos.
- *Un criterio de evaluación*, por medio del cual, las posibles soluciones son comparadas de modo objetivo o subjetivo para que el algoritmo converga a la mejor solución.
- *Un mecanismo de selección*, mediante el cual es posible obtener las soluciones que generarán la siguiente población de individuos y de esta manera, a iniciar un nuevo ciclo evolutivo.

Las cuatro características son importantes en el proceso de evolución; sin embargo, cabe mencionar que el criterio de evaluación es fundamental para que el algoritmo funcione adecuadamente. La razón es que este criterio conocido tambien como función de aptitud, califica el lugar que ocupa cada individuo en el espacio de búsqueda donde aquellos que obtienen los valores más altos tienen mayores probabilidades de ser seleccionados para el proceso de reproducción. La reproducción se lleva a cabo mediante los llamados operadores evolutivos (mutación, cruzamiento). En este proceso se combina la información de uno o varios individuos para dar lugar a nuevos individuos. Este proceso se repite iterativamente, generación tras generación. Para una mejor comprensión de los algoritmos genéticos, ver Goldberg (1989).

A continuación, se presenta nuestro algoritmo de evolución para segmentar imágenes con textura, llamado *EvoSeg* (en inglés, Evolutionary Segmentation) el cual hace uso de un algoritmo genético para optimizar el número de regiones de la imagen. En la Sección II.3.4,

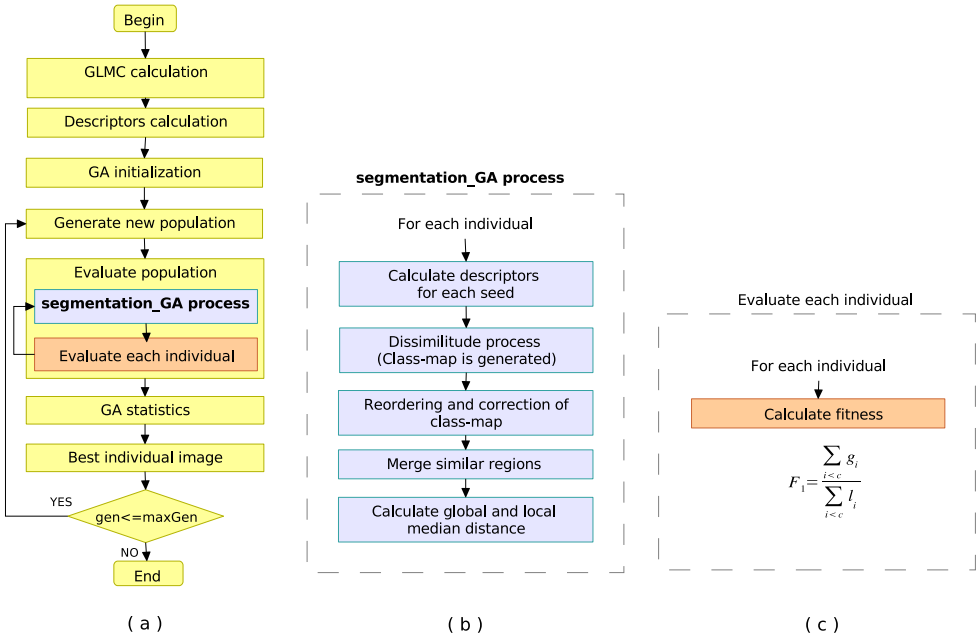


Figura 4. Diagrama de flujo del algoritmo *EvoSeg*. (a) Diagrama Completo de *EvoSeg*. (b) Pasos generales requeridos en el proceso de segmentación. (c) Esquema de la función de aptitud utilizada por el GA.

presentamos una versión mejorada del algoritmo *EvoSeg* donde incorporamos la interacción humana en el proceso de evolución con el fin de mejorar el proceso de búsqueda, llamando a este algoritmo, *I-EvoSeg* (en inglés, Interactive Evolutionary Segmentation).

II.3.1 Evo-Seg, Algoritmo de Segmentación Evolutiva

El algoritmo de segmentación evolutiva, *EvoSeg* intenta identificar cuántas regiones homogéneas existen en la escena sin utilizar información a priori. El algoritmo usa una función objetivo que evalúa las posibles segmentaciones de una imagen basado en la homogeneidad de la región y su distintividad. El algoritmo *EvoSeg* lleva a cabo dos procesos generales, un proceso de análisis de textura estadística y un proceso de segmentación integrado en un algoritmo genético. El diagrama de flujo del algoritmo es presentado en la Figura 4.

Proceso del análisis de textura estadístico

El proceso del análisis de textura estadístico es usado como una forma de obtener una representación compacta de los datos de la textura de la imagen a través de la GLCM y los descriptores estadísticos. De esta manera, para calcular la GLCM, nosotros probamos experimentalmente diferentes valores en los parámetros que requiere la GLCM como: el tamaño de la ventana, dirección y distancia. Los resultados mostraron solamente diferencias considerables cuando se cambia el tamaño de la ventana produciendo mayor difuminado en las imágenes de los descriptores. Finalmente, los valores que utilizamos en la experimentación fueron los siguientes: el tamaño de la ventana fue de 7×7 pixeles, la dirección fue de $0°$, y la distancia fue de 1 pixel. Por otro lado, la información de la GLCM es usada para calcular los descriptores estadísticos de cada pixel de la imagen obteniendo una matriz por cada descriptor. Para ello, se implementaron diferentes descriptores de textura, los cuales fueron usados de manera individual y como una combinación de ellos. Sin embargo, el descriptor con el cual se obtuvieron mejores resultados fue con el momento de segundo orden, el cual es definido como sigue:

$$Mom_k = \sum_i \sum_j (i-j)^2 . M(i,j) \tag{9}$$

Proceso de segmentación integrado en un algoritmo genético

El proceso de segmentación es llevado a cabo utilizando un enfoque basado en regiones donde la similaridad local y global es utilizada para agrupar la información que pertenece a cada región de la imagen. La característica principal de este enfoque es que las regiones se expanden y se fusionan usando la información de la textura, en nuestro caso, los descriptores estadísticos. La inicialización del proceso de segmentación consiste en una selección aleatoria de semillas iniciales las cuales serán el punto de partida para crear las regiones de la imagen. Cada una de estas semillas serán consideradas inicialmente como los centroides de cada región tomando en cuenta que durante el proceso de crecimiento y fusionamiento de regiones cambiarán de

posición dentro de cada región debido a la información agregada. El proceso de crecimiento de regiones consiste en expandir cada región inicial usando el criterio de similaridad local basado en la información de textura contenida en las matrices de los descriptores estadísticos. Posteriormente, el proceso de fusionamiento de regiones se lleva a cabo usando el criterio de similaridad global, el cual se basa en la posición espacial de las regiones dentro del plano de la imagen. Para una mejor comprensión del funcionamiento de nuestro algoritmo *EvoSeg*, a continuación se detalla cada una de sus etapas:

1. **Codificación del genoma y los operadores genéticos**. El cromosoma es representado por su posición de los centroides en el plano de la imagen, ver Figura 5. Cada arreglo del cromosoma contiene M elementos que representan los posibles centroides de las regiones, los cuales son denotados por C_i donde $i = 1...M$. Este arreglo es actualizado dentro del proceso de segmentación indicando las mejores posiciones de los centroides. Los parámetros y operadores del algoritmo permiten una exploración eficiente del espacio de búsqueda; en el caso del método de selección usamos la selección basada en torneo. El operador de cruzamiento se realiza con una probabilidad del 90% mientras que el operador de mutación es del 10%.

2. **Proceso de segmentación**. El proceso de segmentación inicia a partir de los centroides de las regiones con el fin de clasificar cada uno de los pixeles de la imagen dentro de las posibles regiones. El método de clasificación es basado en la técnica de crecimiento y fusionamiento de regiones, la cual usa la información espacial y de la textura para agrupar los pixeles de la imagen en las regiones. Supongamos que tenemos un conjunto $P = \{x_1, x_2, ..., x_N\}$ de N pixeles donde N es la cantidad total de pixeles. Sea M los subconjuntos disjuntos $R_1, R_2, ...R_M$ donde cada subconjunto representa una región con un centroide asociado C_i. Además, tenemos un conjunto $D_T = \{d_{T1}, d_{T2}, ..., d_{TN}\}$ de N valores de los descriptores y sea T el número de descriptores usados en el algoritmo.

Dado esto, el proceso de segmentación es como sigue:

(a) El valor del descriptor s_{C_i} es calculado para cada centroide inicial dado por la media del descriptor d_{pj} dentro de un vecindario 5×5 alrededor del centroide.

(b) El mapa inicial de las clases es formado como una plantilla de números enteros que representan las regiones donde cada pixel pertenecerá, ver Figura 2. Este mapa de clases es creado usando el método de clasificación del vecino más cercano. Dos medidas[1] de distancias son definidas en el paso de similaridad, Δ y δ. La distancia en el espacio del descriptor es determinada por Δ mientras que δ es la distancia dentro del plano de la imagen. De esta manera, un pixel x_j es asignado a una región R_i, si las siguientes dos condiciones son verdaderas: 1) $\Delta(d_{pj}, s_{C_i}) < \Delta(d_{pj}, s_{C_q})$ $\forall$ q y 2) $\delta(x_j, C_i) < t$, donde t es un umbral y $q = 1...M$ con $q \neq i$.

(c) El mapa de clases es reordenado usando el segundo vecino más cercano con el fin de disminuir los errores de la clasificación.

(d) Dos regiones R_i y R_m son fusionadas si ellas satisfacen un criterio de similaridad basado en la mediana de cada región, mientras se considera las posiciones espaciales dentro del plano de la imagen. Además, los centroides son actualizados durante este proceso cuando un elemento es agregado a una región. La siguiente expresión es usada para actualizar los centroides:

$$centroid(x, y) = (\frac{\sum_{I \epsilon R_i} x_j}{\mid R_i \mid}, \frac{\sum_{I \epsilon R_i} y_j}{\mid R_i \mid}), \tag{10}$$

donde (x, y) son las coordenadas del pixel y $\mid R_i \mid$ es el número de elementos en la región R_i. De esta manera, la información genética del cromosoma es ajustada para facilitar la clasificación de los pixeles.

[1] Todas las medidas de distancias son Euclidianas.

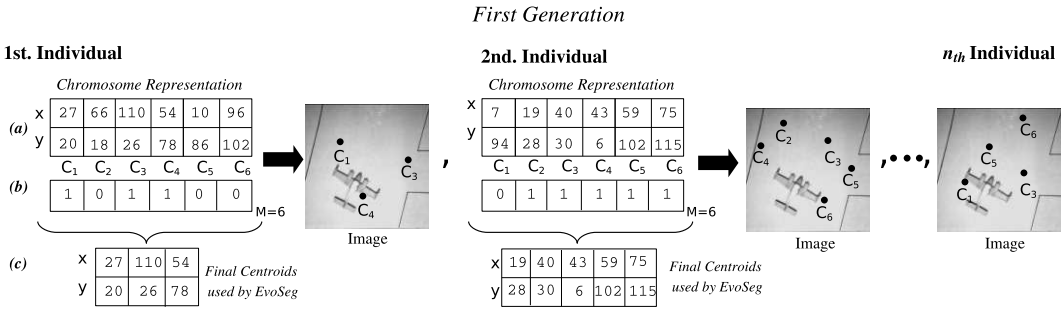


Figura 5. Ejemplo de la representación del cromosoma en la primera generación del algoritmo

3. **Función de Aptitud**. Esta medida de aptitud está basada en la distancia mínima local y global entre regiones. Por un lado, la distancia local está dada por:

$$l = \sqrt{\sum_{i=1}^{c} \sum_{k=1}^{n_i} (D_k - m_i)^2} \tag{11}$$

donde c es el número de regiones, D_k representa la suma de los descriptores d_{pj} de la i^{ma} región, m_i representa la mediana de cada región, m es la mediana total y n_i es el número de elementos en la i^{ma} región. Por otro lado, la distancia global está dada por:

$$g = \sqrt{\sum_{i=1}^{c} (m_i - m)^2} \tag{12}$$

where $m = \frac{1}{N}\sum_{i=1}^{c} (m_i.n_i)$, m es la mediana total y n_i es el número de elementos en la i^{ma} región.

De esta manera, la función de aptitud propuesta quedaría de la siguiente manera:

$$\rho = \frac{\sum\limits_{i=1}^{c} \sqrt{\sum\limits_{i=1}^{c} \sum\limits_{k=1}^{n_i} (D_k - m_i)^2}}{\sum\limits_{i=1}^{c} \sqrt{\sum\limits_{i=1}^{c} (m_i - m)^2}}, \tag{13}$$

Por último, esta función indica que las distancias entre las medianas de las diferentes regiones deben ser maximizadas para mantener la unicidad entre las regiones. Además, las distancias entre los elementos dentro de una región deben ser minimizadas porque

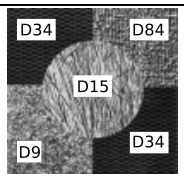
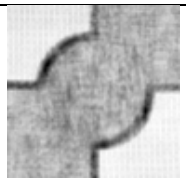
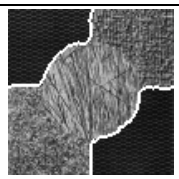
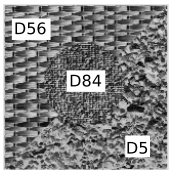
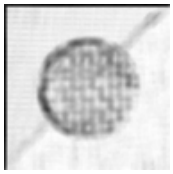
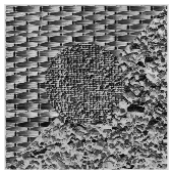
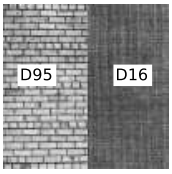
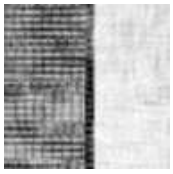
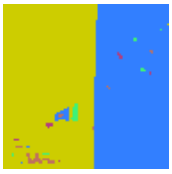
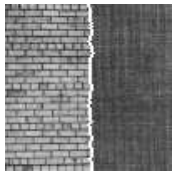
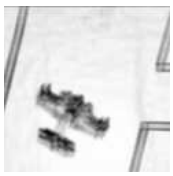
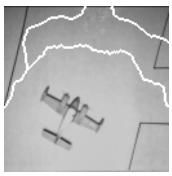
los elemenos más cercanos a un centroide podrían pertenecer a la misma región.

II.3.2 Resultados Experimentales

En esta sección presentamos los resultados experimentales obtenidos con el algoritmo *EvoSeg* y el algoritmo *JSEG*. El algoritmo *JSEG* es un algoritmo de segmentación del estado del arte que considera información de la textura y el color en imágenes y videos, ver Deng y Manjunath (2001). Lo que nos motivó a usar este algoritmo de segmentación para su comparación con *EvoSeg* es debido a su calidad y simplicidad, asi como también su versatilidad en tareas de alto nivel.

La Tabla I presenta los resultados experimentales de estos dos algoritmos; se puede apreciar que ambos algoritmos porducen buenas segmentaciones en las imágenes I(a),(c),(d) y producen una falsa clasificación de los pixeles en la imagen I(g). La segmentación del algoritmo *EvoSeg* en la Tabla I(b) es mejor que el algoritmo JSEG, esto es debido al hecho que JSEG es sensible a los cambios de iluminación. En este caso, JSEG no pudo segmentar la imagen porque las texturas no presentan alto constraste. Por otro lado, el *EvoSeg* es robusto cuando no hay mucho constrate en la imagen y en este caso particular, casi logra segmentar por completo la imagen. La Tabla I(e) muestra los resultados para la imagen de *baboon*, la segmentación de esta imagen utilizando el algoritmo JSEG es superior al *EvoSeg* debido al alto contraste de las texturas que tiene la imagen. Para esta imagen, el algoritmo *EvoSeg* presenta problemas en la segmentación porque los contornos de la imagen del descriptor no estan bien definidos. En este caso, sería interesante agregar al algoritmo *EvoSeg* el método de cuantización de color que utiliza el algoritmo JSEG. En el caso de la imagen del aeroplano mostrada en la Tabla I(f), el algoritmo *EvoSeg* realiza la segmentación satisfactoriamente, mientras que el JSEG se pierde en la iluminación del suelo. En general, podemos decir que las imágenes segmentadas por *EvoSeg* son comparables con un algoritmo que es considerado del es-

Tabla I. Resultados de los algoritmos de segmentación EvoSeg y JSEG usando imágenes con texturas diferentes.

	Imagen Original	**Imagen del Descriptor**	**Resultado *EvoSeg***	**Resultado JSEG**
(a)				
(b)				
(c)				
(d)				
(e)				
(f)				
(g)				

tado del arte y que a su vez, éste podría ser más robusto si se agregara al proceso de segmentación, el método de cuantización de color del algoritmo JSEG antes de generar el mapa de clases. Por último, cabe mencionar que las imágenes que fueron utilizadas en nuestros experimentos fueron obtenidas de internet de la base de datos de USC-SIPI [2]. Esta base datos ha sido utilizada en otros trabajos sobre segmentación de texturas como por ejemplo, las imágenes originales mostradas en la Tabla I(a),(b),(c) y (d) son utilizadas en el trabajo de Yoshimura y Oe (1999).

II.3.3 Discusión de Resultados

En esta sección, presentamos nuestro algoritmo *EvoSeg* como un algoritmo de segmentación evolutiva no supervisado. *EvoSeg* identifica buenas segmentaciones de imágenes de un conjunto de múltiples soluciones que ofrece el algoritmo genético. Después de varios experimentos, decidimos usar el descriptor del segundo momento porque fue con el que mejores resultados obtuvimos. Observamos a lo largo de la fase de experimentación que si el descriptor de la imagen está bien definido, es decir, si sus contornos son muy visibles, sería mucho más fácil para nuestro algoritmo, identificar los límites de las regiones. Por otra parte, el algoritmo genético permitió tener una mejor distribución de las regiones durante el proceso de segmentación debido a que optimiza de manera eficiente las semillas iniciales sin ningún conocimiento a priori. Una vez que el algoritmo genético termina el proceso evolutivo, éste produce una variedad de buenas soluciones de las cuales seleccionamos la mejor. Por último, podemos decir que nuestros resultados fueron comparables con aquellos producidos por el algoritmo JSEG.

II.3.4 I-EvoSeg, Algoritmo de Segmentación Evolutiva-Interactiva

La Computación Evolutiva Interactiva (IEC, Interactive Evolutionary Computation) es un término general que ha sido empleado en métodos de computación evolutiva donde se

[2]Signal and Image Processing Institute: http://sipi.usc.edu/database. Consultado, Agosto 2010

usa el juicio de un experto en el proceso de evaluación. La idea de IEC es involucrar una persona experta durante el proceso de evaluación del algoritmo de manera *on-line*. El juicio de la persona es de gran ayuda cuando el criterio de la función de aptitud no puede ser formulado explícitamente, no está bien definido, o en casos donde es necesario escapar de óptimos locales. Recientemente, se han publicado trabajos interesantes donde hacen uso de la IEC para resolver problemas en el área de medicina (Cagnoni *et al.* (1999); Legrand *et al.* (2006)), artes gráficas (Lutton *et al.* (2005); Lutton (2006)), diseño de modas (Hee-Su y Sung-Bae (2000)), procesamiento de imágenes (Takagi (2001)), entre otros. El objetivo principal de la IEC es permitir la participación de una persona experta para que ayude a simplificar y adaptar el comportamiento aleatorio del sistema y éste pueda ser un problema tratable, Lutton (2006). Por tal motivo, decidimos que sería factible incluir interacción en nuestro algoritmo *EvoSeg* con el fin de ayudar al algoritmo genético a mejorar la función de aptitud. La razón de ello, es porque en algunas generaciones había buenos individuos que el algoritmo obviaba debido a que no tenian asignado un valor de aptitud que correspondiera a lo que visualmente se observaba, lo que para un experto significaría una buena segmentación. En ese sentido, era difícil para el algoritmo ya que el problema de segmentación es un problema complejo *perse*, además de que se planteó de tal manera que el algoritmo no contara con ningún tipo de información a priori. Sin embargo, pudimos observar en la Sección II.3.1 que a pesar de ello, se obtuvieron muy buenos resultados. De esta manera, decidimos llamar a éste algoritmo de segmentación evolutiva con interacción humana, *I-EvoSeg* (en inglés, Interactive EvoSeg).

I-EvoSeg es un algoritmo para la segmentación de imágenes con textura que usa descriptores estadísticos al igual que *EvoSeg* pero con un enfoque evolutivo-interactivo. Nuestro algoritmo extrae las características de la textura de una imagen de entrada con el fin de identificar cuántas regiones homogéneas existen y de esta manera poder segmentar la imagen. Para llevar a cabo la segmentación usando la información de los

descriptores estadísticos usamos un algoritmo genético (GA, Genetic Algorithm) combinándolo con el criterio de evaluación de un especialista en segmentación. La idea de *I-EvoSeg* es considerar la experiencia del usuario en el proceso de optimización del GA, permitiendo que la medida de aptitud de un individuo pueda ser modificada por el experto de forma *on-line*.

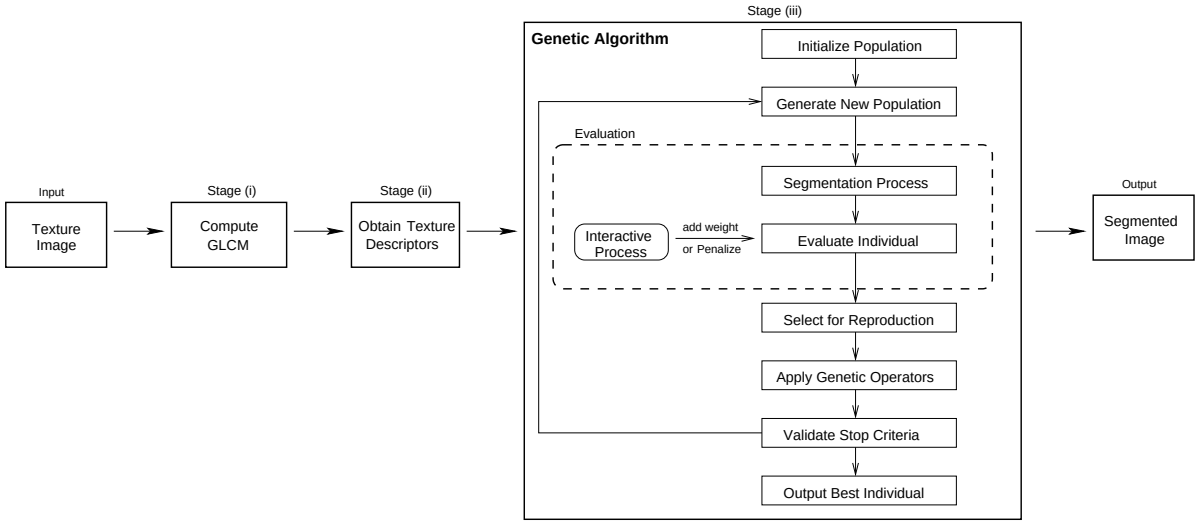


Figura 6. Esquema general de nuestro algoritmo *I-EvoSeg* donde el proceso de interacción es incluido en la etapa de evaluación.

La Figura 6 muestra un esquema general de nuestro algoritmo el cual tiene tres etapas principales:

(a) El cálculo de la GLCM.

(b) El cálculo de los descriptores estadísticos de textura.

(c) El proceso de segmentación que es incluido en el GA interactivo.

De esta manera, *I-EvoSeg* extrae la GLCM de la imagen de entrada para después obtener los descriptores de textura. El GA interactivo, aleatoriamente selecciona las semillas iniciales para cada individuo de la primera generación, subsecuentemente, un proceso de segmentación basado en regiones se lleva a cabo. Después, el usuario "*interactúa*" con el sistema a través de una GUI, ver la Figura 7. Por un lado, el usuario tiene el privilegio de incrementar el valor de la aptitud de un individuo si existe una "*buena*"

segmentación dentro de la población actual de acuerdo a su juicio. Por otro lado, el usuario también puede penalizar a la población si él considera que no existe un individuo que pueda ser "*útil*" en las próximas generaciones. La idea de la interacción es adaptar el comportamiento aleatorio del sistema, basándonos en la experiencia del especialista. De esta manera, el algoritmo es capaz de adaptarse y mejorar el proceso evolutivo.

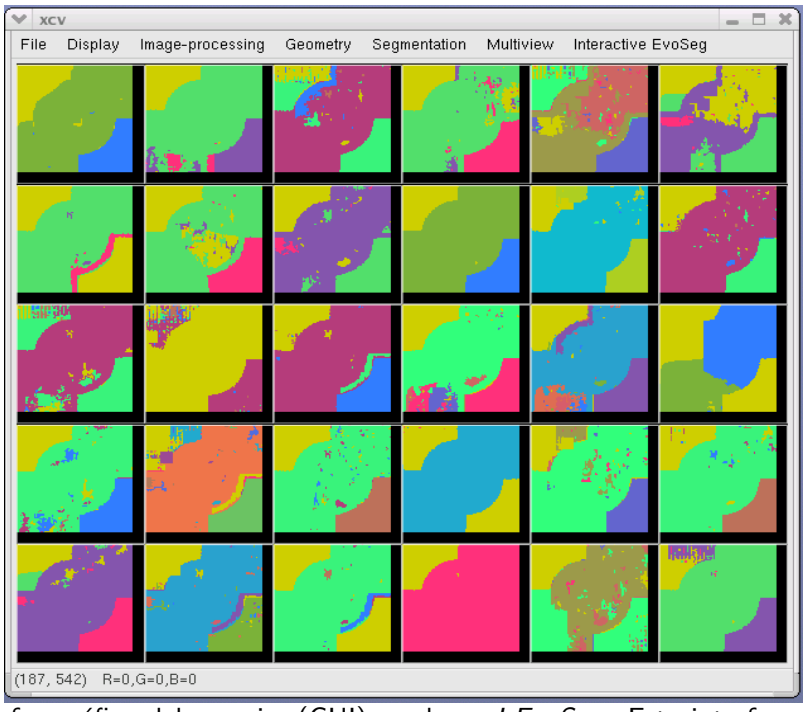


Figura 7. Interfaz gráfica del usuario, (GUI) usada en *I-EvoSeg*. Esta interface gráfica muestra 30 individuos como posibles "*buenas*" segmentaciones de 200 individuos en total.

Cálculo de la Matriz de Co-ocurrencia usando Descriptores Estadísticos

Los descriptores de textura estadísticos son usados como una forma de obtener datos compactos que sean a su vez también representativos ya que un gran número de niveles de gris G implica almacenar una gran de datos temporales; por ejemplo, tendríamos una matriz muy grande de $G \times G$. En ese sentido, La GLCM es muy sensitiva al tamaño de los valores de la textura debido a la gran dimensionalidad que tendría la matriz. Por esta razón, nosotros probamos experimentalmente diferentes tamaños de ventana, direcciones y distancias durante el cálculo de la GLCM usando 256 niveles de gris. El

uso de un número pequeño de niveles de gris es equivalente a ver la imagen a un nivel de escala; mientras que el tener más niveles de gris implica tener más detalles de la imagen pero esto requeriría un esfuerzo computacional extra. Por lo tanto, los descriptores de textura son calculados usando la GLCM obteniendo de esta manera una matriz de datos por cada descriptor, la cual tendría un tamanño igual que la imagen original. Además, en el proceso de segmentación de la imagen, es posible utilizar la información de un descriptor de textura, o bien, la combinación de varios de ellos lo cual implicaría tener mas información acerca de la textura de la imagen. En la fase de experimentación nosotros combinamos la información de algunos descriptores estadísticos para obtener la GLCM, ver Sección II.3.5.

Proceso de Segmentación integrado en un GA Interactivo

El proceso de segmentación es llevado a cabo de la misma manera que el algoritmo *EvoSeg*, ver Sección II.3.1, la diferencia radica principalmente en la evaluación de los individuos, por esa razón me limito a dar la información solamente de la función de aptitud.

4. **Evaluación de la Aptitud** La función de aptitud usada en *I-EvoSeg* se compone de dos partes: (1) *"aptitud interna"* y (2) *"aptitud externa"*.

   - **Aptitud Interna**. Ésta depende solamente de las distancias mínimas locales y globales entre las regiones. La estimación de estas distancias se hace de la siguiente manera:

$$\rho = \frac{\sum_{i=1}^{c} \sqrt{\sum_{i=1}^{c} \sum_{k=1}^{n_i} (D_k - m_i)^2}}{\sum_{i=1}^{c} \sqrt{\sum_{i=1}^{c} (m_i - m)^2}}, \tag{14}$$

   donde c es el número de regiones, D_k representa la suma de los descriptores d_{pj} de

la i^{ma} región, m_i representa la mediana de cada región, m es la mediana total y n_i es el número de elementos en la i^{ma} región.

La función de aptitud interna indica que las distancias entre las medianas de diferentes regiones deberán ser maximizadas para mantener la unicidad entre regiones. Además, las distancias entre los elementos dentro de una región dada deberá ser minimizada debido a que los elementos mas cercanos al centroide podrían pertenecer a la misma región.

- **Aptitud Externa**, ***"Interacción del usuario"***. Ésta depende de la decisión del usuario que generalmente es un especialista en el área. La tarea del usuario es identificar la calidad de la segmentación de la población actual. De esta manera, él podrá decidir si recompensa a un individuo en particular o penaliza a toda la población. Para ello, *I-EvoSeg* evoluciona una población presentando a los individuos en una GUI como imágenes ya segmentadas, ver la Figura 7. Por lo tanto, las dos posibles decisiones del usuario son detalladas a continuación:
  - *Recompensar a un individuo.* El usuario tiene la opción de recompensar un individuo agregando un peso $\omega = [0, 1]$ a su aptitud interna cuando él considere que vale la pena que esa información se transmita a las próximas generaciones. El peso representa qué tan *"bien"* esta la segmentación de la imagen para ese individuo de acuerdo al criterio del usuario. Por ello, si el usuario decide recompensar al individuo, entonces, la aptitud final está dada por:

$$\psi = \rho * (1 + \omega), \tag{15}$$

    donde ρ es la aptitud interna, ver Ecuación III.5.2.
  - *Penalizar a la población actual.* El usuario puede penalizar a toda la población cuando no exista dentro de ella algún individuo que valga la pena para las

próximas generaciones, de esta manera ayuda al proceso a darles menos prioridad a estos individuos. En ese caso, $\mu = [0, 1]$ es usada como un factor de penalización el cual representa qué tan *"mala"* es la segmentación de las imágenes dentro de la población actual de acuerdo al criterio del usuario. Por lo tanto, si el usuario decide penalizar a la población, la aptitud esta dada de la siguiente manera:

$$\psi = \rho * (1 - \mu). \tag{16}$$

Para llevar a cabo esta interacción, el algoritmo *I-EvoSeg* detiene el proceso durante la etapa de evaluación para observar y juzgar a la población actual, de esta forma, se ayuda al GA a converger en mejores soluciones en un menor tiempo.

II.3.5 Resultados Experimentales

Los experimentos fueron llevados a cabo usando imagénes con textura de la base de datos Brodatz[3], donde cada imagen tiene un tamanño de 128×128 pixeles. Los experimentos fueron divididos en cuatro secciones: **Experimento I**, **II**, **III** and **IV**. Los resultados experimentales de nuestro algoritmo *I-EvoSeg* son comparados con los resultados obtenidos de nuestro previo algoritmo *EvoSeg* el cual funciona de la misma manera que *I-EvoSeg* pero sin usar la interacción. Lo anterior, fue con el fin de analizar los beneficios de IEC en este problema. Para el algoritmo *I-EvoSeg* utilizamos 30 generaciones con 30 invidividuos mientras que para *EvoSeg* usamos 50 generaciones con el mismo tamaño de la población. La razón de usar menos generaciones es por la ayuda que daría la interacción al GA convergiendo en menos tiempo. Cada experimento utiliza diferentes descriptores de textura con el fin de observar su beneficio en la segmentación, es decir, los descriptores son seleccionados de acuerdo a las características de la imagen de entrada. De esta manera, probamos diferentes combinaciones de descriptores

[3]http://sipi.usc.edu/database. Consultado, Agosto 2010.

de textura para cada imagen y escogimos los que generaban mejores resultados.

Experimento I

La imagen de entrada usada en este experimento es mostrada en la Figura 8(a) la cual tiene tres diferentes regiones: el fondo y dos círculos, los cuales usan la textura D15 y D34 de la base de datos Brodatz. Los descriptores de textura empleados por el algoritmo fueron: *Contraste*, *Homogeneidad Local*, *Directividad* y el $Momento_{k=1}$. La Figura 8(b-e) illustra estos descriptores aplicados a la imagen original mientras que los resultados del *I-EvoSeg* son presentados en la Figura 9, mostrando algunos de los individuos que fueron recompensados por el usuario y la imagen final segmentada. Por otro lado, la Figura 10 muestra algunos ejemplos de la evolución del *EvoSeg*, presentando mayor sobre-segmentación en los resultados, es decir, varias regiones mal clasificadas. Además, para observar estadísticamente el proceso de la evolución se muestran las gráficas de la aptitud de ambos algoritmos en la Figura 11. Podemos observar en estas gráficas, cómo es que el primer "buen" resultado se obtuvo mas pronto cuando la interacción del usuario fue empleada; es decir, Los mejores individuos fueron obtenidos en la 7^{ma} generación por *I-EvoSeg*, mientras que *EvoSeg* produció resultados comparables solamente hasta la generación 17. Por lo tanto, estos resultados muestran que la interacción ayuda a guiar al GA para encontrar la mejor estrategia de segmentación en menos generaciones.

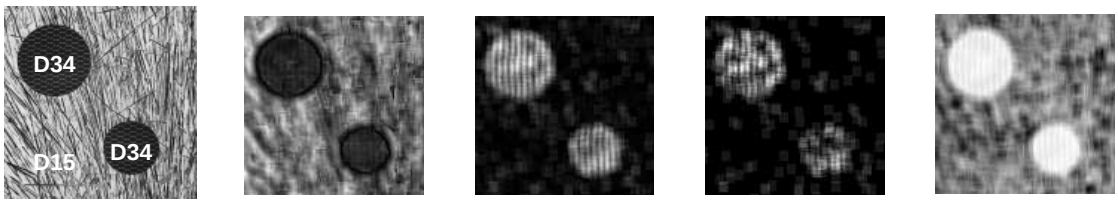


Figura 8. (a) Imagen de entrada usando las texturas D14 y D34. (b-e) Ejemplos de los descriptores de textura aplicados a la imagen de entrada.

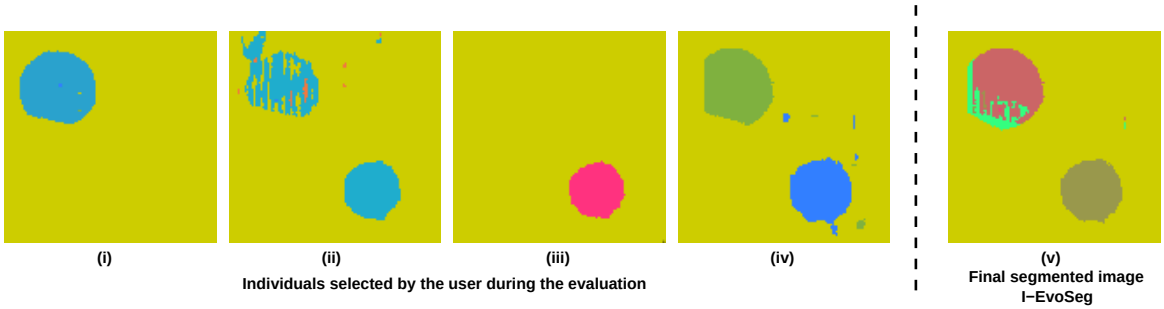


Figura 9. (i)-(iv) Individuos seleccionados por el usuario durante el proceso interactivo; (v) es la mejor imagen segmentada usando *I-EvoSeg* en el experimento I.

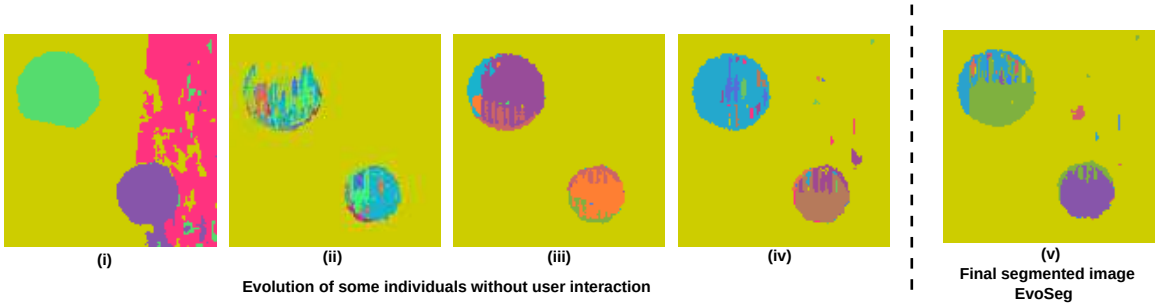


Figura 10. (i),(ii),(iii), y (iv) muestra la evolución del algoritmo *EvoSeg*. (v) representa la imagen final segmentada.

Experimento II

La imagen de entrada del experimento II ha sido usada por Yoshimura y Oe (1999) quienes también han trabajado en segmentación de imágenes con textura. Esta imagen es mostrada en la Figura 12(b-h) junto con las imágenes correspondientes a los descriptores de textura utilizados en este experimento. La Figura 13 muestra cómo la evolución de la población va mejorando durante el transcurso de las generaciones. La mejor segmentación de la imagen producida por *I-EvoSeg* fue obtenida en la 5^{ta} generación, mientras que la mejor segmentación del algoritmo *EvoSeg* fue obtenida hasta la generación 40, ver Figura 15. El mejor invidivuo obtenido en la 5^{ta} generación, posteriormente fue escogido por el proceso evolutivo como uno de los padres de las siguientes generaciones; sin embargo, ningún hijo fue mejor que su padre en las siguientes 25 generaciones restantes. Algunos ejemplos de los individuos que fueron seleccionados durante el proceso interactivo son mostrados en la Figura 13(i-iv). Por otro lado, los resultados del algoritmo *EvoSeg* mostrados en la Figura 14, presentan nuevamente sobre-segmentación en las imágenes.

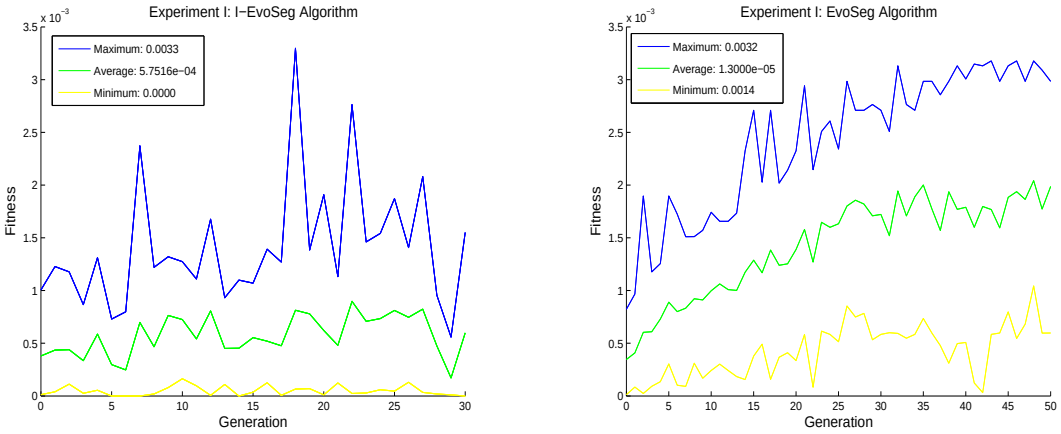


Figura 11. Gráficas de la aptitud del *I-EvoSeg* y *EvoSeg* correspondientes al experimento I.

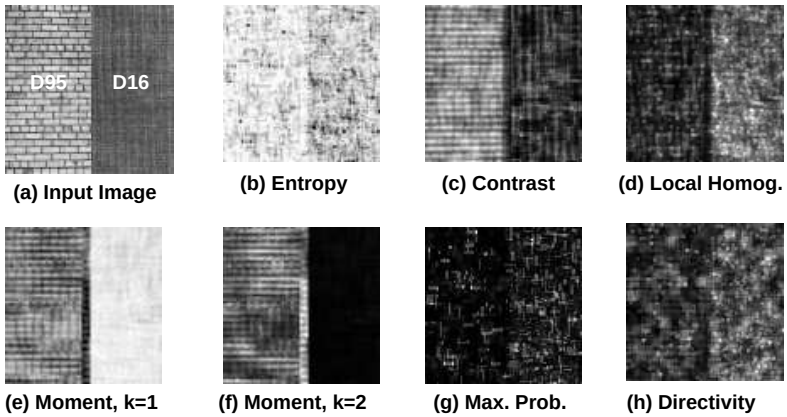


Figura 12. (a) Imagen de entrada para el algoritmo *I-EvoSeg* y *EvoSeg*. (b-h) Ejemplos de los descriptores de textura usados para segmentar la imagen de entrada.

Experimento III y IV

En los experimentos III y IV se usa como imagen de entrada para los algoritmos *I-EvoSeg* y *EvoSeg*, la Figura 16(a), la cual es usada también en el trabajo de segmentación de Yoshimura y Oe (1999). Esta imagen es interesante porque tiene cuatro diferentes texturas de Brodatz (D34,D84,D15,D9) ditribuidas en 5 diferentes regiones (un círculo y cuatro cuadros semi-ocluidos). Por tal motivo, es difícil distinguir las cinco regiones debido a que contienen patrones de textura similares (D9,D84,D15). La diferencia entre el experimento III y IV radica principalmente en el tipo de descriptores que fueron usados para llevar a cabo la segmentación. Los descriptores usados en el experimento III fueron: *Homogeneidad Local*, $Momento_{k=1}$ y $Momento_{k=2}$, ver Figura 16(b-d); mientras que para el experimento IV se

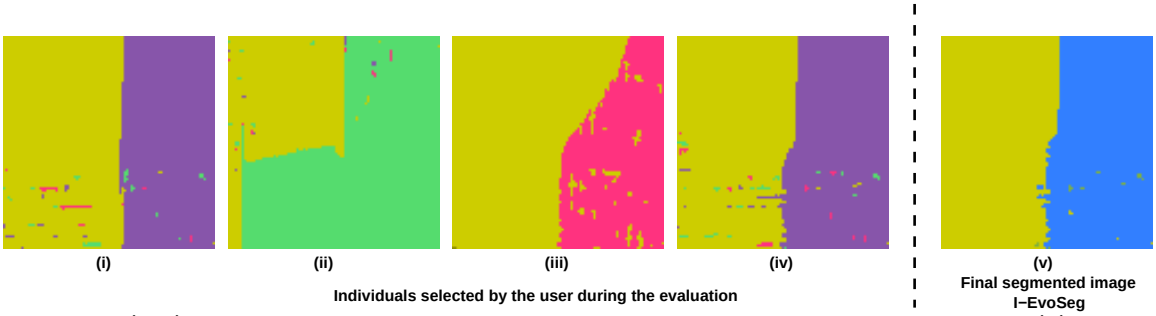


Figura 13. (i-iv) Individuos seleccionados por el usuario durante el proceso evolutivo. (v) Imagen final segmentada usando el algoritmo *I-EvoSeg* en el experimento II.

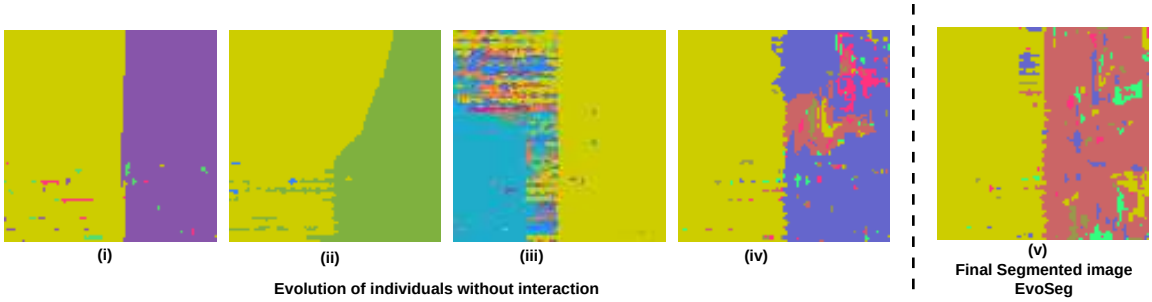


Figura 14. (i-iv) Individuos obtenidos durante la evolución usando el algoritmo *EvoSeg*. (v) Imagen final segmentada.

usó solamente el $Momento_{k=1}$. Observamos en los resultados que ambos algoritmos obtienen resultados similares usando estos tres descriptores o solamente uno, ver Figuras 17, 18, 20 y 21. Por otro lado, las gráficas de aptitud son presentadas en las Figuras 19 y Figura 22. Dado estos resultados, nosotros pensamos que es posible obtener mejores resultados agregando más generaciones para este particular caso porque observamos en el espacio de búsqueda producido por *I-EvoSeg* contaba con individuos muy prometedores para las próximas generaciones.

II.3.6 Discusión de Resultados

En esta Sección se presentó nuestro algoritmo de segmentación evolutiva-interactiva *I-EvoSeg*, con el cual se obtuvieron mejoras significativas en comparación con nuestro algoritmo *EvoSeg* presentado en la Sección II.3.1. El algoritmo propuesto usa un criterio de homogeneidad basado en la información de la textura, aunado con la evaluación de un experto del área en el proceso evolutivo. La interacción del experto con el sistema ayudó a identificar la mejor segmentación de la imagen de un conjunto de soluciones propuestas por el algoritmo. La

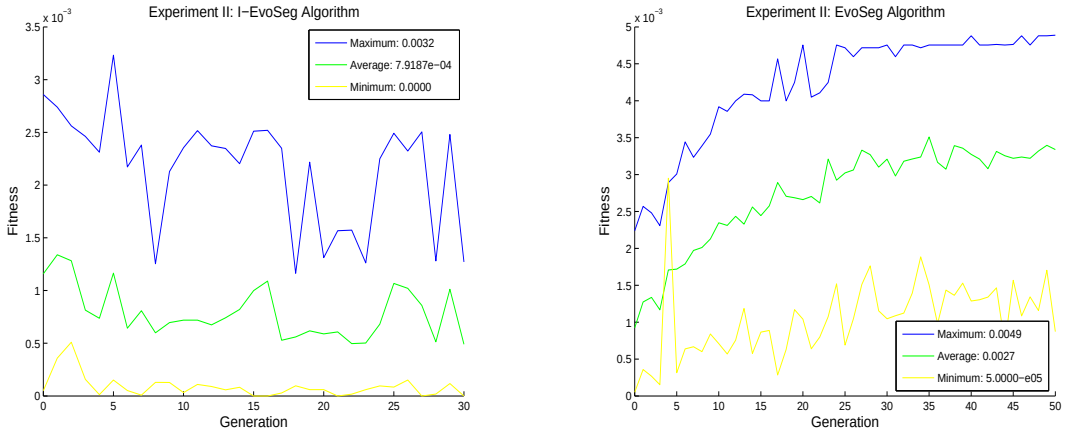


Figura 15. Gráficas de la aptitud generadas por *I-EvoSeg* y *EvoSeg* que corresponden al experimento II.

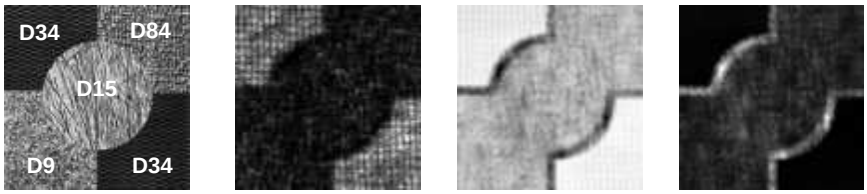


Figura 16. (a) Imagen de entrada para el algoritmo *I-EvoSeg* y *EvoSeg*. (b-d) Descriptores de textura usados en el experimento III. (c) Descriptor usado en el experimento IV.

información proporcionada por la persona fué utilizada como un recurso externo lo que ayudó al algoritmo genético a encontrar mejores soluciones cualitativas. Para ello, el proceso de optimización del algoritmo *I-EvoSeg* consideró una aptitud interna basada en la similaridad de la información local y global; y una aptitud externa, la cual dependió de la experiencia del usuario. De esta manera, la información proporcionada por la GLCM usando diferentes descriptores estadísticos ayudó en gran medida en la aptitud interna para diferenciar las regiones de la imagen mediante el proceso de dividir y fusionar, mientras que la aptitud externa para tener una mejor selección de los individuos. Como resultado, observamos en la experimentación que una de las ventajas de usar el proceso interactivo es que mejora la segmentación de las imágenes en menor tiempo. Por lo tanto, para llevar a cabo la experimentación probamos el algoritmo *I-EvoSeg* con el algoritmo *EvoSeg*, para lo cual usamos imaágenes con textura que han sido ampliamente utilizadas para este tipo de pruebas. Por último, podemos decir que los resultados experimentales mostraron que el enfoque interactivo

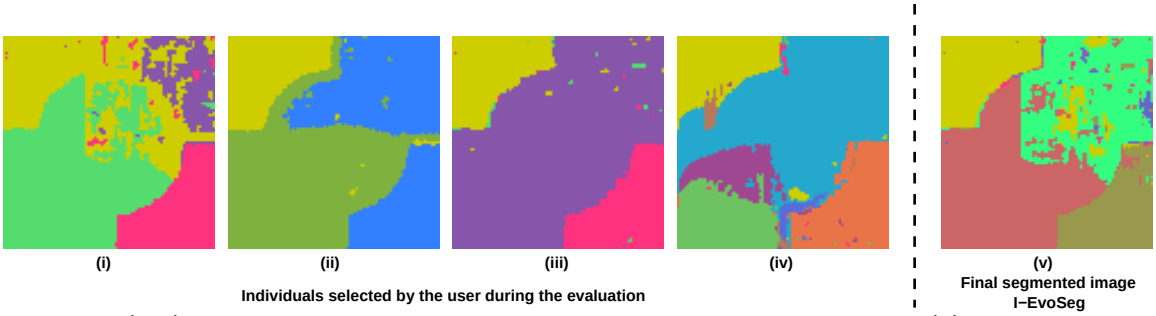


Figura 17. (i-iv) Individuos seleccionados por el usuario en el experimento III. (v) Imagen segmentada por *I-EvoSeg*.

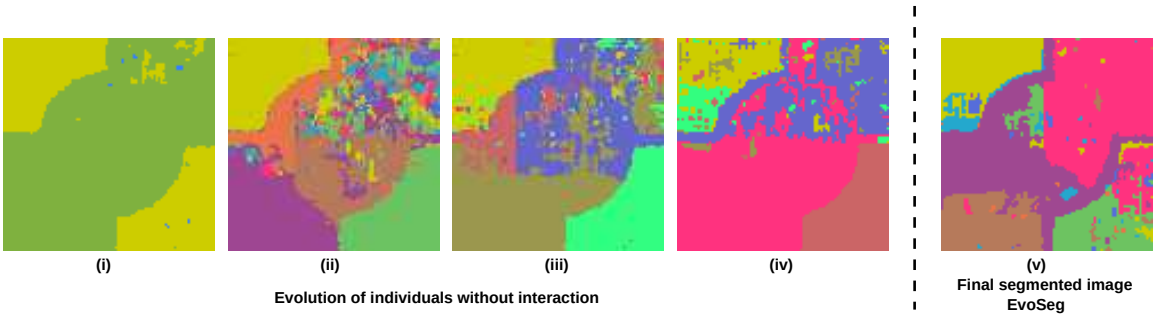


Figura 18. (i-iv) Ejemplos de individuos sin el proceso de interacción. (v) Imagen segmentada obtenida por *EvoSeg*.

produce cualitativamente mejores segmentaciones con un esfuerzo computacional menor. En un futuro, sería interesante analizar más a fondo los descriptores estadísticos con el fin de encontrar la mejor combinación entre ellos de acuerdo a los patrones de textura que tenga la imagen, y de esta manera, obtener una mejor segmentación.

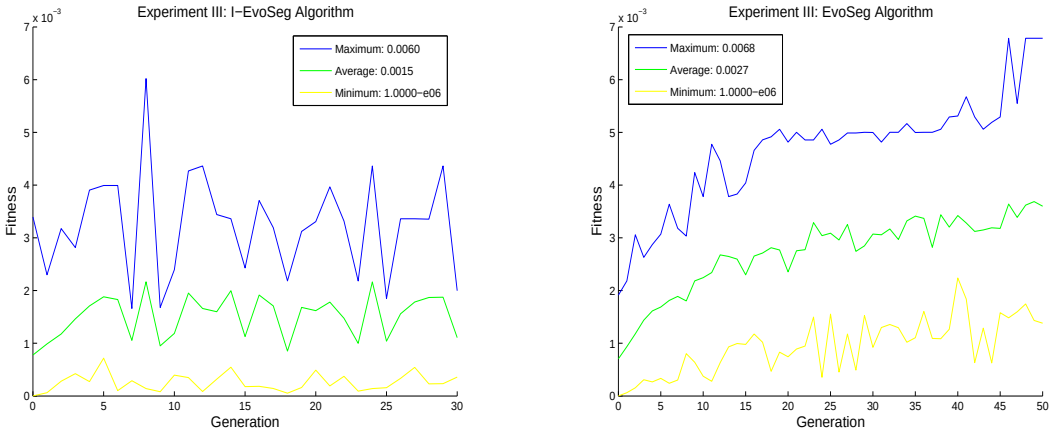


Figura 19. Gráficas de aptitud obtenidas por el algoritmo *I-EvoSeg* y *EvoSeg* que corresponden al experimento III..

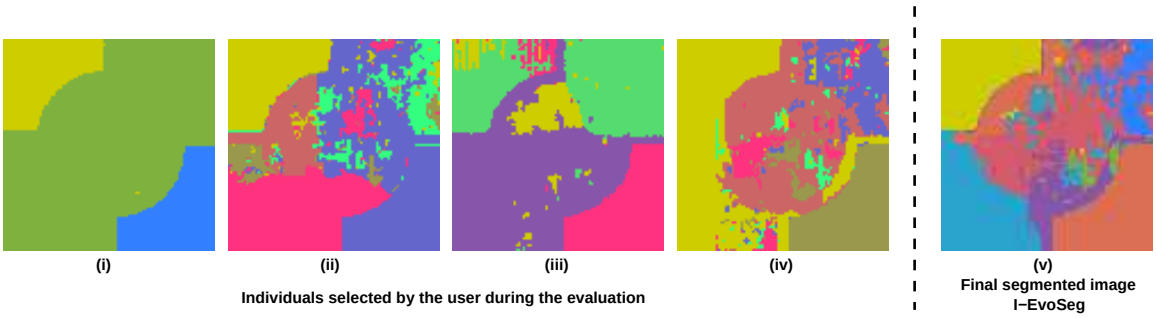


Figura 20. (i-iv) Individuos que fueron seleccionados interactivamente por el usuario durante el proceso de evaluación en el experimento IV. La imagen (v) representa la imagen segmentada por *I-EvoSeg*.

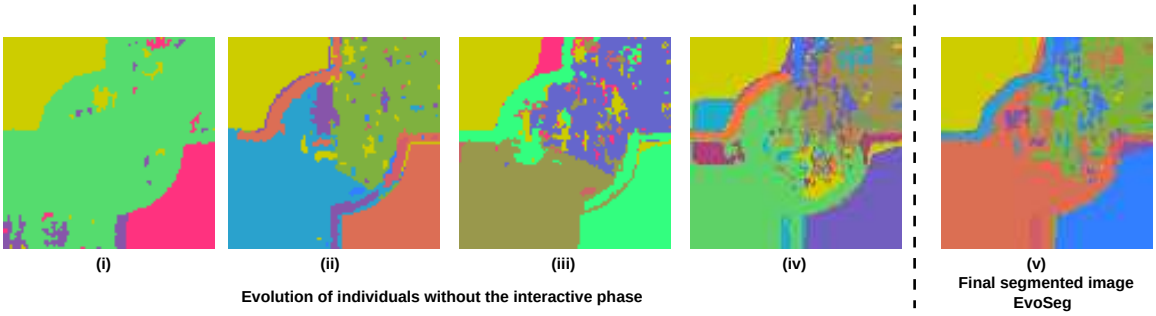


Figura 21. (i-iv) Ejemplos de los individuos obtenidos durante la evolución sin el proceso de interacción. (v) Imagen segmentada por el algoritmo *EvoSeg*.

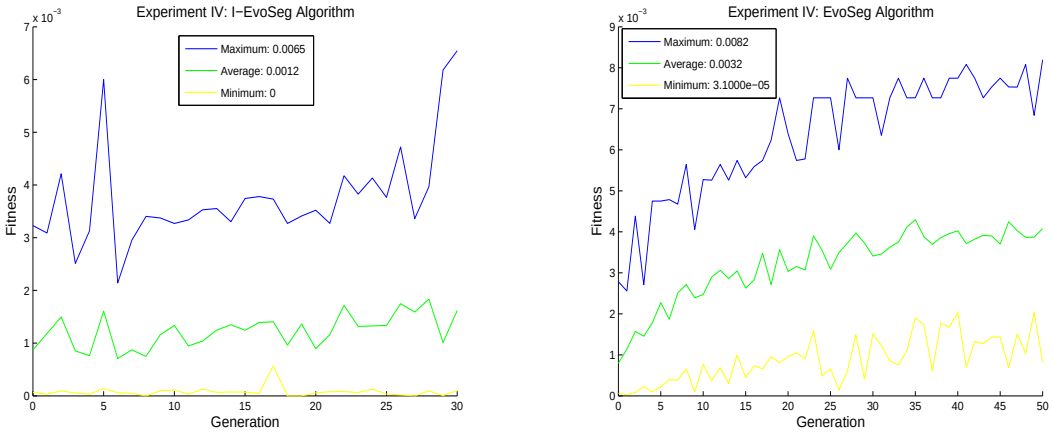


Figura 22. Gráficas de aptitud del algoritmo *I-EvoSeg* y *EvoSeg* correspondientes al experimento IV.

Capítulo III

Aprendizaje Evolutivo de Operadores de Descriptores Locales para el Reconocimiento de Objetos

En este capítulo se presenta un enfoque novedoso para diseñar automáticamente operadores descriptivos para el algoritmo SIFT. La optimización de nuestro algoritmo es llevada a cabo mediante el uso de la programación genética. El algoritmo evoluciona la fórmula de la magnitud del gradiente del descriptor SIFT a través de un determinado conjunto de funciones y terminales relacionadas con la derivación de la imagen, filtros gaussianos y operaciones básicas. Además, en este capítulo se propone un criterio de evaluación cuantitativo para descriptores locales basado en la medida F. Para ello, se evalúa nuestro descriptor $RDGP_2$ bajo diferentes tipos de transformaciones geométricas y fotométricas comparando su rendimiento con tres descriptores del estado del arte. Finalmente, la estrategia propuesta se aplica al reconocimiento de objetos en interiores y exteriores presentando una variedad de resultados experimentales.

III.1 Introducción

La descripción de regiones de interés es un paradigma atractivo y exitoso de la visión por computadora; éste paradigma conmúnmente es aplicado en los sistemas del estado del arte para resolver tareas que se han convertido en todo un reto para la comunidad científica, como por ejemplo: la detección y reconocimiento de objetos, correspondencia de imágenes, navegación de robots, recuperación de imágenes y minería de datos, por mencionar solo algunas aplicaciones reales. En ese sentido, este trabajo propone una metodología basada en el cómputo evolutivo para diseñar automáticamente operadores de descriptores locales usando

programación genética. La aplicación de la programación genética (Genetic Programming, GP) a problemas relacionados con imágenes ha recibido un nivel de atención mayor desde que Koza (1992) introdujo la GP para resolver una tarea de detección de objetos. Los primeros trabajos se remontan a los años 90's cuando Trackett (1993) aplicó GP para resolver una tarea de detección de objetos. Después, Johnson *et al.* (1994) evolucionaron rutinas visuales las cuales fueron capaz de localizar patrones sencillos de las siluetas de las personas. Además, Teller y Veloso (1995) desarrollaron PADO (Parallel Algorithm Discovery and Orchestration) como una prueba del concepto que muestra la GP como un paradigma para construir todo un sistema de reconocimiento. Más tarde, Poli (1996) introdujo la GP en un problema clásico de segmentación de imágenes. Ebner y Zell (1999) introdujo una técnica basada en GP para evolucionar una tarea específica de un operador de la imagen, en particular, para reproducir el ya conocido detector de puntos de interés *Moravec*. En ese sentido, Howard *et al.* (1999) describieron un enfoque basado en GP donde los clasificadores binarios fueron evolucionados con el fin de detectar cosas interesantes en imágenes SAR (Synthetic-Aperture Radar). Más adelante, Zhang *et al.* (2003) usaron GP para realizar detección multiclase de objetos pequeños en imágenes muy grandes. Después, Lin y Bhanu (2005) realizaron detección de objetos en imágenes SAR a través de un enfoque basado en GP donde usaron un sistema de coevolución cooperativa; en este sistema las características más sencillas fueron combinadas de tal manera que las nuevas características generadas por el sistema son llamadas características sintetizadas. Recientemente, Hernández *et al.* (2007) presentaron un enfoque lineal del GP junto con una máquina de soporte vectorial como una estrategia de aprendizaje visual para el reconocimiento de expresiones faciales. Además, Song y Ciesielski (2008) mostraron un método de segmentación de textura rápido y preciso el cual fue desarrollado evolucionando clasificadores con GP. Finalmente, Trujillo y Olague (2008) describieron una metodología basada en GP que sintetiza operadores de bajo nivel de la imagen que detectan puntos de interés en una imagen digital. Todos los trabajos mencionados anteriormente describen métodos para resolver tareas relacionadas con imágenes, las cuales estan basadas en la evolución directa de

programas o algoritmos usando algún tipo de aprendizaje inductivo. En ese sentido, nuestro trabajo está inspirado en la misma idea; nosotros proponemos una metodología basada en GP donde las estructuras de árboles, las cuales corresponden a operadores matemáticos, son evolucionados para mejorar el rendimiento de los descriptores locales.

La investigación sobre descriptores locales ha recibido gran atención en los últimos años debido a sus múltiples ventajas. Por ejemplo, las características locales pueden ser diseñadas para ser muy tolerantes a transformaciones geométricas y fotométricas, oclusiones, asi como también para promover su distintividad. La idea básica, es primero detectar puntos o regiones de interés que sean covariantes a clases de transformaciones y posteriormente, se calcula un descriptor invariante para cada región de la imagen. De este modo, el objetivo de un descriptor local es obtener una descripción compacta y completa de la región, de tal forma, que capture numéricamente en un vector, la información acerca de la estructura local alrededor de cada punto de interés. Luego, los vectores descriptivos pueden ser correspondidos con los descriptores de otras imágenes que contengan diferentes tipos de deformaciones causados por los cambios de la posición de la cámara, luminosidad, y otros como: rotación, escalamiento, cambios de iluminación, compresión JPEG, transformación afín, distorción por lente *eyefish* y deformación no rígida, etc. Existen varios estudios comparativos recientes sobre descriptores de regiones Mikolajczyk y Schmid (2005); Carneiro y Jepson (2003); Moreels y Perona (2007) donde se reportan muy buenos resultados usando este tipo de transformaciones en las imágenes. En general, los mejores resultados son reportados por los métodos o algoritmos que usan histogramas para representar diferentes características de apariencia o forma como el algoritmo SIFT (Scale Invariant Feature Transform). En particular, el descriptor SIFT es considerado como el estado del arte de los descriptores locales.

III.1.1 Motivación y Planteamiento del Problema

El descriptor SIFT propuesto por Lowe (1999), es un histograma en 3D de las posiciones del gradiente y sus orientaciones. Cada región de la imagen es normalizada a una escala y es

representada a través de una concatenación de las orientaciones del gradiente formando un descriptor de 128 dimensiones. Primero, las magnitudes y orientaciones del gradiente son calculados con la información de una región de interés. Después, las magnitudes del gradiente son pesadas con un filtro gausiano sobre la región, ver Figura 23. En ese sentido, para detectar las regiones de interés, el algoritmo SIFT utiliza una diferencia de gaussianas en el espacio de escalas (DoG, Difference of Gaussians) y de esta forma, las regiones de la imagen (posiciones y escala) son seleccionadas por los puntos más sobresalientes de la diferencia de gausianas. Luego, las regiones son normalizadas a escala para calcular las derivadas de la imagen usando la diferencia de pixeles para obtener las magnitudes y orientaciones de la imagen. Ésta información es entonces dividida en 4×4 subregiones con el objetivo de calcular el histograma con las orientaciones del gradiente (8 posibles orientaciones) pesado por su magnitud para cada subregión. De esta manera, el descriptor SIFT es la concatenación de la información obtenida del histograma correspondiente a cada región de la imagen, formando con ello un vector de $4 \times 4 \times 8$ dimensiones. Por otro lado, el uso de la magnitud del gradiente en el proceso descriptivo ha sido criticada Lindeberg (1993, 1998b) especialmente, si el cálculo del gradiente es realizado con la diferencia de pixeles ya que hacerlo de esta manera se obtiene información muy sensible al ruido. En ese sentido, estamos interesados en probar con este trabajo si la magnitud del gradiente utilizada dentro del proceso descriptivo del algoritmo SIFT podría ser reemplazado por otra operación matématica, la cual llamamos *operador* que pudiera ofrecer un rendimiento mejor o comparable con los descriptores del estado del arte. La idea es proponer una metodología basada en la GP que sintetice operadores matemáticos para que sean sustituidos por la magnitud del gradiente utilizada en el descriptor SIFT. Una diferencia significativa con respecto a la investigación previa es que tales características locales han sido diseñadas por expertos usando representaciones tradicionales que tienen una definición clara y bien fundamentada matemáticamente. En este capítulo, mostraremos cómo una estrategia de aprendizaje es capaz de crear operadores compuestos que mejoren el rendimiento de los mejores descriptores locales del estado del arte. Para ello, usaremos el acrónimo RDGP

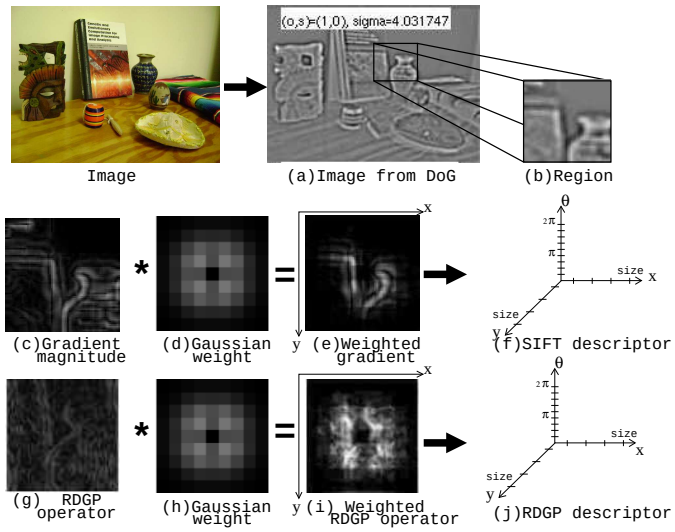


Figura 23. Operador del descriptor local de la imagen usado en el descriptor SIFT: c) Magnitud del gradiente or g) $RDGP_2$.

(Region Descriptor operators with Genetic Programming) para referirnos a nuestro operador del SIFT evolucionado, ver Figura 23. Además, observaremos en los experimentos que este sencillo cambio provoca una mejora significativa en el rendimiento del descriptor con respecto a los descriptores del estado del arte. Dos aspectos principales se abordarán en este capítulo: la selección de un conjunto de funciones matemáticas y operaciones genéticas que son útiles para definir y explorar el espacio de búsqueda, así como también la función de aptitud que es usada para medir el rendimiento de los descriptores locales.

III.1.2 Contribuciones de la Investigación

Este capítulo describe nuestro trabajo de investigación que consiste en un enfoque evolutivo para el cual utilizamos programación genética como una estrategia de aprendizaje, con el fin de crear operadores matemáticos que permitan mejorar el rendimiento del descriptor SIFT. Por lo tanto, las principales contribuciones de nuestro trabajo son las siguientes:

- Este trabajo propone una estrategia basada en la GP para automatizar el diseño de operadores para el descriptor SIFT.

- Proponemos un criterio cuantitativo basado en la medida F para la evaluación de de-

scriptores locales.

- Introducimos el operador $RDGP_2$ el cual mejora considerablemente el rendimiento del descriptor SIFT y de otros descriptores del estado del arte.

- El uso del $RDGP_2$ como descriptor en el reconocimiento de objetos de interiores y exteriores produce menos falsos positivos (en inglés, outliers).

III.2 Reconocimiento de Objetos usando características locales

Reconocer y localizar objetos es un problema clásico en visión por computadora y esencialmente se busca responder la siguiente pregunta: cuáles objetos estan presente en la escena y dónde se encuentran, dado algún conocimiento de cómo ciertos objetos pueden aparecer, más, una imagen de una escena que posiblemente contiene esos objetos. En ese sentido, la mayoría de la investigación en el reconocimiento de objetos involucra cuatro etapas que son: el preprocesamiento de la imagen, la segmentación, la extracción de características y la clasificación. Una de las desventajas de usar este enfoque es que los resultados dependen en gran medida de las etapas iniciales debido que es muy difícil recuperar datos en las etapas finales; y además el proceso de segmentación puede llegar a ser muy complejo y tardar mucho tiempo para llevarlo a cabo. Por ello, en los últimos años se ha optado por un enfoque basado en características locales el cual ofrece una metodología simplificada para el reconocimiento de objetos, donde las etapas de preprocesamiento y segmentación son eliminadas enfocándose solo en el análisis local de la imagen utilizando cantidades relativamente pequeñas de información. Estas etapas para llevar acabo el reconocimiento usando características locales son las siguientes:

1. **DETECCIÓN**. En esta etapa, se identifican los puntos más prominentes de cada imagen para después definir una región alrededor de cada punto. Para ello, se usa

un *detector de puntos de interés*.

2. **DESCRIPCIÓN**. Una vez que se obtienen las regiones de interés, se procede a describir y codificar la información contenida en ellas utilizando un *descriptor de regiones de interés*, obteniendo como resultado un número determinado de descriptores por cada imagen, los cuales estan representados por vectores numéricos de n dimensiones.

3. **CORRESPONDENCIA**. Finalmente, los descriptores de cada imagen son comparados con alguna medida de similitud para identificar las partes que corresponden al objeto dentro de la escena. Es decir, todos aquellos descriptores que obtuvieron un alto valor de similitud significa que se esta relacionando las mismas partes del objeto en ambas imágenes, independientemente de la transformación geométrica o fotométrica que exista entre ellas.

En la actualidad, se han obtenido muy buenos resultados en el reconocimiento utilizando este enfoque, como por ejemplo, en los trabajos de Lowe (1999); Lazebnik *et al.* (2003); Stein y Hebert (2005); Mortensen *et al.* (2005); Bay *et al.* (2006b); Dalal y Trigs (2006); Huan *et al.* (2008); Moreno *et al.* (2009); Ikizler y Duygulu (2009); Carneiro y Jepson (2009). Este enfoque es llamado modelo basado en apariencia el cual captura la información ya sea local o global del objeto que es proyectado en una imagen (dos dimensiones). Schmid y Mohr Schmid y Mohr (1997) introdujeron la idea de usar características locales invariantes para el problema de recuperación de imágenes, lo cual después se extendió para el reconocimiento de objetos con el trabajo propuesto por Lowe (1999), demostrando con ello cuán eficiente puede ser el reconocimiento de objetos usando características locales.

III.2.1 Características Locales

Las características locales representan la información más distintiva del contenido de la imagen; es decir, una característica local es un patrón de la imagen el cual es diferente al de su vecindario más cercano. Generalmente, se entiende por característica local aquella in-

Figura 24. Ejemplo del reconocimiento de un objeto usando características locales.

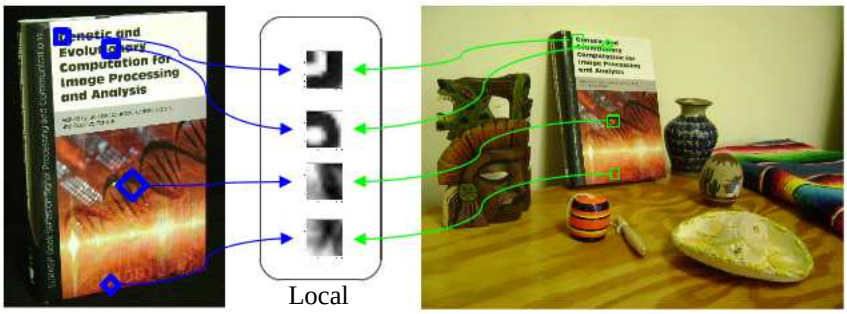


Figura 25. Ejemplo de algunas características locales encontradas en las dos imágenes. Las características en color azul representan al objeto que se desea reconocer y las de color verde representan a las de la escena donde se encuentra el objeto.

formación de la imagen donde existe un cambio en una o varias de sus propiedades. Las propiedades de la imagen que comúnmente son consideradas son la intensidad, color, y textura. De esta manera, una característica local puede ser un punto, contornos o pequeñas partes de la imagen representadas por ventanas cuadradas de información. La importancia de estas características radica en que a través de ellas es posible encontrar estructuras locales en la imagen de una manera repetible y además, es posible codificar esa información en una representación que es invariante a diferentes transformaciones de la imagen, tales como: traslación, rotación, escalamiento, deformación afín, iluminación, difuminación, etc. El resultado de estas características forman entonces la base del modelo basado en apariencia para el reconocimiento de objetos. De esta manera, el propósito de las características locales es proveer una representación que permita eficientemente corresponder estructuras locales entre imágenes para poder reconocer al objeto, ver Figura 25. Para lograrlo, estas características deben cumplir ciertas propiedades las cuales se describen a continuación.

Propiedades de las características locales

Como mencionamos anteriormente, se desea que las características locales representen de manera compacta el contenido de la información de una manera eficiente, de tal manera que al comparar las características del objeto con las de la escena independientemente de la transformación que exista entre las imágenes, se correspondan unas con otras y así poder identificar las mismas partes del objeto en ambas imágenes. Para ello, un buen conjunto de características locales serían aquellas que cumplan con las siguientes propiedades:

- **Invariancia**. Que pueda modelarse matemáticamente la invariancia a transformaciones geométricas y/o fotométricas de tal forma que se detecten las mismas características del objeto en ambas imágenes sin importar por ejemplo, si existe una transformación, tal como: traslación, rotación, escalamiento, iluminación, transformación afín, etc.

- **Repetibilidad**. Dada dos imágenes del objeto y la escena, un alto porcentaje de las características encontradas en una imagen deben ser encontradas en la otra, siempre y cuando representen al mismo contenido de información, es decir, a las mismas partes del objeto.

- **Distintividad**. Los patrones de intensidad de las características encontradas deben mostrar mucha variación de tal forma que puedan ser distinguibles con respecto a las demás.

- **Localidad**. Las características deben ser locales. El uso de la información local permite reducir la probabilidad de oclusión que pueda haber del objeto presente en la escena; además, con la información local se puede modelar las transformaciones geométricas y fotométricas que puedan existir enter las dos imágenes.

- **Cantidad**. El número de características debe ser el adecuado de tal forma que reflejen el contenido de la información de manera compacta. Este número de características debe ser suficientemente grande como para poder reconocer el objeto pero no muy

grande porque habría redundancia de datos y tampoco muy pequeño porque se perdería información valiosa del objeto. De esta manera, el número óptimo depende en gran medida de la aplicación.

- **Precisión**. Que su localización sea lo más precisa posible en ambas imágenes aún existiendo ciertas transformaciones geométricas o fotométricas del objeto en la escena.
- **Eficiencia**. Que su localización y descripción sea rápida para el tipo de aplicación donde vayan a ser usadas.
- **Robustez**. Las características deben ser robustas al ruido, efectos de discretización, compresión, difuminación, rotación, escala, iluminación, transformaciones afines, etc.

Definitivamente la importancia de estas propiedades depende de la aplicación, ya que se necesita hacer un compromiso entre todas ellas. Por ejemplo, la repetibilidad es necesaria en todo tipo de aplicaciones donde se utilicen las características locales ya que depende directamente de otras propiedades como la invarianza, robustez, cantidad, etc. La distintividad y la localidad son propiedades que es dificil satisfacerlas simultáneamente al 100%, si se es muy exigente, ya que entre más local es una característica menos información se tiene acerca de esa parte del objeto, lo cual puede ser mucho más difícil corresponderlas con las características del objeto presente en la escena ya que puede haber otras muy similares que no correspondan al mismo objeto. Por otro lado, si se tienen solamente rotaciones y objetos planares, entonces, no hay problema con las oclusiones ni con las discontinuidades propias de la profundidad ya que las imágenes son relacionadas por una homografía global. En este caso, el tamaño de las características locales pueden ser incrementadas sin ningún problema, resultando con ello, una alta distintividad.

De manera similar, un alto nivel de invariancia o de robustez tiende a reducir la distintividad, debido a que generalmente cuando se desea mucha robustez se descarta información que puede ser considerada como ruido y por lo tanto, se conservan muchas menos características. En el caso de la precisión, es importante tomarla muy en cuenta en aplicaciones donde sea

necesario tener precisión en la correspondencia como en la estimación de la geometría epipolar en la correspondencia estereoscópica, obtener estructura 3D apartir del movimiento de objetos, etc. La propiedad de cantidad por ejemplo, es útil en el reconocimiento de escenas o clases de objetos donde es necesario cubrir densamente el objeto de interés, pero no se debe perder de vista que un número muy alto de características puede requerir demasiado tiempo computacional y posiblemente se pueda caer en redundancia de datos. De hecho, como hemos mencionado anteriormente, es difícil lograr un balance entre todas estas propiedades porque dependen en gran medida de la aplicación ya que si no se toman en cuentan podríamos tener un impacto negativo en los resultados.

III.2.2 Detectores de Puntos de Interés

Los sistemas de reconocimiento de objetos basados en características locales utilizan detectores locales para localizar los puntos más sobresalientes de la imagen, indicando con ello que en esa localización existe información relevante que será utilizada mas adelante por un descriptor local. Un detector de puntos de interés extrae solamente las posiciones exactas de los puntos más prominentes de una imagen. Estos puntos de interés son convertidos en regiones de interés solamente definiendo un vecindario alrededor de cada punto ya que para ciertas aplicaciones un punto da muy poca información acerca del contenido de la imagen. Si el detector regresa como resultado regiones en vez de puntos, entonces nos referiremos a éste como un detector de regiones de interés. Así mismo, los puntos de interés de una imagen generalmente se determinan aplicando una operación matemática a cada punto de la imagen (*operador local* (K)) generando con ello, una imagen de interés $K(I) = I^*$, ver Figura 26. Posteriormente, cada punto $(\mathbf{x})$ es considerado como punto de interés si se cumplen las siguientes dos condiciones:

1. $K(\mathbf{x}) > sup\{K(\mathbf{x}W) \ \mid \ \forall \ \ \mathbf{x}W \in W, \ \ \mathbf{x}W = \mathbf{x}\}$

2. $K(\mathbf{x}) > h$

Figura 26. Ejemplo del proceso de detección de puntos de interés.

donde W es un vecindario de tamaño $n \times n$ alrededor de $\mathbf{x}$, y h es un umbral definido experimentalmente. La primera condición hace referencia a la supresión de no máximos y la segunda es una umbralización que se efectúa para conservar los puntos más prominentes. El valor de h va a depender en gran medida del tipo de detector que se esté utilizando, entre más alto su valor, menor es la cantidad de puntos que se obtienen.

En los últimos años se han propuesto una gran variedad de detectores para extraer características locales que sean invariantes a rotación, traslación, escala y a transformaciones afines.Unos de los detectores invariantes a rotación y traslación más conocidos es el **detector de Harris** propuesto por Harris y Stephens (1988) y el **detector Hessiano** propuesto por Beaudet (1978). El detector de Harris esta basado en una matriz de auto-correlación (M). Esta matriz describe la distribución del gradiente del vecindario local de cada punto, ver Ecuación 17.

$$
\begin{aligned}
M(x, \sigma_I, \sigma_D) &= \sigma_D^2 . G_{\sigma_I} * \begin{bmatrix} I_x^2(\mathbf{x}, \sigma_D) & I_x I_y(\mathbf{x}, \sigma_D) \\ I_x I_y(\mathbf{x}, \sigma_D) & I_y^2(\mathbf{x}, \sigma_D), \end{bmatrix} \\
with \qquad I_x(\mathbf{x}, \sigma_D) &= \tfrac{\delta}{\delta x} G(\sigma_D) * I(\mathbf{x}) \\
G(\sigma) &= \tfrac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}
\end{aligned}
\tag{17}
$$

donde σ_D es la escala de derivación y σ_I la escala de integración. $I_x(\mathbf{x}, \sigma_D)$ es la derivada Gaussiana en la dirección x de la imagen en el punto $\mathbf{x}$ y $I_y(\mathbf{x}, \sigma_D)$ en la dirección y; $G(\sigma)$ es una función Gaussiana de suavizado con desviación estándar σ. Una vez que se obtienen los valores de la matriz (M), éstos son utilizados para obtener el operador K de Harris, el cual

esta dado por:

$$K_{harris} = det(M) - \lambda trace(M)^2, \tag{18}$$

donde $det(M)$ es el determinante de la matriz de auto-correlación y $trace(M)$ es la traza de la matriz M.

Por otro lado, el **detector Hessiano** esta basado en una matriz de segundas derivadas llamada *Hessiana* (H),

$$H(x,\sigma) = \begin{bmatrix} I_{xx}(\mathbf{x},\sigma) & I_{xy}(\mathbf{x},\sigma) \\ I_{xy}(\mathbf{x},\sigma) & I_{yy}(\mathbf{x},\sigma), \end{bmatrix} \tag{19}$$

donde I_{xx}, I_{yy}, I_{xy} corresponden a las segundas derivadas que se calculan para cada punto $\mathbf{x}$.

Los componentes de esta matriz también han sido usados para extraer características y para describir los puntos de interés, ver Beaudet (1978); Kitchen y Rosenfeld (1982); Koenderink (1984) y Schmid y Mohr (1997). Por lo tanto, el operador K del detector Hessiano esta dado por el determinante de la matriz Hessiana, como sigue:

$$K_{hessian} = det(H) = I_{xx}.I_{yy} - I_{xy} \tag{20}$$

El determinante de la matriz alcanza su máximo para las esctructuras "*blob*" presente en las imágenes. De esta manera, los puntos de interés que se obtienen al aplicar el operador $K_{hessian}$ a la imagen, se encuentran en su mayoría localizados en las esquinas y en áreas con mucha textura.

Por otro lado, Trujillo y Olague (2006a) proponen un nuevo enfoque para obtener de manera automática operadores de puntos de interés utilizando como herramienta de optimización, la programación genética. Ellos mostraron que el enfoque propuesto genera operadores confiables y compactos de los cuales se obtuvieron como mejores, el operador IPGP1 y IPGP2.

Estos operadores fueron comparados con el detector Harris y se demostró que para el caso de rotación estos dos operadores fueron mejores de acuerdo a su tasa de repetibilidad. Más adelante, Trujillo y Olague (2007) extendieron su propuesta para obtener detectores invariantes a escala utilizando los operadores evolucionados. De esta manera, los resultados obtenidos fueron mejores en promedio en relación a dos detectores del estado del arte; además, mostraron que estos operadores tan sencillos obtenidos por medio de la evolución pueden obtener mejores resultados que los diseñados por el ser humano.

Detectores invariantes a escala

Los detectores invariantes a escala tienen como principal objetivo identificar los mismos puntos de la imagen a diferentes escalas de manera automática. Witkin (1983) fue uno de los pioneros en formular y proponer las reglas principales de la teoría del espacio de escalas moderna relacionando las estructuras de la imagen representadas a diferentes escalas. Él propuso considerar a la escala como un parámetro continuo, el cual permitiera difuminar la imagen con una máscara de tamaño variable usando un kernel local basado en una función Gaussiana. Desde entonces, la representación del espacio de escalas y sus propiedades han sido ampliamente estudiadas; como resultado de esos estudios se han obtenido importantes contribuciones que han sido propuestas por Koenderink (1984) y Lindeberg (1994). Koenderink (1984) mostró que el espacio de escala debe satisfacer la ecuación de difusión realizando una convolución con un kernel Gaussiano y de esta manera dejó en claro que es la mejor solución para el problema de construir una representación multi-escala. De esta manera, se obtiene como resultado una pirámide de imágenes que simulan el nivel de escalamiento de acuerdo al valor del sigma de la función gaussiana. Entonces, tenemos que el espacio de escala de una imagen es definido como una función $L(x, y, \sigma)$ la cual es el resultado de convolucionar una imagen $I(x, y)$ con una función Gaussiana $G(x, y, \sigma)$ donde $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(x^2+y^2)/2\sigma^2}$. Basado en esta idea, Lindeberg (1998a) propone un detector llamado *LoG (Laplacian of Gaussians)* el cual busca los puntos máximos 3D (ubicación + escala) en el espacio de escalas aplicando una función

de Laplaciano de Gausianas normalizada como se muestra a continuación:

$$L(\mathbf{x}, \sigma) = \sigma^2(I_{xx}(\mathbf{x}, \sigma) + I_{yy}(\mathbf{x}, \sigma)) \tag{21}$$

donde I_{xx} y I_{yy} es la doble derivada en x y y de la imagen, respectivamente, y $\mathbf{x}$ es el punto (x, y) en la imagen. Por otro lado, Lowe (1999) propuso un algoritmo muy eficiente para el reconocimiento de objetos donde su detector esta basado en la localización de puntos máximos 3D en una pirámide de espacio de escalas que es construida con una función de *Diferencias de Gaussianas (DoG)* como su operador principal. De acuerdo con Mikolajczyk y Schmid (2004), estos dos detectores localizan los puntos máximos en los vecindarios que tienen contornos o esquinas, donde el cambio de intensidad esta en una sola dirección, además de ser menos estables debido a que su localización es mas sensible al ruido o a los pequeños cambios en la textura del vecindario. Para resolver ese problema, ellos proponen los detectores **Hessian-Laplace** y **Harris-Laplace** donde la escala es seleccionada de acuerdo al máximo local obtenido por la traza y el determinante de la matriz Hessiana (H) simultáneamente, Mikolajczyk (2002). El detector *Harris-Laplace* combina el operador de Harris (Ecuación 18) con el mecanismo de selección de escalas de Lindeberg (1998a). Este método primero aplica el operador de Harris para localizar los puntos candidatos más prominentes en cada nivel de escalamiento. Después, selecciona aquellos puntos máximos en el espacio de escalas determinados por la función de Laplaciano. De esta manera, los puntos de interés obtenidos son robustos a cambios de escala, rotación, iluminación, y ruido de la cámara. De manera similar, el detector *Hessian-Laplace* realiza el mismo proceso de detección que el detector Harris-Laplace, la diferencia radica en que éste detector usa el operador Hessiano (Ecuación 20) en vez del operador de Harris.

Por otro lado, Bay *et al.* (2006b) proponen un algoritmo de detección y descripción invariante a escala llamado "**SURF**" (Speed Up Robust Features). Este algoritmo se caracteriza por su rapidez de procesamiento y la eficiencia de sus cálculos en las etapas de detección

y descripción, es por ello, que es utilizado para aplicaciones reales. El detector SURF esta basado en un detector Hessiano-Laplace que es aplicado a imágenes integrales las cuales se hicieron populares a partir del trabajo propuesto por Viola y Jones (2001) ya que permiten reducir en gran medida el tiempo computacional. La representación del espacio de escala del detector SURF es un tanto distinta a la propuesta por Lindeberg (1998a) y Lowe (1999) ya que no se construye una pirámide de imágenes como tal (iterativamente no se va reduciendo el tamaño de las imágenes) sino mas bien, el uso de las imágenes integrales junto con los filtros gaussianos permiten un escalamiento a un costo constante.

Detectores invariantes a transformaciones afines

Un detector invariante afín es una generalización de un detector invariante a escala, porque además de ser invariante a rotaciones, traslaciones y escalamiento, es invariante también a una transformación afín de la imagen. En este caso, la transformación afín puede ser vista como un escalamiento no uniforme donde el cambio de escala puede ser diferente en cada dirección debido a que el escalamiento no uniforme tiene una influencia en la localización, la escala y además, en la forma de las estructuras locales. Es por ello, que cuando tenemos una transformación afín importante en la imagen, un detector invariante a escala comúnmente falla. De esta manera, podemos entender que un detector invariante a escala puede ser descrito por un círculo, mientras que en un detector invariante afín el circulo se transforma en una elipse debido al escalamiento uniforme y no uniforme, respectivamente. Como ejemplos de detectores invariantes afines podemos mencionar el detector Harris-Afin y Hessian-Afin, los cuales vienen siendo una adaptación de los detectores invariante a escala Harris-Laplace y Hessian-Laplace. La adaptación a una invariancia afín es llevada a cabo mediando un esquema de estimación iterativa donde el procedimiento es inicializado con una región circular obtenida de un detector invariante a escala. En cada iteración, se construye una matriz de segundo momento (en inglés, second-moment matrix) de la región y se calculan los eigenvalores de esta matriz. Con ello, se obtiene la forma elíptica de la región, la cual corresponde a la deformación afín dibujada en la

imagen, donde el radio de los ejes principales es proporcional al radio entre los eigenvalores de la matriz de transformación. Después se transforma el vecindario de la imagen de tal forma que la elipse es transformada a un círculo para actualizar la estimación de la posición y escala en la imagen transformada. Este procedimiento es repetido hasta que los eigenvalores de la matriz del segundo momento son iguales. Entonces, el resultado de este esquema iterativo es un conjunto de regiones elípticas invariantes a transformaciones afines ocasionadas por los cambios de vista de la cámara. Mikolajczyk y Schmid (2004) propusieron una adaptación afín de los detectores invariantes a escala Harris y Hessian Laplace, llamándolos detectores Harris y Hessian Afín donde la adaptación afín es basada en las propiedades de estimación de formas de la matriz del segundo momento. Ellos mencionan que la optimización de los tres parámetros afines (posición del punto, escala y forma) es muy compleja para que sea práctimente útil. Es por ello, que sugieren una aproximación iterativa de estos parámetros. Para una mejor comprensión acerca del estado del arte y las técnicas que se han implementado en detectores invariantes afines, consultar Mikolajczyk (2002).

III.3 Descriptores Locales

En la actualidad, gran parte de los investigadores del área de visión y de otras áreas tienen especial interés en el descriptor SIFT debido a su habilidad para detectar objetos bajo diferentes condiciones de vista; por tal motivo, se han hecho mejoras al algoritmo SIFT de diferentes maneras con diferentes propósitos, agregando información o cambiando la representación del histograma original. Algunos de los trabajos que han realizado estos cambios o se han inspirado en el algoritmo SIFT son los siguientes: Ke y Sukthankar (2004); Lazebnik *et al.* (2003); Mikolajczyk y Schmid (2003); Stein y Hebert (2005); Mortensen *et al.* (2005); Abdel-Hakim y Farag (2006); Bay *et al.* (2006b); Dalal y Trigs (2006); Bosch *et al.* (2007); Tola *et al.* (2008); Liu *et al.* (2008), entre otros. La idea de nosotros es reemplazar la magnitud del gradiente por una nueva operación matemática para que sea aplicada a las regiones detectadas de la

imagen con el fin de mejorar el rendimiento del descriptor. Dicha operación será desarrollada a través de un proceso de optimización utilizando la programación genética con el objetivo de mejorar el contenido de la información, la cual es una de las propiedades más importantes de los descriptores inspirados en el SIFT. De hecho, la invariancia del contenido de la información determina la distintividad y robustez de los descriptores locales.

La Tabla II presenta un análisis de algunos de los más importantes descriptores del estado del arte. Esta tabla resume los trabajos que son derivados directamente del descriptor SIFT, los cuales son marcados con $\surd$ y aquellos que estan basados en conceptos similiares como los histogramas, teorías, etc, son marcados con $\times$. De hecho, todos los descriptores presentados en esta tabla son organizados de acuerdo a su operador del descriptor, organización de los datos, dimensión del vector, y el tipo de aplicación donde éstos fueron probados.

Tabla II: Análsis de Descriptores Locales

Descriptor	**Operador**	**Organización**	**Aplicación**	**SIFT**
SIFT[1]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 3D	Reconocimiento-Objs.	$\surd$
PCA-SIFT [2]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Directo	Recuperación-Imgs.	$\surd$
RIFT[3]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 2D	Recuperación-Imgs. y Clasificación[4] Recon. Mariposas[5]	$\surd$
GLOH[6]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 3D	-	$\surd$
BSIFT[7]	$I^{(k+1)}(x, y)$	Hist. 3D	Detección-Objs.	$\surd$
SIFT-GC[8]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 3D	-	$\surd$
CSIFT [9]	$\hat{E}_{\lambda^i w} = \sqrt{\hat{E}^2_{\lambda^i x} + \hat{E}^2_{\lambda^i y}}$	Hist. 3D	-	$\surd$
SURF[10]	Haar-Wavelet * G_σ	Directo	Reconstrucción 3D[11] Recon. en Museos[12]	$\surd$
HOG[13]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Bloques	Detección-Personas	$\surd$
PHOG[14]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 2D	Clasificación-Imgs.	$\surd$
DAISY[15]	$G_o^\Sigma = G_\Sigma * \left(\frac{\partial I}{\partial o}\right)^+$	Mapas	Reconstrucción 3D	$\surd$
SIFT-flow[16]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 3D	Detección-Movimiento.	$\surd$
Continuación en la siguiente página				

[1]Lowe (1999) [2]Ke y Sukthankar (2004) [3]Lazebnik *et al.* (2003) [4]Lazebnik *et al.* (2003) [5]Lazebnik *et al.* (2004) [6]Mikolajczyk y Schmid (2005) [7]Stein y Hebert (2005) [8]Mortensen *et al.* (2005) [9]Abdel-Hakim y Farag (2006) [10]Bay *et al.* (2006b) [11]Bay *et al.* (2006b) [12]Bay *et al.* (2006a) [13]Dalal y Trigs (2006) [14]Bosch *et al.* (2007) [15]Tola *et al.* (2008) [16]Liu *et al.* (2008)

Tabla II – Análsis de Descriptores Locales (Continuación)

Descriptor	Operador	Organización	Aplicación	SIFT
Steer. filters[17]	$E_n(\theta) = [G_n^\theta]^2 + [H_n^\theta]^2$	Directo	Anális-Textura	×
(HTD-TBD-EHD) (SCD-CSD-CLD-DCD) [18]	Filtros HSV,HMMD & YCrCb	Histograma	Clasificación-Texturas Recuperación-Imgs.	×
SASI[19]	-	Clique windows	Recuperación-Imgs.	×
GIH [20]	I(x,y)	Histograma de Intensidad	Deformación Sintética Deformación No-Afín	×
MOPs [21]	Haar-Wavelet	Directo	Imgs. Panorámicas.	×
Wiccest[22]	$\hat{E}_{klm}(x, y, \sigma_i)$	Histograma	Reconocimiento-Objs.	×
Hölder[23]	$\alpha_p = sup_s\{f \in C^s(x_o)\}$	Anillos	Correspondencia-Imgs.	×
LESH[24]	Modelo de Energía	Hist. 2D	Reconocimiento-Rostros	×
SMD[25]	I(x,y)	Pares Estables	Correspondencia-Imgs.	×
DLID[26]	$\| \bigtriangledown\phi \| = \sqrt{\phi_x^2 + \phi_y^2}$	Hist. 2D	Deformación-Imgs.	×
WLD[27]	$\xi(I_c)$	Hist. 2D	Clasificación-Texturas Detección-Rostros	×

Ke y Sukthankar (2004) introdujeron un descriptor llamado PCA-SIFT que concatena las derivadas de primer orden de cada subregión en la magnitud del gradiente como en el SIFT; sin embargo, su originalidad está basada en reducir las dimensiones del vector de datos aplicando PCA (Principal Component Analysis). Lazebnik *et al.* (2003), propusieron el descriptor RIFT (Rotation Invariant Feature Transform) el cual generaliza el descriptor SIFT usando anillos concéntricos en una región circular normalizada, y sus datos son organizados con un histograma 2D; la magnitud del gradiente es utilizada como su operador descriptivo tal como el descriptor SIFT. Éste descriptor ha sido probado en aplicaciones como clasificación y recuperación de imágenes con textura. Mikolajczyk y Schmid (2005) propusieron el descriptor GLOH (Gradient Location and Orientation Histogram) el cual usa coordenadas polares en vez de cartesianas para organizar la información en el histograma 3D del SIFT; de nueva cuenta, el operador descriptivo utilizado es la magnitud del gradiente al igual que el SIFT. Stein y Hebert (2005) introdujeron el algoritmo BSIFT (Background-SIFT) el cual incorpora invariancia sobre el fondo de la imagen, además de la invariancia que ya considera el algoritmo del SIFT. Para lograr esto, ellos usan la teoría de difusión de calor para el proceso

[17]Freeman y Adelson (1991) [18]Manjunath *et al.* (2001) [19]Çarkacioglu y Yarman-Vural (2003) [20]Ling y Jacobs (2005) [21]Brown *et al.* (2005) [22]Geusebroek (2006) [23]Trujillo *et al.* (2007) [24]Sarfraz y Hellwich (2008) [25]Gupta y Mittal (2008) [26]Cheng *et al.* (2008) [27]Chen *et al.* (2008)

de detección y de descripción, lo que podría identificarse como la base del operador descriptivo de este algoritmo. Además, este descriptor ha sido usado en para detectar objetos en imágenes naturales y sintéticas. Mortensen *et al.* (2005) presentaron el descriptor SIFT+GC (SIFT+GlobalContext) el cual incorpora un contexto global al algoritmo SIFT agregando información de la forma en el proceso de descripción; su operador y la organización de los datos son implementados de la misma manera que el descriptor SIFT usando la magnitud del gradiente y el histograma en 3D. En el caso de la información obtenida del contexto global, usa un histograma en 2D de 60 dimensiones obteniendo en total un vector descriptivo 188 dimensiones. Abdel-Hakim y Farag (2006) propusieron el algoritmo Colour-SIFT (CSIFT) el cual agrega la invariancia de color al proceso de descripción del SIFT usando un modelo de color RGB Gausiano mediante el cual identificamos la magnitud del gradiente en color como su operador descriptivo y un histograma en 3D como la forma de organizar los datos. Bay *et al.* (2006b) presentaron el algoritmo SURF (Speed Up Robust Features) el cual fue inspirado en el algoritmo SIFT. Sin embargo, ellos mejoraron enormemente el tiempo computacional requerido para procesar la información usando imágenes integrales; además, usaron un detector hesiano muy rápido y las respuestas de Haar-wavalet para formar el vector descriptivo. Este algoritmo ha tenido gran aceptación para usarlo en aplicaciones reales donde se requiere de una respuesta rápida y eficiente para llevar a cabo la tarea de alto nivel.

Dalal y Trigs (2006) describieron un algoritmo inspirado en el descriptor SIFT llamado HOG (Histogram Oriented Gradient), el cual captura la estructura del gradiente en una malla densa usando bloques (cell-blocks) para el problema de detección de personas. Bosch *et al.* (2007) propusieron el descriptor PHOG (Pyramid Histogram of Oriented Gradient) inspirado en la representación de pirámide de imágenes propuesta por Lazebnik *et al.* (2006) y en el descriptor HOG. Ellos organizaron la información de las características en histogramas 2D aplicando los PHOG descriptor en clasificación de imágenes. Tola *et al.* (2008) introdujeron el descriptor DAISY el cual fue inspirado en el descriptor SIFT y GLOH con el fin de llevar a cabo de forma eficiente la correspondencia densa de imágenes. Ellos reemplazaron la

suma pesada de los gradientes por convoluciones de la imagen original con varias derivadas de filtros gausianos, llamando mapas de orientación convolucionados a los resultados de esas convoluciones. De aquí, que el operador descriptivo para el descriptor DAISY está relacionado con los filtros gausianos. En particular, DAISY produce muy buenas reconstrucciones en 3D. Liu *et al.* (2008) propusieron el algoritmo SIFT-flow el cual consiste en alinear imágenes de escenas complejas para lograr buenas correspondencias densas; al igual que el SIFT usaron la magnitud del gradiente y el histograma en 3D para obtener el vector de datos. Este descriptor ha sido usado para predecir movimiento de una sola imagen y para objetos en movimiento. Finalmente, Moreno *et al.* (2009) propusieron el SIFT-Gabor el cual realiza un análisis del gradiente basado en las funciones Gabor impares, cuyos parámetros son selectivamente sintonizados en vez del cálculo de las derivadas de la imagen con el método de la diferencia de pixeles; ellos ilustran la distintividad de su propuesta a través de sus experimentos sobre la correspondencia de imágenes y detección de objetos.

Por otro lado, existen otros descriptores locales que no han sido inspirados directamente del algoritmo SIFT, pero que para algunos trabajos es posible identificar el operador y la organización de los datos que fueron usados para incorporar la invariancia geométrica y fotométrica en el vector descriptivo. Por lo tanto, para algunos de estos trabajos fue difícil identificar el operador del descriptor dado la naturaleza de su proceso de descripción. Por ejemplo, Freeman y Adelson (1991) propusieron los filtros basados en derivadas gausianas (en inglés, steerable filters) las cuales podrían ser consideradas como el operador de este descriptor. Estos filtros usados como descriptores locales, han sido aplicados a la region de la imagen para producir el vector del descriptor, el cual esta constituido por la salida de cada filtro. Manjunath *et al.* (2001) presentaron algunos descriptores de color y textura basados en el estándar MPEG-7, los cuales fueron aplicados para recuperación de imágenes y clasificación de texturas. Los descriptores de color son formados usando histogramas y la información del color dominante, basado en los espacios de color HSV, HMMD, YCrCb. Los descriptores que usan los histogramas capturan la distribución global del color mientras que el descriptor

de color dominante es representado en una representación compacta. Tales descriptores de textura usan un banco de filtros en un espacio de color lo cual puede identificarse como el operador del descriptor.

Çarkacioglu y Yarman-Vural (2003) propusieron un descriptor de textura genérico para recuperación de imágenes llamado SASI (Statistical Analysis of Structural Information) el cual esta basado en la estadística de los coeficientes de correlación clique. El operador del descriptor SASI puede estar relacionado con el cálculo de la estadística de segundo orden referente a los coeficientes de correlación μ y σ definido en su artículo. Ling y Jacobs (2005) propusieron el descriptor del histograma de intensidad geodésica, GIH (Geodesic-Intensity Histogram) el cual captura la invarianza de la deformación sintética y la deformación real no-afín. Esta información es obtenida de la distribución conjunta de las distancias geodésicas y la intesidad de los puntos detectados. Para esto, el valor de intensidad es considerado como el operador del descriptor GIH, el cual es organizado en un histograma 2D. Geusebroek (2006) propuso un descriptor de color compacto llamado Wiccest el cual captura la información de color y de textura basándose en un histograma local de los bordes de color. El operador del descriptor Wiccest $\hat{E}_{klm}(x, y, \sigma_i)$, está relacionado con la convolución de los canales de color oponentes $E, E_\lambda, E_{\lambda\lambda}$, ver Geusebroek *et al.* (2001), con un filtro Gausiano $G_{kl}(x, y, \sigma_i)$. Trujillo *et al.* (2007) propusieron un descriptor local basado en la regularidad de Hölder que es la base de operador descriptivo. El descriptor Hölder organiza la información de la región derivada de la regularidad de la señal punto a punto en cuantro círculos concéntricos obteniendo un vector de 129 dimensiones. Sarfraz y Hellwich (2008) presentaron el descriptor LESH (Local Energy based Shape Histogram) usado para reconocimiento de rostros. Ellos generaron el descriptor LESH concatenando histogramas que acumulan la energía local para diferentes subregiones de la imagen donde la fórmula de la energía podría ser considerado como el operador de este descriptor. Gupta y Mittal (2008) presentaron el descriptor SMD (Stable Monotonic Descriptor) el cual es invariante a cambios monótonicos en las intensidades y robusto al ruido Gausiano. De aqui, que los valores de intensidad son considerados como el operador del

descriptor SMD. Cheng *et al.* (2008) propusieron el descriptor DLID (Deformable Local Image Descriptor) el cual es robusto a deformaciones de la imagen tales como: deformaciones afines, sintéticas, no-rígidas y deformaciones por el lente *eyefish*. Ellos usaron múltiples regiones de soporte de diferente tamaño para cada punto de interés con el fin de calcular un histograma de las direcciones del gradiente para cada región. Entonces, con ello podemos decir que el operador del descriptor DLID es la magnitud del gradiente. Finalmente, Chen *et al.* (2008) propusieron el descriptor local WLD (Weber's Law Descriptor) basándose en la teoría de ley de Weber para la clasificación de texturas y detección de rostros. El operador de este descriptor podría ser definido por la fórmula de la exitación diferencial $\xi(I_c)$ la cual usa las diferencias de intensidades entre los vecinos del pixel a evaluar.

III.4 Criterios de evaluación para descriptores locales

Hoy en dia, la prueba de evaluación para descriptores locales propuesta por Mikolajczyk y Schmid (2005) es ampliamente aceptada por la comunidad para realizar comparaciones entre los descriptores del estado del arte. El protocolo se encuentra disponible a través de su página de internet[1], junto con el código binario de todos los descriptores presentados en su artículo, así como también el conjunto de datos utilizado para la evaluación. Dicho protocolo está compuesto por los siguientes puntos principales:

La prueba de evaluación toma en cuenta imágenes con diferentes transformaciones geométricas y fotométricas y para diferentes tipos de escenarios. En este protocolo son evaluados seis diferentes transformaciones, tales como : cambios de vista, cambios de escala, rotación de la imagen, difuminación de la imagen, cambios de iluminación y compresión JPEG. De aquí, que cada secuencia de imágenes es obtenida tomando fotos de la misma escena en diferentes condiciones. La Figura 27 muestra el conjunto de imágenes de prueba utilizadas para llevar a cabo nuestra experimentación sobre descriptores locales. De la misma manera, éstas imágenes fueron utilizadas por Mikolajczyk y Schmid (2005) para realizar una comparación exhaustiva

[1] http://www.robots.ox.ac.uk/ vgg/research/affine/. Consultado Junio 2010.

de los descriptores del estado del arte con diferentes métodos. Dentro de este conjunto de imágenes, fueron estudiados dos escenearios diferentes, aquellos con imágenes naturales las cuales contienen una gran cantidad de texturas aleatoriamente orientadas; así como también aquellas compuestas de imágenes estructuradas las cuales contienen muchos contornos distintivos. En el caso de la compresión JPEG y la de cambios de iluminación, solo la contienen aquellas imágenes del tipo estructurado. Además, las imágenes son de escenas planares o bien, de cuando se fijó la posición de la cámara durante la adquisición. Por ello, las imágenes siempre son relacionadas con una homografía que es incluida en los datos de prueba. Sin embargo, para crear los datos que son tomados como referencia (ground truth data), la homografía se calcula en dos pasos: primero, se estima una aproximación de la homografía usando puntos seleccionados manualmente; segundo, para deformar la imagen transformada con respecto a la imagen de referencia, se calcula una homografía robusta.

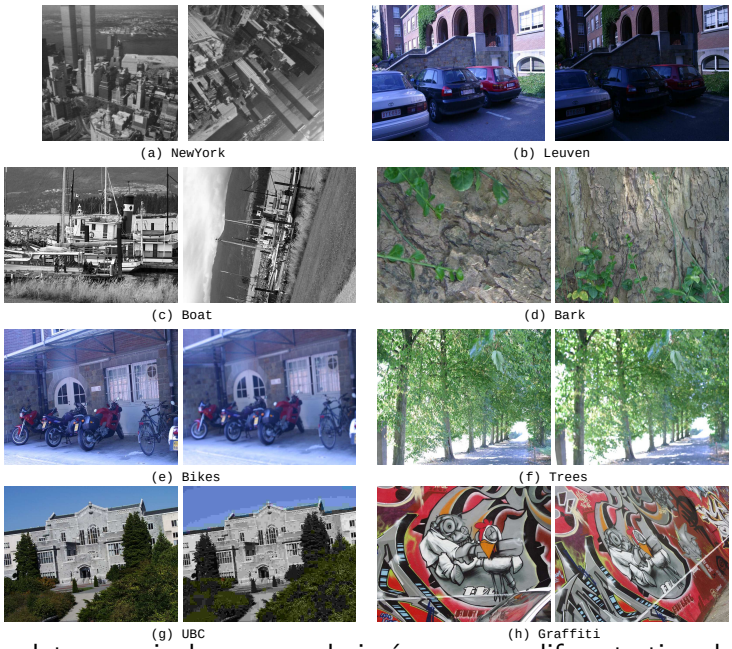


Figura 27. Base de datos que incluye pares de imágenes con diferente tipo de transformaciones. (a) Rotación; (b) Iluminación; (c)&(d) Rotación + Escalamiento; (e)&(f) Difuminación de la imagen; (g) Compresión JPEG; (h) Transformación Afín.

Para mejorar la calidad del descriptor, Mikolajczyk y Schmid (2005) proveen varios detectores de regiones invariantes a escala y a transformaciones afines. En los experimentos,

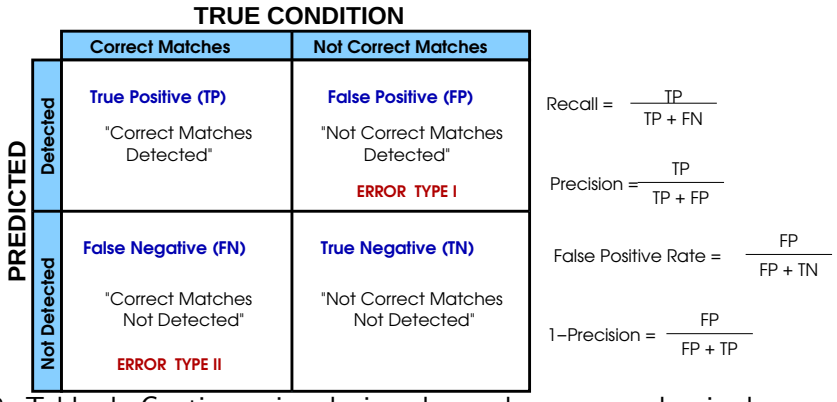


Figura 28. Tabla de Contingencia relacionada con la correspondencia de características.

nosotros usamos el detector original de Diferencias de Gausianas (DoG) propuesto por Lowe (1999), asi como también, los detectores Harris, Harris-Afín(haraff), Hessian-Afín(hesaff) y Hessian-Laplace(heslap). Los últimos tres detectores como se mencionó en la Sección III.2.2 dan como salida regiones elípticas lo cual es mejor para la correspondencia de descriptores cuando se tienen imágenes con transformaciones afines. Estos métodos proveen una alta precisión en la localización en el espacio de escala comparado con el detector DoG cuya detección es inestable. A pesar que la precisión de los detectores afecta el rendimiento de los descriptores, nosotros usamos el detector DoG debido a que viene integrado dentro del algoritmo SIFT. Sin embargo, observaremos en la experimentación que nuestro descriptor evolucionado, el cual usa el detector DoG, es mejor que los demás descriptores que usan un detector más robusto.

Actualmente, las técnicas de evaluación descritas en la literatura proponen diferentes criterios para corresponder descriptores locales. Por ejemplo, estan aquellos que trabajan en el espacio ROC (Receiver Operating Characteristic), Mikolajczyk y Schmid (2003); Chen *et al.* (2008); Ling y Jacobs (2005); Carneiro y Jepson (2002); así como también, en el espacio Recall vs 1-Precision, como por ejemplo, Ke y Sukthankar (2004); Mikolajczyk y Schmid (2005); Bay *et al.* (2006b); Trujillo *et al.* (2007); Cheng *et al.* (2008); Gupta y Mittal (2008); Moreno *et al.* (2009). La información requerida para obtener las curvas en estos espacios es derivada de la tabla de contingencia, ver Figura 28. Las gráficas ROC interpretan el rendimiento del

descriptor trazando los puntos correspondientes a la detección y a los falsos positivos Mikolajczyk y Schmid (2003), mientras que las gáficas de Recall vs 1-Precision dibujan una curva paramétrica que captura la compensación (*trade-off*) entre los datos de recall y 1-precision, ver Mikolajczyk y Schmid (2005). El uso apropiado de cada técnica depende del criterio utilizado para comparar los descriptores locales. Por esta razón, la técnica de Recall vs 1-Precision es usada para evaluar descriptores extraídos de pares de imágenes, mientras que el análisis mediante ROC es usado en el contexto de clasificación o recuperación de imágenes. De esta manera, las curvas Recall vs 1-Precision son más adecuadas para evaluar sistemas de detección debido a que no es necesario predecir los verdaderos negativos (true negatives) dado un par de imágenes. Agarwal *et al.* (2004) mencionaron que las curvas Recall vs 1-Precision son más apropiadas que las curvas ROC para medir el rendimiento de los sistemas relacionados con la detección de objetos. Por tal razón, nosotros trabajaremos en el espacio de Recall vs 1-Precision para evaluar los descriptores locales. En ese sentido, la prueba esta basada en el número de correspondencias correctas y falsas de los descriptores obtenidas de un par de imágenes, ver Figura 29. La idea es crear una curva de Recall vs 1-Precision usando un conjunto de métricas obtenidas de la tabla de contingencia. De esta forma, los verdaderos positivos, TP (true positive) y falsos positivos, FP (false positive) denotan las correspondencias correctas y falsas que fueron detectadas por el sistema; mientras que los falsos negativos, FN (false negative) y verdaderos negativos, TN (true negative) representan las correspondencias correctas y falsas que no fueron detectadas por el sistema respectivamente. Para este problema en particular, los verdaderos negativos nunca fueron calculados. De hecho, esa es una de las razones por la que es mejor trabajar en este espacio ya que no hay necesidad de calcularlos por la naturaleza del problema.

Por lo tanto, esta prueba consiste en contar la correspondencia de dos regiones A y B como correcta; si la distancia entre sus descriptores D_A y D_B están por debajo de un umbral t. Para ello, cada descriptor de la imagen de referencia es comparado con cada descriptor de la imagen transformada para obtener el número de correspondencias correctas y falsas (correct and false

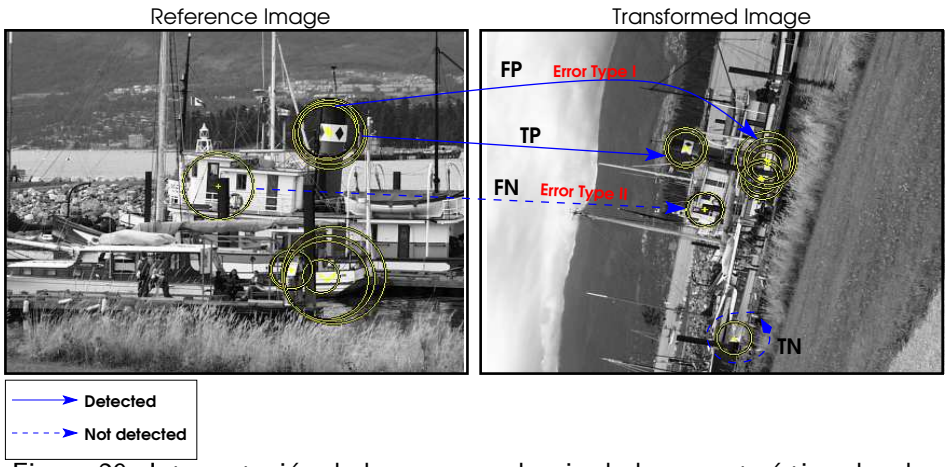


Figura 29. Interpretación de la correspondencia de las características locales

matches). El valor de t es variado para obtener las curvas de Recall vs 1-Precision. Recall es el número de regiones que fueron correctamente correspondidas con respecto al número de regiones que solamente fueron correspondidas entre sí entre las dos imágenes de la misma usando un error de solapamiento.

$$recall = \frac{\#matches \;\; correctos}{\#correspondencias} \tag{22}$$

Este error de solapamiento mide qué tan bien las regiones corresponden entre sí bajo una transformación homográfica de acuerdo al radio de intersección y unión de la región A y B. De aquí, que los autores asumen que una correspondencia es correcta si hay por lo menos un 50% de solapamiento entre las regiones y si la distancia entre ambos descriptores esta por debajo de cierto umbral. Además, 1-Precision es calculado como el número de regiones que fueron falsamente correspondidas con respecto al total de número de regiones que han sido correspondidas.

$$1 - precision = \frac{\#correspondencias \;\; falsas}{\#correspondencias \;\; correctas \;\; + \;\; \#correspondencias \;\; falsas} \tag{23}$$

Notar que las correspondencias falsas son calculadas sobre el número total de correspon-

dencias menos las correspondencias correctas.

Después de haber terminado de realizar los pasos anteriores, entonces, seremos capaces de comparar el rendimiento de la correspondencia para cualquier descriptor usando las curvas Recall vs 1-Precision. Naturalmente, esta comparación solo se puede llevar a cabo visualmente a través de las curvas que representan el rendimiento de cada descriptor donde la que esté mas cerca del eje del Recall y tenga menos puntos en el eje de 1-Precision es la mejor. Es decir, un descriptor perfecto sería aquel que diera como resultado un recall igual a uno para cualquier tipo de precision. En la práctica, el recall incrementa mientras que umbral es relajado teniendo con ello un costo, incrementar el nivel de ruido y decrementar la precision. Las curvas horizontales indican que el recall es alcanzado con un cierto nivel de precision que es limitado por las caracteríticas de la imagen. Entonces, una desventaja de ese enfoque es que implica una interprestación visual subjetiva de las curvas; por ejemplo, un problema ocurre cuando dos o mas gráficas sobrelapan entre sí. Además si queremos mejorar un descriptor local a través de la optimización es necesario definir una función que lo permita.

III.5 Automatización de Operadores Descriptivos usando Programación Genética, RDGP's

La programación genética (GP, Genetic Programming) es quizás la técnica de evolución más avanzada de la computación evolutiva que hasta el momento ya que ha tenido una gran aceptación por la comunidad científica. La GP ha sido inspirada de la evolución biológica para construir automáticamente programas computacionales que puedan realizar una tarea definida por el usuario. La formalización de esta técnica evolutiva se llevó a cabo en los años 90's por John Koza Koza (1992). Debido a los excelentes resultados obtenidos con este enfoque bio-inspirado, la GP ha recibido un gran interés para resolver problemas de visión por computadora en los últimos años; como ejemplo de ello, podemos mencionar en la detección de características (Trujillo y Olague (2006b)), detección de objetos (Zhang *et al.* (2003))

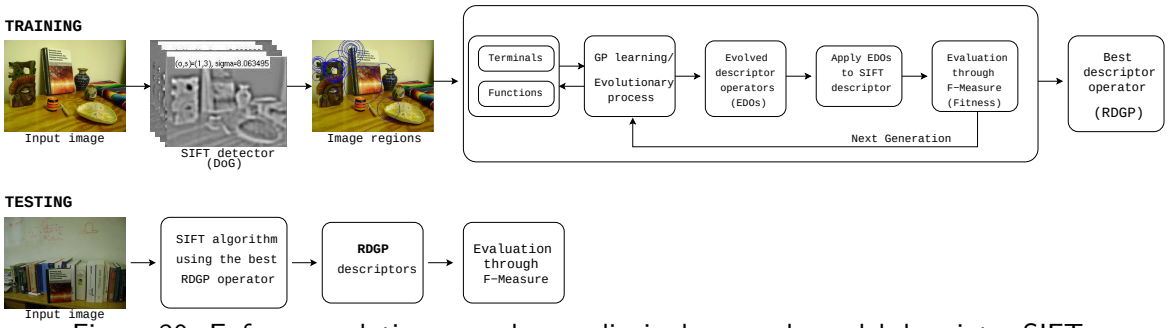


Figura 30. Enfoque evolutivo para el aprendizaje de operadores del descriptor SIFT

y segmentación de textura (Song y Ciesielski (2008); Poli (1996)), entre otros. Poli *et al.* (2008) presentó un libro sobre GP con la finalidad de proveer una guía moderna dirigida a principiantes y expertos sobre el uso y aplicaciones de esta poderosa herramienta evolutiva. La GP puede ser considerada como una rama de los algoritmos genéticos que resuelve problemas automáticamente, es decir, el diseñador no necesita conocer o especificar por adelantado la forma o estructura de la solución. La GP es usada para crear programas computacionales los cuales aprenden una función definida por el usuario donde el proceso de aprendizaje es dividido en dos etapas: la etapa de de entrenamiento y la de prueba. En la etapa de entrenamiento, la evolución es realizada por individuos codificados con una representación de árbol agrupados en una población. De esta manera, la GP comienza con una población inicial de programas generados aleatoriamente donde cada individuo es evaluado por medio de la función de aptitud, la cual verifica qué tan bueno es el individuo para ese problema en particular. En la población inicial los individuos generalmente tendrán una función de aptitud muy baja que tras el curso de las generaciones mejorarán paulatinamente debido a que algunos individuos de la población tendrán mejor rendimiento que otros. Para ello, el principio Darwiniano de reproducción y sobreviviencia del más apto, así como también, las operaciones genéticas de cruzamiento y mutación son usadas para crear una nueva descendencia de la población actual. Por otro lado, en la etapa de prueba el mejor resultado obtenido por la GP despues de varios experimentos son utilizados en el problema para el cual fue diseñado.

En esta Sección, describiremos los tres pasos principales para aplicar la GP en el algoritmo SIFT. Primero, es necesario definir el conjunto de terminales y funciones, asi como también,

la forma de combinarlas para encontrar la solución óptima; segundo, definir la función de aptitud que es crucial para que el programa funcione correctamente; y tercero, los parámetros para controlar el algoritmo y el criterio de terminación del programa. La Figura 30 muestra nuestro enfoque basado en la GP que es utilizado para sintetizar operadores matemáticos que optimicen el descriptor SIFT.

III.5.1 Representación, Espacio de Búsqueda y Operaciones Genéticas

Para definir la representación de la estructura que será evolucionada a través de la GP, hemos revisado cuidadosamente los procesos principales del SIFT. La idea no es mejorar todo el proceso del algoritmo, lo cual sería una tarea más allá de la capacidad de la GP, pero si identificar algo clave que pudiera mejorar el rendimiento y calidad del descriptor SIFT. Este descriptor consiste de cuatro etapas principales donde los descriptores son calculados en la última etapa, ver Figura 31. La descripción de la información local es basada en los gradientes de la imagen que son calculados sobre las regiones detectadas. Cada región detectada es rotada con respecto al ángulo de orientación dominante y escalada al tamaño apropiado de acuerdo a la escala a la que fue detectada esa región. Después, se calcula el histograma en 3D para cada región utilizando la información de la magnitud del gradiente, un peso gausiano y 8 posibles orientaciones para finalmente obtener el vector descriptivo de 128 dimensiones. Para esto, nosotros proponemos reemplazar la magnitud del gradiente que es utilizada por el SIFT y por los descriptores que se inspiraron en él, con una nueva operación evolucionada por la GP. Cabe mencionar que todas las operacones matemáticas evolucionadas por el sistema de aprendizaje basado en GP son probadas en miles de regiones de la imagen calculadas durante el proceso de localización.

Como mencionamos anteriormente, en la GP, los programas son codificados con una representación de árboles, los cuales están formados por nodos internos y hojas llamadas conjunto de funciones (F) y conjunto de terminales (T), respectivamente. De esta manera, el conjunto de funciones y terminales representan el espacio de búsqueda donde el sistema seleccionará

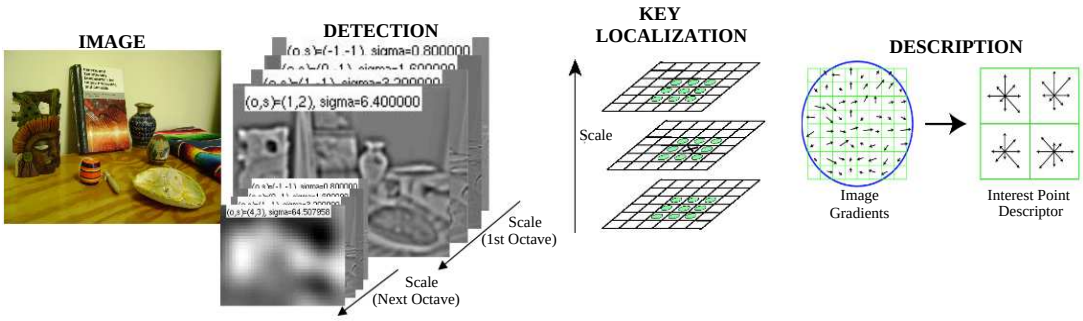


Figura 31. Etapas principales del descriptor SIFT: 1) Detección de los picos en el espacio de escala, 2) Localización de los puntos de interés, 3) Asignación de la orientación, 4) Descripción de los puntos de interés.

individuos de la población actual para realizar combinaciones entre ellos y producir nuevos individuos para la siguiente generación. La Figura 32 muestra una representación del programa $+(Dxx(I), log(Dyx(I)))$ para el cual $\{+, log\}$ son las funciones (nodos internos) y $\{Dxx(I), Dyx(I)\}$ son las terminales (hojas). Históricamente, los operadores diferenciales han sido usados a través de un conjunto de derivadas de la imagen que son calculadas hasta un cierto orden para describir las propiedades de un vecindario. Las propiedades de las derivadas locales (local jet) fueron investigados por Koenderink y van Doorn (1987) y después se propusieron un sin número de enfoques tales como: los filtros orientables (steerable filters) y los invariantes diferenciales que calculas las derivadas por convolución de derivadas gausianas. Particularmente, nosotros decidimos usar tales ideas para establecer nuestro conjunto de funciones y terminales, tal como sigue:

$$
\begin{aligned}
F = \Big\{ & +, |+|, -, |-|, *, \div, \sqrt{I_t}, \\
& \frac{I_t}{2}, log_2(I_t), D_x G_\sigma, D_y G_\sigma, G_\sigma \Big\} \\
T = \{ & D_x(I), D_{xx}(I), D_{yy}(I), D_{xy}(I), D_y(I) \}
\end{aligned}
\tag{24}
$$

donde I es la imagen de entrada y I_t puede ser cualquiera de las terminales en T, asi como también, la salida de cualquier función en F; D_u simboliza las derivadas de la imagen a lo largo de la dirección u, entonces tenemos que $D_u = I * G_{u(\sigma=1)}$; G_σ son los filtros gausianos con $\sigma = 1$ o 2; $D_u G_\sigma$ representa la derivada de un filtro gausiano con una difuminación

de la imagen con valor de σ. Estos conjuntos de funciones y terminales aseguran que la propiedad de cerradura es alcanzada porque todas las terminales y funciones son definidas como imágenes manteniendo así la consistencia de la información. Entonces, como nuestro objetivo es optimizar el operador del descriptor SIFT usamos este conjunto de funciones y terminales que son combinados para producir propiedades estructurales o funcionales no presentes de manera individual. De esta manera, es ampliamente aceptado que la GP es capaz de crear operadores compuestos; de aquí, que la buena elección del conjunto de terminales y funciones es muy importante para el resultado final. Para ello, estamos interesados en obtener operadores que sean simples en cuanto a su estructura y que al mismo tiempo sean capaz de mejorar el rendimiento del descriptor SIFT.

Después de que la generación inicial es creada y la aptitud ha sido asignada a cada individuo, el sistema selecciona probabilísticamente los mejores individuos de la población. Éstos serán los padres de los individuos de la próxima generación utilizando para ello, las operaciones genéticas de selección, mutación y cruzamiento. La función de selección es responsable de escoger los mejores individuos para la reproducción a través de la mutación y cruzamiento. La operación de mutación selecciona aleatoriamente un nodo (sitio de la mutación), el cual es borrado para sustituir esta parte del árbol con una nueva expresión para obtener un nuevo individuo. La Figura 32 muestra cómo el hijo (*operador descriptivo*) es creado con esta operación; por ejemplo, el nodo padre $D_x(I)$ es escogido como el sitio donde se efectuará la mutación y es sustituido por $log(D_{yx}(I))$ para crear un nuevo hijo. Por otro lado, el método de cruzamiento necesita un par de padres para realizar esta operación genética; primero, un sitio de cada padre es seleccionado aleatoriamente como el punto de cruzamiento; entonces, los subárboles son combinados para crear el nuevo hijo. La Figura 33 ilustra la operación final de cruzamiento usado para obtener nuestro operador $RDGP_2$.

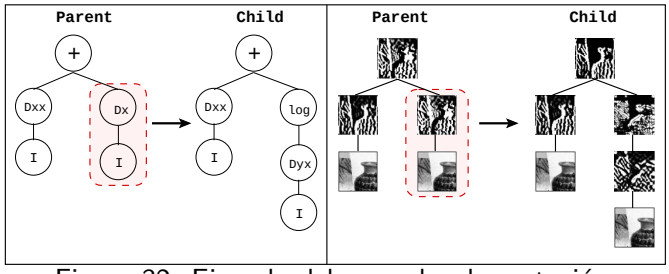


Figura 32. Ejemplo del operador de mutación.

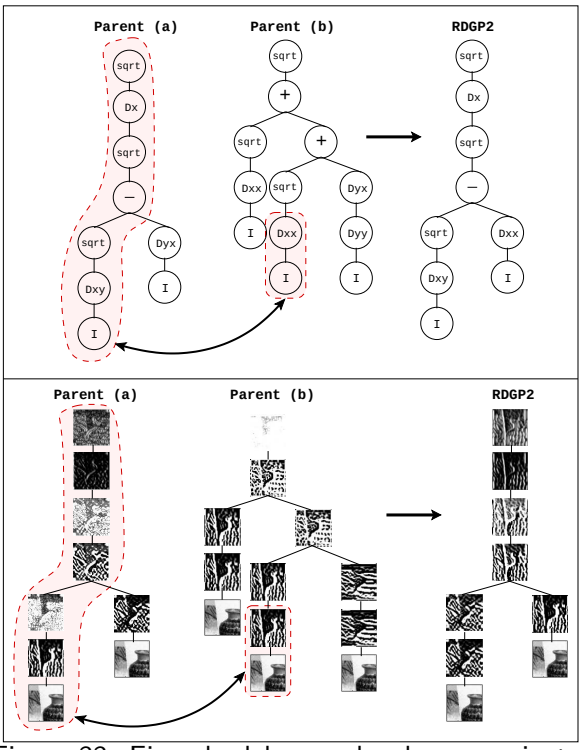


Figura 33. Ejemplo del operador de cruzamiento.

III.5.2 Función de Aptitud

El propósito de nuestra investigación es mostrar que la programación genética es una metodología poderosa capaz de mejorar los descriptores locales los cuales puedan ser usados para mejorar de igual manera el proceso del reconocimiento de objetos. En general, para aplicar la computación evolutiva es necesario idear una función de aptitud bien planteada junto con la representación del problema. En este trabajo, proponemos usar la medida F (*F-Measure*) como la función de aptitud para comparar cuantitativamente descriptores locales. La medida F fue originalmente propuesta en la comunidad de recuperación de información por Van-Rijsbergen (1979) como la función de Efectividad (E). La medida F está basada en media armónica dando el mejor balance entre las métricas de precision y recall, y es ampliamente utilizada como un criterio de evaluación. Recientemente, ha sido utilizada en algunas aplicaciones de visión por computadora tales como la detección de objetos (Agarwal *et al.* (2004)), vigilancia visual para detección de movimiento (Lazarevic-McManus *et al.* (2008)) y segmentación de imágenes (Martin *et al.* (2004); Arbelaez y Cohen (2006); Gimenez y Evans (2008)). De hecho, nuestro trabajo es la primera implementación de esta función para evaluar descriptores locales (Perez y Olague (2008, 2009)). La idea es obtener una medida cuantitativa para evaluar los descriptores locales en el espacio de Recall vs 1-Precision. Por otro lado, es verdad que otros criterios podrian ser usados para comparar cuantitativamente los descriptores tales como el valor de los verdaderos negativos (true negative rate), verdaderos positivos (true positive rate), precision pesada (weighted accuracy), Mediana G (G-Mean), precision y recall por mencionar algunos. De hecho, hay solo un trabajo previo que usa el área bajo la curva como función de evaluación que intenta mejorar descriptores locales usando parendizaje estadístico; sin embargo, Mikolajczyk y Schmid (2005) y Agarwal *et al.* (2004) mostraron que no es apropiado o conveniente usar ese criterio para evaluar descriptores en el contexto del reconocimiento de objetos.

La fórmula general de la medida F es definida por la siguiente ecuación:

$$F_\alpha = \frac{(1+\alpha)\cdot(p\cdot r)}{(\alpha\cdot p + r)} \tag{25}$$

donde p es la precision $\{p : 0 \leq p \leq 1\}$, r es recall $\{r : 0 \leq r \leq 1\}$, y α es el parámetro que controla el balance entre p y r, $\{\alpha : 0 \leq \alpha \leq \infty\}$. Notar que en el caso que $\alpha < 1$ la variable con mayor peso es p, mientras que para el caso que $\alpha > 1$ la variable con mayor peso es r, y cuando $\alpha = 1$ significa que la precision y recall estan bien balanceadas.

En nuestro trabajo, la siguiente ecuación es propuesta como la función de aptitud para nuestro sistema evolutivo la cual está adaptada de acuerdo a la prueba de evaluación propuesta por Mikolajczyk y Schmid (2005):

$$Q = \text{argmax}\left\{F_\alpha(P_i^x, R_i^x) = \sum_{i=1}^{n} \frac{(1+\alpha)\cdot(p_i\cdot r_i)}{(\alpha\cdot p_i)\ +\ r_i}\right\} \tag{26}$$

$$where \quad Q : F_\alpha(P^s, R^s) \quad \geq \quad F_\alpha(P^t, R^t)$$

con n que representa el número de umbrales. Los datos de la precision de un par de imágenes es denotado por $P^x = (p_1, p_2, ..., p_n)$ y los datos del recall por $R^x = (r_1, r_2, ..., r_n)$; Q representa la categorización de los descriptores en orden ascendente donde el valor más alto corresponde al descriptor que obtuvo mejor rendimiento. De esta manera, nosotros afirmamos que la medida F es un criterio sencillo y confiable que provee una evaluación significativa para los descriptores locales lo cual se demuestra en la sección de experimentación.

III.5.3 Inicialización y Parámetros de la Programación Genética

Una vez que definimos el espacio de búsqueda y la función de aptitud, el primer paso es comenzar el proceso evolutivo de manera aleatoria. La población inicial es creada usando el método de *ramped half-and-half* propuesto por Koza (1992), el cual selecciona la mitad de los individuos con el método de crecimiento (grow) y mitad con el método completo (full) para crear la población. El método completo crea árboles balanceados de acuerdo al valor de la profundidad inicial máxima mientras que el método de crecimiento crea árboles no

balanceados permitiendo ramas de longitud variable. Aquí, el tamaño de los individuos no debe exceder un valor de profundidad máximo definido por el usuario, esto con el fin de evitar un crecimiento descontrolado de los árboles conforme pasan las generaciones; es decir, esto ayuda a controlar el problema conocido como inflamiento (bloat). La profundidad de un árbol es definido como el número máximo de aristas que hay entre el nodo raiz y el nodo final. La profundidad del árbol se establece dinámicamente usando dos profundidades máximas que limitan el tamaño de cualquier individuo dentro de la población, ver Tabla III. La profundidad máxima dinámica es un profundidad máxima del árbol que no debe superar a ningún individuo, a menos que su aptitud sea mejor que la mejor solución encontrada hasta ese momento. Si esto pasa, la profundidad máxima dinámica se modifica aumentándola al número máximo de la profundidad del nuevo individuo. Por el contrario, se reduce si el nuevo mejor individuo tiene una profundidad mas baja. La profundidad real máxima es el límite real estricto al que ningún individuo puede sobrepasar bajo ninguna circunstancia. Además, la Tabla III provee otros parámetros para la ejecusión del programa usados en la fase de experimentación. Estos parámetros tienen valores canónicos que fueron establecidos empíricamente despues de un cierto número de pruebas. El parámetro de selección es llevado a cabo usando el método del torneo conservando el mejor individuo siempre. Finalmente, el criterio de terminación fue definido por un número máximo de generaciones; entonces, el proceso evolutivo alcanza un operador óptimo por cada experimento.

Tabla III. Parámetros del algoritmo RDGP.

Parámetros	**Descripción**
Generaciones	50
Tamaño de la Población	50 individuos
Inicialización	Ramped Half-and-Half
Cruzamiento	0.90
Mutación	0.10
Profundidad del Árbol	Selección de profundidad dinámica
Profundidad máxima dinámica	7 niveles
Profundidad máxima real	9 niveles
Selección	Stochastic Universal Sampling
Elitismo	Se conserva el mejor individuo, 1/50

III.6 Resultados Experimentales

Esta Sección presenta tres resultados experimentales realizados para evaluar el impacto de nuestro enfoque sobre el diseño automático de operadores descriptivos para el descriptor SIFT. El primer experimento describe nuestro algoritmo de aprendizaje que sintetiza operadores usando la prueba de evaluación de Mikolajczyk y Schmid (2005) para corresponder los descriptores. El segundo experimento, provee evidencia que nuestro mejor operador optimizado puede mejorar significativamente el rendimiento del descriptor SIFT y de otros dos descriptores del estado del arte. Finalmente, se lleva a cabo la tarea del reconocimiento de objetos usando nuestro mejor operador descriptivo llamado $RDGP_2$.

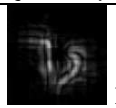
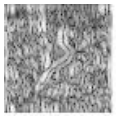
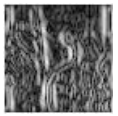
III.6.1 Aprendizaje de los operadores RDGP's

El proceso de aprendizaje es llevado a cabo usando un protocolo de correspondencia de datos que fue diseñado para corresponder miles de regiones de interés entre un par de imágenes, la cual fue mencionada anteriormente en la Sección III.4. En ese sentido, nosotros comparamos la eficiencia de nuestros operadores evolucionados contra tres descriptores del estado del arte: el SIFT original, GLOH y SURF. Además, para esta comparación se aplicaron diferentes detectores de regiones de interés. La implementación del aprendizaje de operadores descriptivos fue programado en Matlab usando la herramienta de programación genética GPLAB[2] mientras que la plataforma central para obtener las características locales del SIFT fue programado en Matlab/C[3]. Para ello, seleccionamos el par de imágenes llamada boat, ver Figura 27(c), para llevar a cabo el aprendizaje de operadores debido a que esta secuencia presenta cambios de rotación y escalamiento. Después, probamos el mejor operador evolucionado en todos los demás pares de imágenes contenidos en la base de datos, y con ello, mostramos que usando esta poderosa herramienta evolutiva es posible obtener resultados impactantantes.

El algoritmo de aprendizaje fue realizado 30 veces usando 50 generaciones y 50 individuos

[2]http://gplab.sourceforge.net/index.html. Consultado Junio 2010.
[3]http://www.vlfeat.org/ vedaldi/code/sift.html. Consultado Agosto 2010.

Tabla IV. Resultados de los cinco mejores operadores RDGP's

 Región sin ninguna operación			 Magnitud del gradiente del SIFT	
Descriptor	**Aptitud**	**Individuo**	**Expresión Matemática**	**Imagen**
$RDGP_1$	7.4158	$sqrt(sqrt(D_x(sqrt(D_x(sqrt(D_{xy}(image)))))))$	$\sqrt{\sqrt{D_x\left(\sqrt{D_x\left(\sqrt{D_{xy}(I)}\right)}\right)}}$	
$RDGP_2$	7.4859	$sqrt(D_x(sqrt(substract(sqrt(D_{xy}(image)), D_{xx}(image)))))$	$\sqrt{D_x\left(\sqrt{\sqrt{D_{xy}(I)}-D_{xx}(I)}\right)}$	
$RDGP_4$	7.3928	$Gauss2(absdif(Gauss2(absdif(absdif(D_y(imagen), D_x(D_x(imagen))), D_y(Logarithm(D_x(D_x(imagen))))), Half(D_x(D_y(imagen)))))$	$G_{\sigma=2}\|G_{\sigma=2}(\|\ D_y(I) - D_{xx}(I)\ \| - D_y(\log(D_{xx}(I)))\ \|) - \frac{D_{xx}(I)}{2}\ \|$	
$RDGP_5$	7.4053	$Gauss1(sqrt(Gauss2(sqrt(sqrt(substract(sqrt(Gauss1(D_y(image))), divide(D_{xx}(image), absadd(D_x(image), D_y(image))))))))$	$G_{\sigma=1}\sqrt{G_{\sigma=2}\sqrt{\sqrt{G_{\sigma=1}(D_y(I))}-\frac{D_{xx}(I)}{\|D_x(I)+D_y(I)\|}}}$	
$RDGP_{11}$	7.3736	$Half(G_2(G_2(sqrt(D_{xx}(Log(D_{xy}(image)))))))$	$\frac{G_{\sigma=2}(G_{\sigma=2}\sqrt{(D_{xx}(log_2(D_{xy}(I))))}}{2}$	

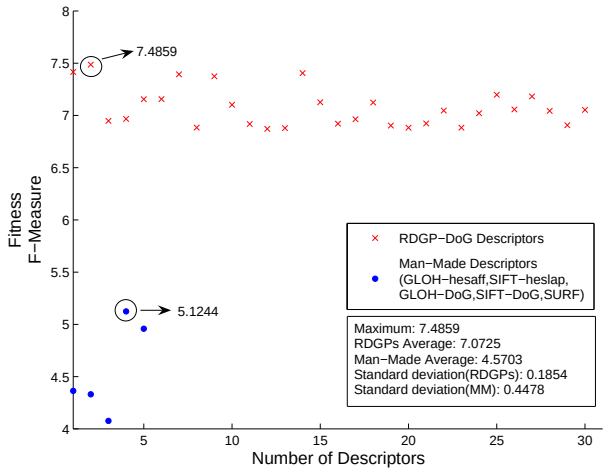


Figura 34. Gráfica que muestra el rendimiento de los 30 descriptores RDGP's y de 5 descriptores diseñados por el ser humano.

para cada corrida. Como resultado de ello, obtuvimos 30 operadores descriptivos que fueron los mejores por cada experimento. La Tabla III.6.1 muestra los 30 mejores operadores de cada experimento usando la notación prefija, y su valor correspondiente de aptitud. De estos 30 operadores seleccionamos el mejor de ellos, $RDGP_2$ para utilizarlo en las pruebas de evaluación y para reconocer objetos. Por otro lado, la Figura 34 muestra que nuestros 30 descriptores que fueron obtenidos automáticamente por la evolución son mejores que los tres descriptores diseñados por el hombre de acuerdo con el valor obtenido de la medida F. Estos valores corresponden a la evaluación de los descriptores usando el par de imágenes boat, ver Figura 27(c). Para esta prueba en particular, nuestro mejor descriptor ($RDGP_2$) logra un valor de 7.4859 mientras que el mejor descriptor diseñado por el hombre tiene un valor de 5.1244. La Tabla III.6.1 muestra los cinco mejores operadores descriptivos obtenidos de los 30 experimentos. Esta tabla presenta cada individuo y su expresión matemática junto con su valor de aptitud. Además, también ilustra la imagen que corresponde a cada fórmula del RDGP. Podemos observar en esta tabla que el $RDGP_1$ y $RDGP_2$ tienen mas detalles de la imagen original, mientras que el $RDGP_7$ y $RDGP_9$ son muy similares a la magnitud pesada del gradiente del SIFT. Por otro lado, el $RDGP_{13}$ no es humanamente interpretable. Finalmente, se puede observar claramente que la mayoría de los operadores mostrados en esta

tabla usan la raiz cuadrada como una de sus operaciones básicas, sucediendo algo similar con los demás operadores evolucionados. Lo curioso de este suceso, es que para la optimización de detectores de puntos de interés propuesto por Trujillo y Olague (2006b) tal operación nunca fue utilizada por el algoritmo.

Como ejemplo de una corrida típica del algoritmo, presentamos cuatro gráficas estadísticas que describen el proceso para sintetizar el operador $RDGP_2$. La Figura 35 ilustra cuatro gráficas, una de ellas muestra los valores de la mediana, el promedio y el valor máximo obtenido de la aptitud del operador $RDGP_2$ y las demás muestran la diversidad de la población, la evolución de la complejidad de la estructura de árbol; y finalmente, las variaciones del uso de las operaciones de cruzamiento y mutación. La gráfica de la diversidad de la población ilustra el porcentaje de los individuos únicos de cada generación; mientras que la complejidad estructural muestra los parámetros relacionados con el tamaño del árbol. Además, la Figura 36 presenta la evolución de la aptitud y la estructura de árbol correspondiente para los operadores $RDGP_1$, $RDGP_7$, $RDGP_9$ y $RDGP_{13}$. Podemos observar que los operadores $RDGP_1$ y $RDGP_7$ alcanzan su valor máximo de aptitud en las últimas generaciones; mientras que el operador $RDGP_1$ y $RDGP_{13}$ lo alcanzan al final de la 30^{va} generación. Por último, la Figura 37 muestra cómo fue usado el operador de cruzamiento para crear el operador $RDGP_2$ en la última generación.

Tabla V: Resultados del Entrenamiento de los RDGP's.

Descriptor	**Aptitud**	**Expresión del Individuo**	**Descriptor**	**Aptitud**	**Expresión del Individuo**
$RDGP_1$	7.4158	$sqrt(sqrt(D_x(sqrt(D_x(sqrt(D_{xy}($ $image)))))))$	$RDGP_2$	7.4859	$sqrt(D_x(sqrt(dif(sqrt(D_{xy}($ $image)),D_{xx}(image)))))$
$RDGP_3$	7.1812	$G_2(G_2(sqrt(D_x(D_y(D_x(D_x($ $image)))))))$	$RDGP_4$	6.9666	$sqrt(div(absadd(D_{yy}(image)),D_{xx}($ $image))),G_1(Half(D_{xx}(image))))))$
$RDGP_5$	7.1557	$sqrt(dif(D_{xx}(image)),D_y(Log($ $sqrt(D_{xx}(image)))))))$	$RDGP_6$	6.9470	$sqrt(dif(sqrt(sqrt(dif(D_{xy}($ $image)),sqrt(D_{xy}(image))))))$ $,sqrt(D_x(image))))$
$RDGP_7$	7.3928	$G_2(absdif(G_2(absdif(absdif(D_y$ $(image),D_{xx}(image))),D_y(Log(D_x($ $D_x(image))))))),Half(D_x(D_y(image)))))$	$RDGP_8$	7.1544	$sqrt(sqrt(D_x(G_2(D_{xy}(image))))))$
$RDGP_9$	7.4053	$G_1(sqrt(G_2(sqrt(sqrt(dif(sqrt(G_1($ $D_y(image))),div(D_{xx}(image)),$ $absadd(D_x(image),D_y(image)))))))))$	$RDGP_{10}$	6.8833	$G_1(dif(sqrt(sqrt(Log(D_{xx}(image$ $))))),Log(G_2(D_{xy}(image))))))$
$RDGP_{11}$	6.9170	$dif(absdif(D_{xy}(image)),div(Half($ $Half(D_y(image))))),D_{xy}(image)))$ $D_x(image)),absadd(D_x(image),$	$RDGP_{12}$	7.1020	$sqrt(dif(sqrt(D_{xy}(image))),D_y(Log($ $D_{xx}(image))))))$
$RDGP_{13}$	7.3736	$Half(G_2(G_2(sqrt(D_{xx}(Log(D_{xy}($ $image)))))))))$	$RDGP_{14}$	6.8704	$absadd(Half(D_{yy}(image))),Log(absadd($ $sqrt(D_{yy}(image))),D_{xy}(D_{yy}(image)))))))$

Continuación en la siguiente página

Tabla V – Resultados del Entrenamiento de los RDGP's. (Continuación)

Descriptor	**Aptitud**	**Expresión del Individuo**	**Descriptor**	**Aptitud**	**Expresión del Individuo**
$RDGP_{15}$	6.9203	$absadd(absadd(Half(Log($ $D_{yy}(image))))),Log(Half(Log($ $D_x(image)))))),Half(D_{yy}(image))))$	$RDGP_{16}$	7.1264	$sqrt(div(dif(D_{xx}(image)),D_x(image)$ $),absadd(absadd(D_{yy}(D_y(image))),$ $D_{yy}(image))),D_x(image))))$
$RDGP_{17}$	6.8778	$sqrt(D_y(sqrt(Log(Log(D_{xx}($ $image)))))))$	$RDGP_{18}$	6.9633	$absdif(sqrt(G_2(sqrt(sqrt(D_{xx}($ $image))))))),D_{xx}(image)))$
$RDGP_{19}$	7.1225	$sqrt(D_y(add(D_{xy}(image)),Log($ $D_{xx}(image))))))$	$RDGP_{20}$	6.9024	$div(Log(D_{yy}(image))),Log(G_1($ $D_x(G_2(D_x(image))))))$
$RDGP_{21}$	6.8824	$Log(Log(G_2(D_{xy}(image)))))$	$RDGP_{22}$	6.9230	$absdif(D_x(image),sqrt(G_2($ $sqrt(D_{xx}(image))))))$
$RDGP_{23}$	7.0466	$absadd(D_{yy}(image)),sqrt(add($ $D_{yy}(image)),D_{xx}(D_x(image))))))$	$RDGP_{24}$	6.8824	$Log(Log(G_2(D_{xy}(image)))))$
$RDGP_{25}$	7.0207	$absdif(sqrt(absadd(absdif(sqrt($ $D_{xy}(image))),sqrt(D_{yy}(image)))),$ $D_{xx}(D_y(image))))),D_{xx}(image)))$	$RDGP_{26}$	7.1978	$sqrt(absdif(dif(D_y(Log(D_{xx}($ $image)))),D_{yx}(image))),G_2(D_x($ $G_1(D_{xx}(image)))))))$
$RDGP_{27}$	7.0570	$sqrt(sqrt(absdif(D_x(image),sqrt($ $D_y(image)))))$	$RDGP_{28}$	7.0433	$sqrt(dif(sqrt(G_2(absadd(D_{xx}(image)),$ $D_{yy}(D_{xx}(image)))))),D_y(image)))$
$RDGP_{29}$	6.9063	$Half(absadd(absadd(Half(sqrt($ $Log(D_{xy}(image))))),D_y(image)),$ $absdif(sqrt(D_{xy}(image))),G_2($ $absadd(D_{xy}(image)),D_{xx}(image)))))))$	$RDGP_{30}$	7.0529	$dif(sqrt(Log(div(D_{xx}(image)),$ $D_{xy}(image))))),D_{yy}(image)))$

III.6.2 Evaluación experimental de descriptores locales

La comparación de descriptores locales consiste en evaluar nuestro descriptor $RDGP_2$ con tres de los mejores descriptores del estado del arte como: SIFT, GLOH y SURF. Para ello, se usaron varios pares de imágenes con diferentes transformaciones, asi como también, diferentes detectores para tener una comparación más confiable. Los resultados son mostrados en la Figura 38 y 39. Los descriptores SIFT y GLOH fueron incluídos en nuestras pruebas debido a que estos dos descriptores fueron los mejores de una evaluación exhaustiva realizada por Mikolajczyk y Schmid (2005), mientras que SURF es considerado como uno de los descriptores más rápidos usado para aplicaciones reales, Bay *et al.* (2006b). Además, cualquier otro descriptor puede ser fácilmente comparado con el de nosotros si usan el mismo proceso de evaluación propuesto por Mikolajczyk y Schmid (2005).

La Figura 38 muestra los resultados al utilizar imágenes con rotación, cambios de iluminación, y compresión JPEG. En el caso de rotación, nuestro descriptor $RDGP_2$ obtiene muy buenos resultados mejorando hasta en un 45.5% al segundo mejor descriptor que fue el descriptor GLOH utilizando el detector DOG. De la misma manera, en el caso de la iluminación nuestro descriptor $RDGP_2$ obtiene el mejor rendimiento seguido del descriptor GLOH-DoG quien obtuvo 5.3919 al evaluarlo con la medida F. Finalmente, en el caso de la compresión JPEG, el $RDGP_2$ obtiene un 6.1477 en su rendimiento mientras que el SURF como segundo mejor descriptor para esta prueba, obtiene un 5.0078. Por otro lado, La Figura 39(d) muestra los resultados sobre las imágenes de entrenamiento, donde el detector DOG detecta un número bajo de correspondencias. El algoritmo $RDGP_2 - DOG$ y $GLOH - DOG$ alcanzan un valor más alto, mientras que $SIFT - DOG$ tiene menos puntuación que $SIFT - heslap$. Después, probamos con las imágenes Bark que también incluyen rotación y escalamiento. En este caso, $GLOH - heslap$ mejora significativamente a $GLOH - DOG$. Sin embargo, nuestro descriptor fue el que mejor desempeño tuvo. Por otro lado, cuando existe solamente rotación, observamos que $GLOH - DOG$ es mejor que el $SIFT - DOG$; sin embargo, $GLOH - hesaff$ obtiene un valor de la medida F más bajo. En el caso cuando existe cambios de iluminación,

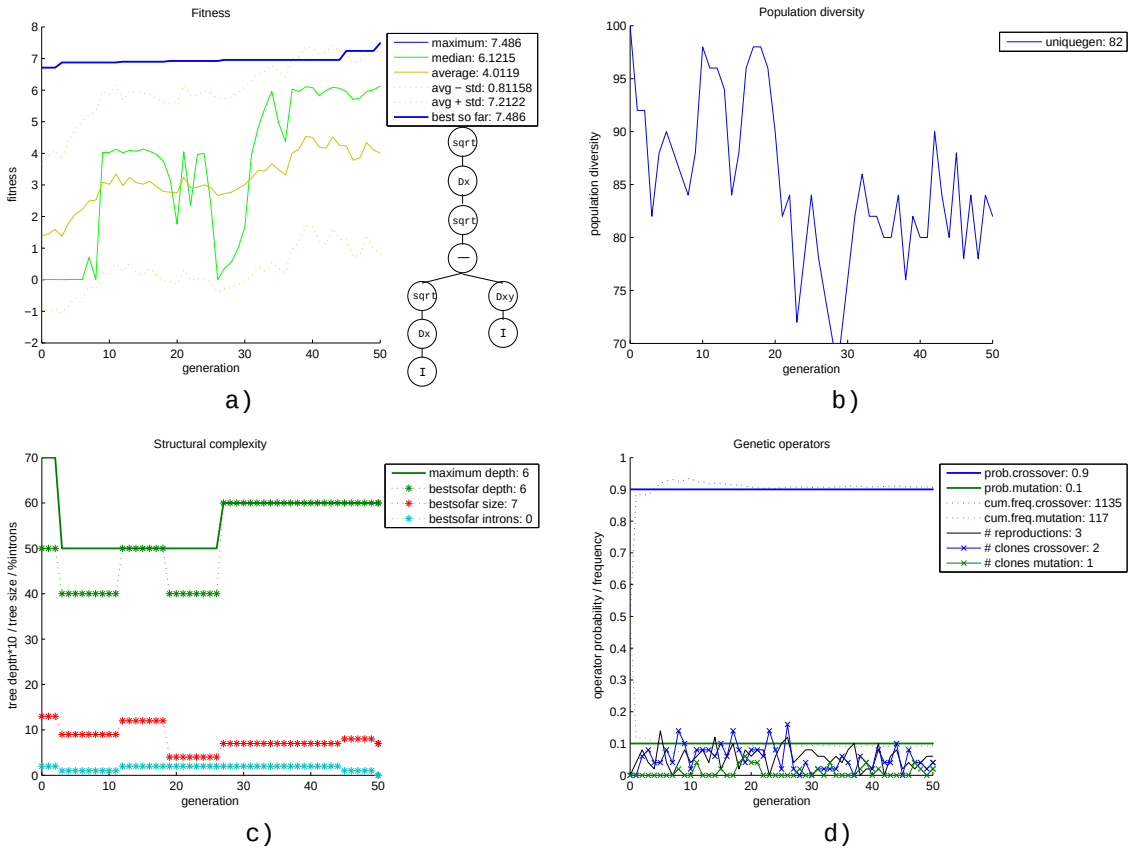


Figura 35. Gráficas de la Evolución del $RDGP_2$. a) Gráfica de la aptitud y su representación de árbol. b) Diversidad de la población durante 50 generaciones. c) Complejidad de la estructura de árbol durante el proceso evolutivo. d) Variaciones de los métodos de mutación y cruzamiento.

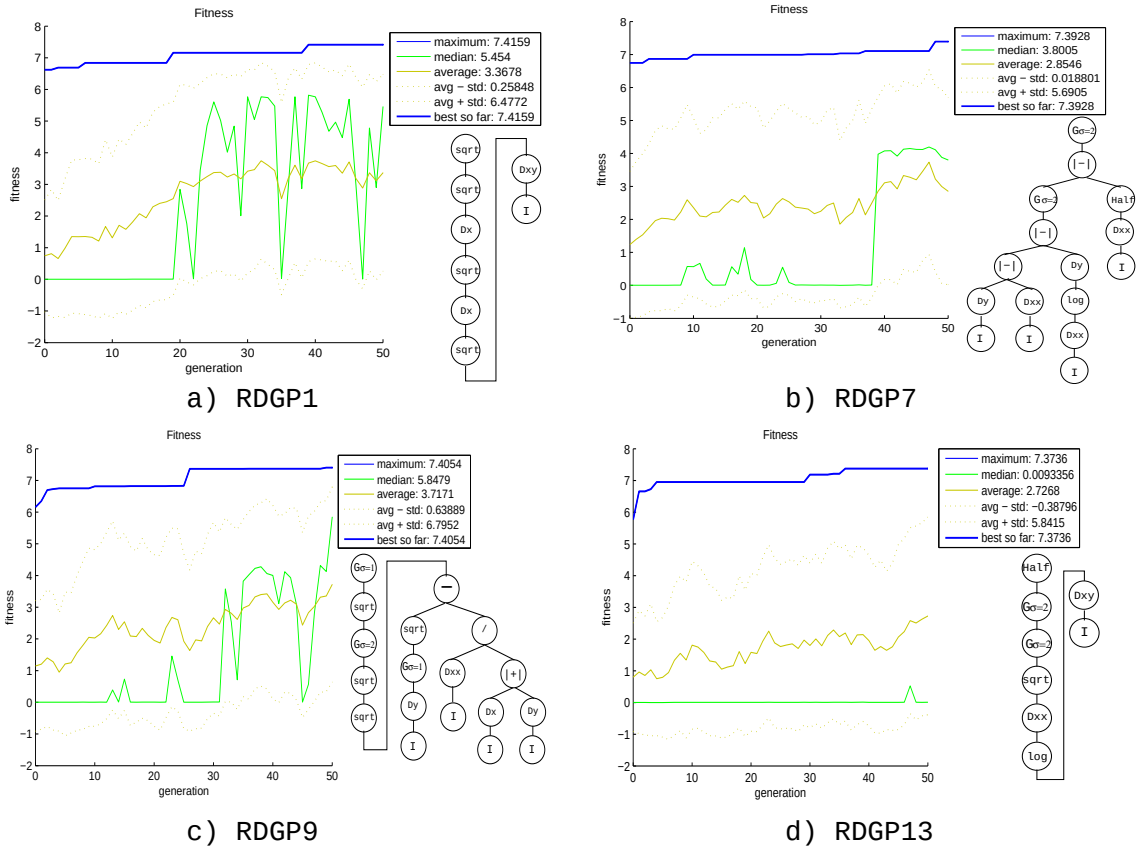


Figura 36. Gráficas de la aptitud del $RDGP_1$, $RDGP_3$, $RDGP_4$ y $RDGP_5$ junto con su representación de árbol

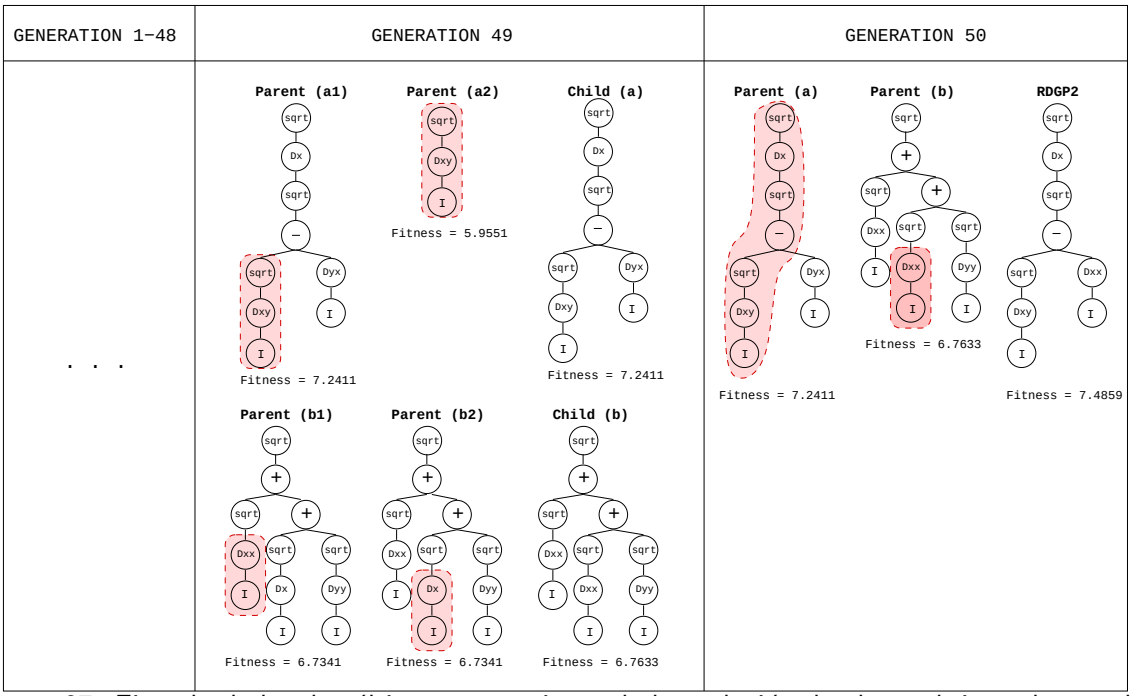


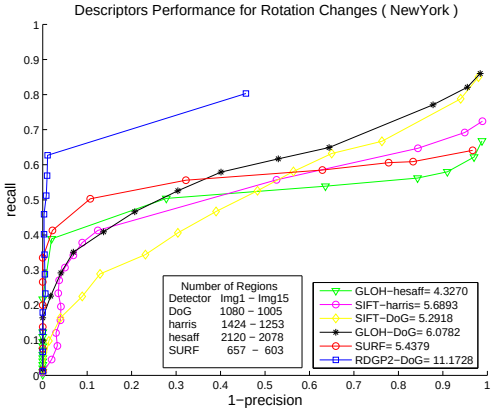
Figura 37. Ejemplo de las dos últimas generaciones de la evolución donde se obtiene el operador del descriptor $RDGP_2$; el hijo (a) y el hijo (b) corresponden a los padres del $RDGP_2$.

$GLOH-haraff$ y $SIFT-hesaff$ logran obtener menor valor de rendimiento con respecto a sus versiones usando DOG. Entonces, podemos decir que para este tipo de transformaciones, DOG representa una buena opción.

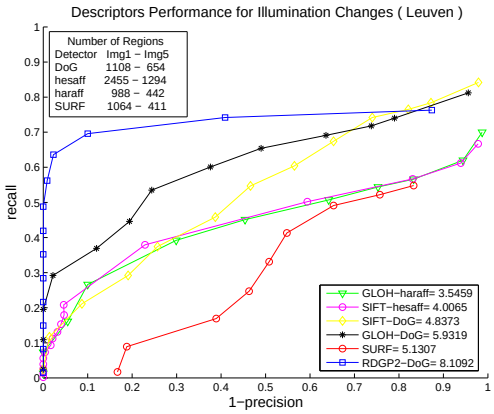
La Figura 39(f) y (g) presenta los resultados de la prueba considerando cambios de difuminación. Aquí, $hesaff$ y $haraff$ obtienen un bajo número de correspondencias; sin embargo, el rendimiento alcanzado por sus descriptores es mayor que aquellos que usan DOG, a excepción del $RDGP_2$. La Figura 39(g) provee resultados similares con el $SURF$ como el segundo mejor descriptor. De nueva cuenta, para este tipo de transformaciones es posible que nuestro descriptor pueda ser aún mejor si usáramos un detector más adecuado para este tipo de transformaciones. En el caso de la compresión JPEG, todos los detectores dan resultados similares para la correspondencia de descriptores; sin embargo, $GLOH-haraff$ y $SIFT-hesaff$ obtienen un valor de la medida F más alto que sus correspondientes versiones con DOG. Para este caso, $SURF$ es nuevamente el segundo mejor descriptor. Hasta ahora, todas las pruebas muestran que nuestro descriptor $RDGP_2$ es el mejor descriptor para ese tipo de transformaciones. Sin embargo, para el caso de las imágenes con transformaciones afines, Figura 39(h) y (i), los descriptores $GLOH-hesaff$, $SIFT-hesaff$ y SURF son los que obtienen el mejor resultado. Es importante remarcar, que para este caso, aún cuando nuestro descriptor utiliza el detector DOG es mejor que las versiones de estos descriptores usando el mismo detector. Por lo tanto, si nosotros utilizamos un detector más adecuado para este tipo de transformaciones, el rendimiento del $RDGP_2$ mejoraría considerablemente.

La Tabla VI presenta un resumen del rendimiento de los descriptores usando la medida F, donde el $RDGP_2$ mejoró acerca de un 33.82% para el caso de la rotación, 21.66% para rotación y escala, 26.85% para la iluminación, 26.43% para difuminación y 18.54 para la compresión JPEG; mientras que para el caso de la transformación afín, los descriptores GLOH y SIFT fueron mejores como mencionamos anteriormente, obteniendo un 10.62% de diferencia con respecto a los demás.

a) NewYork (Rotación)



b) Leuven (Iluminación)



c) UBC (Compresión JPEG)

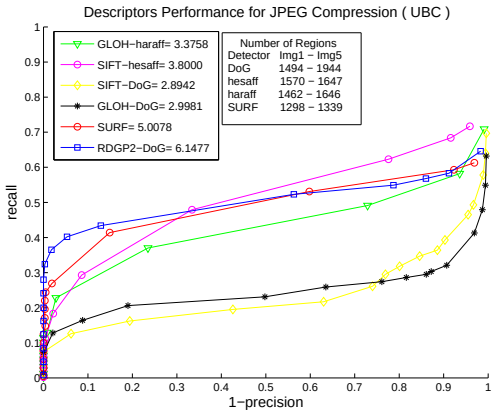
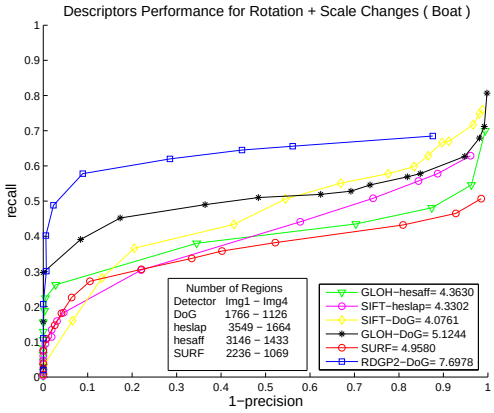
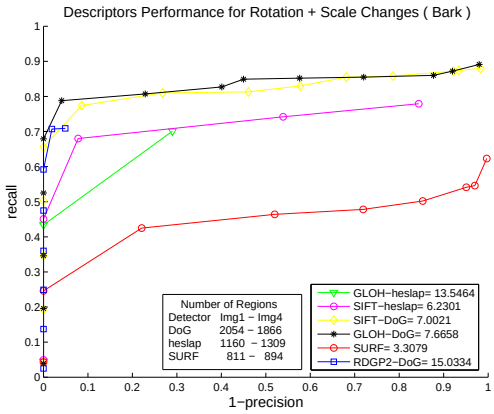


Figura 38. Evaluación de los descriptores $RDGP_2$, SIFT, GLOH y SURF en diferentes tipos de transformación de la imagen como lo es rotación, iluminación y compresión JPEG.

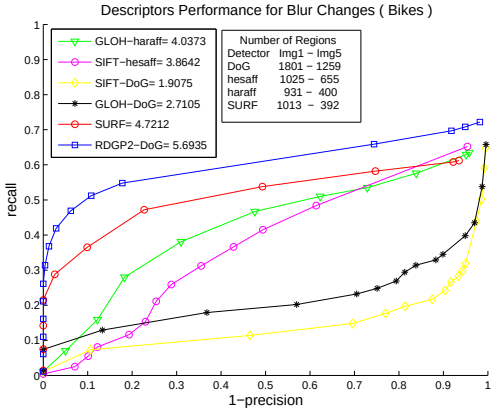
d) Boat (Rotación + Escalamiento)



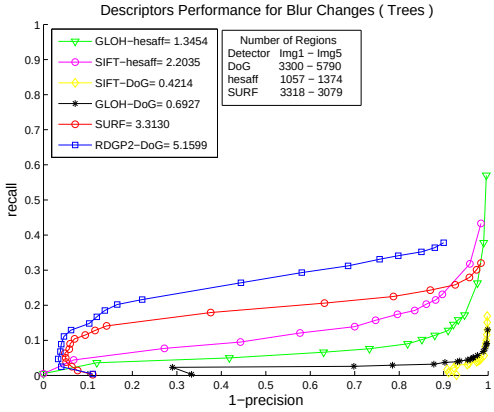
e) Bark (Rotación + Escalamiento)



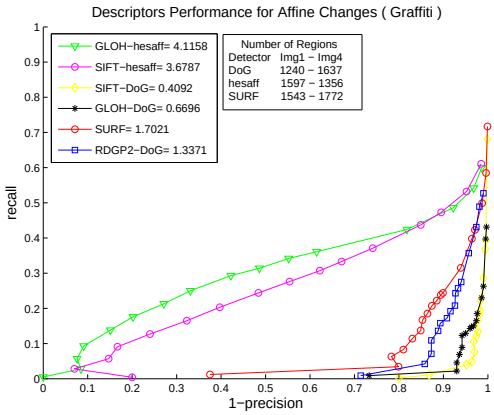
f) Bikes (Difuminación)



g) Trees (Difuminación)



h) Graffiti (Afín)



i) Wall (Afín)

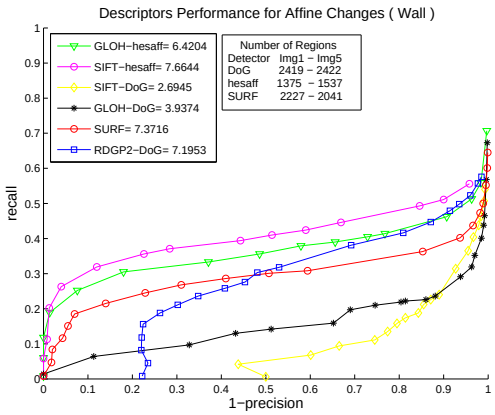


Figura 39. Evaluación de los descriptores $RDGP_2$, SIFT, GLOH y SURF en diferentes tipos de transformación de la imagen como lo es rotación y escalamiento, difuminación y transformación afín.

Tabla VI. Evaluación del rendimiento de los descriptores usando la medida F.

	Medida F								
Descriptor	**NewYork**	**Leuven**	**Boat**	**Bark**	**Bikes**	**Trees**	**UBC**	**Graffiti**	**Wall**
GLOH (hessaff/ haraff/heslap)	4.3270	3.5459	4.3630	**13.5464**	4.0373	1.3454	3.3758	**4.1158**	6.4204
SIFT (harris/ heslap/hesaff)	5.6893	4.0065	4.3302	6.2301	3.8642	2.2035	3.8000	**3.6787**	**7.6644**
SIFT-DoG	5.2918	4.8373	4.0761	7.0021	1.9075	0.4214	2.8942	0.4092	2.6945
GLOH-DoG	**6.0782**	**5.9319**	**5.1244**	7.6658	2.7105	0.6927	2.9981	0.6696	3.9374
SURF	5.4379	5.1307	4.9580	3.3079	**4.7212**	**3.3130**	**5.0078**	1.7021	**7.3716**
$RDGP_2$-DoG	**11.1728**	**8.1092**	**7.6978**	**15.0334**	**5.6935**	**5.1599**	**6.1477**	1.3371	7.1953
Diferencia (%) entre los dos mejores descrip.	**45.59**% ó **83.82**%	**26.85**% ó **36.70**%	**33.43**% ó **50.22**%	**9.89**% ó **10.98**%	**17.08**% ó **20.59**%	**35.79**% ó **55.75**%	**18.54**% ó **22.76**%	**10.62**% ó **11.88**%	**3.8202**% ó **3.9720**%

III.6.3 Reconocimiento de objetos en interiores y exteriores

En esta Sección, describiremos el reconocimiento de objetos similar al propuesto por Lowe (2004a). El objetivo es mostrar el rendimiento de nuestro descriptor $RDGP_2$ en una aplicación real tanto para escenas en interiores como en exteriores. La prueba consiste de un conjunto de fotografías adquiridas con una cámara digital SONY Cyber-shot 12.1MP DSC-W230. Para realizar el reconocimiento de un objeto, seleccionamos dos imágenes, una que es considerada como la base o el objeto a identificar y la segunda, la que representa la escena donde se encuentra el objeto con alguna transformación. Después, los descriptores de las dos imágenes son calculados para luego, corresponderlos con un indexamiento eficiente de vecino mas cercano llamado Best Bin First propuesto por Beis y Lowe (1997). Además, adicionalmente a este proceso de reconocimiento, nosotros calculamos la geometría epipolar usando el algoritmo RANSAC (Random SAmple Consensus) para identificar las correspondencias correctas (inliers) y falsas (outliers).

Experimento I

En este primer experimento, usamos fotografías para objetos con textura y sin textura localizado en escenarios de interiores y exteriores, ver Figura 40. De esta manera, podemos apreciar que la Figura 41 ilustra el reconocimiento de estos objetos utilizando el algoritmo SIFT y el $RDGP_2$ donde las lineas verdes representan las correspondencias correctas mientras que las rojas, las correspondencias falsas. Es fácil observar que el número total de correspondencias falsas es significativamente reducido por nuestro algoritmo conservando la mayoría de las correspondencias correctas. La Tabla VII provee los porcentajes de error de estas correspondencias llevadas a cabo por los dos descriptores donde el descriptor $RDGP_2$ disminuye las correspondencias falsas. Además, al observar este comportamiento en el reconocimiento podemos ver que nuestro descriptor sirve como una especie de filtro en la correspondencia de datos.

Tabla VII. Errores de la correspondencia de los descriptores $RDGP_2$ y SIFT.

	ETAPA DE DETECCIÓN		ETAPA DE CORRESPONDENCIA			
Descriptor	Imagen del Objecto	Imagen de la escena	Total	Correctos	Incorrectos	Error (%)
Book-Cork						
$RDGP_2$	1588	1418	83	50	33	**39.75 %**
SIFT			112	55	57	**50.89 %**
Book-Objects						
$RDGP_2$	1588	2044	255	214	41	**16.08 %**
SIFT			318	240	78	**24.53 %**
Book-Books						
$RDGP_2$	1588	1576	169	146	23	**13.61 %**
SIFT			253	167	86	**33.99 %**
Mask-Objects						
$RDGP_2$	1375	2044	91	73	8	**34.78 %**
SIFT			138	84	40	**68.96%**
Pedestrian-UABC						
$RDGP_2$	189	1579	23	15	18	**62.06 %**
SIFT			58	18	52	**81.25 %**
Stop-Costero Blvd.						
$RDGP_2$	419	1603	29	11	18	**19.78 %**
SIFT			64	12	54	**39.13 %**

Figura 40. Fotografías utilizadas para el reconocimiento de objetos.

Experimento II

Para este experimento utilizamos un conjunto de fotografías más amplio. En el caso del reconocimiento en interiores, adquirimos ocho fotografías de las cuales tres son para objetos y el resto para las escenas, ver Figura 42; mientras que para el reconocimiento en exteriores coleccionamos 150 fotografías, las cuales organizamos en cuatro categorías: BOAT con 52 fotos, CEARTE con 34, HUSSONGS con 28 y PAPAS con 36, ver Figura 44 como ejemplo de algunas de estas fotografías.

Reconocimiento de Objetos en Interiores

En este experimento observamos que para el caso de objetos en interiores el $RDGP_2$ sigue siendo mejor que el algoritmo SIFT en el proceso de correspondencia, tal como lo vimos en el primer experimento. La Figura 43 ilustra el reconocimiento realizado en interiores donde las lineas verdes representan la correspondencia de descriptores que fueron detectadas correctamente; mientras que las rojas representan las falsas correspondencias de este conjunto de descriptores. Por otro lado, la Tabla 43 presenta los errores ocasionados por ambos algoritmos

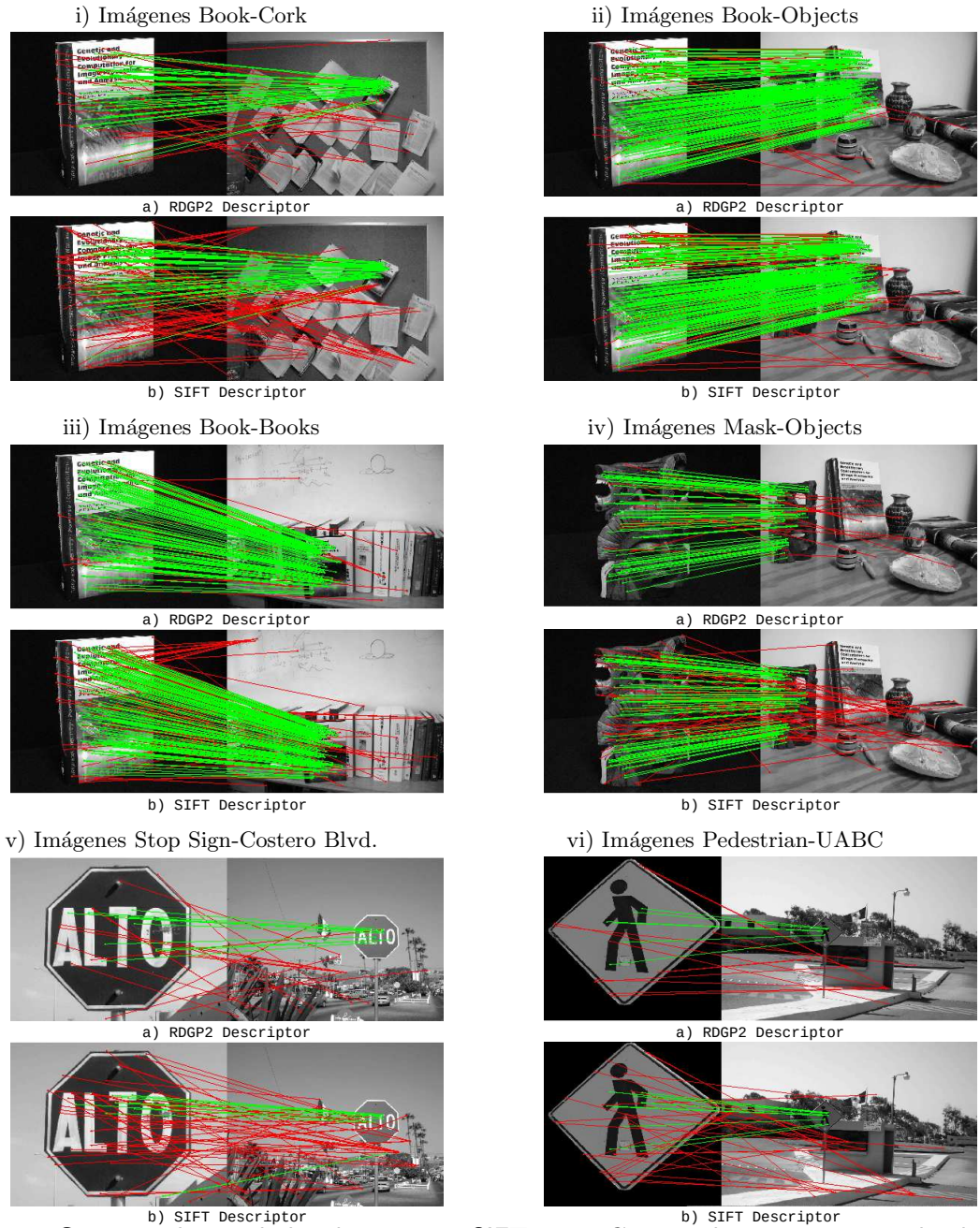


Figura 41. Correspondencia de los descriptores SIFT y $RDGP_2$ en el reconocimiento de objetos en interiores y exteriores.

Figura 42. Conjunto de imágenes usadas para el reconocimiento en interiores.

en la etapa de correspondencia de datos, donde apreciamos que nuestro algoritmo produce menos correspondencias incorrectas que el algoritmo SIFT durante las seis pruebas. Por lo tanto, el algoritmo $RDGP_2$ mejora alrededor de un 20% las correspondencias de descriptores; exceptuando el caso de la prueba con las imágenes de Jessie que logró obtener alrededor de un 10% de mejoría.

Reconocimiento de Objetos en Exteriores

En este experimento, utilizamos 150 fotografías de sitios turísticos en la ciudad de Ensenada, B.C., México. Estas fotografías las organizamos en cuatro categorías: BOAT con 52 fotos, CEARTE con 34, HUSSONGS con 28 y PAPAS con 36. La Figura 44 ilustra las imágenes más representativas del conjunto de fotografías de sitios turísticos. Al llevar a cabo el reconocimiento de todas estas imágenes, resumimos los porcertantes de error de la correspondencia de descriptores en la Tabla VIII mediante la cual podemos observar que nuestro algoritmo sigue produciendo menos errores en la etapa de correspondencia. La Figura 45

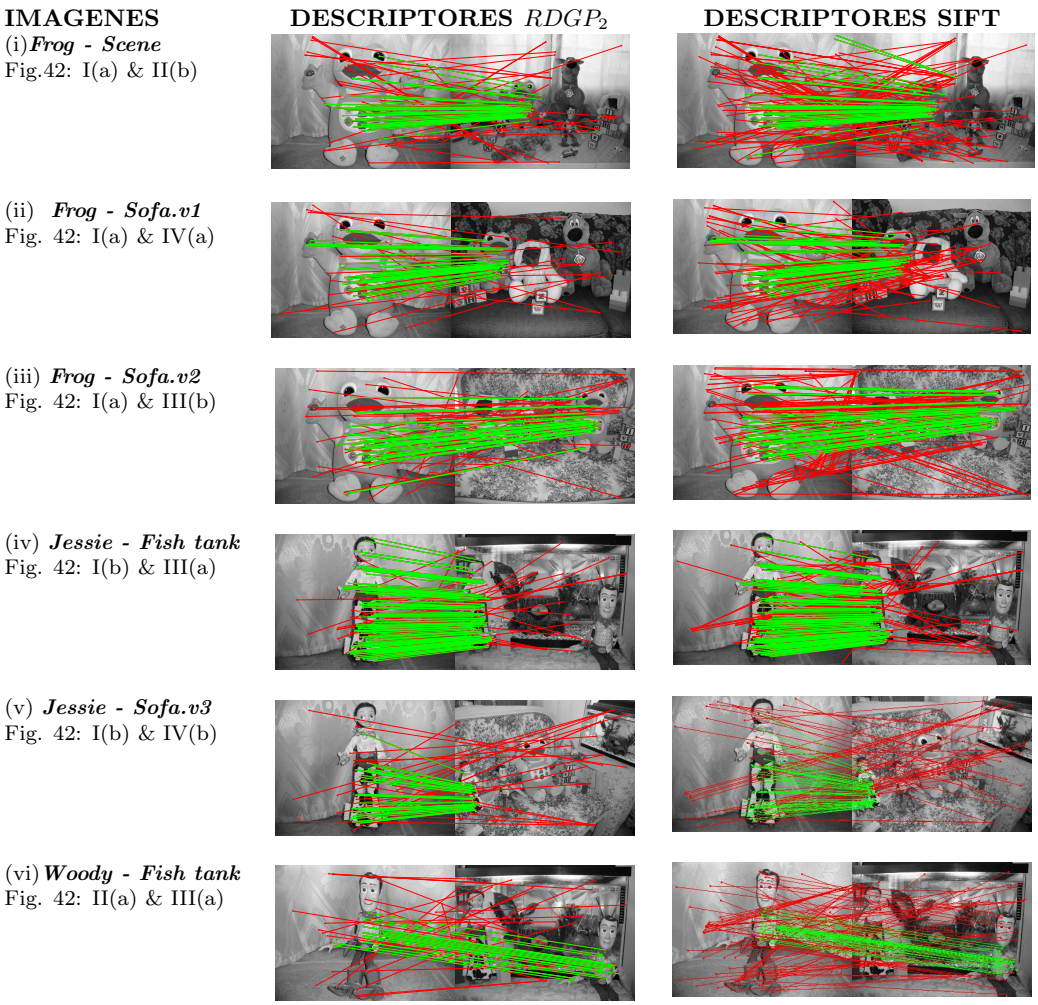


Figura 43. Correspondencia de descriptores para escenarios en interiores.

muestra la correspondencia de descriptores del par de imágenes con las cuales se obtuvieron el mínimo y máximo error durante el proceso de reconocimiento. En este caso, es fácil observar que el máximo error que obtuvimos fue producido debido a la complejidad de la escena; por ejemplo, localizar el barco mostrado en la Figura 44 I(a) con II(a) es una tarea díficil ya que solo se muestra solamente la parte trasera del barco en la segunda imagen. Por otro lado, también es difícil el reconocimiento para las imágenes de CEARTE y PAPAS mostradas en la Figura 44 III(b) y III(d) porque presentan mucha distorción para llevar a cabo el reconocimiento con las imágenes tomadas como base de la Figura 44 I(b) y I(d). Sin embargo, en todos los casos nuestro algoritmo $RDGP_2$ produce un menor nḿero de falsas correspondencias en el proceso de reconocimiento.

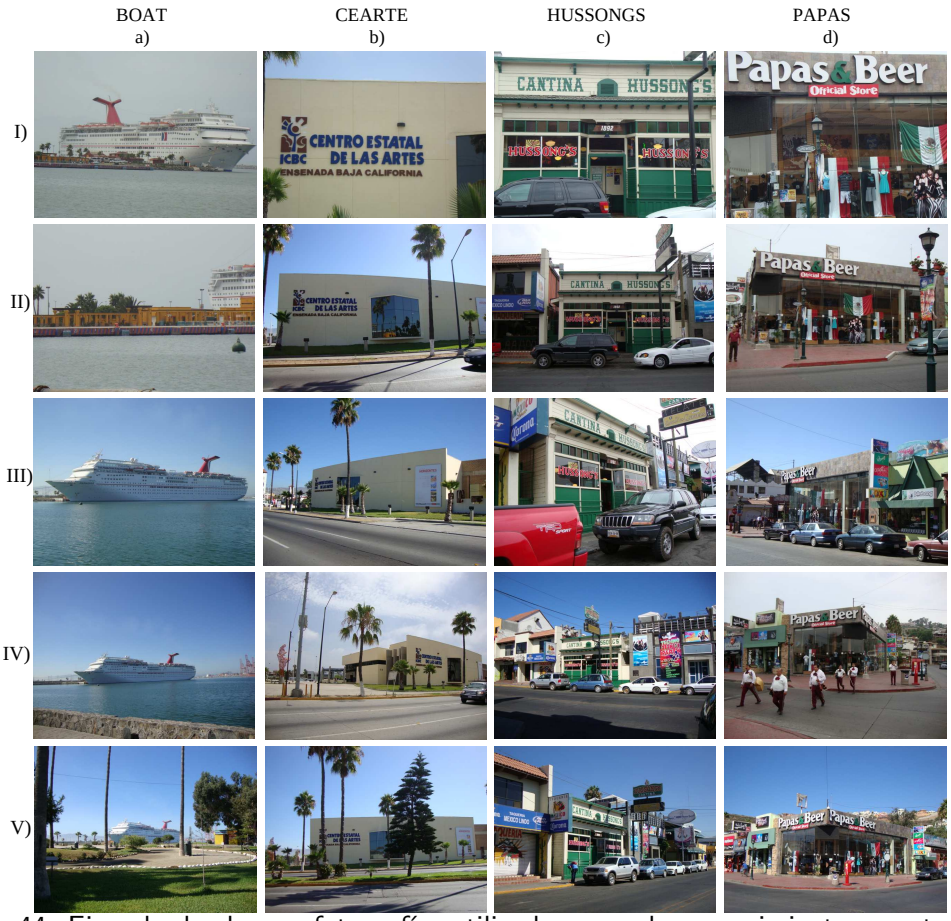


Figura 44. Ejemplo de algunas fotografías utilizadas para el reconocimiento en exteriores.

Tabla VIII. Error en la correspondencia de descriptores entre los algoritmos $RDGP_2$ y SIFT.

	ERROR EN LA CORRESPONDENCIA (%)				Regiones Detectadas	
Descriptor	**Promedio**	**Mediana**	**Mínimo**	**Máximo**	**Img1**	**Img2** (promedio)
BOAT (52)						
$RDGP_2$	35.61 %	28.53 %	11.91 %	84.26 %	1806	1437
SIFT	41.70 %	36.32 %	15.12 %	89.90 %	”	”
CEARTE (34)						
$RDGP_2$	32.64 %	21.97 %	6.45 %	78.13 %	924	1637
SIFT	42.05 %	36.59 %	10.91 %	86.15 %	”	”
HUSSONGS (28)						
$RDGP_2$	43.38 %	37.25 %	12.50 %	71.15 %	2196	2536
SIFT	55.40 %	54.06 %	19.35 %	82.49 %	”	”
PAPAS (36)						
$RDGP_2$	55.26 %	54.89 %	9.63 %	83.02 %	2966	2659
SIFT	67.76 %	69.89 %	17.62 %	94.66 %	”	”

III.7 Discusión de Resultados

En esta Capítulo, describimos un nuevo enfoque evolutivo para el diseño automatizado de operadores descriptivos para el descriptor SIFT, los cuales llamamos $RDGP's$. Nuestra metodología propuesta para sintetizar operadores representa una mejora signitificativa sobre el algoritmo SIFT el cual fue patentado por Lowe (2004b). El objetivo principal de esta investigación fue encontrar de manera automática un conjunto de expresiones matemáticas producidas por la programación genética que fueran igual o mejores que la magnitud del gradiente del descriptor SIFT. Para ello, propusimos llamar a estas expresiones matemáticas como operadores descriptivos. Además, en nuestro trabajo consideramos dos protocolos: 1) Usamos una prueba de evaluación estándar basada en la correspondencia de regiones de interés con el fin de evaluar el rendimiento de nuestro mejor operador evolucionado ($RDGP_2$) y tres descriptores del estado del arte (SIFT, GLOH, SURF) para lo cual nuestro descriptor superó significativamente a los otros descriptores en las diferentes pruebas. 2) Nuestro mejor operador fue probado en una aplicación real, mostrando que el número total de correspondencias falsas en el proceso de reconocimiento es reducido en gran medida. De esta manera, pensamos que esta propuesta puede ser implementada fácilmente para los descriptores inspirados en el SIFT o similares donde un operador matemático es aplicado para describir las regiones de interés. En realidad, para llevar a cabo el primer protocolo en nuestro enfoque evolutivo, propusimos incluir la medida F en el proceso de evaluación para obtener no solamente un resultado gráfico sino más bien, una medida cuantitativa como lo requiere el proceso de optimización utilizando la programación genética. Finalmente, nuestra propuesta abre un nuevo camino en la investigación sobre el aprendizaje evolutivo de descriptores locales.

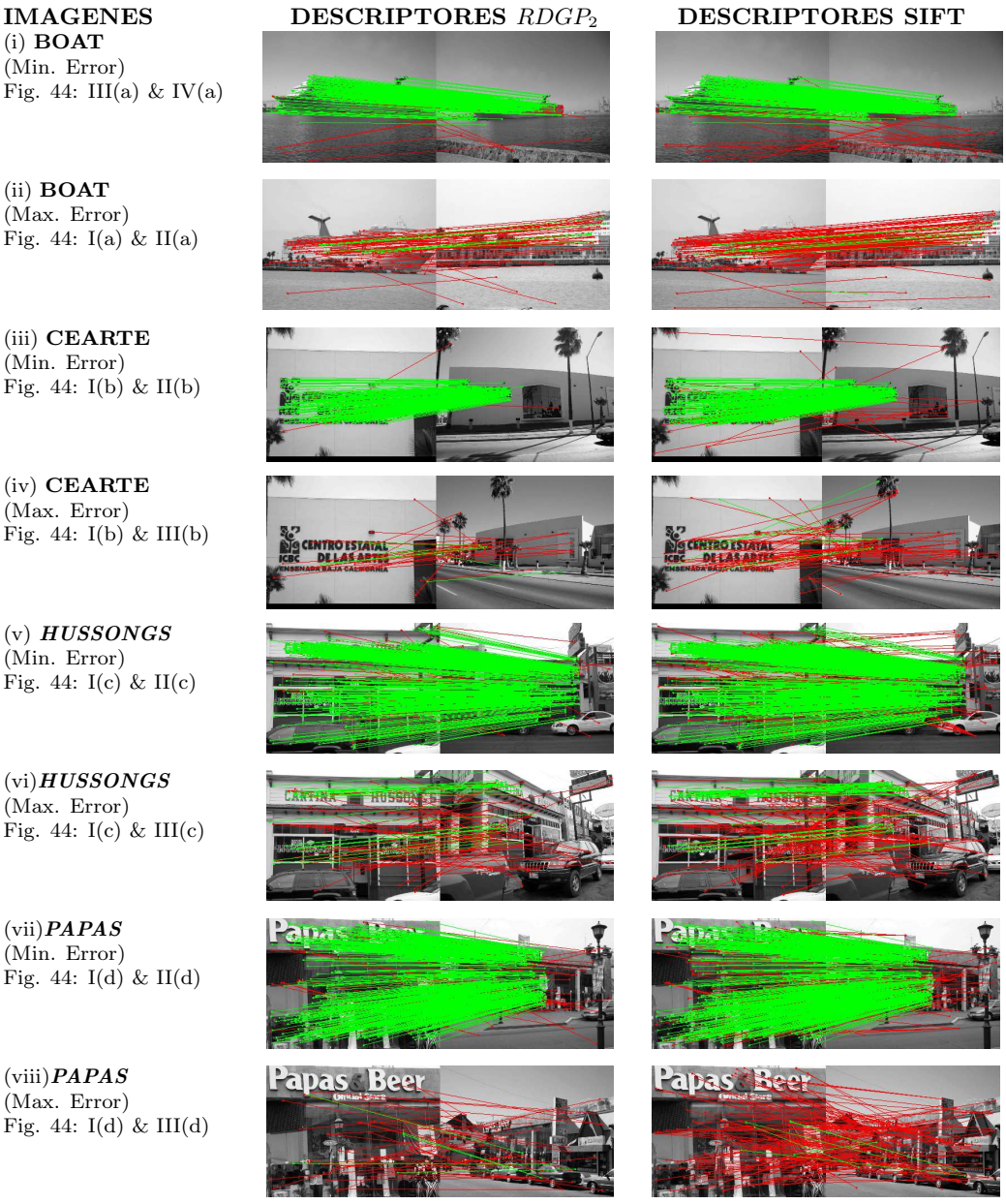


Figura 45. Ejemplo de la correspondencia de descriptores para escenarios al exterior.

Capítulo IV

Conclusiones y Perspectivas

En esta tesis se ha propuesto el uso del cómputo evolutivo como un enfoque novedoso en la descripción del contenido de imágenes digitales para la segmentación y el reconocimiento de objetos. En el primer trabajo se abordó el problema de segmentación de imágenes con textura utilizando descriptores estadísticos en una matriz de co-ocurrencia para analizar su contenido. Además, el proceso de optimización de este algoritmo fue llevado cabo usando un algoritmo genético canónico no supervisado ya que no se tomó en cuenta ninguna información a priori para realizar el proceso de segmentación. De esta manera, en cada generación nuestro algoritmo llamado *EvoSeg* aprende a identificar cuáles son las regiones existentes en la imagen clasificando la información que es homogénea y que a su vez cumpla con la propiedad de conectividad. Posteriormente, se incluyó la interacción en este algoritmo dentro del proceso de optimización. La finalidad de incluir la interacción era mejorar el criterio de evaluación de cada individuo, lo cual ayudó a obtener mejores imágenes segmentadas en un menor tiempo computacional.

Por otro lado, en el segundo trabajo se abordó el reconocimiento de objetos utilizando características locales, es decir, descriptores invariantes a transformaciones geométricas y fotométricas que permitieran describir el contenido de las imágenes con el fin de llevar a cabo el reconocimiento de una manera más eficiente y robusta. En este trabajo se propuso el uso de la programación genética para sintetizar operadores descriptivos para el descriptor SIFT con el fin de obtener rendimiento mejor o comparable con los descriptores del estado del arte. Para ello, se propuso un criterio de evaluación cuantitativa basado en la medida F que permitiera comparar el rendimiento de los descriptores de una manera más clara y concreta que los métodos tradicionales basados en las curvas de recall vs precisión. Una vez que se llevó a

cabo el aprendizaje de operadores descriptivos obtuvimos como resultado 30 operadores para el descriptor SIFT los cuales superaron en gran medida el rendimiento de los descriptores diseñados por el ser humano. Para esto, se eligió el mejor operador llamado $RDGP_2$ para realizar comparaciones entre los mejores descriptores del estado del arte utilizando imágenes con seis diferentes tipos de transformación geométrica y fotométrica, como por ejemplo: rotación, escalamiento, iluminación, difuminado, compresión JPEG y transformación afín. Como resultado de esta evaluación, se obtuvo hasta un 45.59% de mejoría en el caso de la rotación, un 21.66% en rotación y escalamiento, un 28.85% en el caso de iluminación, 26.44% en difuminación de la imagen y un 18.54% en la compresión JPEG. En el caso de la transformación afín nuestro algoritmo ocupó el cuarto lugar debido a que el detector que usamos no está diseñado para este tipo de tranformaciones. Sin embargo, estamos plenamente seguros que si se utilizara un mejor detector, los resultados cambiarían a nuestro a favor. Además, nuestro descriptor $RDGP_2$ fue probado en el reconocimiento de objetos donde se compararon nuestros resultados con el ya conocido descriptor SIFT. En este caso, nuestro descriptor resultó mejor para este tipo de aplicaciones donde la correspondencia de descriptores forma parte del proceso de la aplicación real. De acuerdo con los resultados obtenidos en las diferentes pruebas del reconocimiento, observamos que nuestro descriptor sirve como una especie de filtro para las correspondencias falsas ya que las disminuye considerablemente conservando las correspondencias que son correctas. Finalmente, podemos decir que el enfoque evolutivo en este tipo de problemas puede llegar a ser muy eficiente si se lleva a cabo un buen planteamiento del problema y se define correctamente el criterio de evaluación, lo cual es decisivo para el correcto funcionamiento del algoritmo.

IV.1 Limitaciones del trabajo

Las limitaciones de nuestro trabajo está relacionado con la implementación de los algoritmos. Por un lado, en el caso de la segmentación de imágenes, podemos mencionar las siguientes

limitaciones:

- **Información de la escala de grises**. Nuestro algoritmo *EvoSeg* solamente utiliza la información en escala de grises de la imagen ya que estamos utilizando descriptores estadísticos mediante la matriz de co-ocurrencia en escala de grises. En ese sentido, sería interesante utilizar la información del color con el fin de obtener otro tipo de información del contenido de la imagen pudiendo aplicar estos descriptores a cada banda en diferentes espacios de color.

- **Interfaz de usuario**. En la parte de la interacción es necesario mejorar la interfaz del usuario haciéndola más amigable con el fin de agilizar el proceso de la selección de individuos.

- **Interacción**. Al incluir interacción al sistema, mejora el proceso de optimización, pero a su vez, limita al sistema a depender de una persona experta que tiene que estar presente para analizar las posibles soluciones durante el proceso de evaluación.

Por otro lado, en el caso de la automatización de operadores descriptivos, las limitaciones de nuestro trabajo serían las siguientes:

- **El detector DoG**. Estamos utilizando el detector DoG del algoritmo SIFT en la etapa de detección, el cual es solamente invariante a rotaciones y escalamiento; además de no ser uno de los mejores del estado del arte. Por esta razón, nuestro descriptor no obtuvo el mejor rendimiento para el caso de imágenes con transformacione afines ya que el detector no es el apropiado para este tipo de transformaciones.

- **El lenguaje de programación**. Utilizamos como lenguaje de programación Matlab/C por la rapidez de respuesta para llevar a cabo la implementación. Sin embargo, este lenguaje es computacionalmente muy costoso cuando el algoritmo es ejecutado. Por esta razón, sería muy difícil utilizar el descriptor $RDGP_2$ en aplicaciones reales que requieran un tiempo de respuesta menor a dos segundos a menos que se migrara a otro

lenguaje de programación más eficiente. Un ejemplo del lenguaje que podría utilizarse sería CUDA (Compute Unified Device Architecture) de NVIDIA utilizando una GPU (Graphics Processing Unit). De hecho, en nuestro laboratorio de EvoVision ya se está trabajando sobre esta plataforma por las múltiples ventajas que se obtienen al utilizar este lenguaje en las tareas de alto nivel de Visión por Computadora.

IV.2 Trabajo Futuro

Actualmente, la aplicación de algoritmos evolutivos en distintas áreas de la ciencia ha tomado más fuerza debido a la rapidez de respuesta en comparación con otros métodos de aprendizaje para problemas altamente complejos. Sobretodo, la programación genética ha captado la atención de investigadores y estudiantes para desarrollar sistemas de aprendizaje que nunca antes se hubiera pensado aplicar obteniendo muy buenos resultados con ello. Como ejemplo, podemos mencionar nuestro trabajo basado en la automatización de operadores descriptivos para el descriptor SIFT, el cual nunca antes se habia planteado de esa manera y obtuvimos resultados realmente sorprendentes. En consecuencia, podemos decir que este trabajo sirve como base para futuras aplicaciones reales de la Visión por Computadora el cual puede ser utilizado en nuestro laboratorio de EvoVisión. La condición primordial sería migrar de lenguaje de programación por uno de respuesta más rápida. Algunas de las aplicaciones que se pudieran llevar a cabo en nuestro laboratorio son muy diversas debido a que se cuenta con equipo de alta tecnología que muy pocos laboratorios en nuestro pais e incluso en Europa tienen el privilegio de tener; esto es gracias al esfuerzo que el Dr. Olague ha realizado con el fin de hacer trabajos científicos competitivos con el estado del arte a nivel internacional. De esta manera, el equipo con el contamos en nuestro laboratorio es un robot móvil, un brazo manipulador de seis grados de libertad, una mira de calibración alemana áltamente precisa, dos servidores con equipamiento GPU, cámaras digitales para visión estéreo y cámaras web de alta resolución. Por ello, algunas de las aplicaciones que pudieran suscitarse utilizando como

base el trabajo de detección de Leonardo Trujillo y nuestro trabajo sobre descriptores, son las siguientes: reconocimiento de objetos, recuperación de imágenes, el problema del Next Best View, clasificación de objetos y clases de objetos, detección de rostros o personas en imágenes o vídeo, reconstrucción 3D, seguimiento de objetos o personas en movimiento, entre otros.

Por otro lado, sería interesante aplicar nuestro mejor operador a un algoritmo inspirado en el SIFT que utilice la magnitud del gradiente como su operador descripitvo con el fin de mejorar aún más su rendimiento. Por ejemplo, pudiera ser el descriptor HOG el cual detecta personas en movimiento utilizando una base de datos que contiene hasta 1800 imágenes de personas con posiciones variadas y diferentes fondos. En este caso, nuestro operador $RDGP_2$ pudiera ayudar a mejorar la calidad de la descripción tal como se mejoró para el descriptor SIFT haciéndolo más robusto. Otra de las futuras mejoras para nuestro trabajo sería migrar de lenguaje de programación para mejorar el tiempo de ejecusión de nuestro descriptor $RDGP_2$ ya que actualmente tarda dos segundos al igual que el SIFT programado en Matlab/C. Finalmente, algo que es imprescindible modificar en nuestro algoritmo sería el detector que actualmente usamos, el DoG, ya que no es un detector muy eficiente, no está diseñado para transformaciones afines y además, gran parte del tiempo de cómputo del algoritmo se utiliza en esta etapa.

Bibliografía del Autor

Artículos de Revista

- Olague, G., Pérez, C.B., Fernández, F., y Lutton, E. (2009). An artificial life approach to dense stereo disparity. *Artificial Life and Robotics*. Vol.13(2). Springer Japan.

- Olague, G., Fernández, F., Pérez, C., and Lutton, E. (2005). The Infection Algorithm: An Artificial Epidemic Approach for Dense Stereo Correspondence. *Artificial Life, MIT Press*. Vol. 12(4), pp. 593-615.

Artículos de Conferencias Arbitradas

- Pérez, C.B y Olague, G. Evolutionary Learning of Local Descriptor Operators for Object Recognition. *Genetic and Evolutionary Computation Conference (GECCO)*. ISBN:978-1-60558-325-9. pp. 1051-1058. July 8-12, 2009. Montreal, Canada. **Bronze Medal at the 2009 Human-Competitive Awards, the "Humies".**

- Pérez, C.B. y Olague, G. Evolving Local Descriptor Operators through Genetic Programming. *European Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP)*. LNCS 5484: 414 - 419. Springer-Verlag. April 5-17, 2009. Tübingen, Germany.

- Pérez, C.B y Olague, G. Learning invariant region descriptor operators with genetic programming and the F-Measure. *International Conference on Pattern Recognition (ICPR)*. pp. 1 - 4. ISBN: 978-1-4244-2174-9. December 8-11 2008. Tampa, Florida, USA.

- Pérez, C.B y Olague, G. Unsupervised Evolutionary Segmentation Algorithm based on Texture Analysis. 9^{th} *European Workshop on Evolutionary Computation in Image Analysis and Signal Processing.* LNCS 4448: 407-414. Springer-Verlag. EvoIASP2007. Valencia, España.

- Pérez C., Olague, G., Fernández, F., and Lutton, E. An Evolutionary Infection Algorithm for Dense Stereo Correspondence. 7^{th} *European Workshop on Evolutionary Computation in Image Analysis and Signal Processing.* LNCS 3449: 294 - 303. Springer-Verlag. EvoIASP2005. Laussane, Suiza.

- Olague, G., Fernández, F., Perez, C., and Lutton, E. (2004) The Infection Algorithm an Artificial Epidemic Approach for Dense Stereo Matching. *Parallel Problem Solving from Nature.* X. Yao et al. (Eds.): LNCS 3242: 622-632. Springer-Verlag. Birmingham, UK. September 18-22, 2004.

Capítulos de Libro

- Pérez, C.B. y Olague, G. Lutton E. y Fernández, F. Texture image segmentation using an interactive evolutionary approach. (2009). Título del libro: Studies in Computational Intelligence. Editores: CAGNONI, STEFANO. Editorial: Springer Berlin / Heidelberg. ISBN: 978-3-642-01635-6 Vol. 213. Páginas: 3-19.

- Fernández, F., Olague, G., Pérez, C., y Lutton, E. Advancing Dense Stereo Correspondence with the Infection Algorithm. (2008). Título del libro: Studies in Computational Intelligence. Editores: KACPRZYK, JANUSZ. Editorial: Springer Berlin / Heidelberg. ISBN: 978-3-540-77474-7 Vol. 102. Páginas: 305-324.

Referencias

Abdel-Hakim, A. y Farag, A. (2006). Csift: A sift descriptor with color invariant characteristics. *IEEE Conference on Computer Vision and Pattern Recognition. New York, NY, USA. June 17-22*, **2**: 1978 – 1983.

Agarwal, S., Awan, A., y Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(11): 1475 – 1490.

Arbelaez, P. y Cohen, L. (2006). A metric approach to vector-valued image segmentation. *International Journal of Computer Vision*, **69**(1): 119 – 126.

Bandlow, T., Klupsch, M., Hanek, R., y Schmitt, T. (2000). Fast image segmentation, object recognition and localization in a robocup scenario. *RoboCup-99: Robot Soccer World Cup III*, **LNCS 1856**: 174 – 185.

Bay, H., Fasel, B., y Van Gool, L. (2006a). Interactive museum guide: fast and robust recognition of museum objects. *Proceedings of the first International Workshop on Mobile Vision (IMV 2006). Graz, Austria. May 13*.

Bay, H., Tuytelaars, T., y Van Gool, L. (2006b). Surf: Speeded up robust features. *European Conference on Computer Vision. Graz, Austria. May 7-13*, **LNCS 3951**: 404 – 417.

Beaudet, P. (1978). Rotational invariant image operators. *International Conference on Pattern Recognition (ICPR)*, páginas 579 – 583.

Beis, J. y Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico. June 17-19, 1997*, páginas 1000 – 1006.

Bhandarkar, S. y Zhang, H. (1999). Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation*, **3**(1): 1 – 21.

Bhanu, B., Lee, S., y Ming, J. (1995). Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, **25**(12): 1543 – 1567.

Bosch, A., Zisserman, A., y Munoz, X. (2007). Representing shape with a spatial pyramid kernel. *ACM International Conference on Image and Video Retrieval. Amsterdam, Netherlands. July 9-11*, páginas 401 – 408.

Brown, M., Szeliski, R., y Winder, S. (2005). Multi-image matching using multi-scale oriented patches. *IEEE Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. June 20-26*, **1**: 510 – 517.

Cagnoni, S., Dobrzeniecki, A., Poli, R., y Yanch, J. (1999). Genetic algorithm-based interactive segmentation of 3d medical images. *Image and Vision Computing*, **17**(12): 881 – 895.

Carneiro, G. y Jepson, A. (2002). Phase-based local features. *European Conference on Computer Vision (ECCV). Copenhagen, Denmark. May 27-June 2*, **LNCS 2350**: 282 – 296.

Carneiro, G. y Jepson, A. (2003). Multi-scale phase-based local features. *IEEE Conference on Computer Vision and Pattern Recognition. Madison, WI, USA. June 16-22*, **1**: 736 – 743.

Carneiro, G. y Jepson, A. (2009). The quantitative characterization of the distinctiveness and robustness of local image descriptors. *Image and Vision Computing*, **27**: 1143 – 1156.

Çarkacioglu, A. y Yarman-Vural, F. (2003). Sasi: a generic texture descriptor for image retrieval. *Pattern Recognition*, **33**(11): 2615 – 2633.

Chen, J., Shan, S., Zhao, G., Chen, X., Gao, W., y Pietikäinen, M. (2008). A robust descriptor based on weber's law. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, Alaska. June 24-26*, páginas 1 – 7.

Cheng, H., Liu, Z., Zheng, N., y Yang, J. (2008). A deformable local image descriptor. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, Alaska. June 24-26*, páginas 1 – 8.

Cocquerez, J. y Philipp, S. (1997). *Analyse D'Images Filtrage et Segmentation*. Masson.

Dalal, N. y Trigs, B. (2006). Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition. New York, NY, USA. June 7-22*, **1**: 886 – 893.

Darwin, C. (1859). *On the origin of species by means of natural selection*. Murray, London. 495 páginas, primera edición.

Deng, Y. y Manjunath, B. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **23**(8): 800 – 810.

Duarte, A., Sánchez, A., Fernández, F., y Montemayor, A. (2006). Improving image segmentation quality through effective region merging using hierarchical social metaheuristic. *Evolutionary Computer Vision and Image Understanding*, **27**(11): 1239 – 1251.

Ebner, M. y Zell, A. (1999). Evolving a task specific image operator. *En R.Poli et al. (Eds.), First European Workshops on Evolutionary Image Analysis, Signal Processing and Telecommunications. Göteborg, Sweden.*, **LNCS 1596**: 74 – 89.

Ferrari, V., Tuytelaars, T., y Van Gool, L. (2006). Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, **67**(2): 159 – 188.

Freeman, W. y Adelson, E. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(9): 891 – 906.

Freixenet, J., Muñoz, X., Raba, D., Martí, J., y Cufí, X. (2002). Yet another survey on image segmentation: region and boundary information integration. *7th European Conference on Computer Vision. Copenhagen, Denmark. May 27-June 2*, **LNCS 2352**: 408 – 422.

Geusebroek, J. (2006). Compact object descriptors from local colour invariant histograms. *British Machine Vision Conference. Edinburgh, United Kingdom. September 4-7*, **3**: 1029 – 1038.

Geusebroek, J., van den Boomgaard, R., Smeulders, A., y Geerts, H. (2001). Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(12): 1338 – 1350.

Gimenez, D. y Evans, A. (2008). An evaluation of area morphology scale-space for colour images. *Computer Vision and Image Understanding*, **110**(1): 32 – 42.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning. 412 páginas*. Addison-Wesley, Boston, MA, USA.

Gonzalez, R. y Woods, R. (2002). *Tratamiento Digital de Imágenes*. Prentice Hall, New Jersey. 773 páginas, segunda edición.

Gupta, R. y Mittal, A. (2008). Smd: A locally stable monotonic change invariant feature descriptor. *European Conference on Computer Vision (ECCV).Marseille, France, October 12-18*, **LNCS 5303; Part II.**: 265 –277.

Haralick, R. y Shapiro, L. (1985). Survey: Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, **29**(1): 100 – 132.

Haralick, R., Shanmugam, K., y Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on System, Man and Cibernetics*, **3**(6): 610 – 621.

Harris, C. y Stephens, M. (1988). A combined corner and edge detector. *Alvey Vision Conference*, **15**: 147 – 151.

Hawkins, J. (1969). *Textural properties for pattern recognition*. Academic Press, New York. En Lipkin B, Rosenfeld A (eds) Picture Processing and Psychopictorics.

Hee-Su, K. y Sung-Bae, C. (2000). Application of interactive genetic algorithm to fashion design. *Engineering Applications of Artificial Intelligence*, **13**(6): 635 – 644.

Hernández, B., Olague, G., Hammoud, R., Trujillo, L., y Romero, E. (2007). Visual learning of texture descriptors for facial expression recognition in thermal imagery. *Computer Vision and Image Understanding, Special Issue on Vision Beyond Visual Spectrum*, **106**(2-3): 258 – 269.

Howard, D., Roberts, S., y Brankin, R. (1999). Target detection in sar imagery by genetic programming. *Advances in Engineering Software*, **30**(5): 303 – 311.

Huan, C., Chen, C., y Chung, P. (2008). Contrast context histogram: An efficient discriminating local descriptor for object recognition and image matching. *Pattern Recognition*, **41**(10): 3071 – 3077.

Ikizler, N. y Duygulu, P. (2009). Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing*, **27**(10): 1515 – 1526.

Johnson, M., Maes, P., y Darrell, T. (1994). Evolving visual routines. *Artificial Life*, **1**(4): 373 – 389.

Ke, Y. y Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. washington, dc, usa. june 27- july 2. *IEEE Conference on Computer Vision and Pattern Recognition*, **2**: 506 – 513.

Kitchen, L. y Rosenfeld, A. (1982). Gray-level corner detection. *Pattern Recognition Letters*, **1**(2): 95 – 102.

Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, **50**: 363 – 396.

Koenderink, J. y van Doorn, A. (1987). Representation of local geometry in the visual system. *Biological Cybernetics*, **55**: 367 –375.

Koza, J. R. (1992). *Genetic Programming: On the programming of computers by means of natural selection*. The MIT Press, MA, USA. 849 páginas.

Lazarevic-McManus, N., Renno, J., Makris, D., y Jones, G. (2008). An object-based comparative methodology for motion detection based on the f-measure. *Computer Vision and Image Understanding*, **111**: 74 – 85.

Lazebnik, S., Schmid, C., y Ponce, J. (2003). A sparse texture representation using affine-invariant regions. *IEEE Conference on Computer Vision and Pattern Recognition. Madison, WI, USA. June 16-22*, **2**: 319 – 324.

Lazebnik, S., Schmid, C., y Ponce, J. (2004). Semi-local affine parts for object recognition. *British Machine Vision Conference*, **2**: 959 – 968.

Lazebnik, S., Schmid, C., y Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition. New York, NY, USA. June 17-22*, **2**: 2169 – 2178.

Legrand, P., Bourgeois-Republique, C., Pean, V., Harboun-Cohen, E., Levy-Vehel, J., Frachet, B., Lutton, E., y Collet, P. (2006). Interactive evolution for cochlear implants fitting. *Genetic Programming and Evolvable Machines*, **8**(4): 319 – 354.

Lim, Y. y Lee, S. (1990). On the color image segmentation algorithm based on the thresholding and the fuzzy c-means technique. *Pattern Recognition*, **23**(9): 935 – 952.

Lin, Y. y Bhanu, B. (2005). Evolutionary feature synthesis for object recognition. *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Apps and Revs*, **35**(2): 156 – 171.

Lindeberg, T. (1993). On scale selection for differential operators. *Conference on Image Analysis. Tromso, Norway. May 25-28*, páginas 857 – 866.

Lindeberg, T. (1994). Scale-space theory in computer vision. *Kluwer Academic Publishers. Dordrecht, Netherlands*.

Lindeberg, T. (1998a). Feature detection with automatic scale selection. *International Journal of Computer Vision*, **30**(2): 79 – 116.

Lindeberg, T. (1998b). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, **30**(2): 117 – 154.

Ling, H. y Jacobs, D. (2005). Deformation invariant image matching. *International Conference on Computer Vision. Beijing, China. October 17-20*, **2**: 1466 – 1473.

Liu, C., Yuen, J., Torralba, A., y Sivic, J. (2008). Sift flow: dense correspondence across different scenes. *European Conference on Computer Vision. Marseille, France, October 12-18*, páginas 28 – 42.

Lowe, D. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision. Kerkyra, Greece. September 20-27*, páginas 1150 – 1157.

Lowe, D. (2004a). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**(2): 91 – 110.

Lowe, D. (2004b). Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image. *US Patent 6,711,293 (March 23, 2004). Provisional application filed March 8, 1999. Asignee: The University of British Columbia.*

Lutton, E. (2006). Evolution of fractal shapes for artists and designers. *International Journal on Artificial Intelligence Tools*, **15**(4): 651 – 672.

Lutton, E., Grenier, P., y Levy-Vehel, J. (2005). An interactive ea for multifractal bayesian denoising. *European Workshop on Evolutionary Computation in Image Analysis and Signal Processing. Lausanne, Switzerland. March 30-April 1*, páginas 274 – 283.

Manjunath, B., Ohm, J., Vasudevan, V., y Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Techonology*, **11**(6): 703 – 715.

Martin, D., Fowlkes, C., y Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(5): 530 – 549.

Mikolajczyk, K. (2002). Interest point detection invariant to affine transformations. *PhD thesis. Institut National Polytechnique de Grenoble. 171 páginas.*

Mikolajczyk, K. y Schmid, C. (2003). A performance evaluation of local descriptors. *IEEE Conference on Computer Vision and Pattern Recognition. Madison, Wisconsin, USA. June 16-22*, **2**: 525 – 531.

Mikolajczyk, K. y Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, **1**(60): 63 – 86.

Mikolajczyk, K. y Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(10): 1615 – 1630.

Moreels, P. y Perona, P. (2007). Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, **73**(3): 263 – 284.

Moreno, P., Bernardino, A., y Santos-Victor, J. (2009). Improving the sift descriptor with smooth derivative filters. *Pattern Recognition Letters.*, **30**(1): 18 – 26.

Mortensen, E., Deng, H., y Shapiro, L. (2005). A sift descriptor with global context. *IEEE Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. June 20-26*, **1**: 184 – 190.

Olague, G., Cagnoni, S., y Lutton, E. (2006). Introduction to the special issue on evolutionary computer vision and image understanding. *Pattern Recognition Letters*, **27**(11): 1161 – 1163.

Pal, N. y Pal, S. (1993). A review on image segmentation techniques. *Pattern Recognition*, **26**(9): 1277 – 1294.

Parker, J. (1996). *Algorithms for image processing and computer vision*. John Wiley, New York. 432 páginas.

Pavan, M. y Pelillo, M. (2003). A new graph-theoretic approach to clustering and segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Madison, WI, USA. June 16-22*, **1**: 145 – 152.

Perez, C. y Olague, G. (2007). Unsupervised evolutionary segmentation algorithm based on texture analysis. *9th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing. Valencia, Spain, April 11-13*, **LNCS 4448**: 407 – 414.

Perez, C. y Olague, G. (2008). Learning invariant region descriptor operators with genetic programming and the f-measure. *International Conference on Pattern Recognition. Tampa, Florida, USA. December 8-11*, páginas 1 – 4.

Perez, C. y Olague, G. (2009). Evolutionary learning of local descriptor operators for object recognition. *Genetic and Evolutionary Computation Conference. Montréal, Canada. July 8-11*, páginas 1051 – 1058.

Perez, C., Olague, G., Lutton, E., y Fernandez, F. (2009). Texture image segmentation using an interactive evolutionary approach. *Stefano Cagnoni Eds. Springer Berlin*, **213**: 3 – 19.

Poli, R. (1996). Genetic programming for feature detection and image segmentation. *Proceedings of the AISB 96 Workshop on Evolutionary Computation. Brighton, U.K. April 1-2*, **LNCS 1143**: 110 – 125.

Poli, R., Langdon, W., y Freitag, N. (2008). A field guide to genetic programming. *Published via* `http://lulu.com` *and freely available at* `http://www.gp-field-guide.org.uk`. *Consultado en Junio 2010*.

Sarfraz, S. y Hellwich, O. (2008). Head pose estimation in face recognition across pose scenarios. *Proceedings of VISAPP 2008, International Conference on Computer Vision Theory and Applications. Funchal, Madeira, Portugal. January 22-25*, páginas 235 – 242.

Schmid, C. y Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(5): 530 – 534.

Shi, J. y Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(8): 888 – 905.

Song, A. y Ciesielski, V. (2008). Texture segmentation by genetic programming. *Evolutionary Computation, Special Issue on Evolutionary Computer Vision.*, **16**(4): 461 – 481.

Stein, A. y Hebert, M. (2005). Incorporating background invariance into feature-based object recognition. *IEEE Workshops on Application of Computer Vision (WACV/MOTION'05). Breckenridge, CO, USA; January 5-7*, páginas 37 – 44.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, **89**(9): 1275 – 1296.

Teller, A. y Veloso, M. (1995). Pado: Learning tree structured algorithms for orchestration into an object recognition system. *Technical Report CMU-CS-95-101, Department of ComputerScience, Carnegie Mellon University, Pittsburgh, PA, USA. 30 páginas*.

Tola, E., Lepetit, V., y Fua, P. (2008). A fast descriptor for dense matching. *IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, AK. June 24-26*.

Trackett, W. (1993). Genetic programming for feature discovery and image discrimination. *En S.Forrest (Ed.), Proccedings of the 5th International Conference on Genetic Algorithms, ICGA-93. Urbana-Champaign, IL, USA*, páginas 303 – 309.

Trujillo, L. y Olague, G. (2006a). Synthesis of interest point detectors through genetic programming. *Genetic and Evolutionary Computation Conference. Seattle, WA, USA. July 8-12*, **1**: 887 – 894.

Trujillo, L. y Olague, G. (2006b). Using evolution to learn how to perform interest point detection. *International Conference on Pattern Recognition. Hong Kong, China. August 20-24*, **1**: 211 – 214.

Trujillo, L. y Olague, G. (2007). Scale invariance for evolved interest operators. *European Workshop on Evolutionary Computation in Image Analysis and Signal Processing. Valencia, Spain. April 11-13*, **LNCS 4448**: 423 – 430.

Trujillo, L. y Olague, G. (2008). Automated design of image operators that detect interest points. *Evolutionary Computation, Special Issue on Evolutionary Computer Vision*, **16**(4): 483 – 507.

Trujillo, L., Olague, G., Legrand, P., y Lutton, E. (2007). Regularity based descriptor computed from local image oscillations. *Optics Express*, **15**(10): 6140 – 6145.

Van-Rijsbergen, C. (1979). *Information retrieval*. Butterworth-Heinemann, segunda edición. 224 páginas.

Viola, P. y Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Kauai, HI, USA. December 8-14*, páginas 511 – 518.

Witkin, A. (1983). Scale-space filtering. *International Joint Conference on Artificial Intelligence. Karlsruhe, West Germany*, páginas 1019 – 1023.

Yoshimura, M. y Oe, S. (1999). Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas. *Pattern Recognition*, **32**: 2041 – 2054.

Zhang, M., Ciesielski, V. B., y Andreae, P. (2003). A domain-independent window approach to multiclass object detection using genetic programming. *EURASIP Journal on Applied Signal Processing*, **2003**(8): 841 – 859.