TESIS DEFENDIDA POR

Alina Susana Garza Lozano

Y aprobada por el siguiente comité:

Dr. Gustavo Olague Caballero

Director del Comité

M.C. José Luis Briseño Cervantes

Miembro del Comité

Dr. Jorge Torres Rodríguez

Miembro del Comité

Dr. César Cruz Hernández

Miembro del Comité

Dr. Serguei Miridonov

Miembro del Comité

Dr. Pedro Gilberto López Mariscal

Coordinador del Programa de Posgrado en Ciencias de la Computación

Dr. Raúl Ramón Castro Escamilla

Director de Estudios de Posgrado

21 de Septiembre de 2006

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA

POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

Construcción de un detector de puntos de interés por medio de programación genética

TESIS

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

MAESTRO EN CIENCIAS

Presenta:

Alina Susana Garza Lozano

Ensenada, Baja California, México. Septiembre del 2006.

RESUMEN de la tesis de **Alina Susana Garza Lozano**, presentada como requisito parcial para obtener el grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, Baja California. Septiembre del 2006.

Construcción de un detector de puntos de interés por medio de programación genética

Resumen aprobado por:

Dr. Gustavo Olague Caballero

Director de Tesis

En este trabajo se presenta un novedoso enfoque de aprendizaje máquina para la construcción automática de detectores de puntos de interés útiles para determinadas tareas de alto nivel de la visión por computadora. El problema de detección de características es abordado como un problema de optimización. Para resolverlo se hace uso de la programación genética. El criterio de optimización utilizado promueve la repetibilidad y distribución uniforme de los puntos sobre la imagen. La repetibilidad garantiza estabilidad geométrica bajo diferentes tipos de transformaciones. Los detectores de puntos de interés generados tienen una estructura simple y muestran resultados competitivos con los detectores más utilizados por la comunidad de visión por computadora. La programación genética fue capaz de redescubrir el Laplaciano y compensar sus desventajas, además de que dicho detector presenta un alto nivel de desempeño. Lo que demuestra la capacidad de la programación genética de generar resultados competitivos con los generados por el ser humano.

Palabras clave: Detectores de puntos de interés, Extracción de características, Programación genética, Repetibilidad, Entropía.

ABSTRACT of the thesis presented by **Alina Susana Garza Lozano**, as a partial requirement to obtain the MASTER IN SCIENCE degree in COMPUTER SCIENCES. Ensenada, Baja California. september 2006.

Construction of an Interest Point Detector through Genetic Programming

Abstract approved by:

Dr. Gustavo Olague Caballero

Thesis director

In this work we present a novel approach for machine learning to automatically synthesize interest point detectors useful in certain high level computer vision tasks. The feature detection problem is tackle as an optimization problem. In order to solve it we use genetic programming. The optimization criterion used in this work promotes repeatability and uniform distribution of points all over the image. Repeatability guarantees geometric stability under different types of image transformations. The interest point detectors generated with our approach have simple structure and exhibit competitive performance compared with the detectors most used by the computer vision community. The genetic programming was able to rediscover the Laplacian and compensate its disadvantages, besides, such detector shows a high level of performance. This proves the ability of genetic programming to achieve competitive results compared with the ones generated by the human beings.

Keywords: Interest Point Detectors, Feature extraction, Genetic Programming, Repeatability, Entropy.

Dedicado a mi madre

Cristina Lozano Arroyo

"Aunque siempre soy fuerte confío no me dejarás caer"

Alina.

Agradecimientos

A mi madre, por su apoyo y por emprender conmigo este sueño.

A mi hermano, por el apoyo y ayuda que me brindó, sin la cual no hubiese sido posible la realización de este sueño.

Al Dr. Gustavo Olague, por su colaboración en la realización de esta tesis, por la oportunidad de trabajar junto a él y por la confianza en mi trabajo.

A los miembros del Comité de Tesis, por su interés y aportaciones a este trabajo.

A mis amigos, por su amistad, guía, ayuda y apoyo incondicional en estos dos años.

A mis compañeros del laboratorio de Evovisión, por ayudarme a comprender las bases de la investigación y por sus comentarios y discusiones sobre el tema.

A Leonardo, por su ayuda en la realización de esta tesis.

A mis maestros, por sus conocimientos y enseñanzas a lo largo de mi vida.

Al Centro de Investigación Científica y de Educación Superior de Ensenada y al Consejo Nacional de Ciencia y Tecnología, por las facilidades para estudiar en esta institución.

Al Consejo Nacional de Ciencia y Tecnología, por la ayuda económica y por hacer posible la realización de la maestría.

Ensenada, México
21 de Septiembre del 2006.

Alina Susana Garza Lozano

Tabla de Contenido

Capítulo Página

Tabla de Contenido (Continuación)

Lista de Figuras

Lista de Figuras (Continuación)

Lista de Figuras (Continuación)

Lista de Tablas

Capítulo I

Introducción

Gran parte de la información que los seres vivos recibimos del mundo es a través de la vista. Por medio de la vista el ser humano percibe color, tamaño, textura, forma, distancia, movimiento y posición de los objetos. La visión se produce cuando la información percibida por los ojos es procesada por el cerebro.

El ser humano ha deseado e intentado reproducir la actividad cerebral durante mucho tiempo. Entre las funciones más complejas del cerebro se encuentra la visión. La vista es el sentido más desarrollado. Es por ello, que al proveer a una computadora o robot del sentido de la vista es una tarea compleja que hasta el momento no se ha logrado con plenitud.

"Debido a que la información visual es una de las principales fuentes de datos del mundo real, resulta útil el proveer a una computadora digital del sentido de la vista (a partir de imágenes tomadas con cámaras digitales o analógicas)" (McGlone *et al.*, 2004).

La importancia de que las computadoras cuenten con el sentido de la vista radica en que existen ambientes donde es más apropiado o más eficiente el trabajo realizado por los robots que el realizado por los seres humanos. Por ejemplo, en la industria. Si un robot de ensamblaje pudiese ver, podría tomar las piezas sin importar la posición o el orden en el que fueron acomodadas en la línea de ensamble, a fin de poder tomarla.

La visión por computadora es una rama de la inteligencia artificial que tiene como objetivo resolver el reconocimiento automático de objetos en imágenes digitales, determinar sus figuras y reconstruir la posición y orientación de estos.

La visión por computadora está formada por varias etapas. La primera es adquirir la imagen, para esto se involucran principios de la geometría proyectiva. Después, para obtener una imagen de mejor calidad, se procesa para eliminar el ruido y restaurarla. Posteriormente, durante la etapa de análisis se identifican algunos de los componentes de la imagen mediante métodos como la detección de características y se modela tridimensionalmente la información contenida en la imagen, mediante la reconstrucción. Finalmente, en la etapa de visión artificial, también llamada visión de alto nivel, se comprende el contenido de la escena, el conocimiento de los objetos que la componen y la manipulación de este conocimiento a alto nivel para que un sistema autónomo lo utilice como guía para desenvolverse en su entorno, véase Figura 1.

En la etapa de análisis es donde se obtiene información visual de las imágenes. Para lograrlo es necesario extraer características de ésta, las cuales se procesan para así poder interpretarlas. La extracción de características se realiza a partir de imágenes posiblemente preprocesadas.

La característica de más bajo nivel es el punto. Un punto de interés es un pixel de la imagen que presenta un nivel alto de variación con respecto a una medida local particular. Su detección no es trivial ya que deben cumplir varios criterios. Extraerlos simplifica el análisis de la imágen ya que reducen la abrumadora cantidad de información contenida en las imágenes que durante las tareas de visión de alto nivel se necesita procesar. Así, los puntos de interés son necesarios para varias tareas de

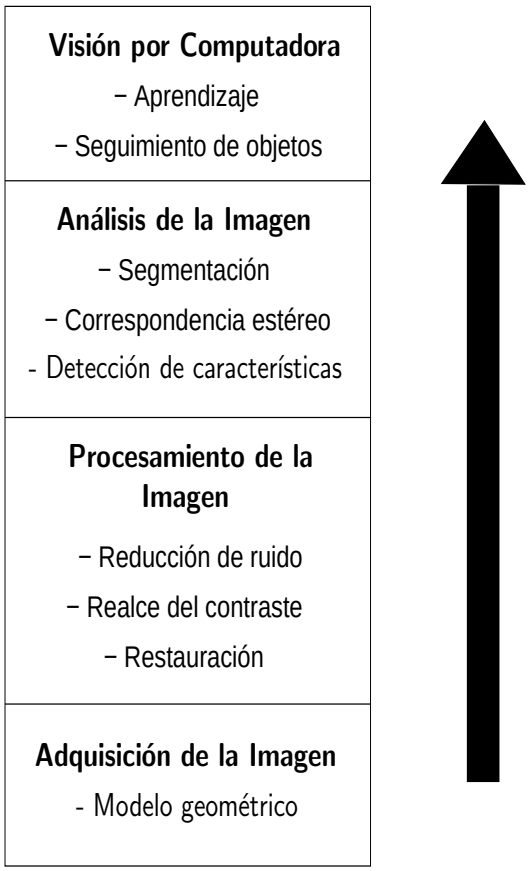


Figura 1: Etapas que conforman la visión por computadora.

alto nivel como por ejemplo, detección y reconocimiento de objetos, reconstrucción 3D, correspondencia y seguimiento, por mencionar algunas.

En la literatura existen diferentes tipos de detectores de puntos de interés, todos ellos resultado de cómo el ser humano ha abordado el problema. Para que los resultados alcanzados por un detector de puntos de interés sean confiables, éste siempre debe detectar los mismos puntos dentro de una misma escena.

Para evaluar los resultados obtenidos por el detector, existen varios criterios, los cuales dependen de la tarea para la cual se requieren los puntos de interés. En este trabajo utilizaremos los criterios de repetibilidad y separación global. Dichos criterios miden

características deseables para tareas de correspondencia, reconocimiento de objetos y reconstrucción 3D.

A pesar de que el problema de extracción de características es muy estudiado actualmente, todavía se está lejos de resolverlo. En este trabajo se aborda el problema de detección de puntos de interés como un problema de optimización. Para solucionarlo se utiliza la programación genética. La razón principal es que permite evolucionar estructuras jerárquicas como las empleadas por los detectores de puntos de interés. Como resultado, se presenta un enfoque novedoso, el cual permite generar detectores de puntos de interés de manera automática a través de la programación genética.

I.1 Objetivos de la Investigación

I.1.1 Objetivo General

Determinar si es posible sintetizar detectores de puntos de interés a partir de la metodología propuesta por la programación genética. Los detectores deben presentar resultados comparables a los obtenidos por los detectores que existen actualmente.

I.1.2 Objetivos Particulares

- Obtener un detector que cumpla con las características de repetibilidad y separación global. Esto permitirá que sea robusto a cambios en las condiciones de observación (invarianza a rotación y cambios de iluminación).

- Determinar cuál es la representación más adecuada de la función aptitud para que el detector de puntos de interés evolucione cumpliendo con las características de repetibilidad y separación global.

- Determinar cuáles son los parámetros estocásticos de ejecución así como el mecanismo de inicialización de los árboles que representan a los individuos más idóneos para obtener el detector de puntos de interés deseado.

- Evaluar qué tan apropiada es la programación genética para obtener un detector de puntos de interés que presente resultados comparables a los detectores con los que se cuenta actualmente.

I.1.3 Importancia de la Investigación

Si los resultados son satisfactorios, se estaría automatizando el proceso de detección de puntos de interés. Esto colocaría a nuestra metodología para sintetizar detectores de puntos dentro del dominio del aprendizaje de máquina. Además, al estar respaldado por una metodología comprensible y que ha presentado buenos resultados como lo es la programación genética, se aseguran su usabilidad, entendimiento y la facilidad de ser modificado para satisfacer diferentes objetivos.

I.1.4 Limitaciones

El detector de puntos de interés que se desarrollará en esta investigación será apropiado para las tareas de correspondencia, reconocimiento de objetos y reconstrucción 3D al cumplir con las características de repetibilidad y separabilidad global. Sin embargo, no será apropiado para otras tareas como la calibración de cámaras, donde otra característica relevante además de las 2 mencionadas anteriormente es la exactitud de la localización. Esta característica podría ser agregada al detector ya que se trata de características complementarias, pero por el momento, dentro del alcance de nuestra

investigación sólo están consideradas las tareas de reconocimiento de objetos y correspondencia. Nuestro detector de puntos de interés sólo trabajará con imágenes en escala de grises.

I.1.5 Contribución de la Investigación

Actualmente se cuenta con varios detectores de características de bajo nivel que obtienen buenos resultados. Estos detectores son sensibles a cambios en las condiciones de observación como ángulo de observación, rotación y escalamiento (Moravec, 1977; Beaudet, 1978; Dreschler y Nagel, 1982; Kitchen y Rosenfeld, 1982; Wang y Brady, 1995; Harris y Stephens, 1988; Förstner y Gülch, 1987). La detección de los puntos de interés por medio de estos detectores depende de las condiciones del ambiente y del problema a tratar. Al desarrollar un detector como el que se pretende en esta investigación, se obtendrá un detector que produzca buenos resultados en las tareas de reconocimiento de objetos y correspondencia, independientemente de las condiciones del ambiente o del problema. A la vez, permitirá que la computadora sea quien se encargue de decidir la manera más apropiada de realizar la detección de los puntos.

I.1.6 Metodología de la Investigación

La metodología empleada en esta investigación es una metodología muy utilizada dentro de la comunidad de visión por computadora, y especialmente dentro del grupo de EvoVisión. Se puede observar esta metodología en la Figura 2. La metodología se divide en dos etapas: Plan de trabajo y ciclo de trabajo.

La descripción de las etapas son las siguientes:

1. **Estudio, análisis e identificación de la problemática.** En esta etapa se hará una revisión de la literatura con el objetivo de analizar y estudiar el problema.

2. **Conceptualización.** Se sigue revisando literatura, ahora con el objeto de entender los planteamientos matemáticos que existan en relación al tema. Para este caso, los modelos matemáticos son las operaciones aplicadas a la imagen para obtener los puntos de interés, como convolución, gradiente, etc.

3. **Formalización.** Una vez que se entendieron los planteamientos matemáticos existentes, en esta etapa se formularán nuestros propios modelos matemáticos para resolver el problema. En este caso, son las funciones primitivas y el conjunto de terminales que son las operaciones aplicadas a la imagen, así como la función aptitud que evaluara los resultados obtenidos del detector con respecto a varios criterios.

4. **Implementación.** Se implementarán los modelos matemáticos obtenidos en la etapa anterior para resolver el problema. Para ello, se emplearan una herramienta de programación genética de C llamada LILGP.

5. **Pruebas.** Una vez implementados los modelos, se probará la eficiencia en la resolución del problema.

6. **Experimentación.** En esta etapa se diseñan los experimentos que permitan evaluar nuestra investigación. Se establecerán el tamaño de la población, los parámetros estocásticos de ejecución, el mecanismo e inicialización, el número de generaciones y de imágenes empleadas en cada experimento, el tipo de imágenes y el número de ejecuciones por experimento.

7. **Análisis de los resultados.** En esta etapa se analizan minuciosamente los

resultados obtenidos. La repetibilidad se analizará por medio de la tasa de repetibilidad y la separabilidad global se hará por medio de la entropía.

8. **Reformulación matemática.** Si los resultados obtenidos no son satisfactorios, se hace una reformulación matemática contemplando los modelos matemáticos que se obtuvieron en la Formalización.

9. **Madurar las ideas, volver a 1.** Se maduran las ideas concluidas. Si en este paso surgen otras incógnitas se analizan para después volver a la primera etapa.

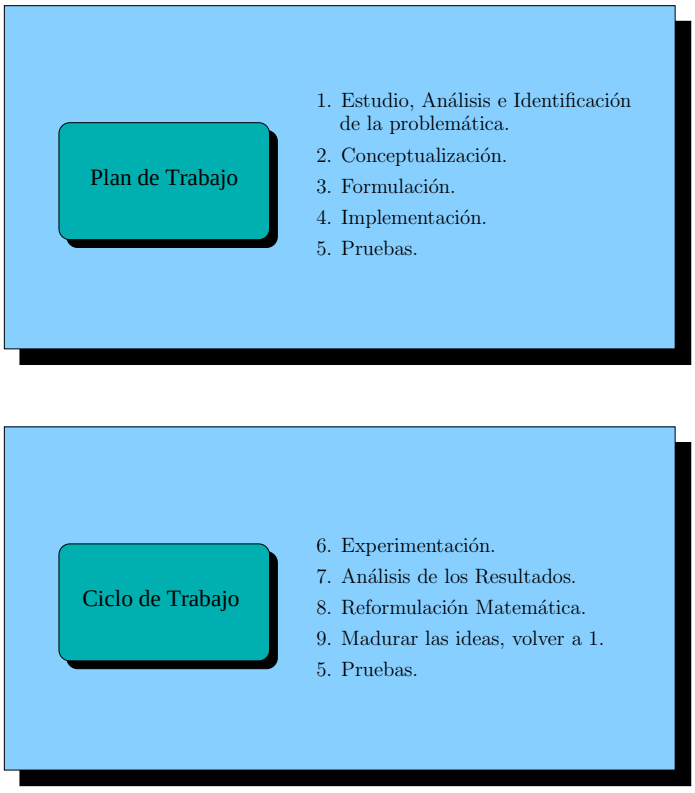


Figura 2: Nuestra metodología puede resumirse en dos etapas de trabajo: plan de trabajo y ciclo de trabajo. Dichas etapas se desarrollan en un proceso iterativo e integral.

I.1.7 Organización de la Tesis

Esta tesis esta estructurada de la siguiente manera:

- **Capítulo II.** Se introducen todos los conceptos relacionados con los puntos de interés que son necesarios para el entendimiento del funcionamiento de los detectores de puntos de interés y que serán utilizados a lo largo de la tesis

- **Capítulo III.** En este capítulo se presentan los fundamentos de la Programación Genética.

- **Capítulo IV.** En él, se exponen los detectores de puntos de interés que existen en la literatura. Incluye resultados de la evaluación de los detectores más utilizados por la comunidad de visión por computadora.

- **Capítulo V**. Se presenta el enfoque propuesto por nuestro trabajo, la arquitectura y los detalles de implementación de la aplicación genética desarrollada en este trabajo. Así como los resultados de las pruebas realizadas para comparar y evaluar nuestro enfoque con los detectores ya existentes.

- **Capítulo VI**. Se comentan las conclusiones a las que se llegan con el presente trabajo y las ideas del trabajo futuro que puede realizarse sobre la misma línea de investigación.

Capítulo II

Puntos de Interés

La extracción de características de bajo nivel (como lo son los bordes, las esquinas y los puntos) es de gran relevancia en problemas tan complejos como: análisis de formas y escenas, reconocimiento de objetos, detección de movimiento, algoritmos de correspondencia entre dos o más imagenes, reconstrucción tridimensional y calibración de cámaras (Hernández y Olague, 2001), entre otros.

Los detectores de características de bajo nivel, son conocidos como detectores de puntos de interés, también llamados operadores de puntos de interés. Estos detectores pueden ser agrupados de acuerdo a la manera en que modelan la información de la imagen. Dicha clasificación es presentada en este capítulo. Además, este capítulo contiene los fundamentos teóricos necesarios para la comprensión de los puntos de interés, el funcionamiento de sus detectores, así como para el desarrollo del trabajo realizado en esta tesis.

II.1 Conceptos Básicos

II.1.1 Concepto de Imagen

El término *imagen* se refiere a una función bidimensional de la luz y/o la intensidad, a la que se indica por $f(x, y)$, donde el valor o amplitud de f en las coordenadas espaciales (x, y) determina la intensidad (iluminación) de la imagen en ese punto (Gonzalez y Woods, 2002). Puesto que la luz es una forma de energia, $f(x, y)$ debe de ser estrictmente mayor que cero y finita, es decir

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R}^+ \, . \tag{1}$$

Con lo anterior, se define a la intensidad de una imagen monocrómatica f en las coordenadas (x, y) como *nivel de gris* l. Es evidente que $l \in [L_{min}, L_{max}]$ donde L_{min} es estrictamente mayor que cero y L_{max} es finita. Es decir, la función $f(x, y)$ es acotada en el intervalo $[L_{min}, L_{max}]$ y se le denomina *escala de grises*.

En la práctica el intervalo $[L_{min}, L_{max}]$ se desplaza numéricamente al intervalo $[0, L]$, donde la escala, $l = 0$ se considera como negro y $l = L$ se considera como blanco. Todos los valores intermedios son tonos de gris que varian de forma continua entre el negro y el blanco.

Para que una imagen $f(x, y)$ pueda ser analizada computacionalmente debe ser digitalizada tanto espacialmente como en su amplitud. Es decir la función $f(x, y)$ se muestrea uniformemente en su dominio (*muestreo de la imagen*) y en su contradominio (*cuantificación del nivel de gris*). Lo anterior se suele representar como una matriz de $N \times M$ (*imagen digital*), donde cada elemento (*pixel*) es una cantidad discreta

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, M-1) \\ f(1,0) & f(1,1) & \cdots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1, M-1) \end{bmatrix} = F(X, Y) \tag{2}$$

Nótese que $(X, Y) \in \mathbb{Z}^2$, es decir

$$F(X, Y) : \mathbb{Z}^2 \longrightarrow \mathbb{Z} \, . \tag{3}$$

En esta tésis se supondrá que los niveles de grises discretos están igualmente espaciados entre $[0, L]$.

Aunque una imagen sea una funcion de $\mathbb{R}^2$ a $\mathbb{R}$ se suele graficar la superficie generada en un plano tridimensional, tomando los valores en el eje z como $f(x, y)$.

II.1.2 Cálculo de la Derivada en una Imagen

El problema fundamental en el cálculo de la derivada en una imagen digital $F(X, Y)$ es que, la definición matemática de la derivada como tal no puede ser aplicada por ser $F(X, Y)$ una función discreta. Por lo tanto, los algoritmos propuestos en la literatura solo son una aproximación a la derivada de la imagen continua $f(x, y)$.

Uno de los métodos más comunes, es utilizar el concepto del gradiente para la diferenciación de una imagen. De esta manera el gradiente de una imagen $f(x, y)$ en el punto (x, y) es el vector

$$\nabla f(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f(x,y)}{\delta x} \\ \frac{\delta f(x,y)}{\delta y} \end{bmatrix} \quad . \tag{4}$$

En este caso el gradiente indica la direción de la máxima variación de $f(x, y)$. El módulo de $\nabla f(x, y)$, es la mágnitud de la máxima variación de $f(x, y)$

$$\|\nabla f(x,y)\| = \left[\left(\frac{\delta f(x,y)}{\delta x} \right)^2 + \left(\frac{\delta f(x,y)}{\delta y} \right)^2 \right]^{\frac{1}{2}} \quad . \tag{5}$$

En la práctica se suele aproximar por

$$\|\nabla f(x,y)\| \approx |G_x| + |G_y| \quad , \tag{6}$$

y la dirección del gradiente está dado por

$$\alpha(x,y) = arctan\left(\frac{G_y}{G_x}\right). \tag{7}$$

En el caso de una imagen digital se suele tomar:

$$\|\nabla F(X,Y)\| = \left[(H_x \otimes F(X,Y))^2 + (H_y \otimes F(X,Y))^2\right]^{\frac{1}{2}} , \tag{8}$$

$$\alpha(X,Y) = arctan\left(\frac{H_y \otimes F(X,Y)}{H_x \otimes F(X,Y)}\right) , \tag{9}$$

donde $\otimes$ representa una convolución y H_x y H_y son máscaras que operan sobre la imagen digital en dirección de los ejes x y y respectivamente.

Con lo anterior se trata de aproximar a la derivada por medio de diferencias. Algunas de las más comunes son las propuestas por Roberts, Prewit y Sobel, ver Tabla I.

Tabla I: Diferentes máscaras utilizadas para derivar una imagen

	Derivada Parcial en x	Derivada Parcial en y
Roberts	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$
Prewitt	$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$
Sobel	$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$

Los operadores de Sobel tienen la ventaja de proporcionar tanto la diferenciación como un efecto de suavizado. Dado que las derivadas realzan el ruido, el efecto de

suavizado es una característica particularmente atractiva. Marr y Hildreth en 1980 propusieron la utilización de un filtro Gaussiano, ver Ecuación (10), para calcular el Laplaciano, ver Ecuación (11).

$$h(x,y) = e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{10}$$

$$\nabla^2 h = \left(\frac{r^2-\sigma^2}{\sigma^4}\right) e^{\left(-\frac{r^2}{2\sigma^2}\right)}. \tag{11}$$

Actualmente se suele utilizar este tipo de filtros por ser más robustos al ruido y eficientes en el cálculo del la primera derivada, la segunda derivada y Laplaciano. En otras palabras, básicamente se encuentra la función de transferencia utilizando la transformada Z de la función Gaussiana convolucionada con la imagen. De esta manera el problema se reduce a encontrar los mejores coeficientes que aproximen esta operación. Esto quiere decir que la operación de derivar la imagen se puede lograr convolucionando la señal de entrada con la primera derivada de la Guassiana para en el caso de aproximar una primera derivada. Lo anterior se puede calcular en una sola expresión lo cual implica que la derivada de cada elemento se puede aproximar en un solo caso. La máscara se propone como:

$$fdg(x) = xe^{\left(-\frac{x^2}{2\sigma^2}\right)} \quad . \tag{12}$$

Deriche (1993) propone la siguiente aproximación a la primera derivada con un error cuadrático estimado de $\epsilon^2 = 2.765396E-06$

$$\begin{aligned} fdg_a(x) = & \left(-0.6472cos\left(0.6719\frac{x}{\sigma}\right) - 4.531sin\left(0.6719\frac{x}{\sigma}\right)\right) e^{\left(-1.527\frac{x}{\sigma}\right)} \\ & + \left(0.6494cos\left(2.072\frac{x}{\sigma}\right) + 0.9557sin\left(2.072\frac{x}{\sigma}\right)\right) e^{\left(-1.516\frac{x}{\sigma}\right)} \end{aligned} \tag{13}$$

Para una segunda derivada la aproximación se logra modelando la función de transferencia de la imagen convolucionada con la segunda derivada de la Gaussiana. Deriche (1993) propone una aproximación con un error cuadrático estimado de $\epsilon^2 = 2.941315E-05$ donde las expresiones son las siguientes:

$$sdg(x) = \left(1 - \left(\frac{x}{\sigma}\right)^2\right) e^{\left(-\frac{x^2}{2\sigma^2}\right)} \tag{14}$$

,

$$\begin{aligned} sdg_a(x) = & \left(-1.331cos\left(0.748\frac{x}{\sigma}\right) + 3.661sin\left(0.748\frac{x}{\sigma}\right)\right) e^{\left(-1.24\frac{x}{\sigma}\right)} \\ & + \left(0.3225cos\left(2.166\frac{x}{\sigma}\right) - 1.738sin\left(2.166\frac{x}{\sigma}\right)\right) e^{\left(-1.314\frac{x}{\sigma}\right)} \end{aligned} \tag{15}$$

En la Figura 3 se muestra la función Gaussiana, así como su primera y segunda derivada con $\sigma = 1$.

Autores como Lindeberg (1993); van Vliet *et al.* (1998); Yokono y Poggio (2004), entre otros han propuesto mejoras a los algoritmos recursivos que propone Deriche, los cuales son cada vez más eficientes en el procesamiento como en la aproximación numérica a una derivada.

De esta manera, en el presente trabajo se decidió utilizar los filtros recursivos propuestos por Deriche para calcular la primera y segunda derivada en las imagenes digitales utilizadas.

II.1.3 Función de Autocorrelación

La función de autocorrelación mide cambios locales en una señal. Esta medida es obtenida por la correlación de un elemento de la imagen con sus vecinos. Esto es por corrimientos en diferentes direcciones alrededor de un elemento para obtener las diferencias con sus vecinos. En el caso de un punto de interés y en particular de una esquina,

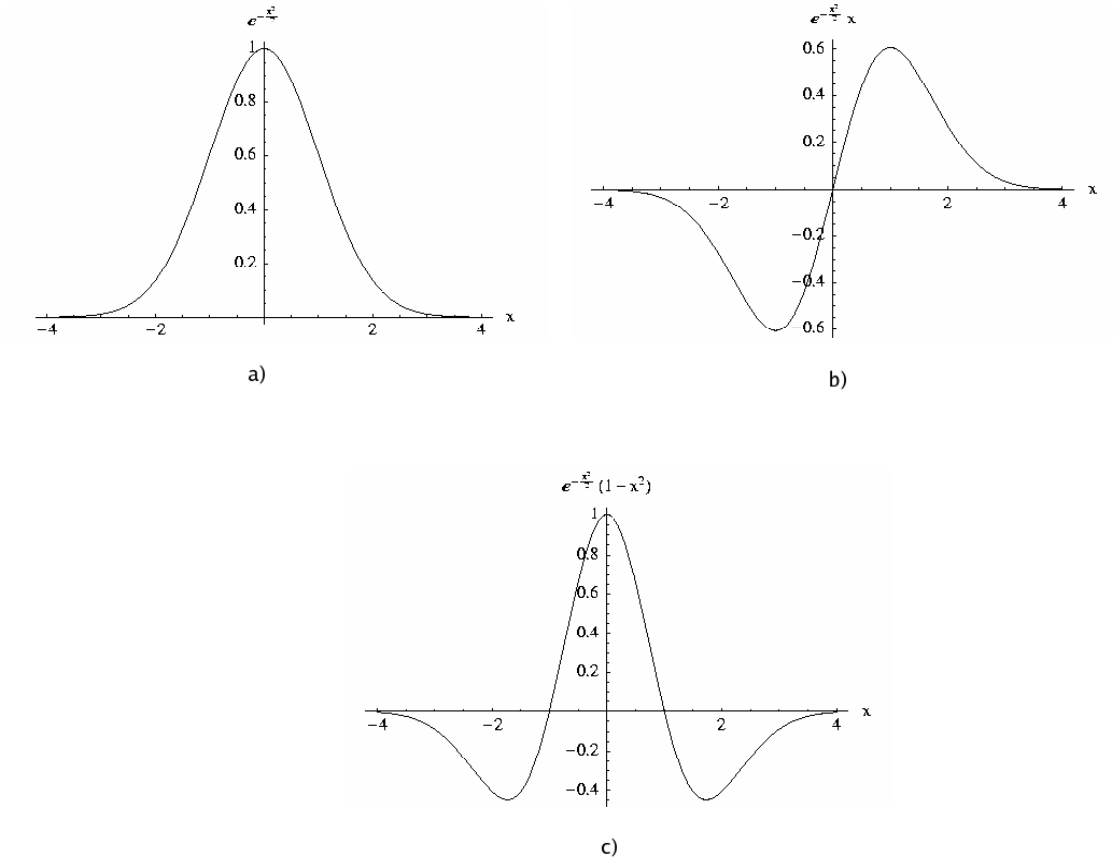


Figura 3: a) Función Gaussiana. b) Primera derivada de la Función Gaussiana. c) Segunda derivada de la Función Gaussiana

la función de autocorrelación es alta para todos los cambios de dirección (Schmid *et al.*, 2000). La función de autocorrelación se define como

Definición 1 (Función de Autocorrelación)**.** Dados $(\Delta x, \Delta y)$ y un punto $(x, y) \in \mathbb{R}^2$. La función de autocorrelación es dada por la siguiente relación:

$$f(x,y) = \sum_{(x_k,y_k)\in W} [I(x_k,y_k) - I(x_k+\Delta x, y_k+\Delta y)]^2 \tag{16}$$

donde (x_k, y_k) pertenecen a una región W centrada en el punto (x, y) de la imagen I.

Para utilizar esta función como un detector de puntos de interés o en su defecto como

detector de esquinas, se requiere realizar una integración sobre todas las direcciones donde se llevaron acabo los corrimientos. La integración sobre corrimientos discretos puede ser sustituida por la matriz de autocorrelación. Esta matriz se deriva al aplicar una aproximación de primer orden de la expansión de Taylor.

$$I(x_k + \Delta x, y_k + \Delta y) \approx I(x_k, y_k) + \begin{pmatrix} I_x(x_k, y_k) & I_y(x_k, y_k) \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad , \tag{17}$$

donde $I_x(x_k, y_k)$ y $I_y(x_k, y_k)$ indican la primera derivada parcial de la función de imagen I con respecto a x y y respectivamente. Substituyendo (17) en (16) se obtiene:

$$\begin{aligned} f(x,y) &= \sum_{(x_k,y_k)\in W} \left[\begin{pmatrix} I_x(x_k, y_k) & I_y(x_k, y_k) \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right]^2 \\ &= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \cdot \\ &\quad \begin{bmatrix} \sum_{(x_k,y_k)\in W} (I_x(x_k, y_k))^2 & \sum_{(x_k,y_k)\in W} I_x(x_k, y_k) I_y(x_k, y_k) \\ \sum_{(x_k,y_k)\in W} I_x(x_k, y_k) I_y(x_k, y_k) & \sum_{(x_k,y_k)\in W} (I_y(x_k, y_k))^2 \end{bmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \cdot \mathbf{M}(x,y) \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \end{aligned} \tag{18}$$

De esta forma la función de autocorrelación dada por la Ecuación (16) puede aproximarse por la matriz $\mathbf{M}(x, y)$ la cual describe el vecindario del punto (x, y). En la práctica se suele dar un peso extra a la matriz de autocorrelación $\mathbf{M}(x, y)$ por medio de un filtro Gaussiano $G(\sigma)$, esto es descrito en la Ecuación (19),

$$\mathbf{M}(x,y) = G(\sigma) \otimes \begin{bmatrix} (I_x(x_k, y_k))^2 & I_x(x_k, y_k) I_y(x_k, y_k) \\ I_x(x_k, y_k) I_y(x_k, y_k) & (I_y(x_k, y_k))^2 \end{bmatrix} \quad . \tag{19}$$

En este trabajo se calculará la matriz de autocorrelación $\mathbf{M}(x, y)$ como en la Ecuación (19).

II.1.4 Concepto de Punto de Interés

Uno de los procesos fundamentales de la visión por computadora es la detección y extracción de características de una imagen. Algunas características son los bordes, las esquinas y los puntos. La característica de más bajo nivel es el punto. Puede describirse por medio de dos coordenadas:

$$p = p(x, y) \quad \text{en 2D y} \quad p = p(x, y, z) \quad \text{en 3D donde} \quad x, y, z \in \mathbb{R} \, .$$

Los puntos de interés son identificados dentro de una imagen digital en forma de un pixel distinto a sus vecinos. Para una imagen en niveles de gris esto significa los extremos, el mínimo y el máximo. Es decir, son los puntos *prominentes* en la imagen pues muestran una propiedad distintiva, lo que los hace apropiados o de *interés* para varias tareas dentro la visión por computadora. Véase Figura 4. Por ejemplo, los bordes y las esquinas son puntos de interés. Las esquinas de tipo L, T e Y son un tipo específico de puntos de interés (Schmid *et al.*, 2000).

Los puntos de interés son necesarios para varias tareas de alto nivel como la detección y reconocimiento de objetos, correspondencia, y reconstrucción 3D, por mencionar algunos. Es decir, son los elementos base, indispensables, sin los cuales no se podrían desarrollar otras aplicaciones dentro de la visión por computadora. Sin embargo, su detección no es trivial, ya que los puntos de interés deben cumplir con ciertos criterios (McGlone *et al.*, 2004):

- **Distintivo** - Los puntos deben ser diferentes de sus puntos vecinos. Algunos autores proponen que los puntos de una línea recta no son apropiados.

- **Inusual** - Deben tener una separación global entre puntos de interés de una misma escena. Este criterio puede ayudar a reducir la complejidad en la correspondencia de imágenes al reducir el número de posibles correspondencias.

Figura 4: Puntos de interés detectados sobre una escena.

- **Invariante** - Debe ser invariante a distorsiones geométricas y radiométricas, por ejemplo, traslaciones, escalas y rotaciones.

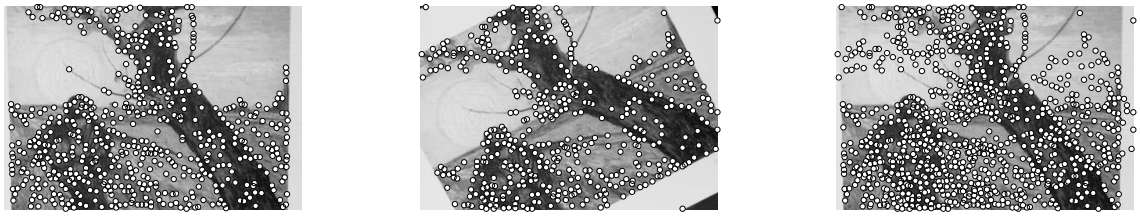
- **Estable** - Debe ser estable con respecto al ruido.

II.2 Detectores de Puntos de Interés

Un detector de puntos de interés ubica la posición (a nivel pixel o subpixel) de los puntos de interés dentro de una imagen.

Los detectores deben ser capaces de localizar características que sean simples de detectar, proveean información útil en nuevos procesos y presenten estabilidad geométrica bajo diferentes transformaciones. Además, para que los resultados alcanzados por el detector sean confiables, dentro de una misma escena siempre se deben detectar los mismos puntos, véase Figura 5. En este caso, una detección robusta de los puntos de interés

bajo transformaciones geométricas es posible bajo traslación, rotación y variación en la iluminación.



(a) (b) (c)

Figura 5: Secuencia de imágenes rotadas. (a) La imagen de referencia. (b) Imagen rotada 38 grados. (c) Imagen rotada 116 grados. Se puede observar que en las tres imágenes se detectaron los mismos puntos.

Los métodos de detección de puntos de interés y esquinas se pueden clasificar en 3 categorías:

- Los basados en contornos,
- Los basados en intensidad y
- Los basados en modelos paramétricos.

Los detectores basados en intensidad son referidos más apropiadamente como detectores de puntos de interés. En el capítulo IV se muestran varios detectores de cada categoría.

II.2.1 Métodos Basados en Contornos

Los métodos basados en contornos, operan sobre todos aquellos puntos que describen el contorno (i.e., puntos de frontera) de un objeto. Para ello, primero es necesario extraer los contornos. El punto de interés se identifica como el punto de frontera (PF) que tiene máxima curvatura o donde se localiza el punto de inflexión, de la estructura representada por la unión de todos los PF.

II.2.2 Métodos Basados en Intensidad

Los métodos basados en intensidad operan sobre los valores de intensidad en tonos de gris en una región de la imagen. El punto de interés es detectado a partir del cálculo de diversas propiedades geométricas de la superficie, basados en los principios de geometría diferencial.

A los métodos basados en intensidad, se les conoce en la literatura como métodos directos (Rohr, 1992), debido a que trabajan directamente con los valores de intensidad de la región donde esta ubicado el punto. La precisión de detección del punto de interés con estos métodos es a nivel de pixel (números enteros). Esta limitante, provoca que no sean aplicados en procesos de mayor nivel, como por ejemplo, la calibración de una camára, donde una precisión a nivel subpixel es requerida.

Algunos de los detectores basados en intensidad Harris y Stephens (1988); Förstner (1994); Shi y Tomasi (1994) utilizan como medida de interés, la distribución del gradiente alrededor de cada punto capturado por la matríz de autocorrelación.

Su funcionamiento es básicamente el siguiente: definen una función que opera sobre un vecindario local y extrae una medida de *interés* para cada pixel de la imagen. Esta operación produce una nueva imagen que puede ser referida como *imagen de interés*, véase Figura 6 , a la cual, se le aplica el método de *Supresión de no máximos* (Canny, 1983) para obtener los puntos con la medida de interés más alta. Una operación de umbralización es necesaria para obtener los puntos definitivos, debido a que la supresión de no máximos regresa algunos puntos falsos, esto, ocasionado por el ruido y las texturas finas.

Supresión de No Máximos

Canny (1983) propusó este método con el objetivo de reducir la frontera de los bordes a

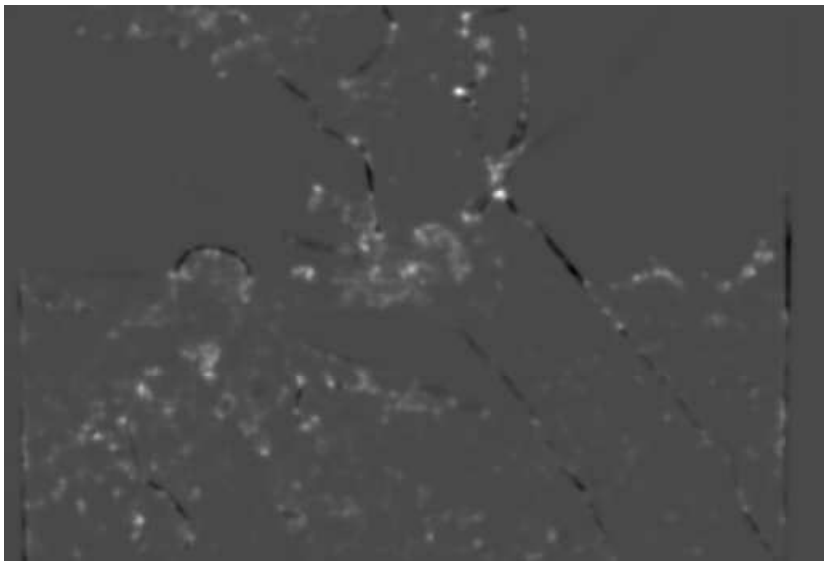


Figura 6: Imagen de interés antes de la supresión de no máximos.

un pixel. Consiste en obtener el punto más sobresaliente en una región (i.e., el máximo local) y eliminar los no sobresalientes (i.e., no máximos). Al final se obtiene una imagen que sólo contiene a los máximos locales, el resto de los pixeles fueron descartados al asignarles el valor de cero. Sin embargo, la supresión de no máximos puede detectar puntos falsos debido al ruido. Dado que el contraste de estos puntos es pequeño, se pueden reducir aplicando una umbralización.

En este trabajo la implementación de la supresión de no máximos se realiza de la siguiente manera:

1. Se establece el tamaño de la región sobre la cual se buscará el máximo local. Los más utilizados para este propósito son de 3×3, 5×5 y 7×7. En este trabajo la región se estableció de 7×7, debido a que un vecindario pequeño puede provocar que no se identifique al máximo local correcto en picos anchos (i.e., regiones de interés) donde varios pixeles adyacentes tienen el mismo valor.

2. Se realiza una dilatación morfológica de la imagen. Esto es, para cada pixel de la imagen, se examinan sus vecinos en busca del máximo, posicionando dicho pixel como el centro del vecindario, véase Figura 7. Una vez encontrado, los $n \times n$ elementos de la región son igualados al máximo de la región encontrado. Véase Figura 8.

3. Se compara la *imagen de interés* con la imagen dilatada. Los puntos que son iguales en ambas imágenes (i.e., máximos locales) conservan su valor, de lo contrario se igualan a cero. Así, se obtiene una imagen que sólo contiene a los máximos locales.

4. Se aplica una operación de umbralización a la imagen que sólo contiene a los máximos locales. Los puntos que resulten de esta operación son los puntos de interés. En esta tesis, la umbralización la implementamos de 2 maneras. A continuación se explican ambos criterios.

7	3	8
4	i, j	2
6	1	5

Figura 7: Ejemplo de un vecindario de 3×3, teniendo al pixel i, j como centro.

Los dos criterios que implementamos para determinar un umbral en esta tesis son:

1. Establecer un Valor Mínimo para considerarlo Punto de Interés o

2. Establecer un Número Fijo de Puntos a obtener.

Establecer un Valor Mínimo para considerarlo Punto de Interés.- Es importante determinar cual es el valor apropiado, ya que si el umbral es bajo, se estarán detectando

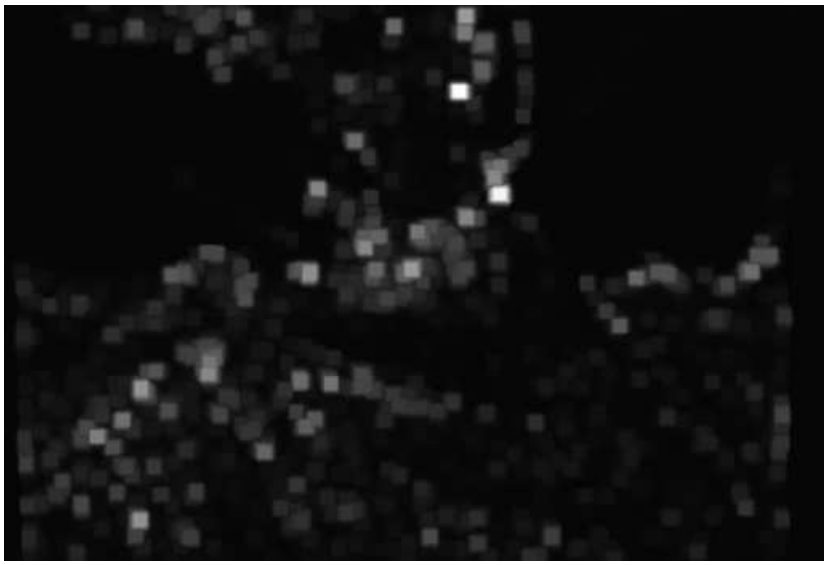


Figura 8: Imagen dilatada morfológicamente.

falsos positivos, y si por el contrario, el umbral es muy alto, se discriminarán puntos correctos.

Así, el valor del umbral se determina, estableciendo un porcentaje con respecto al punto con la mayor medida de intensidad observada. Así, todos los puntos que sean mayores al umbral, serán considerados puntos de interés. Por ejemplo, el detector de Harris Harris y Stephens (1988) establece un umbral del 1% del punto con mayor medida de intensidad.

Establecer un Número Fijo de Puntos a obtener.- Con este criterio, el umbral es determinado de acuerdo a las necesidades del usuario. El umbral se establece como el n número de puntos que se requiere obtener. Para ello, los puntos (i.e., máximos locales) deben ser ordenados descendentemente de acuerdo a su medida de intensidad y así se toman los n puntos al inicio de la lista.

En el presente trabajo utilizamos ambos métodos. En el capítulo IV utilizamos el

criterio 1, en el capítulo V utilizamos ambos, el criterio 2 en la aplicación genética y el criterio 1 en las pruebas realizadas.

II.2.3 Métodos Basados en Modelos Paramétricos

Los detectores paramétricos se basan en expresiones analíticas que modelan las variaciones de intensidad en una imagen digital. Proveen precisión a nivel sub-pixel, pero están limitados a ciertos tipos de puntos de interés, como las esquinas en L. Los parámetros de los modelos describen los efectos ópticos, físicos y geométricos del objeto. Estos parámetros generalmente son:

- El ángulo de apertura de la esquina,
- La magnitud de la intensidad fuera y dentro de la esquina,
- Posición espacial de la esquina y
- El grado de incertidumbre (difuminado) de la esquina.

Algunos de estos detectores contienen los siguientes elementos:

1. Una función escalón unitaria que representa a un borde ideal.
2. Un filtro Gaussiano o exponencial que emula el efecto de difuminado de la estructura.
3. Un proceso de convolución entre el filtro Gaussiano y el borde ideal. El proceso de convolución es necesario para representar el efecto de difuminado en el modelo de esquina ideal.
4. Una descripción de los parámetros modelados.

5. Una región de la imagen en la vecindad donde se ubica el punto de esquina. Generalmente una región cuadrada.

6. Un método para la obtención de los mejores valores de los parámetros que se ajusten a la imagen en tonos de gris en la región en estudio. Generalmente, los parámetros se ajustan a la imagen a través de métodos de optimización multidimensionales, (descenso de gradiente, mínimos cuadrados).

Por tanto, la ubicación de la esquina es conocida una vez realizado el ajuste de los parámetros de los modelos a la imagen original. El ajuste proporciona una primera aproximación en la ubicación de la esquina. Es importante enfatizar, *que el proceso de convolución de la esquina ideal y el fitro Gaussiano provoca un desplazamiento del punto de esquina* (Rohr, 1992).

II.3 Evaluación de Detectores de Puntos de Interés

Como se mecionó en la sección II.1.4, los puntos de interés deben cumplir con ciertos criterios. Por ello, es necesario contar con un criterio eficiente y razonable para evaluar la confiablidad de los resultados obtenidos por los detectores de puntos de interés. El único criterio para el cual existe una métrica ampliamente aceptada (i.e., tasa de repetibilidad) es el de estabilidad (Schmid *et al.*, 2000). El cual, probablemente sea el más importante de ellos. Sin embargo, también existe el criterio de contenido de información (Schmid *et al.*, 2000). Estos dos criterios miden la calidad de las características para tareas como reconocimiento de objetos, correspondencia y reconstrucción 3D. Se aplican a cualquier tipo de escena, y no dependen de un modelo específico de característica o interpretación de alto nivel de esta. Existen otros criterios como la precisión en la localización, el cual es importante para tareas como la calibración de cámaras. Sim embargo, para el objetivo del presente trabajo, sólo se incorporan el criterio de

repetibilidad.

Antes de presentar la repetibilidad, es necesario introducir el concepto de homografía.

II.3.1 Homografía

Dado un conjunto de puntos x_i en $\mathbb{P}^2$ y su conjunto de puntos correspondientes x_i' también en $\mathbb{P}^2$, la proyección proyectiva que lleva cada x_i a x_i' es la homografía 2D. Los puntos x_i y x_i' son puntos en dos imágenes, considerando cada imagen como un plano proyectivo $\mathbb{P}^2$ (Hartley y Zisserman, 2004).

Así, para un conjunto de puntos correspondientes $x_i \leftrightarrow x_i'$ entre dos imágenes, existe una matríz H de tamaño 3×3 tal que

$$Hx_i = x_i'$$

para cada i.

A fin de conocer la homografía entre dos imágenes se realizan los siguientes pasos. Primero es necesario establecer cuantos puntos correspondientes $x_i \leftrightarrow x_i'$ son requeridos para calcular la transformación proyectiva H. Un límite inferior surge al considerar el número de grados de libertad y el número de restricciones. Por un lado, la matríz H contiene 9 entradas, pero se encuentra sujeta a un factor de escala. Así, el número total de grados de libertad es 8. Por el otro lado, cada correspondencia punto a punto cuenta con dos restricciones, ya que para cada punto x_i en la primera imagen los dos grados de libertad del punto de la segunda imagen deben corresponder al punto mapeado Hx_i. Un punto 2D tiene dos grados de libertad correspondientes a los componentes x y y, cada uno de los cuales puede ser especificado por separado. Alternativamente, el punto es especifíficado como un 3-vector homogéneo, el cual también tiene dos grados de libertad puesto que la escala es arbitraria. Como consecuencia es necesario especificar 4

puntos correspondientes con el objetivo de restringir H completamente.

Si cuatro puntos correspondientes son dados, entonces una solución exacta para la matríz H es posible. Esta es la solución mínima. Sin embargo, ya que los puntos son medidos inexactamente, debido al ruido, si se proporcionan más de 4 puntos correspondientes "inexactos", tal vez estos puntos no sean completamente compatibles con ninguna transformación proyectiva, y así, se estará frente a la tarea de determinar la "mejor" transformación posible.

Existen varios métodos para realizar el cálculo de H. A continuación se presentan tres métodos. Uno de ellos es un método lineal y los otros dos son métodos para el cálculo robusto de la homografía.

Algoritmo de Transformación Lineal Directa (DLT)

Este es un método lineal para determinar H dado un conjunto de 4 puntos correspondientes de 2D a 2D $x_i \leftrightarrow x_i'$. La transformación está dada por la ecuación $x_i' = Hx_i$. Nótese que está ecuación involucra vectores homogéneos; así los 3-vectores x_i' y Hx_i no son iguales, tienen la misma dirección pero difieren en magnitud por un factor de escala diferente de cero. La ecuación puede ser expresada en términos del producto cruz de vectores como $x_i' \times Hx_i = 0$. Esta forma permite derivar una solución lineal simple para H.

Si el j-ésimo renglón de la matríz H es denotado por h^{jT}, entonces podemos escribir

$$Hx_i = \begin{pmatrix} h^{1T}x_i \\ h^{2T}x_i \\ h^{3T}x_i \end{pmatrix} \quad .$$

Escribiendo $x_i' = (x_i', y_i', w_i')^T$, el producto cruz puede ser dado explícitamente en

notación de Einstein como

$$x'_i \times Hx_i = \begin{pmatrix} \mathbf{y'_i} h^{3T} x_i - \mathbf{w'_i} h^{2T} x_i \\ \mathbf{w'_i} h^{1T} x_i - \mathbf{x'_i} h^{3T} x_i \\ \mathbf{x'_i} h^{2T} x_i - \mathbf{y'_i} h^{1T} x_i \end{pmatrix} \quad .$$

Puesto que $h^{jT} x_i = x_i^T h^j$ para $j = 1, ..., 3$, esto da un conjunto de 3 ecuaciones en las entradas de H, que pueden ser escritas en la forma

$$\begin{bmatrix} 0^T & -\mathbf{w'_i} x_i^T & \mathbf{y'_i} x_i^T \\ \mathbf{w'_i} x_i^T & 0^T & -\mathbf{x'_i} x_i^T \\ \mathbf{y'_i} x_i^T & \mathbf{x'_i} x_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0. \tag{20}$$

Estas ecuaciones tienen la forma $A_i h = 0$, donde A_i es una matríz de tamaño 3×3, y h es un 9-vector conformado por las entradas de la matríz H,

$$h = \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix}, \quad H = \begin{bmatrix} \mathbf{h_1} & \mathbf{h_2} & \mathbf{h_3} \\ \mathbf{h_4} & \mathbf{h_5} & \mathbf{h_6} \\ \mathbf{h_7} & \mathbf{h_8} & \mathbf{h_9} \end{bmatrix} \quad , \tag{21}$$

con $\mathbf{h_i}$ el i-ésimo elemento de h. Tres comentarios con respecto a estas ecuaciones se presentan en orden:

1. La ecuación $A_i h = 0$ es una ecuación lineal para h. Los elementos de la matríz A_i son cuadráticos en las coordenadas conocidas de los puntos.

2. Aunque existen tres ecuaciones en (20), sólo 2 de ellas son linealmente independientes (dado que el tercer renglón es obtenido, por factor de escala, de la suma de x'_i veces el primer renglón y y'_i veces el segundo). Así cada punto correspondiente proporciona 2 ecuaciones en las entradas de H. Es usual omitir la tercer ecuación (Sutherland, 1963). Entonces, el conjunto de ecuaciones se convierte en

$$H = \begin{bmatrix} 0^T & -\mathbf{w'_i} x_i^T & \mathbf{y'_i x_i^T} \\ \mathbf{w'_i x_i^T} & 0^T & -\mathbf{x'_i} x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0. \tag{22}$$

Esto es escrito

$$A_i h,$$

donde A_i es ahora la matríz de tamaño 2×9 de la Ecuación 22. Si, alguno de los puntos x'_i es un punto ideal, tal que $\mathbf{w'_i} = 0$, entonces el par de Ecuaciones (22) colapsan a una sóla ecuación. El conjunto de Ecuaciones (20) aún tiene 2 ecuaciones lineales independientes, y en este caso la tercer ecuación de 20 no debe ser omitida (aunque quizá alguna de las dos primeras pueda serlo). Es posible que el incluir las tres ecuaciones resulten en un conjunto de ecuaciones mejor condicionado para conjuntos de puntos generales.

3. Las ecuaciones se mantienen para cualquier representación de coordenadas homogéneas $(x'_i, y'_i, w'_i)^T$ del punto x'_i. Se puede seleccionar $w'_i = 1$, lo que significa que (x'_i, y'_i) son las coordenadas medidas en la imagen.

Resolviendo para H

Cada punto correspondiente da lugar a dos ecuaciones independientes en las entradas de H. Dado un conjunto de cuatro puntos correspondientes, se obtiene un conjunto de ecuaciones $Ah = 0$, donde A es la matríz de los coeficientes de las ecuaciones, la cual es creada a partir de los renglones A_i de cada punto correspondiente, y h es el vector de incognitas de H. Entonces se busca una solución h diferente de cero, ya que la solución trivial $h = 0$ no es de interés para nuestro propósito. Si la Ecuación (20) es usada, entonces A tiene un tamaño de 12×9, y si se usa la Ecuación (22) el tamaño es de 8×9. En cualquier caso, A tiene rango 8, y así tiene un espacio nulo de 1 dimensión que proporciona una solución para h. Tal solución h, sólo puede ser determinada a un factor de escala diferente de cero. Se puede seleccionar arbitrariamente una escala para h, pero cumpliendo con el requerimiento de que su norma sea $\|h\| = 1$.

A continuación se presenta el algoritmo paso por paso del método DLT:
Dado $n \geq 4$ puntos correspondientes de 2D a 2D, $x_i \leftrightarrow x_i'$, determinar la matríz de homografía H tal que $x_i' = Hx_i$.

1. Para cada punto correspondiente $x_i \leftrightarrow x_i'$ calcular la matríz A a partir de la Ecuación (20). Sólo los primeros dos renglones necesitan utilizarse en general.

2. Ensamblar las n matrices A_i de tamaño (2×9) en una sola matríz A de tamaño $2n \times 9$.

3. Obtener SVD de A. El vector singular correspondiente al valor singular más pequeño es la solución h. Específicamente, si $A = UDV_T$ con diagonal D la cual contiene entradas positivas ordenadas de forma descendente sobre la diagonal. Por lo que se considera h como la última columna de V.

4. La matríz H es determinada a partir de h como en la Ecuación(21).

Normalización

El método de normalización, consiste en una traslación y escalamiento de las coordenadas de la imagen. La normalización debería realizarse antes de aplicar el algoritmo para encontrar la homografía. Subsecuentemente, una corrección apropiada del resultado expresa H con respecto al sistema coordenado original.
Además de mejorar la exatitud de los resultados, la normalización de los datos provee un segundo beneficio deseado. Así el algoritmo que incorpora una normalización inicial de datos será invariante con respecto a elecciones arbitrarias de escala y origen coordenado. Esto, debido a que la normalización deshace el efecto de cambios en las coordenadas al selecionar un marco coordenado canónico para los datos medidos. Así,

una minimización algebraica es efectuada en un marco canónico fijo y el algoritmo utilizado es practicamente invariante a transformaciones expresadas por una similitud. Es importante mencionar, que algunos sistemas coordenados son mejores para el cálculo de homografías, de ahí la importancia de llevar a cabo la normalización.

Algoritmo de Transformación Lineal Directa (DLT) Normalizado

El resultado del algoritmo DLT depende del marco coordenado en el cual se expresan los puntos. De hecho, el resultado no es invariante a transformaciones de similitud en la imagen. Para lograr que el resultado sea invariante, se le aplica normalización a los datos antes de utilizar el algoritmo DLT.

El algoritmo DLT normalizado es el siguiente:
Dado $n \geq 4$ puntos correspondientes de 2D a 2D, $x_i \leftrightarrow x_i'$, determinar la matríz de homografía H tal que $x_i' = Hx_i$.

1. Normalización de x: Calcular una transformación de similitud T consistente de una traslación y escalamiento, que toma puntos x_i a un nuevo conjunto de puntos $\tilde{x_i}$, tal que el centroide de los puntos $\tilde{x_i}$ es el origen coordenado $(0,0)^T$ y su distancia promedio del origen es $\sqrt{2}$.

2. Normalización de x_i': Calcular una transformación de similitud T' para los puntos en la segunda imagen, transformando x_i' a $\tilde{x_i'}$.

3. DLT: Aplicar el algoritmo básico DLT a las correspondencias $\tilde{x_i} \leftrightarrow \tilde{x_i'}$ para obtener la homografía $\tilde{H}$.

4. Denormalización: Establecer $H = T'^{-1}\tilde{H}T$.

Algoritmo RANSAC

Hasta este punto se ha asumido, que la única fuente de error está en la medida de la posición de los puntos, la cual sigue una distribución Gaussiana. Sin embargo, esta suposición no es válida porque existen puntos que no encajan en el error de distribución Gaussiana. A estos puntos se les llama *aberrantes* (del inglés *outliers*). Los puntos aberrantes pueden alterar la homografía estimada y por ello deben ser identificados. El objetivo es determinar un conjunto de puntos congruentes (del inglés *inliers*) de las "correspondencias" presentadas, de tal manera que la homografía pueda ser estimada en una manera óptima (i.e., robusta). Se dice que es una *estimación robusta* (i.e., tolerante) a puntos aberrantes (los cuales son medidas que siguen una distribución de error diferente que posiblemente no ha sido modelada).

El algoritmo *Random Sample Consensus* que significa Consenso de muestra aleatoria (RANSAC) es un método robusto para el cálculo de una homografía 2D, ya que puede lidiar con una gran porción de puntos aberrantes (Fischler y Bolles, 1981).

La idea es simple: dos puntos son seleccionados aleatoriamente; estos puntos definen una línea. El *soporte* para esta línea es medido por el número de puntos que caen a una distancia umbral de la línea. Esta selección aleatoria es repetida un número de veces y la línea con más soporte es designada como el ajuste al modelo más robusto. Los puntos dentro del umbral de distancia son los puntos congruentes (y constituyen el conjunto de consenso). La intuición es que si uno de los puntos es un punto aberrante la línea no tendrá mucho soporte, véase Figura 9. El umbral de distancia por lo general, es seleccionado empíricamente. Sin embargo, el método más sencillo para determinar el error de una correspondencia (umbral de distancia) es usar el error simétrico de transferencia (i.e., $d^2_{transfer} = d\left(x, H_{-1}x'\right)^2 + d\left(x', Hx\right)^2$) donde $x \leftrightarrow x'$ son los puntos

correspondientes.

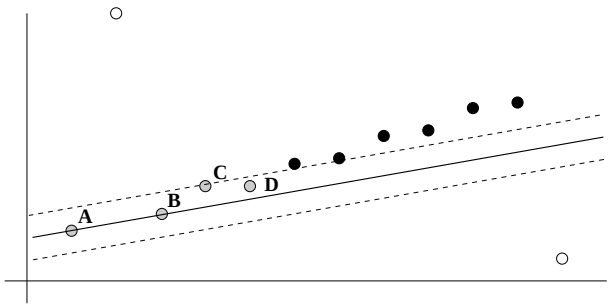


Figura 9: Técnica RANSAC. Los puntos grises representan *inliers*, en este caso el modelo al que deben ajustarse los puntos es una línea recta, en líneas punteadas se aprecia el margen de holgura.

Dicho más generalmente, se desea ajustar un modelo (en este caso una línea) a datos y la muestra aleatoria consiste en un subconjunto mínimo de los datos (en este caso se representan como dos puntos) para determinar el modelo. Si el modelo es una homografía planar y los datos un conjunto de puntos correspondientes 2D; entonces el conjunto mínimo consiste de cuatro puntos correspondientes.

El algoritmo de RANSAC es el siguiente:
El objetivo es encontrar el ajuste robusto de un modelo al conjunto de datos S que contiene puntos aberrantes.

1. Seleccionar aleatoriamente una muestra de s puntos de S y representar el modelo con este subconjunto.

2. Determinar el conjunto de puntos S_i que se encuentran dentro de un umbral de distancia t del modelo. El conjunto S_i es el conjunto de consenso de la muestra y define los puntos congruentes de S.

3. Si el tamaño de S_i (el número de inliers) es mayor que un umbral T, reestimar el modelo usando todos los puntos en S_i y terminar.

4. Si el tamaño de S_i es menor que T, seleccionar un nuevo subconjunto y repetir los pasos anteriores.

5. Después de N intentos el conjunto de consenso más grande S_i es seleccionado, y el modelo es reestimado usando todos lo puntos en el subconjunto S_i.

Medianas Mínimas Cuadradas (Least Median Squares)

Este método asume que todas las medidas pueden ser interpretadas con el mismo modelo. Por ello, detecta muy fácilmente los datos que se encuentran fuera de la norma, lo que lo convierte en un método de estimación robusta. El método realiza una búsqueda en el espacio de soluciones a partir de los subconjuntos obtenidos con el mínimo de correspondencias requeridas. Si se necesitan un mínimo de cuatro puntos correspondientes para calcular la transformación proyectiva y se tienen un total de n correspondencias, entonces el espacio de búsqueda estará formado por las combinaciones de n elementos tomados de cuatro en cuatro. Como este espacio puede ser muy grande, varios subconjuntos de 4 puntos son seleccionados aleatoriamente.
El algoritmo para obtener la homografía con este método es el siguiente:

1. Utilizar una técnica Monte-Carlo para seleccionar m subconjuntos de cuatro puntos.

2. Para cada subconjunto S, se calcula una solución H.

3. Para cada H, se calcula la mediana M_S de los cuadrados de las distancias geométricas con respecto a todos los puntos correspondientes.

4. Se selecciona la homografía H con menor M_S.

Una vez que se obtiene la solución, los puntos aberrantes se seleccionan de entre aquellos con mayor distancia geométrica. Los puntos correspondientes buenos (i.e., puntos

congruentes) son seleccionados de entre aquellos con menor distancia a un umbral establecido. Cuando los puntos malos han sido rechazados, una mejor solución puede ser obtenida con los puntos seleccionados (puntos congruentes) y utilizando el algoritmo de las medianas mínimas cuadradas. Este es el método que se utiliza para calcular las homografías empleadas en esta tesis.

II.3.2 Repetibilidad

La repetibilidad compara explícitamente la estabilidad geométrica de los puntos de interés detectados entre diferentes imágenes de una escena tomada bajo condiciones de vista variantes. Un punto es "repetido", si el punto 3D de la escena detectado en la primera imagen también es detectado con exactitud en la segunda imagen. Así, la repetibilidad significa que la detección es independiente de cambios en las condiciones de la imagen, como los parámetros de la cámara, su posición relativa en las escena y en las condiciones de iluminación.

Dado un punto 3D "X" y 2 matrices de proyección P_1 y P_i, las proyecciones de "X" en las imágenes I_1 e I_i son

$$x_1 = P_1 X \quad \text{y} \quad x_i = P_i X.$$

Un punto x_1 detectado en la imagen I_1 es repetido en la imagen I_i si el punto correspondiente x_i es detectado en la imagen I_i.

Para medir la repetibilidad, se tiene que establecer una relación única entre x_1 y x_i. Esto es difícil para escenas 3D, pero en el caso de escenas planas esta relación esta definida por una homografía

$$x_i = H_{1i} x_1 \text{ donde } H_1 i = P_i P_1^{-1} \ .$$

Una Homografía 2D es una matríz de tamaño 3×3 que realiza la transformación proyectiva que lleva a cada punto de I_i a I_i'. Tanto I_i como I_i' están en el plano proyectivo, véase Figura 10.

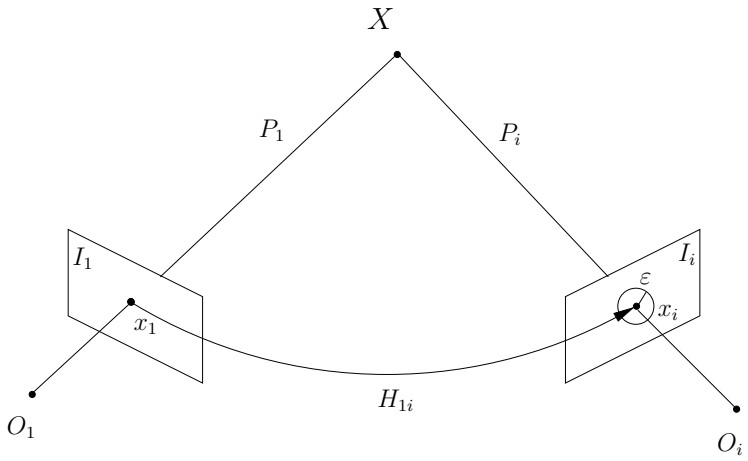


Figura 10: En el caso de escenas planas, x_1 y x_i están relacionadas por la homografía H_{1i}.

La tasa de repetibilidad esta definida como el número de puntos repetidos entre 2 imágenes con respecto al número total de puntos detectados. Los puntos que debido a los cambios en las condiciones de la imagen no puedan ser observados en ambas imágenes corrompen la medida de repetibilidad. Esta parte común de la escena es determinada por la homografía. Los puntos $\bar{x}_1$ y $\bar{x}_i$ que están en la parte común de la imágenes I_1 e I_i están definidas por

$$\{\bar{x}_1\} = \{\bar{x}_1 | H_{1i} x_1 \in I_i\} \text{ y } \{\bar{x}_i\} = \{\bar{x}_i | H_{i1} x_i \in I_1\} .$$

Por lo general, un punto no es detectado exactamente en la posición x_i sino en un vecindario de x_i el cual es definido por el tamaño del vecindario ε. Así el par de puntos ($\bar{x}_1$,$\bar{x}_i$) que corresponden dentro de un vecindario ε esta definido por

$$R_i(\varepsilon) = \{(\bar{x}_1, \bar{x}_i) | dist(H_{1i}\bar{x}_1, \bar{x}_i) < \varepsilon\}.$$

La tasa de repetibilidad $r_i(\varepsilon)$ para la imagen I_i esta definida por

$$r_i(\varepsilon) = \frac{|R_i(\varepsilon)|}{min(n_1, n_i)},$$

donde $n_1 = |\{\bar{x}_1\}|$ y $n_i = |\{\bar{x}_i\}|$ son el número de puntos detectados en las partes comunes de las imágenes I_1 e I_i.

La repetibilidad en este trabajo es implementada para evaluar la capacidad del detector para determinar regiones correspondientes. El error de localización ε es de 2.5 pixeles. Las regiones son elípticas, y están definidas por $a(x-u)^2 + 2b(x-u)(y-v) + c(y-v)^2 = 1$ donde u y v son las coordenadas del punto de interés.

La implementación se realiza de la siguiente manera:

1. Se realiza la proyección de las regiones de ambas imágenes. Para esto, se efectúa una transformación afín de la imagen 1 a la imagen 2 y viceversa. Es necesario contar con las homografías de las imágenes.

2. Se busca la parte en común de ambas imágenes. Esto es, que las regiones de la imagen 1 se encuentren en la imagen 2, y viceversa.

3. Se tomá el número mínimo de regiones que ambas imágenes tienen en común.

4. Una vez que se tienen las regiones en común, se calcula el traslape de cada una contra las demás. El traslape esta dado por la razón *interseccion/union*. Para considerar que un punto de las elipses (i.e., regiones) se encuentra en la intersección o unión de 2 elipses, se calculan las distancias de ese punto al centro de ambas elipses. Si ambas distancias son menor a 1, el punto se encuentra en la intersección. Si sólo una de las distancias es menor a 1, el punto forma parte de la unión.

5. Se calcula el error de traslape de cada región, el cual esta dado por $e = 1 - (interseccion/union)$.

6. El resultado devuelto por esta implementación es el porcentaje de regiones que presentan el 10%, 20%, 30%, 40%, 50% y 60% de error de traslape. Para ello se divide el número de regiones con un determinado porcentaje de error entre el número mínimo de regiones que ambas imágenes tienen en común.

II.3.3 Contenido de Información

El contenido de información es una medida de qué tan distintivo es un punto de interés. Esto se basa en el vecindario del descriptor calculado para el punto de interés. Los descriptores caracterizan la forma local de la imagen para cada punto de interés. El contenido de información mide la distribución de estos descriptores. Si todos los descriptores se encuentran juntos, no proporcionan ninguna información. Esto es, la información de contenido es baja. Así pues, la correspondencia fallaría, ya que cada punto podría corresponder a cualquier otro punto. Si por el contrario, los descriptores estuviesen distribuidos por toda la imagen, la información de contenido sería alta, y la correspondencia no fallaría.

El contenido de información de los descriptores es medido usando la entropía. La entropía mide la aleatoriedad de una variable. Mientras más aleatoria sea una variable más grande es la entropía. La entropía de una partición $A = \{A_i\}$ es

$$H(A) = -\sum_i p_i \ log(p_i),$$

donde p_1 es la probabilidad de A_i. Nótese que el tamaño de la partición influye en el resultado. Si B es una nueva partición formada de las subdivisiones de los conjuntos de A entonces $H(B) \geq H(A)$. En el caso de puntos de interés, se desea conocer cuanta información de contenido promedio "transmite" un punto de interés.

II.4 Conclusiones

Los conceptos básicos tratados en este capítulo permiten comprender la operación de los detectores de puntos de interés que se exponen en el capítulo IV. Así como la comprensión de las diferentes métricas que permiten conocer el desempeño de cada detector para determinadas tareas de la visión por computadora. Lo anterior es parte fundamental de este trabajo ya que los conceptos de repetibilidad y contenido de información son claves para la propuesta de esta tesis.

En los capítulos siguientes las definiciones descritas se utilizan constantemente. Estos conceptos en conjunto con la teoría de la programación genética, la cual se presenta en el siguiente capítulo, permiten desarrollar el enfoque propuesto en este trabajo.

Capítulo III

Programación Genética

¿Cómo pueden las computadoras aprender a resolver problemas sin ser programadas explícitamente? En otras palabras: ¿Cómo pueden construirse computadoras para hacer lo que se necesite hacer, sin decirles exactamente lo que se tiene que hacer?. La respuestas a este tipo de preguntas, son respondidas por primera vez por el paradigma de Programación Genética (PG) el cual surge como una metodología con la que se formaliza el procedimiento de búsqueda de soluciones desde el punto de vista de la inducción de programas, que busca encontrar un programa de computadora dentro del espacio posible de programas de computadora.

A continuación se describe el surgimiento de la Programación Genética, los puntos principales de su metodología, asi como algunas de sus características.

Para la presentación de estos conceptos, primeramente se describe el surgimiento del paradigma del computo evolutivo, dentro del cual se encuentra la PG. Enseguida se habla de la inducción de programas y la manera como se implementa utilizando la PG. Posteriormente, se define la representación del espacio de soluciones utilizando un conjunto de funciones primitivas, y se comentan las principales funciones que controlan la ejecución así como algunas características de la PG.

III.1 Surgimiento del Paradigma

El estudio de sistemas adaptativos artificiales en varios campos de la ciencia e ingeniería que utilizan técnicas empleadas por sistemas adaptativos naturales (Fleming y Fonseca, 1993; Fogel, 1994) representa hoy en día un área de interés en el desarrollo de sistemas que se comporten de una manera más inteligente (Forrest *et al.*, 1993) y robusta en ambientes impredecibles (Fleming y Fonseca, 1993; Fogel, 1994). Este paradigma se propone con el fin de lograr desarrollar máquinas que participen junto con el diseñador en la resolución de problemas (Forrest, 1990; Forrest *et al.*, 1993). Esta área es conocida como Computación Evolutiva.

Tradicionalmente, la investigación en los campos de inteligencia artificial, sistemas de auto-mejora y auto-organización, aprendizaje de máquina e inducción en general, utiliza métodos correctos, consistentes, justificables, ciertos (deterministas), metódicos, parsimoniosos y decisivos (que tengan un punto de terminación bien definido). Una razón del escepticismo inicial de las personas que abordan la Programación Genética es que es difícil imaginar que estos siete principios, que están virtualmente integrados en nuestro entrenamiento y pensamiento, no deben ser usados para resolver todos los problemas, ya que han jugado roles invaluables en la solución exitosa en muchos problemas de ciencias, ingeniería y matemáticas. Dado que las ciencias de la computación están fundadas en la lógica, es especialmente difícil para sus practicantes imaginar que estos siete principios guía no deban usarse para resolver cada problema. Sin embargo, en la PG no se utilizan en el sentido tradicional sino que se emplean los principios de la naturaleza (i.e., la evolución natural).

III.1.1 La Evolución Natural

Los seres vivos estan divididos en especies. Con el tiempo, el cual puede ser corto o largo, las especies cambian (i.e., evolucionan). Existen varios componentes esenciales

en la evolución natural. Los individuos dentro de una población de una especie son diferentes. Como resultado, algunos viven más tiempo y son más probables de tener hijos que sobrevivan a la edad madura con respecto a otros (selección natural). Se dice que estos individuos son más aptos (supervivencia de los más aptos). Algunas veces la variación tiene un componente genético, (i.e., los hijos tal vez lo hereden de sus padres). El desarrollo de cada individuo es controlado en parte por sus genes. El ambiente se encarga de decidir si las variantes incorporadas a la especie permanecen o no. Consecuentemente después de un número de generaciones la proporción de individuos dentro de la especie con estas características hereditarias favorables tienden a incrementarse. Así, las especies, después de un tiempo cambian o "evolucionan", preservándose aquellas que poseen mejores características.

Muchas de las especies que estamos acostumbrados a ver utilizan la reproducción sexual para producir hijos. La reproducción sexual significa que los genes de cada hijo son una combinación de los genes de sus padres. Al proceso en que los genes de ambos padres son seleccionados, manipulados y unidos para producir un nuevo conjunto de genes para los hijos es llamado cruzamiento ("crossover" en inglés) o recombinación.

El por qué la naturaleza inventó la reproducción sexual no es claro, pero tiene varias propiedades importantes. Primero, los hijos son genéticamente diferentes de ambos padres. Segundo, si los individuos en la población contienen diferentes conjuntos de genes asociados con alta aptitud, el hijo de 2 padres con alta aptitud tal vez herede ambos conjuntos de genes. Así, la reproducción sexual proporciona un mecanismo por el cual combinaciones beneficiosas de genes existentes pueden ocurrir en un solo individuo. Como este individuo puede tener hijos, la nueva combinación de genes puede distribuirse por toda la población.

El proceso de copiar genes es problemático. A pesar de que la naturaleza ha evolucionado mecanismos que al copiar genes limitan los errores, estos se siguen presentando. Estos errores son conocidos como mutaciones. Si los errores ocurren al copiar genes a celdas que serán utilizadas para producir hijos, la mutación será transmitida a los hijos. Se piensa que la mayoría de mutaciones son dañinas, sin embargo, algunos cambios incrementan la aptitud de los individuos. Al transmitir esto a los hijos, la mejora de aptitud tal vez se distribuya a toda la población en generaciones posteriores.

Estas ideas inspiraron a numerosos científicos a aplicar conceptos similares para la resolución de problemas en sus respectivos campos. Actualmente se han desarrollado un gran número de técnicas consideradas como evolutivas.

III.1.2 Computación Evolutiva

El término Computación Evolutiva ha sido creado específicamente para una descripción global de todos los diferentes estilos de computación, o métodos basados en adaptación evolutiva (i.e., el proceso evolutivo natural). El objetivo es lograr desarrollar inteligencia artificial en base a un estudio detalladado de las diferentes características, procedimientos y circunstancias que se presentan en el proceso evolutivo de la vida que permitan una mejor comprensión así como su aplicación en la generación de tecnología.

Algunos de los estilos predominantes de la computación evolutiva son:

- Estrategias de Evolución (ES),
- Algoritmos Genéticos (AG),
- Programación Genética (PG),
- Programación Evolutiva,

En este capítulo se hace una descripción detallada de la Programación Genética. A continuación se presenta un breve resumen del resto de los estilos predominantes de la computación evolutiva.

Estrategias de Evolución.- Define un estilo de computación evolutiva frecuentemente asociado con problemas de optimización en ingeniería. Las estructuras, las cuales van más allá de la adaptación son típicamente conjuntos de valores reales, variables-objetivo, las cuales están asociadas con valores reales, o variables de estrategias en lo individual. La aptitud es determinada con la tarea de ejecutar rutinas específicas y algoritmos usando variables-objetivo como parámetros. Las variables de estrategia controlan la forma en la cual la mutación hace variar cada variable-objetivo durante la producción de nuevos individuos.
La recombinación es característica de este estilo de programación (conocida como "crossover" en los algoritmos genéticos) es usualmente empleada tanto por las variables objetivo como por las variables de estrategia. Además posee las siguientes características: Enfatiza el lazo conductual entre padres e hijos. Realiza una selección estrictamente determística. La representación de las estructuras son análogas a los individuos, por lo que usa operaciones de recombinación para generar nuevas estructuras. Rechenberg en 1965 comenzó el campo. Para mayor información ver, (Bäck *et al.*, 1991) y (Bäck y Scwefel,1993) citados en (Forrest, 1993).

Algoritmos Genéticos.- Operan sobre una cadena de caracteres de longitud fija, binaria, en tanto la estructura sobrevive a la adaptación. La aptitud es determinada por la ejecución de una rutina, utilizando una interpretación de la cadena de caracteres como el conjunto de parámetros. La recombinación sexual o "crossover", es el principal operador genético empleado, junto con mutación usualmente incluido como un

operador de importancia secundaria. Otras estructuras representacionales son posibles (representaciones octales, hexadecimales, utilizando el alfabeto, etc.). Enfatiza el lazo conductual a nivel genético. Ver también el trabajo de Holland (Holland, 1992) y el libro de Goldberg (Goldberg, 1993) para futuras referencias.

Programación Evolutiva.- Este estilo opera sobre una variedad de representación de estructuras, frecuentemente valores reales, variables-objetivo aunque estructuras más complejas han sido usadas (e.g., máquinas de estado finito). De nuevo, las variables-objetivo funcionan como argumentos a tareas de rutina específicas y algoritmos diseñados para resolver el problema de interés. El único operador genético empleado es el de mutación, con una estrategia significativa de construcción en el algoritmo global para dirigir la búsqueda en una dirección fructífera. Además, posee las siguientes características: enfatiza el lazo conductual entre poblaciones reproductivas así como la naturaleza probabilística de selección para la conducción de un torneo estocástico hacia la supervivencia de cada generación. La representación de las estructuras es análoga a las distintas especies (poblaciones reproductivas) por lo que no existen poblaciones de recombinación, ya que no hay comunicación sexual entre especies. El campo fue comenzado por Fogel, Owens y Walsh en 1966. Se puede encontrar más información en (Fogel, 1993) citado en (Forrest, 1993).

Las primeras aplicaciones de estas técnicas evolutivas aparecen en los años 60's. En la década de los 80's hubo un período de indiferencia, pero posteriormente se dió un resurgimiento en los años 90's.

III.1.3 Programación Genética

El paradigma de Programación Genética, fue propuesto por John Koza en octubre de 1987. La idea de la PG le surgio en un vuelo al regresar de una conferencia de Inteligencia Artificial. La aplicó por primera vez en la resolución de un par de ecuaciones lineales e induciendo la secuencia Fibonacci.

La PG es una extensión de los algoritmos genéticos (AG). Pero es considerada más una generalización que una especialización de estos. Difiere de los AG en la forma en que representa a los individuos de la población, pues utiliza programas de computadora en lugar de cadenas de longitud fija. En la PG, los individuos, es decir, los programas, son representados como árboles sintácticos o como su correspondiente expresión en notación prefija.

La meta de la PG es lograr que las computadoras aprendan a resolver problemas sin ser explícitamente programadas (i.e., programación automática). Un impedimento para que las computadoras resuelvan problemas sin ser explícitamente programadas es que los métodos actuales de aprendizaje de máquina, inteligencia artificial, sistemas de auto-mejoramiento y redes neuronales buscan soluciones en formas que pueden facilitar la solución a ciertos problemas, y muchos de ellos facilitan análisis matemáticos que de otra manera no podrían ser posibles. Sin embargo, estas estructuras especializadas son una manera innatural y restringida de hacer que las computadoras resuelvan problemas sin ser explícitamente programadas, ya que las computadoras no son programadas en lenguajes de vectores de peso, árboles de decisión, gramáticas formales, clusters conceptuales, coeficientes polinomiales, reglas de producción o cadenas de cromosomas. La realidad es que si se desea que las computadoras resuelvan problemas sin ser explícitamente programadas, las estructuras que se deben emplear son programas

de computadora.

Los programas de computadora ofrecen la flexibilidad de:

- Realizar operaciones en una manera jerárquica,
- Realizar cálculos alternativos condicionados al resultado de los cálculos intermedios,
- Realizar iteraciones y recursiones,
- Realizar cálculos con variables de diferentes tipos, y
- Definir valores intermedios y subprogramas de tal manera que puedan ser reusados subsecuentemente.

Cuando se dice que las computadoras deben resolver problemas sin ser programadas explícitamente, se tiene en mente que no se debe especificar de antemano el tamaño, la forma, y la complejidad estructural de la solución. En su lugar, estos atributos de la solución deben emerger durante el proceso de resolución del problema como resultado de las demandas del problema. El tamaño, forma, y complejidad estructural deben ser parte de la respuesta producida por la técnica de resolución del problema y no parte de la pregunta. Por estas razones, la PG trata de generar soluciones a problemas a partir de la inducción de programas. Así, la PG es un proceso general de búsqueda en el espacio de todos los posibles programas que solucionen un problema.

III.2 Inducción de Programas

Dentro del espacio de posibles programas de computadora, la inducción de programas involucra el descubrimiento inductivo de un programa de computadora que produzca alguna salida deseada cuando se le presenta alguna entrada en particular. Y esto es precisamente lo que la metodología de PG realiza de una manera sistematizada. Con base

en este planteamiento un programa de computadora puede ser llamado una fórmula, un plan, una estrategia de control, un procedimiento computacional, etc. Similarmente, las entradas del programa de computadora pueden ser llamadas variables independientes, variables de estado, valores de sensores, argumentos de una función, etc. A su vez, las salidas del programa de computadora pueden denominarse variables dependientes, un movimiento, un actuador, el valor regresado por una función, etc.

Sin importar la diferencia en terminología, las razones para reformular los problemas planteados por la inducción de programas como la búsqueda de un programa de computadora, son tres:

1. Los programas de computadora tienen la flexibilidad necesaria para expresar soluciones a una amplia variedad de problemas.

2. Los programas de computadora pueden tomar el tamaño, la forma y la complejidad estructural necesarios para resolver estos problemas.

3. La PG proporciona una metodología para realizar inducción de programas.

Con esto se ve que los programas pueden ser el lenguaje para expresar varios problemas. En la Tabla II se presentan diversas áreas que pueden ser expresadas como un problema de inducción de programas.

La Tabla II usa la terminología empleada para describir las entradas, salidas y el programa de computadora de diferentes áreas de problemas, desde la perspectiva de un problema de inducción de programas. Se observa que diferentes problemas pueden ser reformulados de esta manera. Por ejemplo, el control óptimo consiste en encontrar una estrategia (de control), representada por un programa de computadora, que use las variables del estado actual de un sistema para escoger un valor en las variables de

control que causen que el estado del sistema se mueva hacia la meta deseada, mientras se maximiza o minimiza alguna medición de costo.

Tabla II: Algunas áreas de problemas que pueden ser expresados como un problema de inducción de programas

Area del Problema	Programa de Computadora	Entrada	Salida
Control óptimo	Estrategia de Control	Variables de Estado	Variables de Control
Planeación	Plan	Valores de Sensores o de Detectores	Acciones de Actuadores
Diseño de Filtros Adaptativos	Filtro FIR o IIR	Señal de Datos y Señal de Ruido	Señal de Datos
Descubrimiento de Identidades Matemáticas	Nuevas Expresiones Matemáticas	Muestra Aleatoria de Valores de las Variables Independientes de la Expresión Matemática Dada	Valores de la Expresión Matemática Dada
Programación Automática de un Automáta Celular	Reglas de Transición de Estado para la Célula	Estado de la Célula y sus Vecinos	Próximo Estado de la Célula

Un ejemplo de un problema de control óptimo simple involucra descubrir una estrategia de control para centrar un carro en una pista en un tiempo mínimo. En este ejemplo, las variables de estado del sistema son la posición y la velocidad del carro. La estrategia de control especifica cómo escoger la fuerza que es aplicada al carro, pues

la aplicación de la fuerza causa que el estado del sistema cambie. El estado destino deseado es que el carro descanse en el punto central de la pista.

III.3 Áreas de Aplicación de la PG

Las áreas en las cuales se puede aplicar la PG son las que presentan algunas de las siguientes características :

1. Las interrelaciones entre las variables relevantes son pobremente entendidas (o donde se sospecha que el actual entendimiento pueda estar equivocado).

2. Determinar el tamaño y forma de la solución es una parte muy importante del problema.

3. El análisis matemático convencional no puede, o no provee soluciones analíticas.

4. Una solución aproximada es aceptable (o es el único resultado probable de obtener).

5. Pequeñas mejoras en el desempeño son medidas rutinariamente (o medibles fácilmente) y altamente valoradas.

6. Existe una gran cantidad de datos, que se encuentran en un formato legible por las computadoras, que requieren examinación, clasificación e integración (tales como la biología molecular de las secuencias de proteínas y ADN, datos astronómicos, datos de observación satelital, datos financieros, datos de transacciones mercantiles o datos en la world wide web).

III.4 Descripción Detallada de la PG

Las diferentes técnicas evolutivas, incluidas la PG, tienen como componentes fundamentales los siguientes:

- Una población de individuos, cada uno de los cuales representa una solución al problema planteado,
- Un mecanismo de reproducción con el que se producen cambios en las características de los diferentes individuos,
- Un criterio de evaluación de las diferentes soluciones del problema en cuestión y
- Un mecanismo de selección por el cual ciertas soluciones son destinadas a ser las generadoras de la siguiente población de individuos y a iniciar un nuevo ciclo evolutivo.

Una vez que hemos visto el objetivo y los componentes de la programación genética, veremos cómo se formaliza el problema de inducción de programas en este paradigma. Para la implementación de la PG es necesario especificar:

1. El conjunto de terminales (constantes y variables),
2. El conjunto de funciones primitivas,
3. La medición de la Aptitud,
4. Los parámetros de las operaciones genéticas y
5. El criterio para terminar la ejecución.

III.4.1 Conjuntos de Terminales y Funciones Primitivas

Las estructuras que evolucionarán en la PG, como ya se mencionó, son programas de computadora estructurados jerarquicamente. Los conjuntos de terminales y las funciones primitivas integran la fase de representación del dominio del problema por medio de un programa de computadora. El conjunto de las estructuras utilizadas en la PG, son el conjunto de todas las posibles composiciones de funciones que pueden ser creadas

recursivamente a partir del conjunto de funciones $\in F = \{f_1, f_2, ..., f_{N_{func}}\}$ y el conjunto de terminales $\in T = \{a_1, a_2, ..., a_{N_{term}}\}$ (i.e., $C = F \cup T$). Cada función f_i del conjunto de funciones F tiene un número específico $z(f_i)$ de argumentos $z(f_1)$,$z(f_2)$,...,$z(f_{N_{func}})$. Así, la función f_i tiene aridad $z(f_i)$.

Las funciones del conjunto de funciones F pueden incluir:

- Operaciones aritméticas (+,-,*,/.),
- Funciones matemáticas (sen,cos,exp,log,etc.),
- Operaciones Booleanas (AND,OR,NOT,etc.),
- Operadores condicionales (IF-THEN-ELSE,etc.),
- Funciones que causan iteraciones (DO-UNTIL,etc.),
- Funciones que causan recursión y,
- Funciones de dominio específico que puedan ser definidas.

El conjunto de terminales, está compuesto por variables (que pueden representar, tal vez, las entradas, sensores, detectores, o variables de estado de algunos sistemas) o constantes (como el número 3 o la constante Booleana NIL).

Tanto el conjunto de funciones F como el de terminales T, deben satisfacer las propiedades de *cerradura* y *suficiencia*. La propiedad de cerradura requiere que cada una de las funciones pueda aceptar como argumentos cualquier valor y tipo de datos que posiblemente sea regresado por cualquier función, así como cualquier valor y tipo de datos que cualquier terminal pueda tomar. Es decir, que cada función del conjunto de

funciones F debe estar bien definida y *cerrada* para cualquier combinación de argumentos que pueda encontrar. Por ejemplo, si la operación aritmética de división puede encontrar como su segundo argumento (i.e., divisor) el valor numérico de 0, la propiedad de cerradura no se cumplirá, salvo, que se realice algún arreglo para lidiar con la posibilidad de la división por 0. Un enfoque para satisfacer la cerradura, es definir una función de división protegida. La cual, recibe 2 argumentos, regresa el valor numérico de 1 si intenta realizar división entre 0, si no es asi, realiza la división y regresa el resultado obtenido.

La propiedad de suficiencia requiere que el conjunto de terminales y funciones sea capaz de expresar una solución al problema. El usuario de la PG debe saber o creer que alguna composición de las funciones y terminales son *suficientes* para encontrar la solución al problema. El paso de identificar las variables que tienen poder explicativo suficiente para resolver un problema en particular, es común en prácticamente todos los problemas de la ciencia. A pesar de que existan ejemplos ilustrativos de cómo seleccionar un conjunto de funciones y terminales suficientes en (Koza, 1992), es el usuario quien debe proporcionar el conjunto C apropiado para su problema.

Una vez definido el conjunto de funciones primitivas y terminales que conformarán el programa genético, el siguiente paso es conocer el algoritmo o ciclo evolutivo de la PG.

III.4.2 Algoritmo de la PG

Existen dos enfoques para la ejecución de un programa genético, el enfoque generacional y el enfoque de estado estable. Tradicionalmente, la PG utiliza el enfoque generacional (Banzhaf *et al.*, 1998). A continuación se presentan los algoritmos de ambos enfoques.

Algoritmo del Enfoque Generacional

En este enfoque existen generaciones bien definidas. Cada generación es representada por una población completa de individuos. Una población nueva es creada a partir de la generación anterior. La nueva generación reemplaza a la generación anterior y el ciclo continua. El ciclo de ejecución consiste en los siguientes pasos:

1. Inicializar la población.
2. Evaluar a los individuos de la población existente. Asignándole una aptitud a cada individuo.
3. Repetir los siguientes pasos hasta que la nueva población sea completada:
    - Seleccionar un individuo o individuos de la población usando un método de selección.
    - Realizar operaciones genéticas en el individuo o individuos seleccionados.
    - Insertar el resultado de las operaciones genéticas en la nueva población.
4. Si el criterio de paro es cumplido, ir al siguiente paso. De lo contrario, reemplazar la población existente con la nueva población y repetir los pasos del 2 al 4.
5. Presentar al mejor individuo de la población como el resultado obtenido del algoritmo.

Algoritmo del Enfoque de Estado Estable

Este enfoque es la principal alternativa al generacional. En el enfoque de estado estable no existen las generaciones. En su lugar, existe un flujo continuo de individuos que se reproducen. Los hijos reemplazan individuos dentro de la misma población. Este

método es fácil de implementar y presenta grandes beneficios tales como la paralelización.

El algoritmo consiste en los siguientes pasos:

1. Inicializar la población.
2. Seleccionar aleatoriamente un subconjunto de la población que formará parte del torneo (i.e., competidores).
3. Evaluar la aptitud de cada competidor.
4. Seleccionar al ganador o ganadores del torneo usando algún método de selección.
5. Aplicar operadores genéticos al ganador o ganadores del torneo.
6. Reemplazar a los perdedores del torneo con los resultados que se obtuvieron del paso anterior.
7. Repetir los pasos del 2 al 7 hasta que el criterio de paro se cumpla.
8. Seleccionar al mejor individuo de la población como el resultado obtenido del algoritmo.

Este método es llamado de estado estable porque los operadores géneticos son aplicados asíncronamente y no existe un mecanismo centralizado para generaciones explícitas.

En esta tesis se emplea el enfoque generacional. En la Figura 11 se muestra el ciclo evolutivo de la PG y en la Figura 12 se observa el diagrama de flujo.

A continuación se explica cada una de las etapas del ciclo evolutivo.

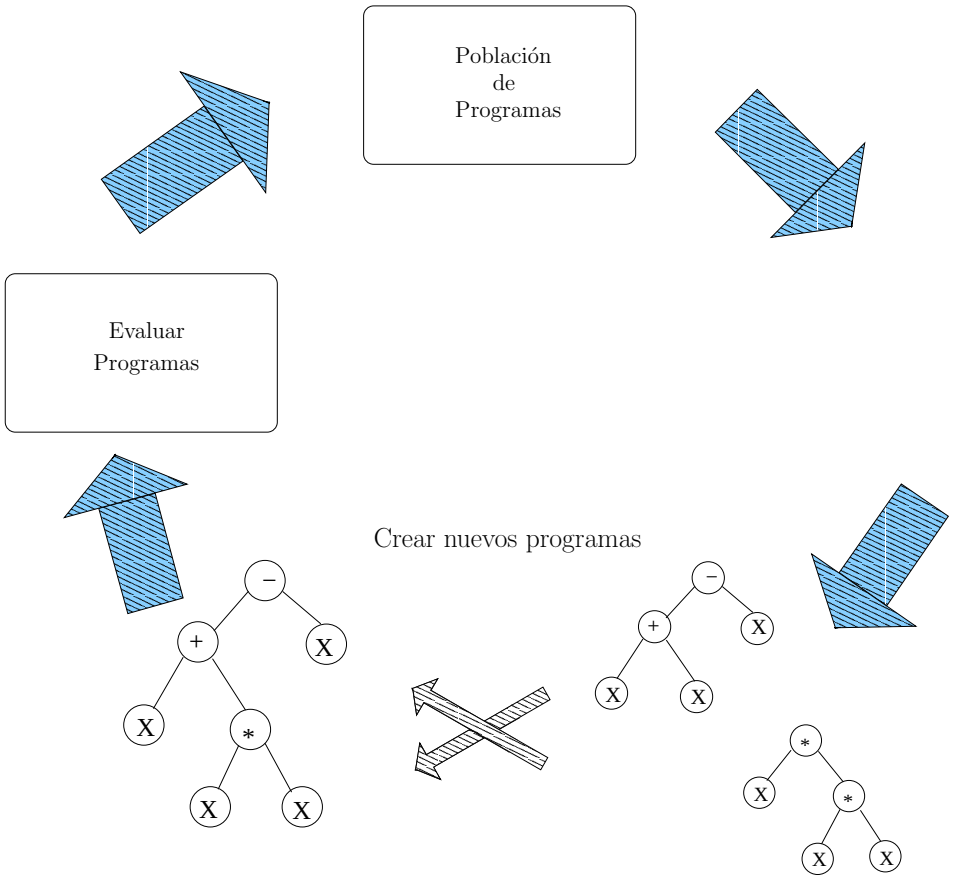


Figura 11: Ciclo evolutivo de la PG. El ciclo de PG es como todo proceso evolutivo. Nuevos individuos (en la PG nuevos programas) son creados. Son evaluados. Los más aptos de la población triunfan al crear hijos. Los menos aptos mueren y son removidos de la población.

Generación de la Población

La PG utiliza una población de individuos (i.e., soluciones) con el propósito de que el proceso de búsqueda de la solución al problema en cuestión se realice en paralelo.

Una nueva población (o generación) es creada a partir de la población anterior. Esto por medio de la evolución. Sin embargo, es necesario contar con una población inicial. La creación de una buena población inicial, esto es, que presente diversidad de variedad, es decisiva para una ejecución exitosa o no del programa genético.

Cada individuo de la población es creado de selecciones aleatorias del conjunto de

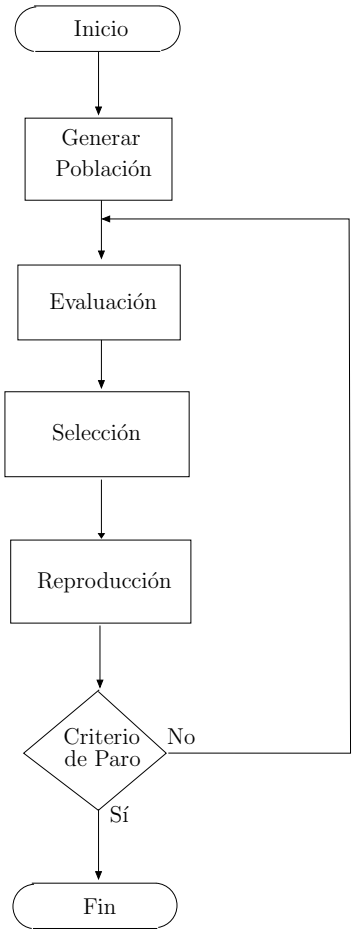


Figura 12: Diagrama de flujo general del algoritmo de enfoque generacional.

funciones F y terminales T. La creación de individuos duplicados en la población inicial es improductivo, ya que se desperdician recursos computacionales. Ademas de que reduce la variedad genética de la población. Por ello, es deseable, pero no necesario, evitar los duplicados durante la inicialización de la población.

En la PG existen 3 métodos para inicializar una población. Estos métodos son:

- Completo (del inglés Full).
- Crecimiento (del inglés Grow).
- Expansión Mitad-y-Mitad (del inglés Ramped Half-and-Half).

Completo (Full).- Este método crea árboles donde la longitud de la ruta desde cualquier nodo terminal hasta el nodo raíz es igual a la profundidad máxima. Esto se logra, restringiendo la selección sólo al conjunto de funciones F, de los nodos que se encuentren en una profundidad menor a la máxima, y restringiendo la selección sólo al conjunto de terminales T de los nodos que se encuentran en la profundidad máxima. El resultado, es que cada rama del árbol llega hasta la máxima profundidad. Con este método se generan árboles que presentan la misma forma. Esto es, que todos los árboles tienen en un mismo nivel a los nodos terminales u hojas. En la Figura 13 se muestra un árbol construido con este método.

Crecimiento (Grow).- Con este método se obtienen árboles que presentan formas variadas. La longitud de la ruta desde cualquier nodo terminal hasta el nodo raíz no es mayor a la profundidad máxima. Esto se logra haciendo la selección aleatoria de los nodos con profundidad menor a la máxima, del conjunto combinado $C = F \cup T$, el cual consiste de la unión del conjunto de funciones con el conjunto de terminales. Mientras que la selección de los nodos que se encuentran en la profundidad máxima, se restringe al conjunto de terminales T.En la Figura 14 se muestra un árbol construido con este método.

Expansión Mitad-y-Mitad (Ramped Half-and-Half).- Este es un método mixto que incorpora tanto el método Completo como el de Crecimiento. Consiste en crear un número igual de árboles usando un parámetro de profundidad, cuyo rango varia de 2 a la máxima profundidad especificada. Por ejemplo, si la profundidad máxima especificada es de seis, entonces 20% de los árboles tendrán profundidad dos, 20% tendrán profundidad tres y así sucesivamente hasta la profundidad seis. Posteriormente, para cada valor de profundidad (i.e., 2,3,4,5 y 6), el 50% de los árboles son creados con

el método Completo y el 50% restante con el método Crecimiento. Este método crea árboles que presentan una amplia variedad de tamaños y formas. En la Figura 15 se muestran varios árboles construidos con este método.

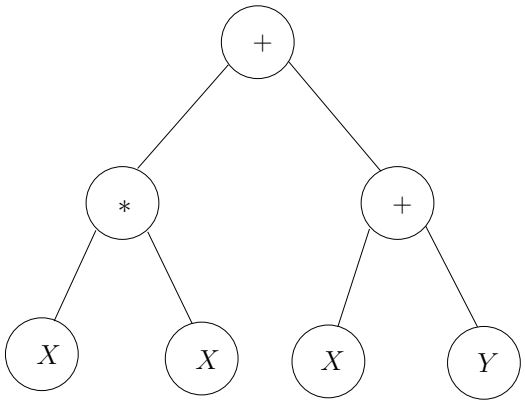


Figura 13: Árbol creado con el método Completo.

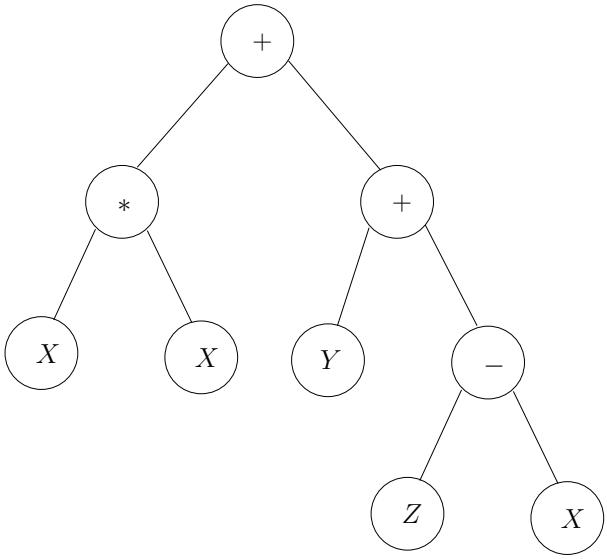


Figura 14: Árbol creado con el método de Crecimiento.

Evaluación

En la naturaleza, la aptitud de un individuo es la probabilidad de que sobreviva a la fase de la evolución y se reproduzca. En el mundo artificial de los algoritmos genéticos,

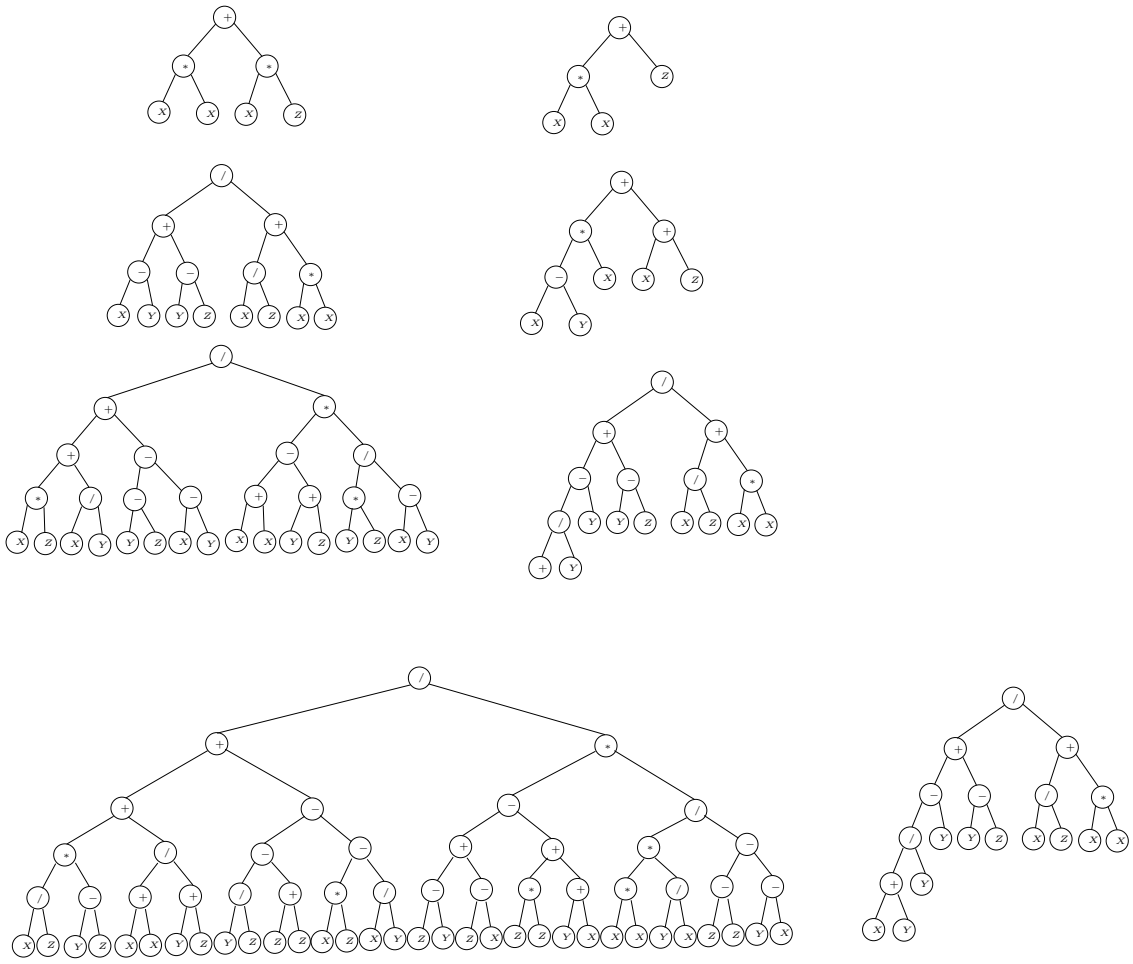


Figura 15: Población de 8 individuos compuesta de árboles creados con el método Expansión Mitad-y-Mitad. Los niveles de profundidad son 2,3,4 y 5. Por lo tanto, cada nivel de profundidad debe tener a 25% de los individuos (i.e., 2 individuos), 50% de ellos (i.e., 1 individuo) se generan con el método Completo y 50% con el método de Crecimiento.

como se describe en Koza (1992), la aptitud de los individuos es evaluada y entonces se usa para controlar la aplicación de las operaciones que modificaran las estructuras en nuestra población artifical.

El hecho de que los individuos existan y sobrevivan en la población así como el que se reproduzcan exitosamente puede ser indicativo de su aptitud, tal como lo es en la naturaleza.

En la PG, cada individuo en la nueva población de programas de computadora es evaluado por la *Función Aptitud* (del inglés *Fitness Function*), que mide qué tan bien se desempeña cada estructura-solución en función del problema que se desea resolver, es decir, califica el lugar que ocupa en el espacio de búsqueda. La función aptitud es el elemento con el cual se decide quién perdura o no, repitiéndose el proceso por muchas generaciones.

En general, este algoritmo producirá poblaciones de programas de computadora que a lo largo de muchas generaciones tenderán a exhibir un incremento en el valor de la aptitud promedio con respecto a su ambiente. Además, estas poblaciones de programas de computadora pueden rápida y efectivamente adaptarse a los cambios en el ambiente. El mejor individuo que aparece en cada generación de la corrida se designa como el resultado producido por la programación genética.

A cada individuo se le asigna una aptitud escalar a través de un procedimiento de evaluación bien definido. El cual permitirá seleccionar a los individuos sobre los que se aplicarán los operadores genéticos.

En PG se utilizan cuatro casos de aptitud, los cuales son:

- Cruda (del inglés Raw).
- Estandarizada (del inglés Standardized).
- Ajustada (del inglés Adjusted).
- Normalizada (del inglés Normalized).

Cruda.- Es la medida de aptitud que se encuentra en la terminología natural del problema a resolver. Por ejemplo, la aptitud cruda en el problema de la hormiga es el número de piezas de comida encontrada. La mayor cantidad de comida encontrada, es la óptima. Así, la aptitud cruda tiene un rango de 0 (i.e., la menor cantidad de comida encontrada, por lo tanto, el peor caso) a 89 (i.e., la mayor cantidad de comida posible de encontrar, por lo tanto, es el mejor caso). La aptitud es usualmente, pero no siempre, evaluada sobre un conjunto de casos de aptitud. Estos casos de aptitud proveen una base para evaluar los individuos de la población sobre un número diferente de situaciones representativas suficientemente grande, de manera que se puede obtener un rango de diferentes valores de aptitud cruda. Los casos de aptitud son típicamente solo una pequeña muestra de todo el espacio del dominio (que usualmente es muy grande o infinito). Para funciones booleanas con pocos argumentos, es práctico usar todas las posibles combinaciones de los argumentos como casos de aptitud. Los casos de aptitud deben ser representativos de todo el espacio del dominio, porque ellos forman las bases que permiten generalizar los resultados obtenidos al espacio del dominio.

La aptitud cruda también puede ser usada como el error que presenta un individuo. Cuando se define como un error, la aptitud cruda $r(i,t)$ de un individuo i de una

población de tamaño M en cualquier paso generacional t es

$$r(i,t) = \sum_{j=1}^{N_e} \mid S(i,j) - C(i,j) \mid \ ,$$

donde $S(i,j)$ es el valor regresado por el individuo i para el caso de aptitud j de N_e casos y donde $C(i,j)$ es el valor correcto para la aptitud del caso j.

Debido a que la aptitud cruda indica la terminología natural del problema, el mejor caso puede ser tanto pequeña (en el caso de ser un error) o grande (cuando representa comida encontrada, beneficio alcanzado, etc.)

Estandarizada.- La aptitud estandarizada $s(i,t)$ redefine la aptitud cruda, tal que un valor pequeño siempre es un buen valor. Si para un problema en particular, un valor pequeño de aptitud cruda es mejor, la aptitud estandarizada es igual a la aptitud cruda. Esto es,

$$s(i,t) = r(i,t).$$

Es conveniente y deseable que el mejor valor de la aptitud estandarizada sea igual a cero. Si este no es el caso, puede hacerse sustrayendo o sumando una constante. Si para un problema en particular, un valor grande de aptitud cruda es mejor, la aptitud estandarizada debe ser calculada a partir de la aptitud cruda. Por ejemplo, en el problema de la hormiga, se trata de maximizar la cantidad de comida descubierta a lo largo de una ruta, así, un valor grande de aptitud cruda es mejor. En este caso, la aptitud estandarizada es igual al máximo valor posible de aptitud cruda (denotada por r_{max}) menos la aptitud cruda encontrada. Esto es, necesitamos revertir la aptitud cruda,

$$s(i,t) = r_{max} - r(i,t) \ .$$

Si la hormiga encuentra 5 de las 89 piezas de comida usando un programa de computadora dado, la aptitud cruda es de 5 y la estandarizada es de 84.

Ajustada.- La aptitud ajustada $a(i,t)$ es calculada a partir de la aptitud estandarizada $s(i,t)$ de la siguiente manera

$$a(i,t) = \frac{1}{1+s(i,t)} \ ,$$

donde $s(i,t)$ es la aptitud estandarizada del individuo i en el tiempo t.

La aptitud ajustada se encuentra entre 0 y 1. Los mejores individuos de la población presentan un valor grande de aptitud ajustada.

No es necesario usar la aptitud ajustada en la PG, sin embargo, es generalmente útil. Esto debido a que tiene el beneficio de exagerar la importancia de pequeñas diferencias en el valor de la aptitud estandarizada a medida que esta se aproxima a cero (como frecuentemente ocurre en las últimas generaciones de la ejecución del programa genetico). Así, a medida que la población mejora, se enfatizan las pequeñas diferencias que hacen la diferencia entre un buen individuo y uno muy bueno. Esta exageración es especialmente potente si la aptitud estandarizada alcanza cero cuando es encontrada una solución perfecta. Por ejemplo, si el rango de la aptitud estandarizada varía entre 0 (el mejor) y 64 (el peor), la aptitud ajustada de dos individuos malos con 64 y 63 de aptitud estandarizada es de 0.0154 y 0.0159 respectivamente. La aptitud ajustada de dos buenos individuos con 4 y 3 de aptitud estandarizada es de 0.20 y 0.25 respectivamente. Este efecto es menos potente (aun así valioso) cuando el valor de aptitud estandarizada no puede definirse para alcanzar 0 en caso del mejor individuo, como sucede en problemas de optimización donde el mejor valor mínimo diferente de cero no es conocido por adelantado.

Nótese que para ciertos métodos de selección diferentes a la selección por aptitud

proporcional (e.g. selección de torneo y selección por rango), la aptitud ajustada no es relevante ni usada.

Normalizada.- Si el método de selección esta basado en la proporción de la aptitud de los individuos, el concepto de aptitud normalizada es necesario. La aptitud normalizada $n(i,t)$ de una población de tamaño N es calculada a partir de la aptitud ajustada $a(i,t)$ de la siguiente manera

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{N} a(k,t)} .$$

La aptitud normalizada tiene 3 características deseables:

- Que su rango se encuentre entre 0 y 1,
- Su valor es grande para los mejores individuos de la población,
- La suma de las aptitudes normalizadas es 1.

Nótese que para ciertos métodos de selección diferentes a la selección por aptitud proporcional (e.g. selección de torneo y selección por rango), la aptitud normalizada no es relevante ni usada.

Selección

La selección es uno de los operadores genéticos. Su principal objetivo es identificar buenas soluciones y determinar qué individuos dejarán descendencia y en qué cantidad. De esta manera la búsqueda se orienta hacia aquellas soluciones más promisorias. La selección permite que los individuos más aptos se reproduzcan con mayor frecuencia y con menor frecuencia los menos aptos, negando en ocasiones el derecho de reproducción de algunos individuos. Existen varios métodos de llevar a cabo la selección. Los más comunes son:

- Selección Proporcional o Ruleta.
- Selección por Rango.
- Selección por Torneo.

Selección Proporcional o Ruleta.- En este tipo de selección, el número de veces en que cada individuo obtiene el derecho de reproducción es proporcional a su valor de aptitud. El mecanismo de este operador funciona como una ruleta (ver Figura 16, inciso (b)), donde la ruleta tiene N divisiones que representan el tamaño de la población. El ancho de cada división está en proporción a la función aptitud de cada miembro. La superficie de la ruleta que le corresponde a cada individuo, (i.e., la probabilidad de un individuo de ser seleccionado) es

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad ,$$

donde p_i es la probabilidad del individuo i de ser seleccionado, $\sum_{j=1}^{N} f_j$ es la sumatoria de los valores de aptitud de toda la población y f_i es el valor de aptitud del individuo i. La ruleta girará M veces, según el número de padres que necesite la nueva población, eligiendo cada vez al individuo que señale el indicador de la ruleta.

Selección por Rango.- En este método, los individuos se ordenan en función a su aptitud. A esto se le llama rango. El rango varia de 1 (el mejor individuo) a el tamaño de la población (el peor individuo). Una vez que se tiene a la población por rango, a cada individuo se le asigna un nuevo valor de aptitud proporcional al rango. Esto se hace mediante una función lineal o exponencial.

Algunas distribuciones de aptitudes contienen anomalias, tales como que algún individuo presente una aptitud extraordinariamente buena en comparación al resto de los

individuos de la población o a una gran cantidad de ellos, quienes presentan una aptitud diferente pero indistinguible. La selección por rango tiene el efecto de separar la distribución de la aptitud del proceso de selección. Si muchos individuos tienen aptitudes casi idénticas, el programa genético tal vez requiera una gran cantidad de generaciones para diferenciar los individuos ligeramente mejores de los individuos ligeramente peores. La selección por rango exagera la diferencia entre aptitudes casi idénticas y acelera la convergencia. Si un individuo tiene un muy buen valor de aptitud relativo a los otros individuos (e.g., quizá la presencia de un individuo mediocre entre muchos individuos muy pobres en una generación temprana), la selección por rango minimiza el efecto de este relativo buen individuo y no le permite reproducirse excesivamente para evitar que domine la población (i.e., previene la sobrevivencia del mediocre). Así, la selección por rango previene la convergencia prematura.

Selección por Torneo.- La selección por torneo es homóloga a la competencia entre individuos por su derecho a reproducirse tal y como se presenta en la natulareza. En la versión más simple de la selección por torneo, dos individuos son seleccionados aleatoriamente, y el individuo con mejor aptitud es seleccionado para reproducirse (ver Figura 16).Si se desea, el número de individuos seleccionados para realizar el torneo puede ser mayor a dos. Este proceso se repite hasta que se obtengan el número de ganadores necesarios para generar la nueva población.

El mejor individuo de la población tiene una probabilidad de 1 de predominar en la competencia, el peor individuo tiene 0 de probabilidad y un individuo que se encuentre a la mitad del rango de la población tiene una probabilidad de 0.5 de predominar en la competencia. Así, la selección por torneo, es en efecto, una versión probabilística de la selección por rango. Tiene la ligera ventaja de ser más rápido que la selección por rango.

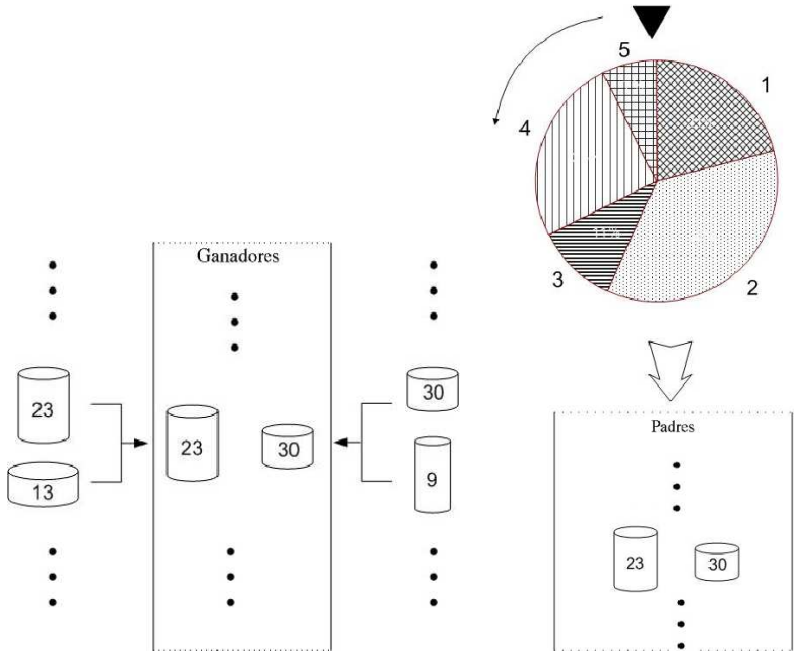


Figura 16: Algunos operadores de selección. (a) Selección por torneo. (b) Selección proporcional.

Reproducción

La reproducción consiste en aplicar uno o más operadores $\mathcal{O}$ sobre los individuos seleccionados en la etapa de selección (llamados *padres:* x_i) y obtener la creación de los individuos que conformarán la siguiente generación (llamados *hijos:* y_i). La aplicación del operador se hace por cada componente del vector de valores del individuo

$$y_i = \mathcal{O}(x_i) \quad .$$

A estos operadores se les conoce como *operadores evolutivos o genéticos.* Los operadores genéticos son los que permiten modificar las estructuras en busca de la adaptación a su ambiente dentro la PG.

Existen 2 operadores primarios:

- Reproducción Darwiniana,

- Cruzamiento (crossover).

Reproducción Darwiniana.- Este operador es asexual. Opera sobre un sólo padre y genera un sólo hijo cada vez que es utilizado. La operación de reproducción consiste en dos pasos. Primero, un individuo de la población es seleccionado por medio de algún método de selección. Segundo, el individuo seleccionado es copiado, sin ninguna alteración, a la nueva población (i.e., la siguiente generación). La aptitud de tal individuo no cambia, por lo que no requiere ser calculada de nuevo. Si la reproducción es aplicada, digamos, a 10% de la población cada generación, el uso de este operador reduce en 10% los calculos necesarios para determinar la aptitud en cada generación. Esto es una ventaja, ya que, el cálculo de aptitud es lo que más tiempo consume en un problema no trivial.

Cruzamiento.- La operación de cruzamiento (reproducción sexual) crea variaciones en la población al producir nuevos descendientes, los cuales estan formados de partes tomadas de cada padre. El cruzamiento comienza con dos individuos (i.e., padres) y crea dos descendientes (i.e., hijos).

Ambos padres son seleccionados por medio de algún método de selección. El operador de cruzamiento comienza seleccionando por medio de una distribución de probabilidad uniforme, un punto aleatorio en cada padre para ser el *punto de cruzamiento*. Los puntos de cruce son diferentes para cada padre. Los padres por lo general son de diferentes tamaños.

El fragmento de cruzamiento consiste en el subárbol que se encuentra a partir del punto de cruce, siendo este la raíz. Este fragmento a veces consiste en un sólo nodo, véase Figura 17.

El primer hijo es creado al borrar el fragmento de cruzamiento del primer padre e insertándole el fragmento de cruzamiento del segundo padre a partir del punto de cruce del primer padre. Es decir, se intercambian los fragmentos de cruzamiento del padre 1 por el del padre 2. El segundo hijo es construido de la misma manera. Así, los hijos están compuestos de subexpresiones (i.e., subárboles, subprogramas, subrutinas) de sus padres. Intuitivamente, si dos programas de computadora (i.e., padres) son algo efectivos en la resolución de un problema, entonces algunas de sus partes probablemente tendrán algún mérito. Por la recombinación aleatoria de partes de algunos programas efectivos, algunas veces produciremos nuevos programas de computadora (i.e., hijos) que sean más aptos en la resolución del problema dado, en comparación a los padres. En la Figura 17 se observa un ejemplo de cruzamiento.

Si en los puntos de cruce de ambos padres se encuentran terminales, el resultado es una mutación en ambos individuos. Por lo que ocasionalmente, la mutación es inherente en el cruzamiento. Cuando dos individuos idénticos se cruzan, los hijos resultantes serán diferentes, debido a que los puntos de cruce de cada padre son diferentes. Si el cruzamiento entre dos padres crean hijos de tamaño superior al permitido, la operación de cruzamiento es abortada y uno o ambos padres son copiados a la nueva población.

En adición a los operadores genéticos primarios de cruzamiento y reproducción, existen cinco operadores secundarios opcionales, los cuales son usados ocasionalmente:

- Mutación,
- Permutación,
- Edición,
- Encapsulación y
- Diezmado.

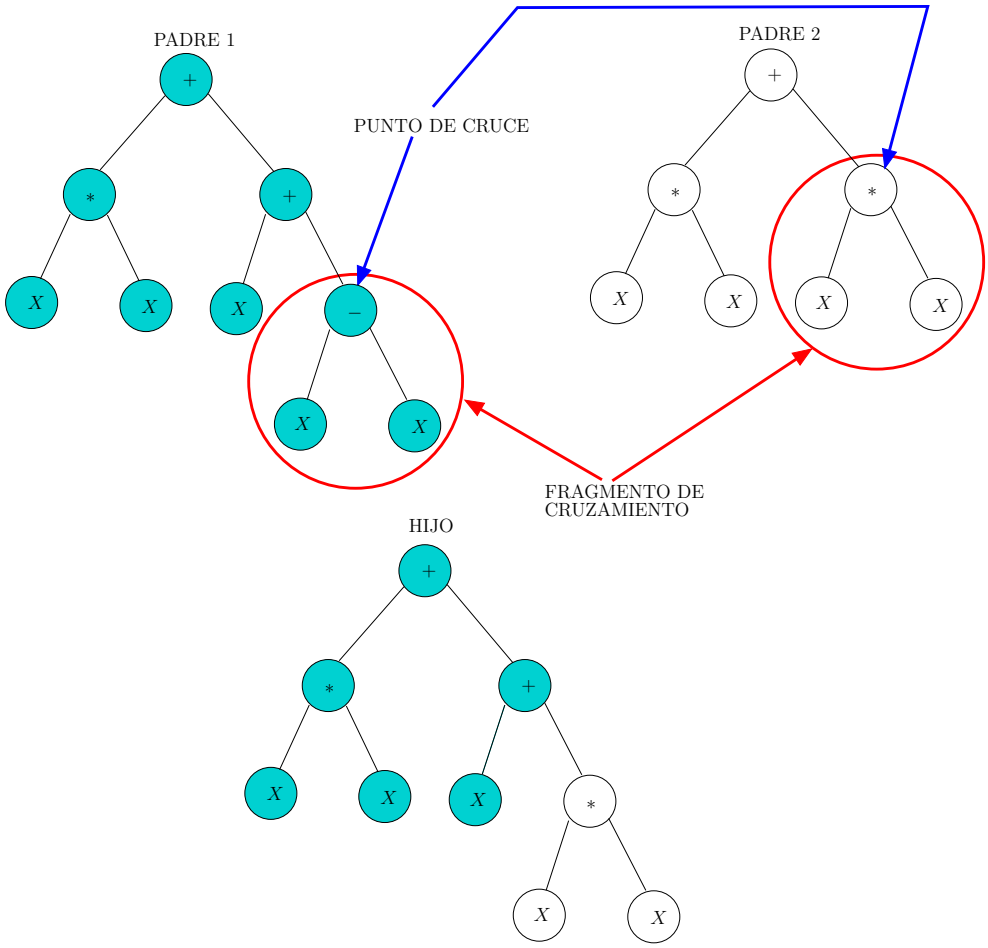


Figura 17: Cruzamiento de un subárbol en programación genética: $x^2 + (x + (x - x))$ se cruza con $2x^2$ para producir $2x^2 + x$. El subárbol del padre 2 $(*xx)$ es copiado e insertado en una copia del padre 1, reemplazando al subárbol $(-xx)$ para crear el programa hijo $(+(*xx)(+x(*xx)))$.

Mutación.- El operador de mutación introduce cambios aleatorios a los individuos de la población. Su importancia radica en que evita la pérdida de diversidad en la población y por lo mismo, se le considera un operador de exploración del espacio de búsqueda. Sin embargo, en la PG, es relativamente raro que una función o terminal desaparezca por completo de una población, debido a que no estan asociadas con posiciones o estructuras fijas como lo serían los cromosomas en los Algoritmos Genéticos. Por ello, la mutación no es necesaria en la PG.

La mutación es un operador asexual, el cual opera sobre un sólo padre y genera un sólo hijo. Comienza al seleccionar un padre por medio de alguno de los métodos de seleción. Posteriormente, se selecciona aleatoriamente un *punto de mutación* dentro del individuo seleccionado. El punto de mutación puede ser un punto interno (i.e., función) o externo (i.e., terminal) del árbol. La operación de mutación remueve el subárbol que se encuentre a partir del punto de mutación, (i.e., remueve el punto de mutación y todo lo que se encuentre por debajo de el) e inserta en el punto de mutación un subárbol generado aleatoriamente.

Permutación.- La permutación es un operador asexual. Opera sobre un sólo padre y genera un sólo hijo. Comienza seleccionando un padre por medio de alguno de los métodos de seleción. Posteriormente, se selecciona aleatoriamente un punto interno (i.e., función) del individuo. Si la función seleccionada tiene k argumentos, se selecciona aleatoriamente una permutación del conjunto de posibles $k!$ permutaciones. Así, se permutan los argumentos de la función de acuerdo con la permutación seleccionada. Si la función es conmutativa, no habra efecto inmediato como resultado de la permutación. Por ejemplo, si la función, es una división, que opera con los argumentos a y b, tal que el resultado sea $\frac{a}{b}$, si se aplica la permutación, los argumentos seran b y a obteniéndose $\frac{b}{a}$.

Edición.- La edición es un operador asexual. Opera sobre un sólo padre y genera un sólo hijo. Comienza seleccionando un padre por medio de alguno de los métodos de seleción. Este operador aplica recursivamente un conjunto preestablecido de *reglas de edición* dependientes del dominio y son específicas a cada individuo de la población.
La regla de edición universal independiente del dominio es la siguiente: Si cualquier

función que no tiene efectos secundarios y no es dependiente del contexto, tiene terminales constantes como argumentos, entonces el operador de edición evaluará la función y la reemplazará con el valor obtenido de dicha evaluación. Por ejemplo, la expresión (+12) será reemplazada por 3 y la expresión booleana ($AND \ \ TRUE \ \ TRUE$) será reemplazada por ($TRUE$). En adición, el operador de adición aplica reglas preestablecidas de dominio específico. La aplicación recursiva de reglas de edición consume mucho tiempo de ejecución.

Este operador puede ser usado de dos maneras: la primera, es cosmética (i.e., completamente externa a la ejecución) con el objetivo de que la salida del programa genético sea legible. La segunda, es durante la ejecución del programa, para simplificar la salidad (sin sacrificar los logros de los resultados) o para mejorar el desempeño del programa genético.

Encapsulación.- El encapsulamiento permite identificar automáticamente un subárbol potencialmente útil y darle un nombre, de manera que después pueda ser referenciado y utilizado.

Una manera de resolver un problema grande es descomponiéndolo en una jerarquía de subproblemas pequeños. La clave para ello es identificar subproblemas pequeños que descompongan útilmente el problema. Una meta importante de la inteligencia artificial y del aprendizaje de máquina es realizar esta identificación de manera automática.

El encapsulamiento es asexual, opera sobre un sólo padre y genera un sólo hijo. El padre es seleccionado por alguno de los métodos de selección. Este operador comienza seleccionando aleatoriamente un punto interno del árbol (i.e., función). Remueve el subárbol localizado en el punto seleccionado y lo encapsula en una función, la cual no tiene argumentos. Esta nueva función permite hacer referencias al subárbol borrado.

Las nuevas *funciones encapsuladas* reciben el nombre de $E0, E1, E2, ..$, conforme son creadas. Un nodo llamando a la función recién creada es insertado en el padre en el punto seleccionado en vez del subárbol.

El conjunto de funciones del programa genético es aumentado al incluir la nueva función, de manera que pueda ser utilizada más adelante durante la ejecución por el operador de mutación.

El efecto de este operador es, que el subárbol seleccionado ya no esta sujeto a la separación que resulta del cruzamiento, debido a que ahora es un punto indivisible. El cual puede proliferar en la población de las siguientes generaciones.

Diezmado.- Para algunos problemas complejos, la distribución de los valores de aptitud puede ser asimétrica, teniendo como resultando un gran porcentaje de individuos con una aptitud muy pobre. En tales problemas, una gran cantidad de tiempo de ejecución es gastado en individuos pobres de generaciones muy tempranas. Cuando la asimetría es muy grande, los pocos individuos con mejor aptitud inmediatamente empiezan a dominar la población y la variedad empieza a disminuir. A pesar de que el cruzamiento reincorpora variedad a la población, la selección de padres utilizada por el cruzamiento está basada en la aptitud, con ello, el cruzamiento se concentra en pocos individuos.

El operador de diezmado ofrece una manera rápida de lidiar con esta situación. La operación es controlada por dos parámetros: un porcentaje y una condición que especifíca cuándo será invocada. El objetivo del operador de diezmado es eliminar un porcentaje de individuos de la población en función de su aptitud, para evitar el

evaluar soluciones que serán desechadas en generaciones posteriores. Por ejemplo, si el porcentaje es de 10% y la condición es que sea invocada en la generación 0, toda la población con excepción del 10% es borrada. La reselección no es permitida, con el fin de maximizar la diversidad en la población restante. Así, si no existian duplicados en la generación cero y el operador de diezmado es aplicado después de calcular las aptitudes para la generación 0, la población seguirá sin presentar duplicados en la generación 1.

Criterio de Paro

El programa genético avanzará generación tras generación evolucionando la población hacia la o las mejores soluciones en el espacio de búsqueda. Sin embargo, en ocasiones, no se sabe con certeza la región donde se encuentra la solución óptima o quizá esta no exista o no se conozca, por lo que la ejecución podría seguir indefinidamente. Por esto es necesario definir un criterio de paro adicional. Este criterio se define *a priori*, y es el número máximo de generaciones permitido. Este criterio permite ademas, llevar a cabo pruebas estadísticas analizando la convergencia de los valores de aptitud a través de las generaciones, con lo que se puede estimar el número de generaciones apropiado y utilizarlo en el futuro.

III.5 Conclusiones

En este capítulo se presentó una metodología (i.e., la Programación Genética), capaz de alcanzar la *programación automática* de una computadora, objetivo buscado por largo tiempo por varias campos dentro de la inteligencia artificial. También se pudo ver, que la PG debe ser considerada como un sistema de aprendizaje de máquina.

El rápido crecimiento del campo de la PG, se debe a que, es una metodología fácil de usar e implementar, así como de propósito general, ya que puede aplicarse a diferentes

áreas y problemas sin realizar cambios significativos a su estructura. Otra ventaja de la PG, es que no es necesario un gran conocimiento del problema a resolver, pues el conocimiento emerge a través de la evolución.

En el siguiente capítulo se propone un enfoque novedoso para afrontar el problema de detección de puntos de interés. Para lograrlo, se ha decidido utilizar la programación genética por las razones expuestas en este capítulo.

Capítulo IV

Trabajo Previo

El propósito de este capítulo es ubicar el presente trabajo dentro del marco científico de las diferentes propuestas encontradas en la literatura.

Como se menciona en la sección II.2, los métodos de detección de puntos de interés pueden clasificarse en tres categorías: los basados en contornos, los basados en intensidad y los basados en modelos paramétricos. En este capítulo se presentan diferentes detectores de cada una de las categorías.

Debido a que el detector propuesto en este trabajo se encuentra clasificado dentro de los métodos basados en intensidad. Por esta razón se desarrollan más a detalle, con excepción del método propuesto por Moravec. Inclusive, con el propósito de conocer a profundidad el funcionamiento particular de cada uno de ellos, son implementados y evaluados en su desempeño.

IV.1 Detectores de Puntos de Interés

IV.1.1 Métodos Basados en Contornos

Tsai *et al.* (1999) proponen una medida de la curvatura basada en los valores propios de la matriz de covarianza de los puntos de frontera. La máxima curvatura ocurre en los valores significativamente grandes de los valores propios.

Sohn *et al.* (1998) proponen un método de suavizado de los puntos de frontera para

estimar la curvatura, empleando aproximaciones determinísticas basadas en el recocido simulado. Los máximos de esta función de curvatura una vez aplicado el método, identifican a cada una de las esquinas.

Medioni y Yasumoto (1987) emplean B-splines para aproximarse al contorno descrito por los puntos de frontera. La esquina se ubica en el máximo de la curvatura una véz calculado los coeficientes de los B-splines.

Horaud *et al.* (1990) extraen segmentos de línea de los contornos de la imagen. Estos segmentos son agrupados y las intersecciones de los grupos de segmentos de línea son usados como puntos de interés .

IV.1.2 Métodos Basados en Intensidad

Los métodos que serán discutidos son:

- Detector de Moravec.
- Detector de Beaudet.
- Detector de Dreschler y Nagel.
- Detector de Kitchen y Rosenfeld.
- Detector de Wang y Brady.
- Detector de Harris y Stephens.
- Detector de Förstner.
- Detector IPGP1.
- Detector IPGP2.

Conceptualmente cada uno de estos operadores, con excepción de los detectores IPGP1 y IPGP2, fueron diseñados como detectores de esquinas. Es por esto, que los detectores que a continuación se presentan, hablan del concepto de *esquina* en vez de punto de interés. Sin embargo, su capacidad de detección no está limitada a puntos que forman estrictamente una esquina. Este tipo de detectores extraen todos los puntos donde la variación de intensidad en la imagen es alta con respecto a una medida particular. En otras palabras, el tipo de características que se extraen de una imagen son *puntos de interés*.

IV.1.3 Detector de Moravec

Moravec (1977) desarrolló uno de los primeros detectores de puntos de interés basados en la señal. Esta basado en la función de auto-correlación. Mide las diferencias de valores de gris entre ventanas desplazadas en varias direcciones: horizontal, vertical y ambas diagonales. Localiza puntos donde el mínimo de la sumatoria de la diferencia de los cuadrados entre pixeles adyacentes son un máximo local.

IV.1.4 Detector de Beaudet

Beaudet (1978) propone el Operador Rotacional Invariante (DET), basado en el cálculo del determinante del Hessiano de una región de la imagen con el propósito de ubicar las esquinas. La expresión propuesta por Beaudet es la siguiente

$$K_{Beaudet} = \Delta(H) = \begin{vmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{vmatrix} , \tag{23}$$

donde $I(x, y)$ es la función de intensidad en tonos de gris de una imagen. $\Delta(H)$ es el determinante del Hessiano de $I(x, y)$. I_{xx}, I_{yy} e I_{xy} son las segundas derivadas parciales

de $I(x, y)$ con respcto a xx, yy y xy respectivamente.

La localización de una esquina consta de dos pasos:

1. Se calculan los extremos del operador rotacional invariante $K_{Beaudet}$ y se toman sus correspondientes valores absolutos.

2. Si el valor calculado en la Ecuación (23) sobrepasa un umbral definido por el usuario, entonces la esquina se encuentra en las coordenadas (x,y) del máximo encontrado.

La Figura 18 muestra el comportamiento tridimensional del operador rotacional invariante para el caso de una esquina-L, con un ángulo de apertura de $\pi/2$ y un factor de difuminado $\sigma = 1$.

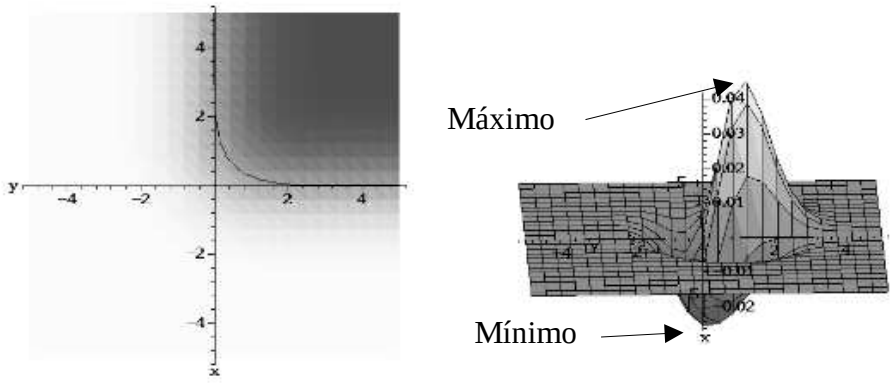


Figura 18: Detector de Beaudet. a) Esquina-L. b) Superficie generada por el determinante del Hessiano $K_{Beaudet}$ operado sobre la imagen sintética.

Las características de este detector son:

1. El detector $K_{Beaudet}$ otorga una máximo positivo y un mínimo negativo.

2. El mínimo negativo detecta una posición falsa del punto de esquina, vea Figura 18.b. Por tanto se debe tener cuidado en el momento de calcular los extremos.

3. El máximo positivo de $K_{Beaudet}$ determina la ubicación de la esquina. Se observa que el máximo se encuentra desplazado hacia la cresta de la esquina.

4. El detector no es estable en el espacio de escala lineal Gaussiano. Esto es, si aplicamos distintos factores de difuminado la ubicación de la esquina se altera.

5. El detector es sensible al ruido. El operador rotacional invariante no contempla algún elemento que suavize las regiones de intensidad dentro y fuera de la esquina.

6. Se requiere definir intuitivamente un umbral para considerar la existencia de una esquina, $máximo(K_{Beaudet}) \geq Umbral$.

IV.1.5 Detector de Dreschler y Nagel

Dreschler y Nagel (1982) proponen un operador basado en la *curvatura Gaussiana principal* de una superficie. El aporte radica en aplicar el concepto de curvatura Gaussiana de una superficie a una esquina en la imagen. La curvatura Gaussiana principal $k_{min}k_{max}$ esta dada por

$$K_{D\&N} = k_{min}k_{max} = \frac{\Delta(H)}{(1 + I_x^2 + I_y^2)} \ , \tag{24}$$

donde I_x e I_y son la primeras derivadas parciales de la función de intensidad $I(x, y)$ con respecto a x y y respectivamente. $\Delta(H)$ es el determinate del Hessiano, equivalente al detector $K_{Beaudet}$.

El comportamiento del detector $K_{D\&N}$ es similar al propuesto por Beaudet, en su forma y propiedades. Sin embargo, el determinante del Hessiano no es la expresión exacta que denota a la curvatura Gaussiana $k_{min}k_{max}$ (Deriche y Giraudon, 1993). El punto de esquina se localiza como sigue:

1. Se calcula la curvatura Gaussiana dada por la Ecuación (24).

2. Se identifica el máximo positivo y el mínimo negativo de la curvatura, puntos P_a y P_b de la Figura 19. El punto P_a representa el extremo cuando $k_{min}k_{max} > 0$. En la literatura, P_a es un punto elíptico. Por otro lado, el punto P_b representa el extremo cuando $k_{min}k_{max} < 0$ y se le conoce como punto hiperbólico (Deriche y Blazka, 1993).

3. El punto de esquina P_e se localiza en la intersección de la línea que une a P_a y P_b con la curva $k_{min}k_{max} = 0$. La curva $k_{min}k_{max} = 0$ es la curvatura Gaussina principal. La Figura 19 muestra gráficamente estos conceptos.

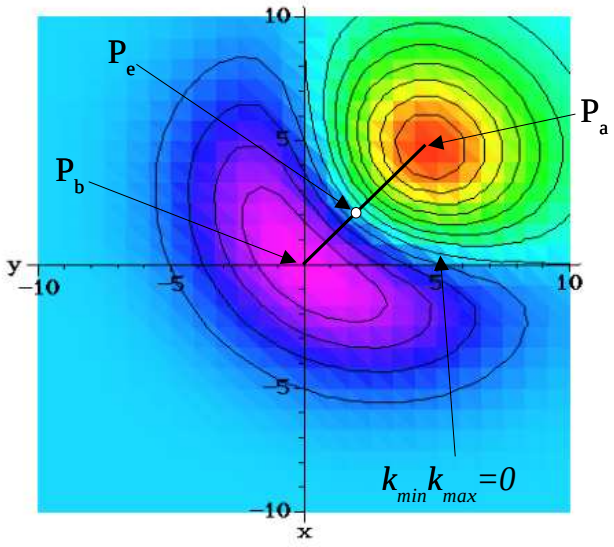


Figura 19: Ubicación del punto de esquina del detector de Dreschler y Nagel, para una esquina con un ángulo de apertura de $\pi/2$ y $\sigma = 1$.

Características del detector, ver Zheng *et al.* (1999):

- Sensible al ruido.
- Se requiere definir un umbral para discernir la existencia de una esquina.
- No es estable en el espacio de escala lineal Gaussiano.

IV.1.6 Detector de Kitchen y Rosenfeld

Kitchen y Rosenfeld (1982) propusieron uno de los detectores más populares que trabajan con imágenes digitales. El detector está basado en el cambio de la dirección del gradiente a lo largo del borde que define una esquina, multiplicado por la magnitud del gradiente. La expresión que representa este detector esta dada por

$$K_{K\&R} = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_yI_y}{I_x^2 + I_y^2} \quad . \tag{25}$$

La ubicación de la esquina se encuentra en las coordenadas (x,y) donde se localiza el extremo dado por la Ecuación (25), ver Figura 20. Las características son:

1. Existe sólo un mínimo o máximo. El punto de esquina P_e corresponde al extremo de $K_{K\&R}$, ver Figura20.a.

2. El máximo del detector $K_{K\&R}$ está ubicado fuera de la curva que representa la curvatura principal. La Figura 20.c muestra gráficamente este efecto.

3. No es estable en el espacio de escala lineal Gaussiano.

4. Es sensible al ruido.

5. Se requiere definir intuitivamente un valor de umbral para discernir la existencia de una esquina.

IV.1.7 Detector de Wang y Brady

Wang y Brady (1995) proponen otra medidada de curvatura $K_{W\&B}$. El valor de la curvatura $K_{W\&B}$ es proporcional a la segunda derivada tangencial en la dirección del borde e inversamente proporcional a la magnitud de su velocidad de cambio. El detector de Wang y Brady esta dado por

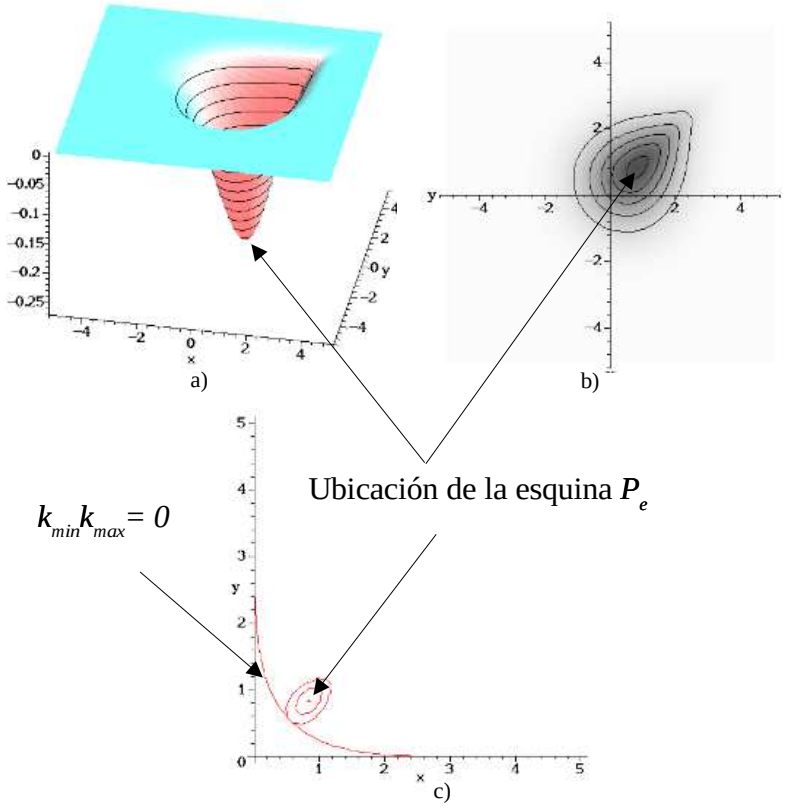


Figura 20: Ubicación del punto de esquina del detctor de Kitchen y Rosenfeld, para una esquina con un ángulo de apertura de $\pi/2$ y $\sigma = 1$. a) Modelo tridimensional de $K_{K\&R}$. b) Curvas de contorno de $K_{K\&R}$. c) Ubicación del punto de esquina P_e.

$$K_{W\&B} = \frac{\frac{\partial^2 I}{\partial t^2}}{|\nabla(I)|} \qquad para \qquad \nabla^2(I) >> 1 \tag{26}$$

donde $\frac{\partial^2 I}{\partial t^2} = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_yI_y}{|\nabla(I)|^2} = K_{K\&R}$, es la segunda derivada tangencial de I. $\nabla(I)$ es el gradiente de I. $\nabla^2(I)$ es el Laplaciano de I. Aplicando la supresión de repuesta falsa de esquina, la Ecuación (26) se reduce a la siguiente expresión

$$K_{W\&B} = (\nabla^2(I))^2 - S|\nabla(I)|^2 \ , \tag{27}$$

donde S en una constante en la supresión de respuesta falsa de una esquina. S es propuesta empíricamente por el usuario. La Figura 21 muestra el comportamiento

tridimensional de $K_{W\&B}$.

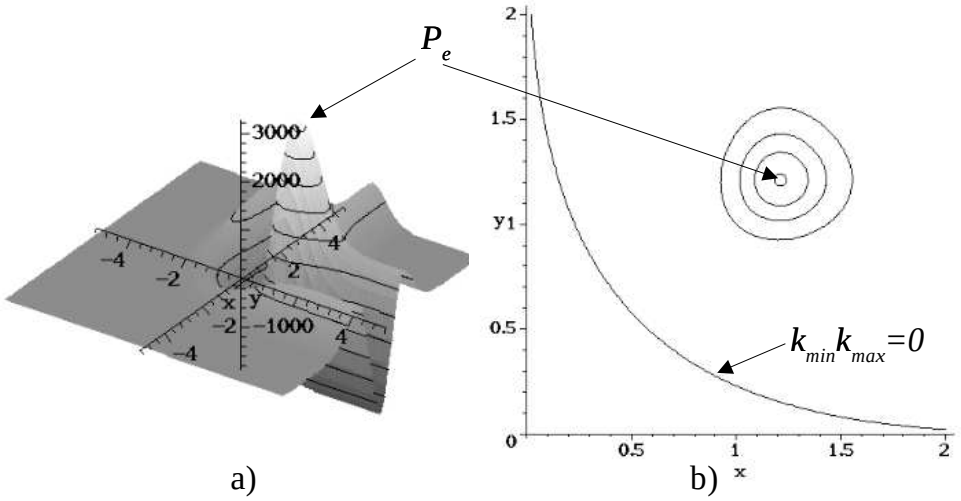


Figura 21: Detector de Wang y Brady, para una esquina con un ángulo de apertura de $\pi/2$ y $\sigma = 1$. a) Modelo tridimensional de $K_{W\&B}$. b) Ubicación del punto de esquina P_e.

La ubicación de la esquina se localiza calculando el máximo de la Ecuación (27). El detector de Wang y Brady satisface los requerimientos de tiempo real, para la estimación de escenas en movimiento (Zheng *et al.*, 1999). Sin embargo, la ubicación de la esquina se encuentra desplazada en la dirección de la cresta de la esquina, como se observa en la Figura 21.b.

IV.1.8 Detector Harris y Stephens

Harris y Stephens (1988) estaban interesados en utilizar técnicas de análisis de movimiento para interpretar el ambiente de robots basados en imágenes de una sola cámara móvil. Por ello, necesitaban un método para realizar la correspondencia de puntos en cuadros consecutivos de la imagen como el detector de Moravec, pero estaban interesados en rastrear tanto esquinas como bordes. Está basado en el detector de puntos

propuesto por Moravec y en la función de autocorrelación. Originalmente, Harris y Stephens escriben la función de autocorrelación como se expresa en la Ecuación (28), donde W es una región de la imagen I. Nótese que la Ecuación (29) es equivalente a la Ecuación (17),

$$E_{x,y} = \sum_{u,v} W_{u,v} \cdot [I_{x+u,y+v} - I_{u,v}]^2 \tag{28}$$

$$\approx \sum_{u,v} W_{u,v} \cdot [xX + yY + O(x^2, y^2)]^2 \tag{29}$$

donde los gradientes son aproximados como:

$$X = I \otimes [-1, 0, 1] = \frac{\partial I}{\partial x} \ , \tag{30}$$

$$Y = I \otimes [-1, 0, 1]^T = \frac{\partial I}{\partial y} \tag{31}$$

Para pequeños cambios en E se sigue que:

$$E(x, y) = Ax^2 + 2Cxy + By^2 \ . \tag{32}$$

Donde

$$A = X^2 \otimes \omega$$

$$B = Y^2 \otimes \omega$$

$$C = XY \otimes \omega \ .$$

Tomando

$$\omega_{u,v} = e^{-\frac{u^2+v^2}{2\sigma^2}} \ .$$

Finalmente los cambios en E, para pequeños corrimientos en (x, y) puede ser escrito consistentemente como:

$$E(x, y) = (x, y)\mathbf{M}(x, y)^T \ . \tag{33}$$

Donde $\mathbf{M}$ es una matriz simétrica de tamaño 2×2 dada por:

$$\mathbf{M} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad . \tag{34}$$

Nótese que $\mathbf{M}$ es la matriz de autocorrelación escrita en la Ecuación (18). Las curvaturas principales de la función de autocorrelación están descritas por los eigenvalores α y β de la matriz de autocorrelación $\mathbf{M}$ y forman una descripción rotacionalmente invariante de $\mathbf{M}$. Con lo cual se pueden determinar tres casos principales:

1. Si ambas curvaturas son pequeñas, entonces la autocorrelación local es uniforme y consecuentemente la region W de la imagen esta compuesta por una intensidad aproximadamente constante (i.e., si existe un cambio en el vecindario causa un pequeño cambio en E);

2. Si una curvatura es considerablemente mayor que la otra, entonces la función de autocorrelación se puede modelar como una cresta, esto es que solo los cambios a lo largo de la cresta (i.e., a lo largo del borde) provocan pequeños cambios en E. Con lo cual el punto está situado sobre un borde.

3. Si ambas curvaturas son grandes, la función es un pico pronunciado, entonces los cambios en cualquier dirección incrementan E, lo cual indica una esquina.

Los eigenvalores α y β se pueden aproximar como:

$$Tr(\mathbf{M}) = \alpha + \beta = A + B \; , \tag{35}$$

$$Det(\mathbf{M}) = \alpha\beta = AB - C^2 \tag{36}$$

donde $Tr(\mathbf{M})$ indica la Traza de $\mathbf{M}$, y $Det(\mathbf{M})$ es el Determinante de $\mathbf{M}$. Con lo cual el detector de Harris y Stephens está dado por:

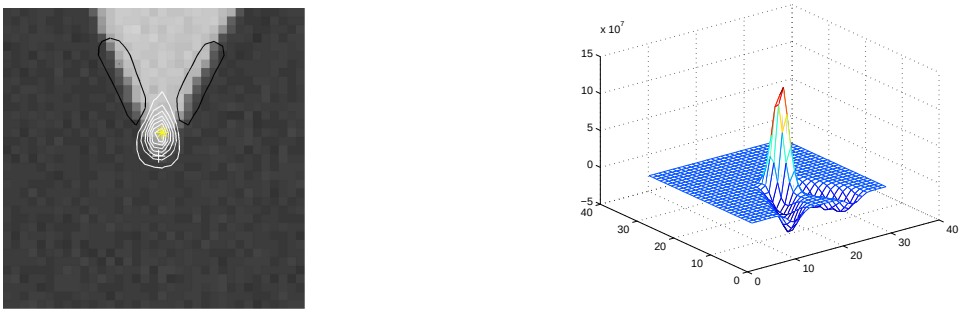
$$R = Det(\mathbf{M}) - k \cdot Tr(\mathbf{M}) \tag{37}$$

donde k es la llamada constante de Harris, y es propuesta por el usuario, aunque en la práctica suele tener un valor aproximado de 0.04. De esta forma el criterio de ubicación del punto esquina se simplifica a tomar los puntos que estén arriba de un umbral definido por el usuario.

La Figura 22 muestra el detector de Harris aplicado a una Esquina tipo L, asi como la superficie generada por R. La implementación de este detector fue realizada aproximando el gradiente con un filtro Gaussiano como se describe en la sección II.1.2 y la matriz de autocorrelación como se describe en la Ecuación (19).

IV.1.9 Detector de Förstner

El operador de Förstner puede ser visto como un desarrollo estadístico y analítico y puede ser formulado como un problema de encontrar puntos apropiados para una descripción de la imagen (Förstner y Gülch, 1987). Esto implica que dada una región, esta debe contener estructuras sobresalientes que describan la imagen. Förstner a partir de observaciones hechas en problemas como: correspondencia de imágenes por medio de mínimos cuadrados, detección de esquinas a partir de la intersección de bordes, la



(a) (b)

Figura 22: a) Curvas de nivel de R sobre la imagen, el * muestra el punto esquina propuesto por Harris. b) Superficie generada por R

búsqueda del centro de gravedad de una región de la imagen, así como la búsqueda del centro de elementos circulares, nota la similitud de estos procesos y propone una métrica a partir de la mátriz de la ecuación normal N definida en la Ecuación (38),

$$N = \begin{bmatrix} \sum g_x^2 & \sum g_x g_y \\ \sum g_x g_y & \sum g_y^2 \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \quad , \tag{38}$$

donde g_x y g_y es la imagen convolucionada con la primera derivada de un filtro Gaussiano. En otras palabras es equivalente a la matriz de autocorrelación $\mathbf{M}$ propuesta en la Ecuación 18

En este caso si se calcula $Q = N^{-1}$, una métrica de error para el punto (x, y) puede determinarse apartir de los eigenvalores λ_1 y λ_2 de la matriz Q. Los eigenvalores definen los semiejes de una elipse de error, la cual caracteriza al punto. De esta forma:

- Si la elipse se aproxima a una forma circular, entonces el punto puede estar ubicado dentro de un borde.
- En el caso en que los semiejes de la elipse sean pequeños se hablará de un punto de interés.

Förstner determina que los eigenvalores de N y Q son iguales, con lo cual propone las siguientes aproximaciones. Para determinar la razón de los semiejes de la elipse y medir la *circularidad* de la misma se propone la Ecuación (39) y con la Ecuación (40) se determina su orientación.

$$q = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 r} \right)^2 = 4 \frac{Det(N)}{(Tr(N))^2} \quad , \tag{39}$$

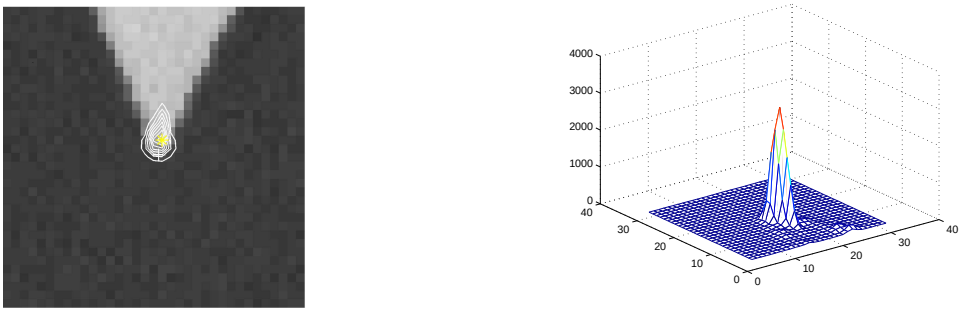
$$tan(2\Phi) = 2 \frac{2N_{12}}{N_{11} - N_{22}} \quad . \tag{40}$$

De forma similar Förstner propone una medida del tamaño de la elipse de error

como:

$$w = \frac{1}{\lambda_1 + \lambda_2} = \frac{Det(N)}{Tr(N)} \quad , \tag{41}$$

con lo cual los puntos que exceden un cierto umbral en el valor de q y w representan puntos de interés. En el caso del presente trabajo el umbral se determina como un porcentaje del valor más alto de w. La Figura 23 muestra la aplicación de este operador sobre una esquina tipo L.



(a) (b)

Figura 23: a) Curvas de nivel de w sobre la imagen, el * muestra el punto esquina propuesto por Förstner. b) Superficie generada por w

IV.1.10 Detector IPGP1

Trujillo y Olague (2006) proponen dos nuevos detectores de esquinas y puntos de interés, IPGP1 y IPGP2 (de sus siglas en inglés *Interest Point Genetic Programing*). Al mismo tiempo se propone un nuevo enfoque al problema de detección de puntos de interés como lo son las esquinas. Es decir, se aborda como un problema de optimización, donde el criterio que se utiliza para encontrar la solución es descrito por 3 características principales:

1. **Distinción Global** entre los puntos extraidos. Es decir, los puntos seleccionados como prominentes en una imagen, no sean los mismos.

2. **Alta Información de Contexto** en comparación a otros pixeles.

3. **Estabilidad** sobre ciertos tipos de transformación en la imagen, i.e., la rotación. Una métrica importante en este aspecto es la repetibilidad.

El problema es resuelto a travéz de la Programción Genética (GP). Donde cada individuo en la población del GP representa un operador candidato de puntos de interés.

El detector IPGP1 esta definido como:

$$IPGP1 = G(\sigma = 2) \otimes [G(\sigma = 1) \otimes I - I] \tag{42}$$

Nótese que IPGP1 es un detector con una estructura muy sencilla. Lo cual tiene ventajas como un procesamiento rápido y fácil de implementar. Básicamente IPGP1 extrae el punto en dos pasos:

1. Extraer las frecuencias altas de la imagen. Esto se lleva a cabo al substraer de la imagen original una imagen que ha sido convolucionada con un filtro Gaussiano, el cual tiene el efecto de suprimir las frecuencias más altas.

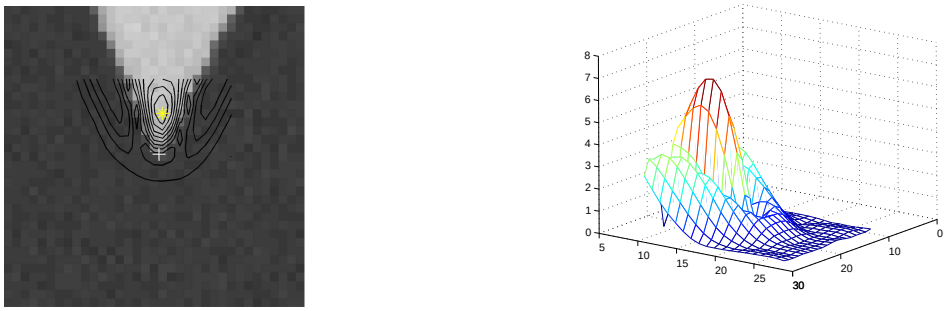
2. Suavizar la Imagen resultante de la operación anterior. Es decir se convoluciona la imagen resultante con un filtro Gaussiano con un efecto de suprimir las frecuencias más altas.

IPGP1 tiene algunas características interesantes como:

- No utilizar derivadas para la extración de puntos.

- Tener una repetibilidad del 95% que es superior a detectores como el propuesto por Harris y Stephens.

El criterio de ubicación de la esquina es por medio de un umbral propuesto por el usuario. La Figura 24 muestra la ubicación de la esquina en una imagen tipo L, asi como



(a) (b)

Figura 24: a) Curvas de nivel de IPGP1 sobre la imagen, el * muestra el punto esquina propuesto por IPGP1. b) Superficie generada por IPGP1

la superficie generada al aplicar IPGP1 a la imagen.

IV.1.11 Detector IPGP2

IPGP2 al igual que IPGP1 es un detector de punto de interés elaborado por medio de Programación Genética. Representa una versión modificada del Operador DET propuesto por Beaudet (1978), el cual se analiza en la sección IV.1.4. IPGP2 tiene la misma estructura que el DET, con la diferencia que es promediada alrededor de un vecindario local con una función de suavizado Gaussiana (Trujillo y Olague, 2006). De esta forma se podría decir que el detector de Beaudet fue propuesto nuevamente, es decir, un redescubrimiento hecho previamente por el hombre. La Ecuación muestra la expresión para el IPGP2

$$IPGP2 = G(\sigma = 1) \otimes [L_{xx} \cdot L_{yy}] - G(\sigma = 1) \otimes [L_{xy} \cdot L_{yx}] \tag{43}$$

El criterio para seleccionar los puntos de interés es apartir de un umbral propuesto por el usuario.

IV.2 Métodos Basados en Modelos Paramétricos

Los detectores propuestos por Rohr (1992), Deriche y Giraudon (1993), Deriche y Blazka (1993), Baker y Nayar (1998) y Olague y Hernández (2005) son conocidos como *detectores paramétricos de características de una imagen*.

Rohr (1992) fue el primer modelo paramétrico propuesto. Propone un modelo basado en una función escalón ideal convolucionada con un filtro Gaussiano. Para el caso de una esquina-L, el eje de simetría de dicho escalón está centrado en el eje x. Las coordenadas del punto de esquina corresponden al origen del sistema de coordenadas (x,y).

Rohr analiza el desplazamiento de la esquina-L causado por el efector de difuminado del filtro Gaussiano. Propone una ecuación implicita, para determinar el desplazamiento. Finalmente, concluye que la ubicación de la esquina es independiente de la altura de la esquina y que tiene una relación líneal con respecto al factor de difuminado σ y una relación no líneal con respecto al ángulo de apertura de la esquina β.

Deriche y Giraudon (1993) proponen un modelo similar al de Rohr. Ambos trabajos parten de un modelo de esquina ideal convolucionado con un filtro Gaussiano. Su desarrollo inicia con una función escalón unitaria del borde ideal. La diferencia fundamental entre estos dos métodos radica en la forma de localizar el desplazamiento del punto de esquina. Deriche basa su análisis en el comportamiento de su modelo y los detectores de Beaudet y Dreschler y Nagel, en el espacio de escalas lineal Gaussiano. Dichos detectores reportan que la posición exacta de la esquina puede ser detectada en el cruzamiento por cero del modelo en el espacio de escalas Gaussiano, es decir, cuando el Laplaciano del modelo es igual a cero.

Baker y Nayar (1998) desarrollan un algoritmo que construye un detector para un número arbitrario de parámetros. El trabajo hace un interesante énfasis en los efectos ópticos y de percepción en los sistemas de adquisición de imágenes. Cada característica es representada como un valor de densidad en un subespacio múltiple de muestreo simplificado. Las muestras del subespacio son tomadas por cada uno de los parámetros del modelo. Una característica es detectar si la proyección de los valores de intensidad alrededor del punto de interés, la esquina, está lo suficientemente cerca del subespacio de muestreo.

Deriche y Blazka (1993) desarrollaron un modelo simplificado del detector de Rohr. Proponen un filtro exponencial sustituyendo el filtro Gaussiano y caracterizan a cada borde en lugar del eje de simetría que emplea Rohr.

Olague y Hernández (2005) desarrollaron un modelo llamado USEF que significa función unitaria de borde. El modelado de una esquina se hace a partir de la función error que es tratada como una distribución de probabilidad, la cual modela las características físicas y ópticas encontradas en sistemas de imágenes digitales. De esta forma, el criterio de ubicación de la esquina se encuentra determinado por la distancia mínima de la curva de nivel a la intersección de las asíntotas que caracterizan la orientación de cada borde. Este modelo proporciona flexibilidad y generalidad para modelar esquinas complejas. Además, determinaron que un sensor CCD produce un efecto de difuminado diferente sobre las dos principales direcciones de la imágen (x, y).

IV.3 Evolución de Detectores de Puntos de Interés

Ebner (1998) aplicó la programación genética para detectar puntos de interés. Se trató de reproducir la respuesta del detector Moravec al especificarlo directamente en la función aptitud. El conjunto de terminales empleado es la representación en escala

de grises de una imagen de entrada. Las intensidades de la imagen fueron escaladas al rango [0,1]. El resultado obtenido fue un detector que se aproximó al Moravec, pero que presenta un error de localización de 15% en comparación con Moravec. La diferencia fue debido a que el filtrado y umbralización que se realiza con el detector Moravec no se aplicó a la función aptitud.

Ebner y Zell (1999) continuaron con la evolución de detectores de puntos de interés, pero esta vez lo hicieron para una tarea específica como lo es el flujo óptico y sin tratar de reproducir un detector ya existente. El flujo óptico es calculado al establecer los puntos correspondientes entre la imagen previa y la actual. El flujo óptico puede ser muy grande si la cámara se mueve muy rápida o muy lenta. El propósito fue extraer sólo los puntos que puedan ser localizados en la siguiente imagen. Para lograrlo se especificaron las siguientes características en la función aptitud: El número de correspondencias debe ser grande y con buena calidad. La relación de puntos correspondidos y los no correspondidos debe ser alta y no deben ser ambiguos. Los resultados se compararon con los detectores Kitchen-Roselfeld, Det(Hessiano), Moravec, SUSAN y Filtros Gabor. En la función aptitud el detector evolucionado mostró mejor aptitud que el resto de los detectores. Sin embargo, el criterio de optimización utilizado no garantiza generalidad en otras tareas de visión donde los puntos de interés son requeridos.

Trujillo y Olague (2006) Presentan un nuevo enfoque para la generación automática de un extractor de características útil en tareas de alto nivel de visión por computadora. Como parte de la función aptitud se introduce el criterio de repetibilidad. En sus resultados, se redescubrió una versión modificada del operador de Beaudet, presentando un mejor desempeño.

IV.4 Evaluación de Detectores de Puntos de Interés

La evaluación de detectores de características se ha concentrado en detectores de bordes. Sólo unos pocos autores han evaluado los detectores de puntos de interés:

Deriche y Giraudon (1993) examinan el comportamiento de los detectores para modelos de características teóricas. Este método (i.e., Análisis teórico) es limitado, ya que se aplica a características muy específicas. Ellos estudiaron analíticamente el comportamiento de 3 detectores de punto de interés utilizando un modelo de esquina L. Su estudio les permitió proponer un criterio para corregir el sesgo de localización.

Schmid *et al.* (2000) estableció una medida efectiva que captura la esencia de las características deseadas de estabilidad y robustez de los detectores de puntos de interés. Introdujeron 2 criterios de evaluación: la tasa de repetibilidad y la información de contenido. Su métrica se basa primordialmente en la tasa de repetibilidad. Estos dos criterios miden la calidad de los puntos de interés para tareas como la correspondencia, reconocimientos de objetos y reconstrucción 3D. En este contexto al menos un subconjunto de puntos debe ser repetido con el fin de permitir la correspondencia de puntos. Mas aún, si las medidas basadas en la imagen, como la correlación, son usadas para comparar puntos, los puntos de interés deben tener patrones diferentes.

Los detectores evaluados fueron Harris mejorado, Foerstner, Cottier, Heitger y Horaud. Se evaluó bajo diferentes transformaciones (rotación, escalamiento, iluminación y ángulo de observación). El detector que mostró mejores resultados es el Harris mejorado. Sin embargo, sus resultados permiten concluir que los detectores evaluados sólo son estables frenta a la rotación. Ya que la invariancia a otro tipo de transformaciones, requiere consideraciones adicionales.

IV.5 Experimentación

En esta sección se lleva a cabo una comparación de los 8 detectores de puntos de interés presentados en este capítulo. Su desempeño es evaluado por medio de la tasa de repetibilidad. La base de datos de imágenes utilizada para estos experimentos es la empleada en (Schmid *et al.*, 2000). Los motivos son varios, principalemente, porque debido a lo importante de la aportación de Schmid *et al.*, se desea tenerlo como principal referencia para la comparación de resultados. También porque son imágenes que presentan varias transformaciones. Para los experimentos de esta sección se utilizan las imágenes con rotación, las cuales, tienen como característica extra riqueza en texturas. Esto debido a que los detectores en cuestión no son apropiados para imágenes con otro tipo de transformaciones (Schmid *et al.*, 2000). Las imágenes rotadas se obtuvieron girando la cámara sobre su eje óptico usando un mecanismo especial. Las imágenes con variación en la iluminación fueron obtenidas cambiando la apertura de la cámara. El cambio es cuantificado por el "valor de gris relativo" (i.e., la razón del valor medio de gris de una imagen con el valor medio de la imagen de referencia) (Schmid *et al.*, 2000).

No fue necesario realizar el cálculo de las homografías requeridas para obtener la tasa de repetibilidad, debido a que la base de datos de las imágenes ya las incluye.

"Para asegurar una tasa de repetibilidad precisa, el cálculo de homografía debe ser preciso e independiente de los puntos detectados. Para ello, se requiere una localización a nivel sub-pixel. Para obtener las homografiías se tomó una segunda imagen por cada posición de la cámara, con puntos negros proyectados en la imagen. Véase Figura 25. Los puntos negros son extraídos con mucha precisión ajustando una plantilla y sus centros son usados para calcular la homografía. Un método de medianas mínimas

cuadradas hace el cálculo más robusto.", (Schmid *et al.*, 2000).

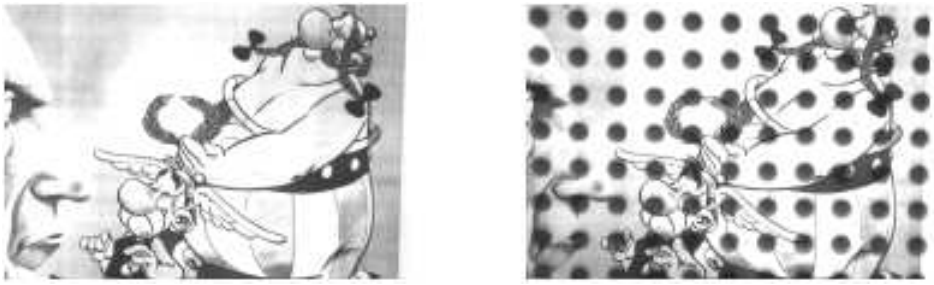


Figura 25: Dos imágenes. En la izquierda la imagen de la escena, en la derecha, la escena con los puntos negros proyectados.

Los 8 detectores de puntos de interés, así como la repetibilidad, fueron desarrollados con el lenguaje de programación c++ para su ejecución desde VXL. La tasa de repetibilidad se implementa como se describe en la sección II.3.2. El umbral utilizado para la supresión de máximos se determinó con el método de *Establecer un Valor Mínimo para considerarlo Punto de Interés* como se explica en la sección II.2.2.

Cada detector se aplicó a 16 imágenes rotadas, cuyos ángulos de rotación varian entre 0°y 180°.

Los tres detectores que presentaron mejor desempeño son IPGP1, IPGP2 y Harris mejorado, respectivamente. El IPGP1 presenta una repetibilidad promedio de 98.41%, seguido por IPGP2 con 93.84% y Harris mejorado con 89.13%. En la tabla III se presentan los promedios de las tasas de repetibilidad de los 8 detectores evaluados. En la Figura 26 se muestra la tasa de repetibilidad obtenida con cada una de las 16 imagenes.

Con los tres detectores que presentaron mejores resultados, se realizó otro experimento. Éste consistió en disminuir los umbrales que determinan cuáles pixeles son

Tabla III: Tasa de repetibilidad promedio obtenida con las 16 imágenes con rotación de la escena VanGogh.

Detector	Tasa de repetibilidad %
IPGP1	98.4154
IPGP2	93.8404
Harris	89.1310
Beaudet	84.8601
Förstner	76.6577
Wang y Brady	74.4834
Kitchen y Rosenfeld	63.4099
Dreschler y Nagel	23.1581

puntos de interés. Este experimento permite observar, véase Tabla IV, lo importante de la selección de un umbral adecuado y cómo se ve afectado el desempeño de los detectores por dicho umbral. Al disminuir el umbral, los puntos detectados aumentan y es muy probable que la tasa de repetibilidad disminuya. El desempeño de IPGP1 e IPGP2 disminuyó y el de Harris aumentó, sin embargo, los detectores IPGP1 e IPGP2 siguen presentando mejor desempeño que el detector de Harris. En la Figura 27 se muestra la tasa de repetibilidad obtenida con cada una de las 16 imagenes despues de disminuir los umbrales.

Tabla IV: Tasa de repetibilidad promedio obtenida con las 16 imágenes con rotación de la escena VanGogh despues de modificar umbrales a los detectores IPGP1, IPGP2 y Harris.

Detector	Tasa de repetibilidad %
IPGP1	96.4144
IPGP2	93.7467
Harris	90.7172
Beaudet	84.8601
Förstner	76.6577
Wang y Brady	74.4834
Kitchen y Rosenfeld	63.4099
Dreschler y Nagel	23.1581

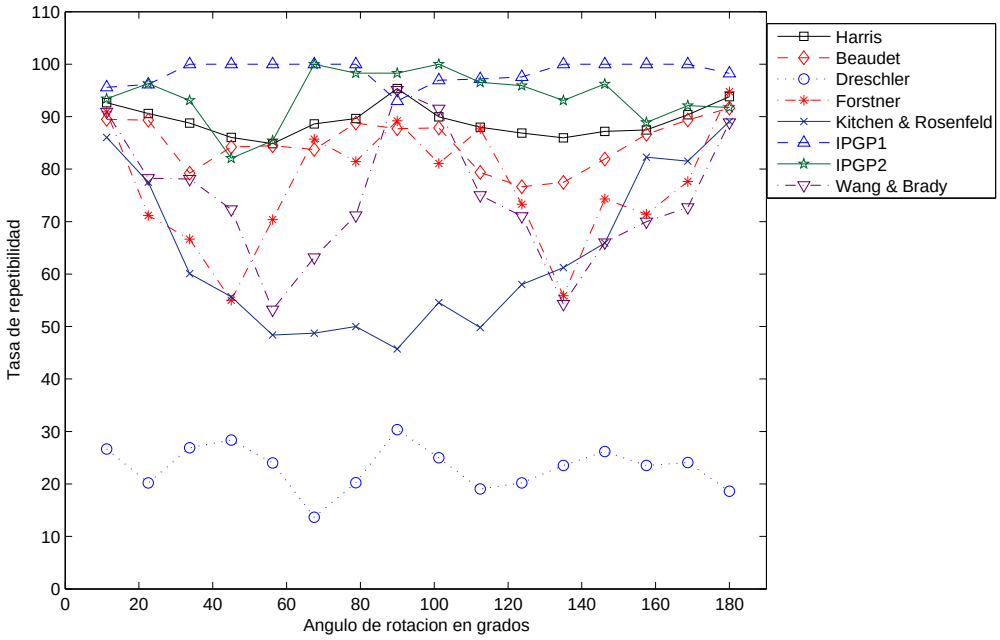


Figura 26: Medida de desempeño: Tasa de repetibilidad graficada contra la rotación para las 16 imágenes de VanGogh.

Como se puede observar, los detectores obtenidos por Programación Genética superan al detector de Harris, el cuál es el más utilizado por la comunidad. Esto se debe a que tanto el IPGP1 como el IPGP2 fueron desarrollados específicamente con el objetivo de ser estables a ciertos tipos de transformación, en especial a la rotación, ver Figura 28.

El detector que obtuvó peor desempeño es Dreschler y Nagel, con una tasa de repetibilidad promedio de 23.15%. Esto puede deberse a que la implementación del modelo analítico es muy dependiente de la imagen, además de ser susceptible al ruido, véase Figura 29.

Un resultado notable es el desempeño del detector de Beaudet, el cual obtuvó un promedio de tasa de repetibilidad de 84.86%. Sólamente después del detector de Harris. La importancia de este resultado radica en que al no existir un evaluación previa de

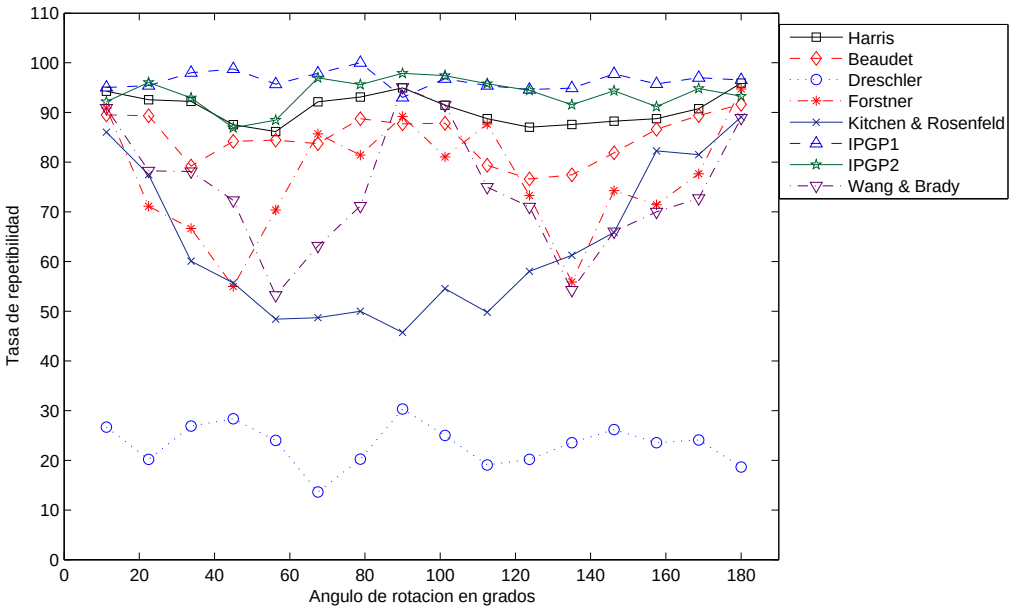


Figura 27: Medida de desempeño: Tasa de repetibilidad graficada contra la rotación para las 16 imágenes de VanGogh despues de modificar umbrales a los detectores IPGP1, IPGP2 y Harris.

él, se desconocia su efectividad para ciertas tareas. Hoy, se sabe que es un detector suficientemente estable.

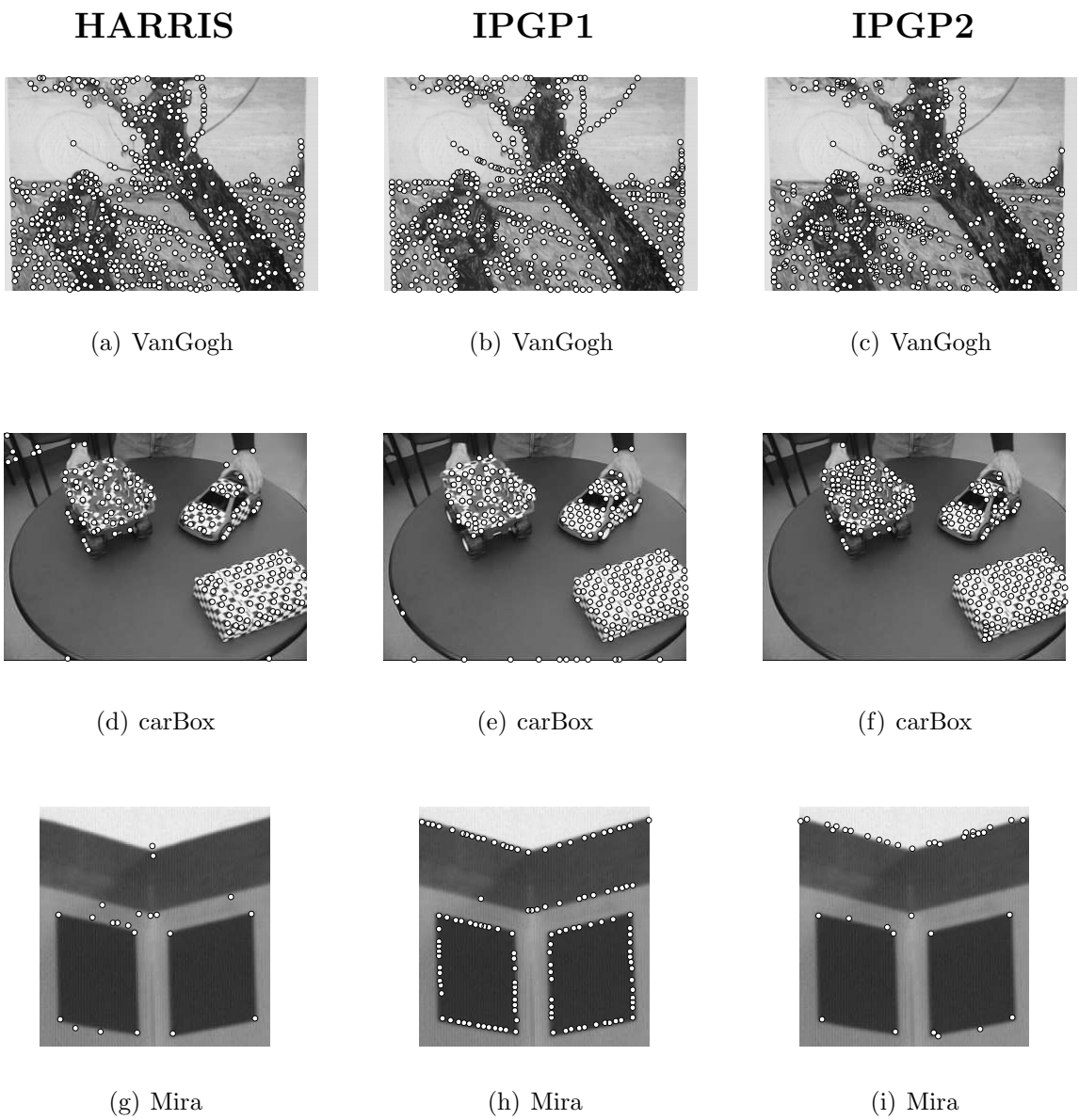
En las Figuras 30, 31, 32, 33, 34, 35, 36 y 37 se presentan una secuencia de 2 imágenes con los puntos de interés detectados por cada uno de los 8 detectores evaluados.

IV.6 Conclusiones

En este capítulo se ha presentado el estado del arte de los detectores de puntos de interés. Esto permite observar que los detectores de puntos de interés que han sido formulados hasta el momento, con excepción de IPGP1 e IPGP2, han sido producto del

análisis e interpretación del problema por la mente humana. Por esto, una comparación analítica es sumamente difícil. Lo que confirma la idea de aplicar una tendencia actual en el estudio de las ciencias computacionales, como lo es la síntesis automática de programas. Esta metodología permite descubrir soluciones que la mente del ser humano no hubiera podido considerar.

También se realizó la implementación y evaluación del desempeño de ocho detectores de puntos de interés frente a imágenes que presentan rotación. Esto permite precisar las operaciones útiles, necesarias o idóneas para desarrollar el detector propuesto en este trabajo. Esto será tratado en el siguiente capítulo.

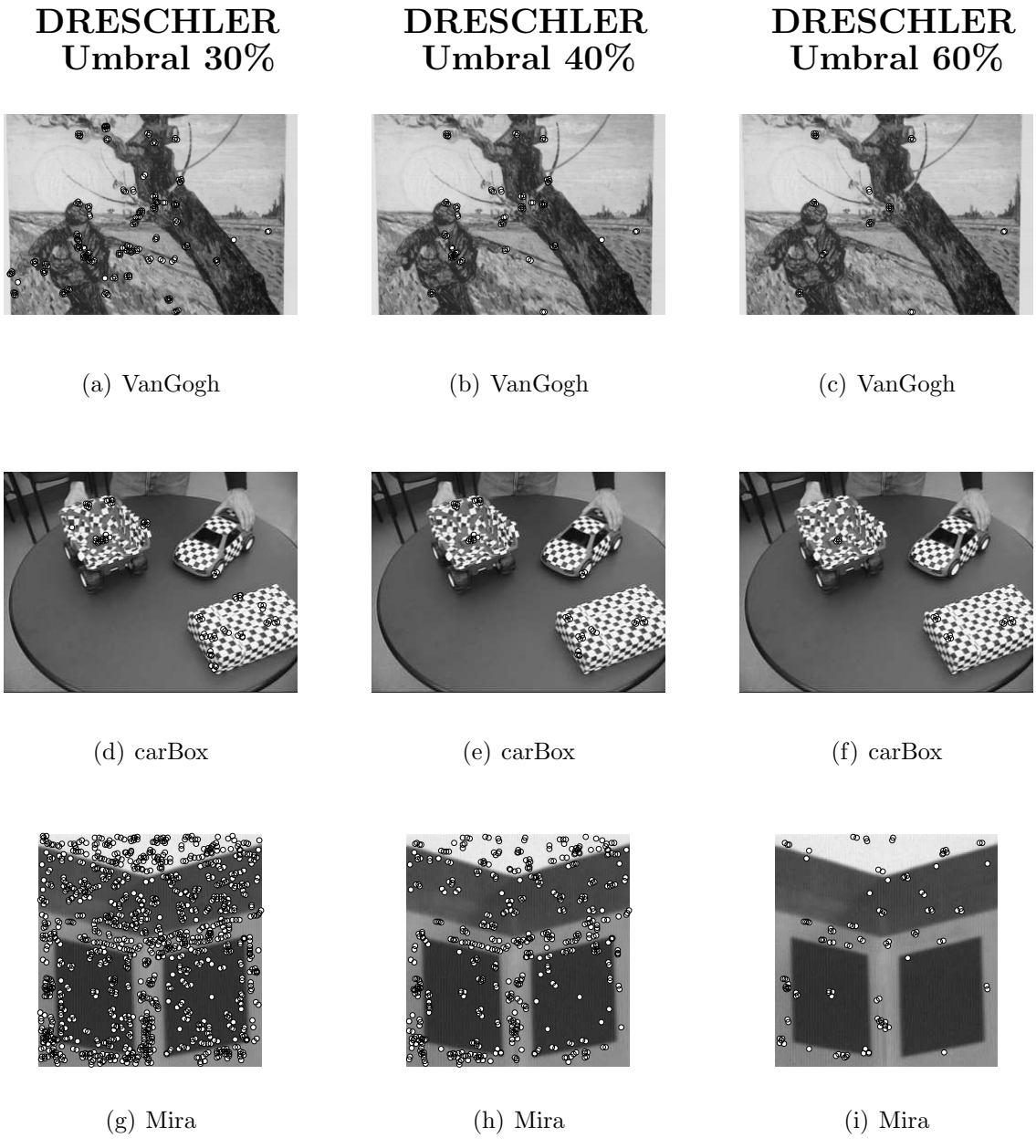


(a) VanGogh (b) VanGogh (c) VanGogh

(d) carBox (e) carBox (f) carBox

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

Figura 28: Puntos detectados en tres diferentes escenas (i.e. VanGogh, carBox y Mira) con los tres detectores que mostraron mejor desempeño.

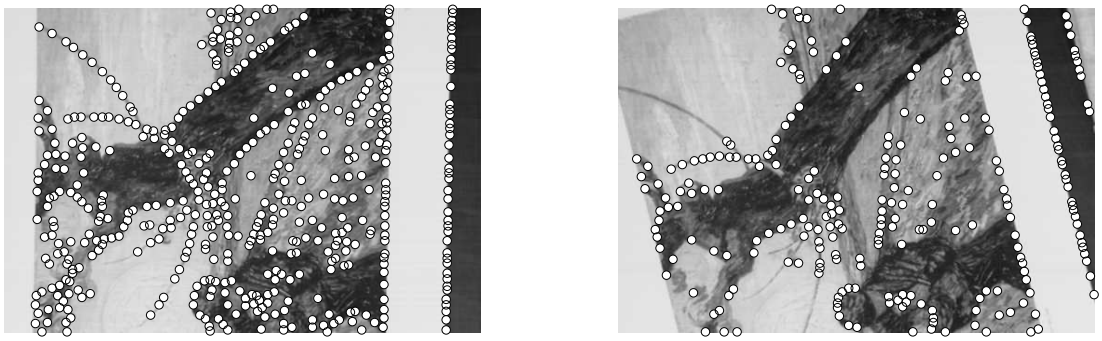


(a) VanGogh (b) VanGogh (c) VanGogh

(d) carBox (e) carBox (f) carBox

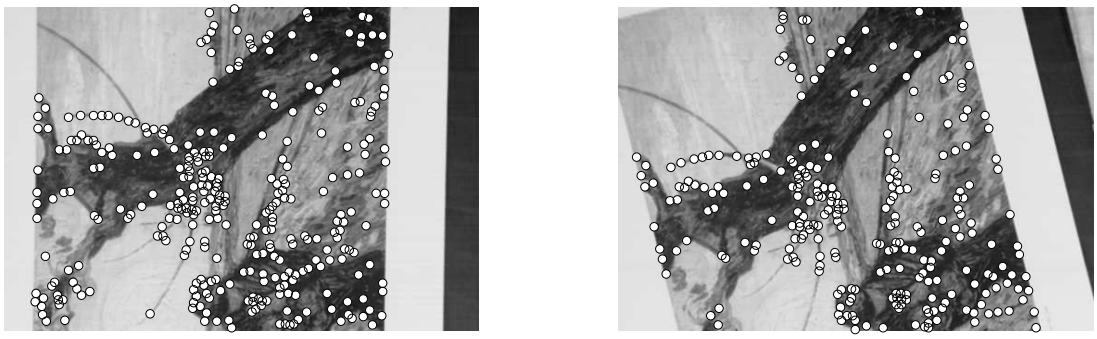
(g) Mira (h) Mira (i) Mira

Figura 29: Puntos detectados en tres diferentes escenas (i.e. VanGogh, carBox y Mira) por el operador de Dreschler y Nagel variando el umbral.



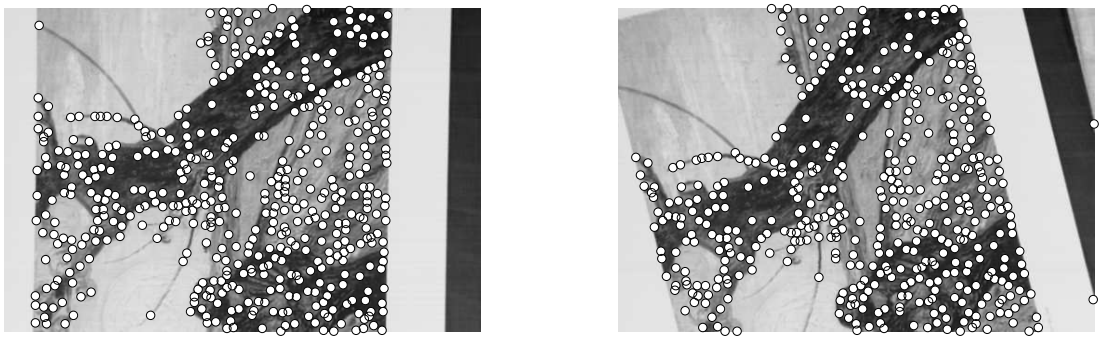
(a) (b)

Figura 30: Puntos detectados por el detector IPGP1. (a) Rotada 90°(b) Rotada 101°.



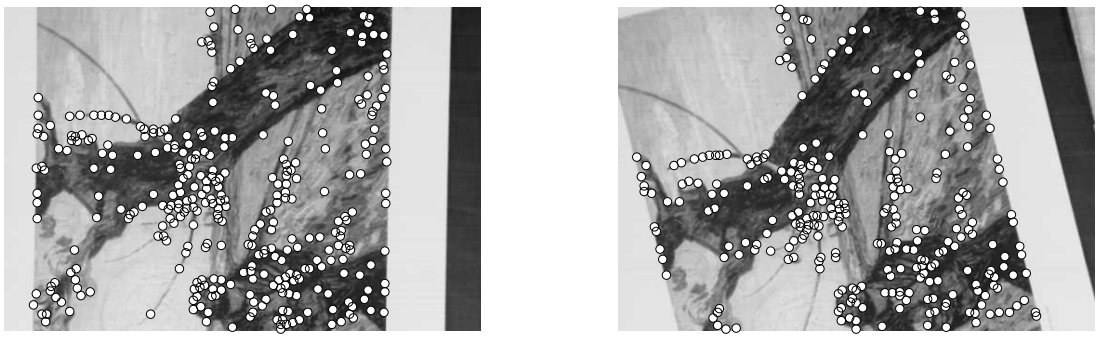
(a) (b)

Figura 31: Puntos detectados por el detector IPGP2. (a) Rotada 90°(b) Rotada 101°.



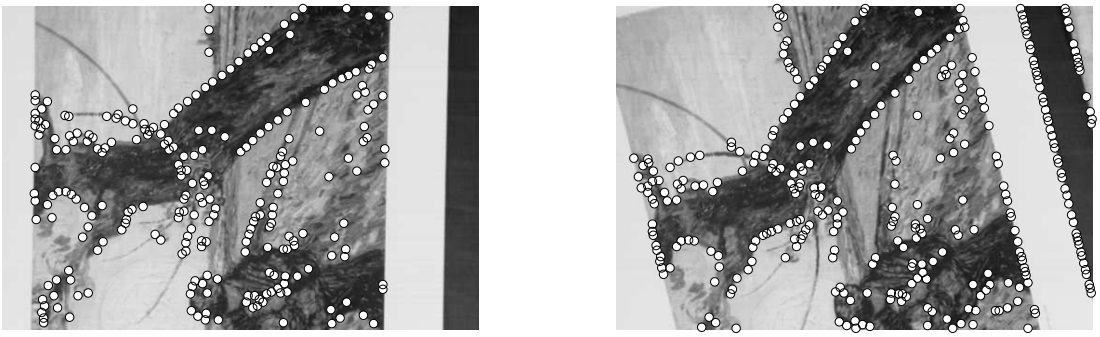
(a) (b)

Figura 32: Puntos detectados por el detector Harris y Stephen. (a) Rotada 90°(b) Rotada 101°.



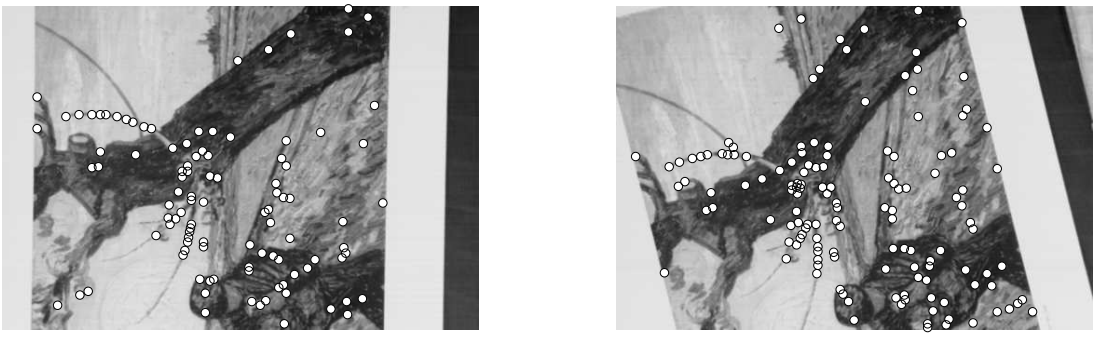
(a) (b)

Figura 33: Puntos detectados por el detector Beaudet. (a) Rotada 90°(b) Rotada 101°.



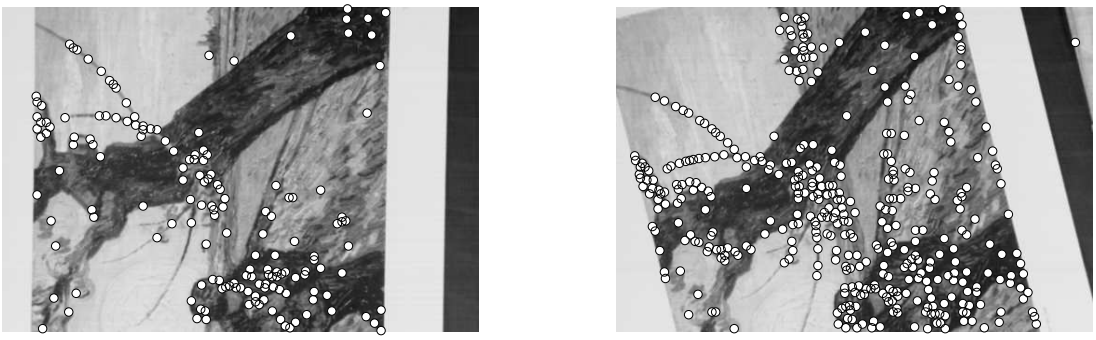
(a) (b)

Figura 34: Puntos detectados por el detector Förstner. (a) Rotada 90°(b) Rotada 101°.



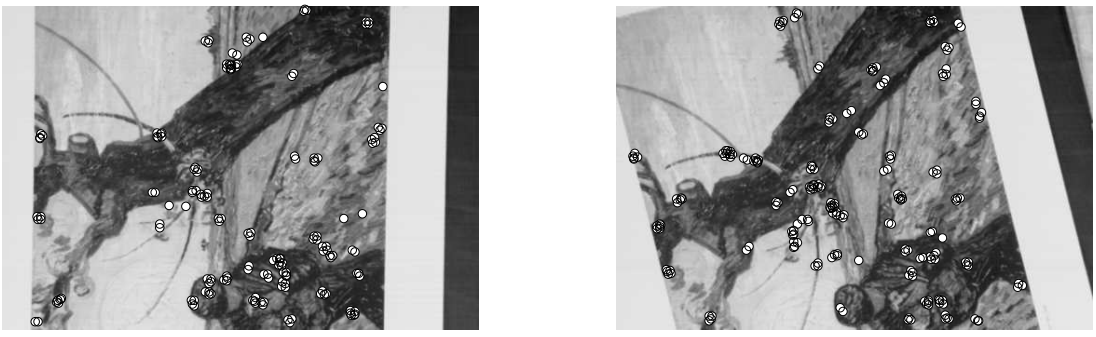
(a) (b)

Figura 35: Puntos detectados por el detector Wang y Brady. (a) Rotada 90°(b) Rotada 101°.



(a) (b)

Figura 36: Puntos detectados por el detector Kitchen y Rosenfeld. (a) Rotada 90°(b) Rotada 101°.



(a) (b)

Figura 37: Puntos detectados por el detector Drescler y Nagel. (a) Rotada 90°(b) Rotada 101°.

Capítulo V

Detectores de Puntos de Interés por Medio de Programación Genética

En este capítulo se aborda el problema de detección de puntos de interés como un problema de optimización. Para ello se hace uso de la programación genética, pues ha demostrado una poderosa capacidad de generar automáticamente soluciones eficientes y coherentes a problemas de optimización. Como resultado, se presenta un enfoque de aprendizaje de máquina, el cual permite construir automáticamente un detector de puntos de interés apropiado para ciertas tareas de alto nivel de la visión por computadora. Además de ser soluciones competitivas con las desarrolladas por el ser humano. Sin embargo, nuestro enfoque permite adecuar la obtención del detector de acuerdo a las necesidades del problema o a las tareas para las cuales se desea emplearlo. En este trabajo, nos interesa un detector "general", es decir, que sea útil en varias tareas. Dichas tareas son la correspondencia, reconocimiento de objetos y reconstrucción 3D. Para ello, el detector debe cumplir con ciertas características, principalmente, debe presentar estabilidad frente a transformaciones, pero también es necesario que exista separación global entre los puntos. La métrica para la estabilidad a tranformaciones es la repetibilidad, propuesta por (Schmid *et al.*, 2000), ver sección II.3.2. La separabilidad global fue abordada por (Trujillo y Olague, 2006). Para ello, aplican el concepto de entropía, ver sección II.3.3, con el objetivo de alcanzar la distribución "uniforme" de los puntos sobre la imagen.

En el presente trabajo se diseñó y desarrolló una aplicación genética que permite la

obtención automática de un detector de puntos de interés, utilizando la tasa de repetibilidad y entropia como los criterios de evaluación de aptitud de cada detector obtenido por la evolución genética. Dicha aplicación, permite utilizar una serie de operadores genéticos, el uso de ADFs (del inglés *Automatic Define Function*), pero principalmente, permite adecuar el criterio de evaluación de acuerdo a las necesidades de la tarea donde se desea utilizar el operador.

En este capítulo se explica a detalle la implementación de dicha aplicación genética. Así como los resultados obtenidos utilizando como conjunto de entrenamiento dos series de imágenes. Una serie de imágenes presenta transformaciones de rotación, la otra serie presenta cambios de iluminación y rotación. También se presentan los resultados obtenidos al aplicar a varias imágenes los detectores descubiertos por la evolución genética.

V.1 Aplicación Genética

Cada individuo de la población representa a un detector de puntos de interés. Para definir una aplicación de programación genética es necesario definir tres elementos: Función Aptitud, un Conjunto de Funciones y un Conjunto de Terminales.

V.1.1 Función Aptitud

Como se ha mencionado, nuestra función aptitud involucra los criterios de repetibilidad y entropía. Para ello, nuestro enfoque utiliza una función aptitud proporcional a la tasa de repetibilidad y entropía en las direcciones x y y.

La tasa de repetibilidad $r_J(\epsilon)$ es calculada para un conjunto $J = \{I_i\}$ de n imágenes de entrenamiento donde $i = 1...n$. Una imagen base I_i es utilizada como referencia para

calcular la repetibilidad con el resto de las imágenes en J. La tasa de repetibilidad entre la imagen de referencia y alguna otra imagen perteneciente a J se realiza como se explica en la sección II.3.2. En este trabajo, la tasa de repetibilidad expresa el porcentaje de regiones repetidas con un error de traslape de 30%. Así, el rango de la tasa de repetibilidad es de 0% a 100%. Sin embargo, la búsqueda de la programación genética fácilmente podría perderse en un máximo no deseado si no se toman en cuenta algunas consideraciones apropiadas. Por ejemplo, pueden existir individuos que detecten puntos agrupados en regiones sin textura, lo cual constituye información no útil, pero aún así, podrían obtener una repetibilidad alta. Más importante aún, las imágenes de entrenamiento utilizadas tienen regiones con gran textura distribuidas en el plano de la imagen. Por ello, un "buen" detector debería extraer puntos uniformente distribuidos sobre toda la imagen. Para lograrlo, se mide la entropía del histograma de localización de los puntos.

Entropía

Como se explica en la sección II.3.3, la entropía mide la aleatoriedad de una variable. Es necesario recordar, que el significado de aleatorio en probabilidad, significa que un elemento de un conjunto tiene la misma probabilidad de ser seleccionado con respecto al resto de los elementos del conjunto. Dicho en otras palabras, al medir la entropia de los subconjuntos o particiones (i.e., variables) de un conjunto, se esta midiendo que tanta igualdad existe de seleccionar algún subconjunto (i.e., aleatoriedad de la variable). Implícitamente, se esta midiendo que tan distribuida se encuentra la probabilidad de cada subconjunto o partición. Por esta razón, el concepto de entropía es aplicado al histograma de localización de los puntos, pues permite evaluar la distribución espacial de los puntos detectados sobre la imagen. Una mayor entropía representa una mejor distribución de los puntos sobre la imagen.

Para este trabajo, el histograma de localización de puntos se realiza de la siguiente manera:

1. Se obtienen las clases del histograma para las direcciones x y y. Cada clase tiene un intervalo de 8 pixeles. Por ello, el número de clases depende de las dimensiones de la imagen.

2. Se obtiene el número de pixeles pertenecientes a cada clase, es decir, se obtiene la frecuencia.

3. Se asigna la probabilidad de cada clase como

$$P(i) = \frac{\text{frecuencia de region(i)}}{\text{puntos detectados en la imagen}}$$

Así, la entropia en la dirección H_x y H_y del detector es calculada como

$$H_{(x,y)} = -\sum_{i=1}^{\frac{1}{8}dim_{(x,y)}} P(i) log_2[P(i)] \tag{44}$$

Sin embargo, debido a la naturaleza logarítmica de la entropía, los valores se encuentran en un rango muy pequeño. Con el propósito de ampliar este rango y promoviendo puntos de disperción a lo largo de las direcciones x y y y de posicionarlo en un intervalo de 0 a 1, se aplica una función sigmoidal, ver Figura 38,

$$\phi = \frac{1}{1 + e^{-x}}$$

Por lo tanto, la entropía del detector esta dada por las expresiones

$$\phi_x = \frac{1}{1 + e^{-\alpha(H_x - c)}} \tag{45}$$

$$\phi_y = \frac{1}{1 + e^{-\alpha(H_y - c)}} \tag{46}$$

Los parámetros a y c fueron estimados experimentalmente de la entropia media producida por los detectores de Harris y Beaudet.

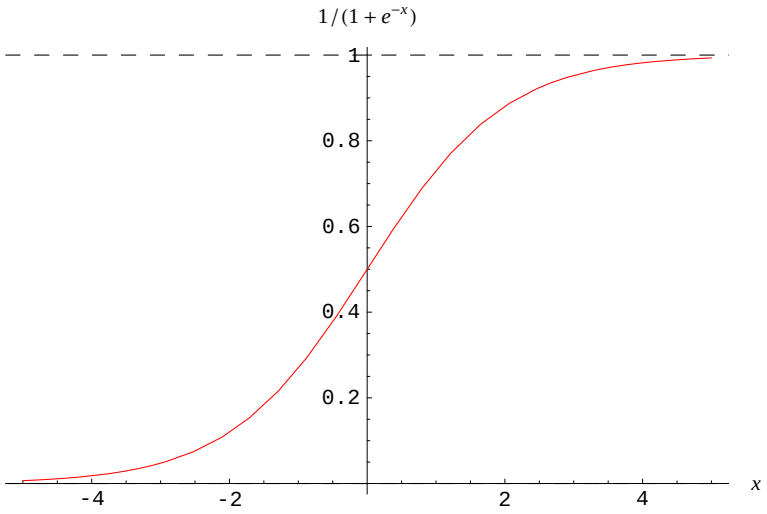


Figura 38: Función Sigmoidal.

A pesar de las consideraciones expuesta hasta este punto, la búsqueda de la programación genética aún puede generar individuos poco útiles, que pueden detectar tanto demasiados como insuficientes puntos de interés. Si se detectan demasiados puntos, la información es inútil, además de consumir los recursos computacionales. Para evitarlo, en este trabajo se seleccionó como método para determinar si un punto es de interés o no durante la supresión de no máximos, el *Establecer un Número Fijo de Puntos a obtener*, ver sección II.2.2. El número de puntos solicitados es de 500. Pero aún falta el caso cuando los puntos detectados son insuficientes. Para lidiar con él, se decidió agregar un término de penalización a la función aptitud. Este término consiste en penalizar a los individuos que detecten un número menor al solicitado. Esta dado por

$$N_{\%} = \frac{\text{número de puntos encontrados}}{\text{número de puntos requeridos}} \tag{47}$$

Así, una vez expuestas las consideraciones realizadas para el diseño de nuestro enfoque,

la función aptitud resultante es

$$f(x) = r_J(\epsilon) \cdot \phi_x^\alpha \cdot \phi_y^\beta \cdot N_\%^\delta \tag{48}$$

En este caso, los parámetros α, β y δ son para darle un peso a cada término y así controlar la influencia que cada uno ejerce sobre la función aptitud. Estos parámetros tambien fueron establecidos por experimentación.

V.1.2 Conjunto de Funciones

Las funciones que integran el conjunto F de Funciones, son aquellas que por su naturaleza consideramos apropiadas para la detección de puntos de interés. Ademas de que forman parte de los detectores que el ser humano ha desarrollado, por lo que han probado su utilidad para tal propósito. A pesar de que las consideramos suficientes para la construcción de un detector, no se descarta que otras funciones puedan generar iguales o mejores resultados.

Las funciones propuestas tienen aridades uno y dos, es decir, son funciones unarias y binarias, respectivamente. El conjunto de funciones unarias son cinco, y las binarias son siete. El conjnto de funciones es el siguiente:

$$F_{binaria} = \{+, -, |-|, *, /\} \tag{49}$$

$$F_{unaria} = \{A^2, \sqrt{A}, log_2, EQ, G(\sigma = 1), G(\sigma = 2), 0.25 * A\} \tag{50}$$

donde EQ es el histograma de equalización de la imagen y $G(\sigma = x)$ es un filtro Gaussiano con $\sigma = x$.

Así, el conjunto F de Funciones propuesto es:

$$F = F_{binaria} \cup F_{unaria} \tag{51}$$

V.1.3 Conjunto de Terminales

Para determinar el conjunto de terminales T apropiado para nuestra aplicación genética, se han estudiado detenidamente varios detectores de puntos de interés. Gracias al entendimiento de las propiedades analíticas de los detectores de esquinas descritos por (Schmid *et al.*, 2000; Olague y Hernández, 2005), podemos concluir que un detector de puntos de interés efectivo requiere información relativa a la tasa de cambio de los valores de intensidad de la imagen (i.e., las derivadas de una imagen). Se podría dejar que la evolución genética decidiera si el uso de las derivadas es útil para obtener un detector competitivo, al incluirlas en el conjunto de funciones F. Sin embargo, debido a que ya se conoce que las derivadas y el suavizado de imágenes son, efectivamente, muy útiles en la detección de características, se decidió "ahorrarle" a la evolución dicho descubrimiento y proporcionarle directamente las imagenes derivadas y suavizadas. El conjunto de terminales propuesto es el siguiente:

$$T = \{I_i, I_{i,\sigma=1}, L_{i,x}, L_{i,y}, L_{i,x,x}, L_{i,y,y}, L_{i,x,y}\} \tag{52}$$

donde I_i es la imagen original, $L_{i,w}$ es la derivada de la imagen en la dirección w calculada con filtros recursivos que se aproximan a las derivadas de la función Gaussiana y $I_{i,\sigma=1}$ es la imagen suavizada con una aproximación al filtro Gaussiano con $\sigma = 1$. El conjunto es dependiente de la imagen, por lo que cada imagen $I_i \in J$ tiene un conjunto de terminales correspondientes T_i. Al igual que con el conjunto de funciones, creemos que este conjunto de terminales es suficiente para alcanzar el objetivo del presente trabajo, pero no afirmamos que sea el conjunto óptimo.

V.2 Experimentación

Para realizar la experimentación, se diseño e implementó una aplicación genética que incorpora lo propuesto en la sección V.1. En esta sección se muestran la arquitectura

de dicha aplicación genética, los detalles de implementación así como los resultados de los experimentos y pruebas realizadas.

V.2.1 Arquitectura de la Aplicación Genética

Nuestro programa genético fue desarrollado con las herramientas computacionales: VXL, LILGP y CImg. Para describir la arquitectura de nuestra aplicación, es necesario que primero se presente la descripción de las herramientas utilizadas.

VXL

Vision-Something-Libraries por sus siglas en inglés, en español significa Bibliotecas-De algo-para Visión). Es una colección de bibliotecas en C++ diseñadas para la investigación de visión por computadora. VXL está programado en ANSI/ISO C++ y está diseñado para ser portable en diferentes plataformas. Las bibliotecas fundamentales de VXL son:

- **vnl (numéricas).** Matrices, vectores, descomposiciones, etc.
- **Vil (imágenes).** Carga, almacenamiento y manipulación de imágenes en formatos de archivos comunes. Incluyendo imágenes muy grandes.
- **Vgl (geometría).** Geometría para puntos, curvas y otros objetos elementales en una dos o tres dimensiones.
- **Vsl (streaming I/O), vbl (basic templates) y vul (utilities).** Funcionalidades independientes de plataforma.

LILGP

Es una herramienta genérica de programación genética desarrollada en lenguaje C. Fue

diseñada con un número de metas en mente: velocidad, facilidad de uso y soporte para un número de opciones incluyendo:

- Facilidad de uso: Los parámetros son especificados por medio de una archivo de parámetros.
- Programas genéricos en C para propocionar portabilidad, que pueden ejecutarse tanto en estaciones de trabajo como en PC's.
- Soporte para múltiples subpoblaciones con topologías de intercambio arbitrarias.
- Manipulación de árboles cuyos nodos son apuntadores a funciones. Los árboles son localizados en un gran bloque de memoria para obtener mayor velocidad y evitar el intercambio de memoria. Por ello, permite un gran número de individuos y de generaciones.
- Siete métodos de selección: Proporcional a la aptitud, Selección voraz, Aptitud inversa, Torneo con tamaño variable, Aleatorio, Mejor individuo y Peor individuo.
- Tres operadores genéticos: Cruzamiento, Reproducción y Mutación.
- Archivos de salida extensos, incluyendo estadísticas.
- Soporte para ADF (*Automatic Defined Function*).

Biblioteca CImg

La biblioteca CImg es una herramienta de código abierto para el procesamiento de imágenes. Provee clases simples y funciones para cargar, guardar, procesar y desplegar imágenes desde cualquier código C++. Es portátil, pues trabaja en diferentes sistemas operativos (Unix/X11, Windows, MacOS X, FreeBSD) con diferentes compiladores de C++. Consiste únicamente en un archivo de cabecera CImg.h, que debe ser incluido en el programa fuente donde desea ser utilizado. Depende de un número mínimo de

bibliotecas estandar de C++. Contiene algoritmos de procesamiento de imágenes muy útiles como las derivadas y filtros Gaussianos, utilizados en este trabajo.

Debido a que el trabajo previo del grupo de Evovisión, dentro del cual se encuentra el presente trabajo, ha sido realizado en VXL, era necesario que nuestra aplicación genética se ejecutara desde VXL. Para ello, fue necesario agregar LILGP a VXL. Como ya se explicó, LILGP se encuentra en lenguaje C, mientras que vxl en C++. Por lo que fue necesario tener ciertas consideraciones al diseñar nuestra aplicación, así como realizar el ligado apropiado para que el compilador utilizado por VXL pudiese compilar el código de LILGP.
Nuestra aplicación se podría dividir en básicamente 3 secciones: las funciones en lilgp, el programa principal en vxl y diversas funciones.

Funciones en LILGP

LILGP tiene una estructura que facilita su uso para diferentes problemas, por ello se divide en dos partes: *kernel* y *app*. El Kernel es el núcleo, en el se encuentran todos los programas que realizan las operaciones "genéricas" de cualquier herramienta de programación genética; como por ejemplo, los metodos de selección y los operadores genéticos, por mencionar algunos. Por esto, los programas del kernel no deben ser modificados. App es la parte de LILGP que debe ser modificada por el usuario para resolver algún problema. Esta es la razón de que sólo nos concentremos en App. App consiste en los programas appdef_dpi.h, app_dpi.c, app_dpi.h, function_dpi.c, function_dpi.h y el archivo de parámetros input.file. A continuación se explican:

- **input.file** Contiene parámetros como tamaño de la población, número de generaciones, método para inicializar la población, profundidad de inicialización, profundidad máxima, operadores genéticos a utilizar así como sus probabilidades

y su método de selección, nombre de archivos de salida, si se emplean adf, etc. Los parámetros utilizados para la obtención de nuestros resultados se presentan en la sección V.2.2.

- **appdef_dpi.h** Se define el tipo de datos que cada nodo devuelve, el cual es llamado DATATYPE, así como el número máximo de argumentos de cada función. Para nuestra aplicación, el dato devuelto son las imágenes, por lo que el tipo es una matriz bidimensional dinámica de tipo flotante con dimensiones iguales a las de las imágenes. El número máximo de argumentos son dos.

- **app_dpi.h** Se define una estructura de datos llamada "globaldata" necesaria para pasar información entre la función de evaluación de los individuos y las funciones y terminales. En nuestro caso, esta estructura incluye el número de terminales, número de imágenes y sus dimensiones, la matriz que contiene al conjunto de terminales y las variables auxiliares para las terminales.

- **app_dpi.c** Contiene un serie de funciones que:

  - Construyen el conjunto de terminales T y funciones F (si se utilizan ADF's, se deben tener en cuenta consideraciones especiales),
  - inicializan los datos específicos del problema,
  - evaluan la aptitud del individuo,
  - realizan acciones: al terminar la evaluación de los individuos, al finalizar cada generación (por ejemplo, se podría mostrar al mejor individuo de la generación) y al terminar la creación de la nueva población y
  - liberan las estructuras utilizadas.

  Para nuestra aplicación, en la función de inicialización, se cargan las imágenes de entrenamiento y se procesan para obtener las terminales. Todos los datos

de la estructura "globaldata" son inicializados en esta función. En la función de creación del conjunto de funciones y terminales, se especifíca una tabla de conjuntos con ciertas consideraciones para permitir el uso de ADF's. En la función de evaluación de aptitud, se evalua el individuo con todas las imágenes que componen el conjunto de entrenamiento, se obtienen la tasa de repetibilidad media y la entropía y se cálcula la función aptitud.

- **function_dpi.h** Se definen los prototipos de las funciones contenidas en function_dpi.c.

- **function_dpi.c** Por cada función del conjunto F y cada terminal del conjunto T, se define una función que recibe como argumentos un dato int y un arreglo de tipo farg y devuelve un tipo DATATYPE. Los argumentos que debe recibir son *int tree* y *farg *args*. Esto no debe ser modificado, pues es la estructura que deben de tener para la ejecución por parte de LILGP.

Programa Principal en VXL

Está conformado por los archivos llamados xcv_dets.cxx y xcv_dets.h. En ellos, se define la clase xcv_dets, la cual tiene un método llamado *interes()*. Este método, es el que realiza la ejecución de la aplicación genética y no requiere modificación alguna. Salvo que la ubicación del archivo de parámetros cambie. Este método llama varias funciones de LILGP, tanto del kernel como de app. Por medio de dichas funciones, se crea la población inicial, los archivos de salida, se cargan los parámetros del archivo de parámetros, se crea la tabla de conjuntos de funciones y terminales, se realiza la evolución hasta alcanzar el criterio de paro (i.e., creación de poblaciones por medio de selección, cruzamiento, reprodución, mutación y evaluación de aptitud) y al finalizar se liberan las estructuras y memoria utilizadas.

Diversas Funciones

Se encuentran en el archivo varias_func.cxx y varias_func.h. Contiene las funciones para el cálculo de entropía y tasa de repetibilidad, crear las terminales, realizar la supresión de no máximos y para aplicar varias funciones de la biblioteca CImg como suavizado y equalización. La función de entropia es la versión en C++ de la función codificada en MATLAB utilizada por (Schmid *et al.*, 2000). Esta se realiza de acuerdo a como se presenta en la sección II.3.2.

Las funciones diversas que se mencionan, son en gran parte, consecuencia de la adaptación de LILGP a VXL. Esto debido a que existen ciertas estructuras de C++ que no pueden ser utilizadas desde C. Por ello, fue necesario crear funciones "puente" entre C y C++. Dichas funciones, principalmente son para utilizar la biblioteca CImg. Esta biblioteca es crucial para la obtención de las terminales y funciones del programa genético y por ello deben ser llamadas desde LILGP. También fue necesaria una función "puente" para utilizar las bibliotecas de VXL dentro del cálculo de repetibilidad.

V.2.2 Detalles de Implementación

Las imágenes de entrenamiento y prueba fueron obtenidas del sitio web del Visual Geometry Group. Las imágenes de entrenamiento consisten en el conjunto de imágenes VanGogh y Leuven. VanGogh presentan transformaciones de rotación y Leuven presenta tanto rotación como cambios de iluminación. Para las imágenes de prueba se utilizaron cinco conjuntos de escenas: Monet, Mars, New York, Graph y Mosaic. Monet, Mars y New York presentan transformaciones de rotación y Graph y Mosaic cambios de iluminación. En la sección IV.5 se explica como fueron tomadas las imágenes que presentan rotación. Las imágenes con variación en la iluminación fueron obtenidas cambiando la apertura del lente de la cámara. El cambio es cuantificado por el "valor

de gris relativo", esto es la razón de la media de una imagen con respecto a la media de la imagen de referencia.

Se realizaron dos tipos de experimentos, el experimento 1 utiliza como imagen de entramiento la escena VanGogh y el experimento 2 utiliza dos imágenes de entrenamiento, la escena VanGogh y Leuven. Para el experimento 1 las imágenes de prueba son seis: Monet, Mars, New York, Graph, Mosaic y Leuven. Para el experimento 2 las imágenes de prueba son cinco: Monet, Mars, New York, Graph y Mosaic.

Para las pruebas, se seleccionaron además de los detectores obtenidos en el presente trabajo, los tres detectores que mostraron mejor desempeño en los experimentos realizados en el capítulo IV (i.e., Harris, IPGP1 e IPGP2). Cada detector seleccionado utilizó como método para determinar el umbral, el de *Establecer un Valor Mínimo para considerarlo Punto de Interés*, ver sección II.2.2. El umbral establecido no fue modificado para adaptarse al tipo de escena. Por ejemplo, el DPIPG1 utilizó un umbral de 30% del valor máximo observado, para las ocho escenas. Esto, con el propósito de observar qué tan "generales" son al aplicarse con diferentes imágenes, sin "personalizarlos". En algunas tareas como segmentación, detección de características y otras del procesamiento de imágenes, por mencionar algunas, es práctica usual adaptar el umbral a las imágenes.

Los parámetros del programa genético que se utilizaron para la realización de los experimentos que llevaron a la obtención de los detectores de puntos de interés, cuyos resultados son reportados en esta sección, son los siguientes:

- **Tamaño de la Población e Inicialización:** El tamaño de la población fue de 200 individuos inicializados con el método Ramped Half-and-Half. La profundidad de inicialización fue de 2-6. La profundidad máxima de los individuos fue de 7 niveles.

- **Operadores Genéticos:** Se utilizaron los operadores de cruzamiento, mutación y reproducción. Las probabilidades fueron $P_c = 0.85$, $P_m = 0.10$ y $P_r = 0.10$ respectivamente.

- **Método de Selección:** Se utilizó el método de torneo, con un tamaño de 4 individuos.

- **Parámetros de la Función Aptitud:** Como ya se explicó, estos fueron determinados experimentalmente y son los siguientes: $a_x = 7$, $c_x = 5.05$, $a_y = 6$, $c_y = 4.3$, $\alpha = 10$, $\beta = 10$, $\delta = 2$.

V.2.3 Resultados

Se presentan cuatro detectores de puntos de interés generados con el enfoque propuesto en el presente trabajo: DPIPG1 y DPIPG2 obtenidos en el experimento 1 y DPIPG5 y DPIPG6 obtenidos en el experimento 2 (***D****etectores de* ***P****untos de* ***I****nterés por medio de* ***P****rogramación* ***G****enética*). Cada uno fue generado en una ejecución diferente de la aplicación. A pesar de haber realizado varios experimentos y encontrado varios detectores de puntos, presentamos los detectores que obtuvieron mejor desempeño. Además, creemos que estos ejemplos son suficientes para demostrar la utilidad de nuestro enfoque.

Cada detector fue probado con una serie de escenas. Algunas de estas, son imágenes que observan características similares a las que se utilizaron para el entrenamiento, es decir, con rotación y texturas. Estas escenas son Mars, New York y Monet. También se utilizaron dos escenas que muestran variaciones en iluminación, llamadas Graph y Mosaic. Y por último, se utilizó una escena que presenta tanto rotación como iluminación llamada Leuven.

DPIPG1

Como se aprecia en la Figura 39, la estructura de este detector es muy simple. Consiste básicamente en dos pasos: el primero es aplicar el Laplaciano a la imagen, ver Ecuación (26), el segundo es suavizar varias veces las frecuencias extraidas con una Gaussiana. Al aplicar el Laplaciano se extraen las altas frecuencias de la imagen, y al suavizarlo, se eliminan los puntos aislados (i.e., ruido).

El utilizar varias veces la función de suavizado equivale a suavizar con una σ grande. Con esto, los detalles se pierden y en su lugar se obtienen regiones donde la frecuencia es alta. Algo notable, es que sin haber declarado el Laplaciano como una función, nuestro enfoque de aprendizaje lo "descubrio" por si mismo como una operación útil en la detección de características. Es conocido que una desventaja del Laplaciano, es su susceptibilidad frente al ruido. Por ello, resulta aun más notable, que la evolución genética "compensará" dicha desventaja eliminando el ruido por medio de un filtro de suavizado gaussiano. Este detector es bueno con imágenes que no presentan variaciones abruptas de intensidad, debido al Laplaciano.

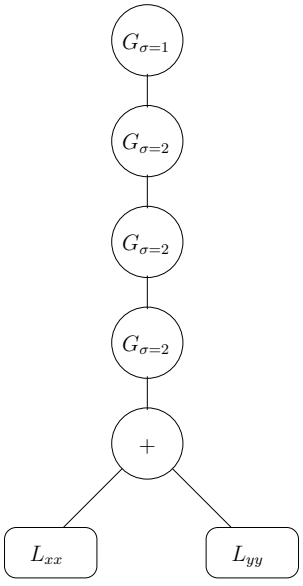


Figura 39: Representación en árbol del DPIPG1.

A pesar de la simpleza de operación de este detector, se obtuvo una tasa de repetibilidad media de 98.3378% con el conjunto de imágenes de entrenamiento, el cual es el principal marco de referencia en (Schmid *et al.*, 2000). Con el conjunto de imágenes de prueba presentó, 86.7308% con Monet, 95.5344% con Mars, 94.2436% con New York, 94.6304% con Graph, 96.5193% con Mosaic y 70.7416% de repetibilidad media con Leuven. En la Tabla V y Figuras 50, 43, 44,45, 46, 47, 48 y 49 se presenta una comparación de los resultados obtenidos por varios detectores para ambos conjuntos de imágenes.

DPIPG2

El segundo detector encontrado por nuestro enfoque, véase Figura 40, presenta también una estructura simple. No utiliza derivadas para extraer regiones con alta frecuencia. Su funcionamiento consiste en primero extraer las frecuencias altas al restar la imagen suavizada con $\sigma = 1$ de la imagen suavizada con $\sigma = 2$. Después, al aplicar los filtros de suavizado a dichas frecuencias, obtiene los puntos más sobresalientes, pues elimina los detalles. Con el conjunto de entrenamiento obtuvo 97.7527% de tasa de repetibilidad media, con el conjunto de prueba obtuvo 89.4079% con Monet, 95.7237% con Mars, 94.6484% con New York, 93.6785% con Graph, 96.4592% con Mosaic y 71.2243% de tasa de repetibilidad media con Leuven. En la Tabla V y Figuras 50, 43, 44,45, 46, 47, 48 y 49 se presentan una comparación de los resultados obtenidos por varios detectores para ambos conjuntos de imágenes.

DPIPG5

Es el primer detector encontrado con dos imágenes de entrenamiento, véase Figura 41, presenta una estructura un poco más compleja que los detectores anteriores. Utiliza derivadas de segundo orden. Su funcionamiento es básicamente el siguiente: extrae las frecuencias altas al restarle a la segunda derivada la imagen original, aplica filtros de

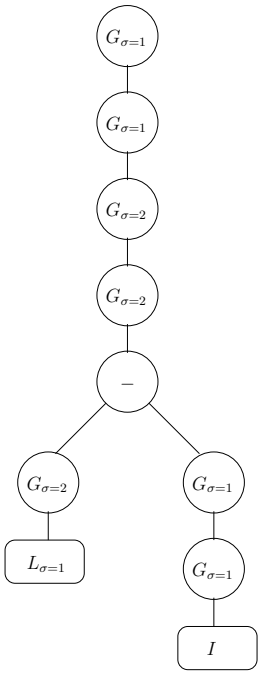


Figura 40: Representación en árbol del DPIPG2.

suavizado y el resultado es extraido de ;la imagen. Por último aplica filtros de suavizado para obtener los puntos más sobresalientes. Este detector tiende a detectar los bordes. Con el conjunto de entrenamiento obtuvo 96.4974% de tasa de repetibilidad media con VanGogh y 76.2103% con Leuven, con el conjunto de prueba obtuvo 86.6968% con Monet, 96.0086% con Mars, 95.3701% con New York, 93.7633% con Graph y 95.9872% con Mosaic. En la Tabla V y Figuras 50, 43, 44,45, 46, 47, 48 y 49 se presentan una comparación de los resultados obtenidos por varios detectores para ambos conjuntos de imágenes.

DPIPG6

Es el segundo detector encontrado por el experimento 2, véase Figura 42, presenta una estructura sumamente sencilla. No utiliza derivadas, sólo extrae las altas frecuencias de la imagen al restar la imagen suavizada de la imagen original. Posteriormente, lo suaviza varias veces para obtener las regiones con altas frecuencias. Al igual que DPIPG5,

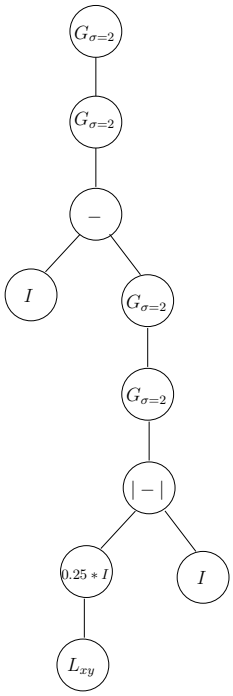


Figura 41: Representación en árbol del DPIPG5.

tiende a detectar los bordes. Con el conjunto de entrenamiento obtuvo 95.9042% de tasa de repetibilidad media con VanGogh y 76.0811% con Leuven, con el conjunto de prueba obtuvo 82.7285% con Monet, 95.8533% con Mars, 96.5738% con New York, 95.0355% con Graph y 95.8226% con Mosaic. En la Tabla V y Figuras 50, 43, 44,45, 46, 47, 48 y 49 se presentan una comparación de los resultados obtenidos por varios detectores para ambos conjuntos de imágenes.

Rotación

Las escenas a las que se aplica una rotación son VanGogh, Monet, Mars y New York. Todas ellas presentan, como característica adicional, riqueza en texturas.

Los detectores que obtuvieron mejor desempeño en todas las escenas, con excepción de la escena Monet, fueron los cuatro detectores obtenidos con nuestra aplicación, el DPIPG1, DPIPG2, DPIPG5 y DPIPG6. Con Monet, fue el IPGP2 el que presentó

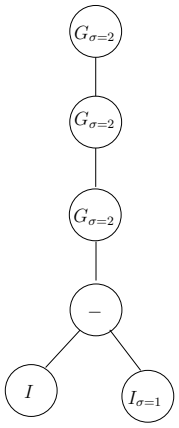


Figura 42: Representación en árbol del DPIPG6.

mejores resultados. Sin embargo, para obtener mejores resultados con la escena de Monet,es necesario disminuir el umbral para detectar las variaciones que no son tan drásticas. Esto, debido a que es una escena con textura pero con regiones que tienen pocos contrastes, por lo que todos los detectores, incluyendo IPGP2, se concentran en las regiones de mayor variación, las cuales son escasas. En las Figuras 43,44,45,46,47,48 se presentan las gráficas del desempeño de los cinco detectores seleccionados con las imágenes rotadas.

Como conclusión, se puede decir, que debido al entrenamiento y a los resultados obtenidos, los detectores DPIPG1, DPIPG2, DPIPG5 y DPIPG6 son muy estables y apropiados para imágenes con las características antes expuestas.

Cambios de Iluminación

Las escenas utilizadas son Graph y Mosaic. Graph presenta una variación de iluminación muy pequeña. Mosaic, observa una variación un poco mayor, y sus imágenes tienen poco contraste pero además se encuentran en niveles bajos de gris, es decir, son oscuras, aun en la imagen con mayor iluminación.

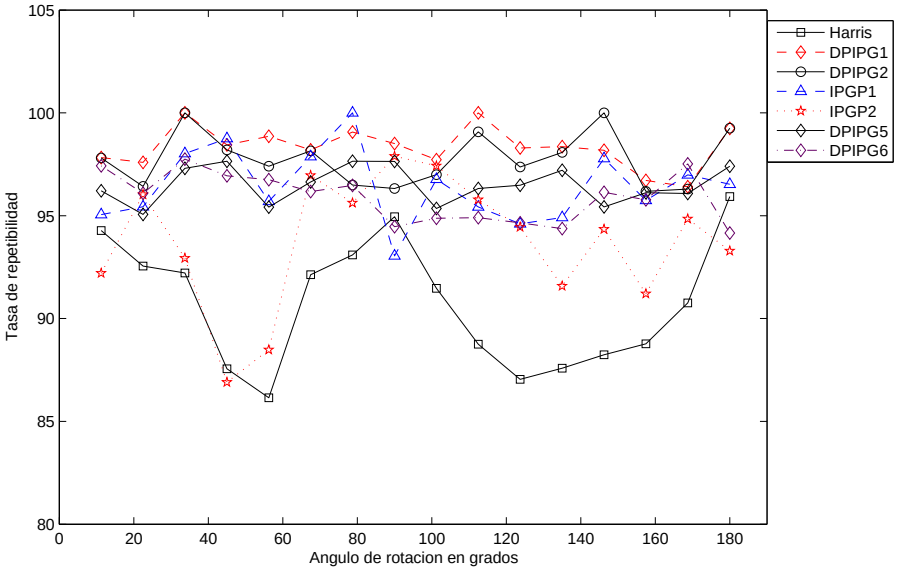


Figura 43: Medidas de desempeño: Tasa de repetibilidad graficada contra rotación de las 16 imágenes de la escena VanGogh. Rotación de 180°.

En la escena Graph, el detector que presentó mejor desempeño fue Harris. Esto debido a que las imágenes de Graph tienen bordes y esquinas muy definidas; características para lo cual Harris es muy apto. Sin embargo, para esta prueba la variación en la iluminación es tan pequeña y uniforme, que no representa un gran problema. Por lo que sería bueno probarlos bajo condiciones de iluminación más demandantes.

En la escena Mosaic, el mejor detector fue DPIPG1, seguido por DPIPG2. En las Figuras 47 y 48 se presentan las gráficas del desempeño de los cinco detectores seleccionados con las imágenes que presentan variaciones en la iluminación.

Rotación y Cambios de Iluminación

La escena utilizada es Leuven. Leuven presenta rotación y cambios de iluminación muy grandes. Los detectores que presentarón mejor desempeño fueron DPIPG5, DPIPG6,

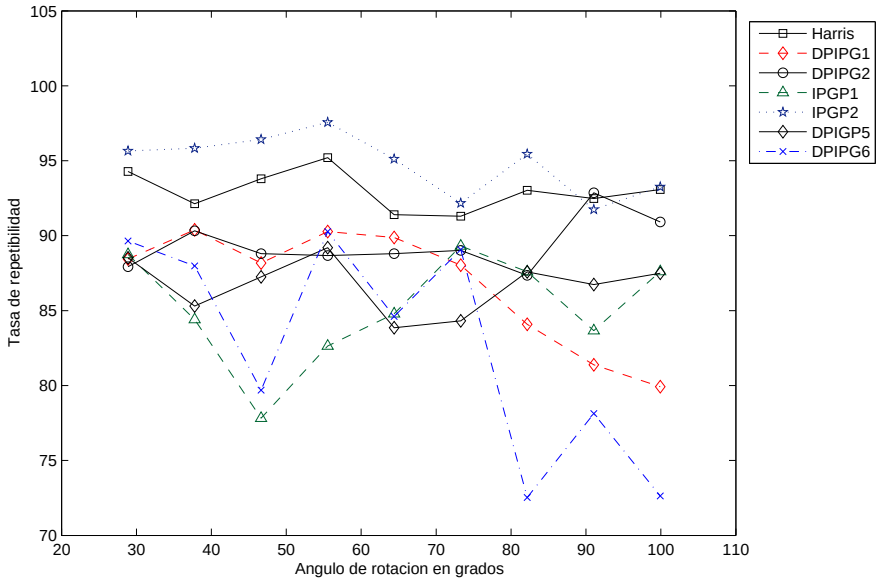


Figura 44: Medidas de desempeño: Tasa de repetibilidad graficada contra rotación de las 10 imágenes de la escena Monet. Rotación de 100°.

DPIPG2 y DPIPG1. Hay que recordar, que para el experimento 2, esta escena fue parte del conjunto de entrenamiento, por ello, los detectores DPIPG5 y DPIPG6 son los que presentan mejor desempeño. En la Figura 49 se presenta la gráfica del desempeño de los cinco detectores seleccionados con la escena Leuven.

Nuestros detectores mostraron un gran desempeño en todas las escenas que se utilizaron. Sin embargo, se demuestra, que no se puede afirmar que un detector de puntos de interés es el mejor de todos. Su desempeño depende de las características de la imagen y de las condiciones de observación. Algunas veces será suficiente con adaptar el umbral a la imagen, pero en otras ocasiones, lo más recomendable será seleccionar otro detector. En las Figuras 51, 52, 53, 54, 55, 56 y 57 se muestran los puntos detectados para una imagen de cada una de las secuencias estudiadas.

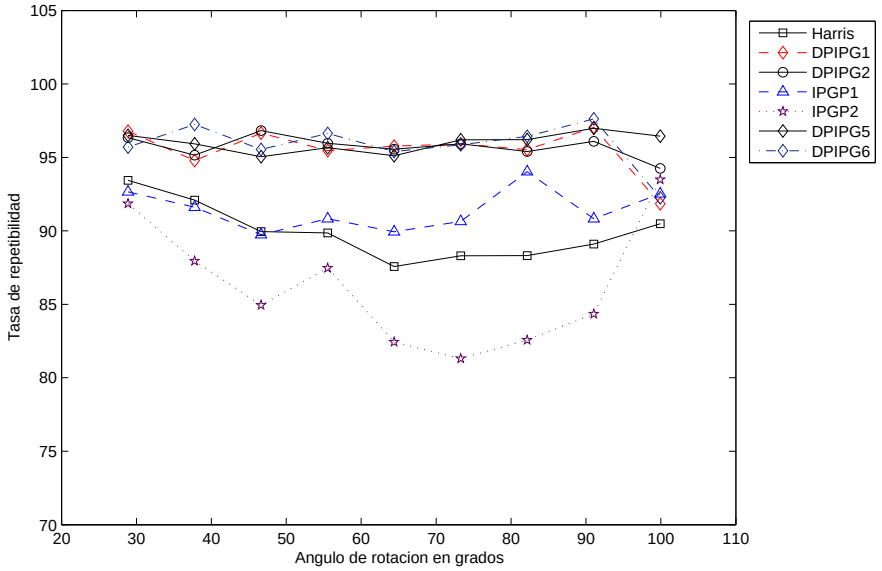


Figura 45: Medidas de desempeño: Tasa de repetibilidad graficada contra rotación de las 10 imágenes de la escena Mars. Rotación de 100°.

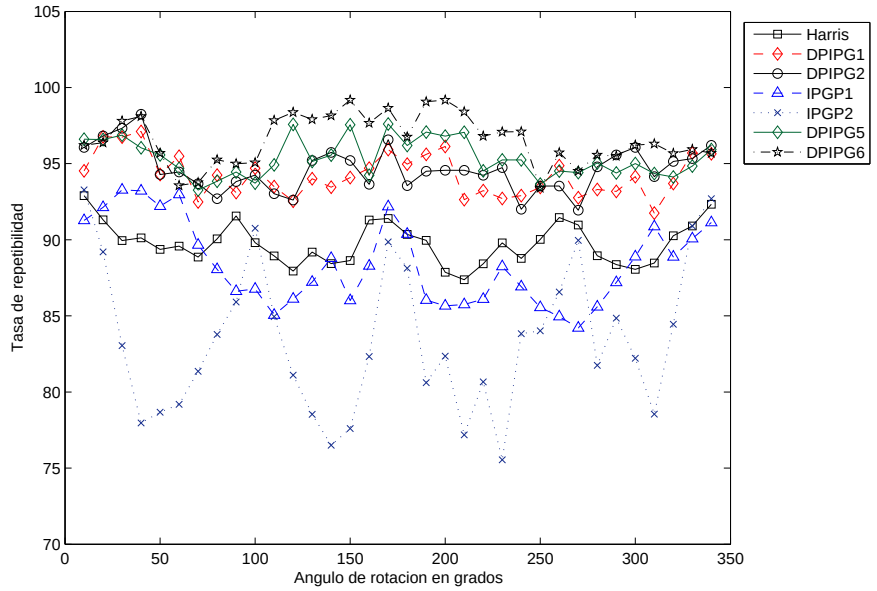


Figura 46: Medidas de desempeño: Tasa de repetibilidad graficada contra rotación de las 35 imágenes de la escena New York. Rotación de 340°.

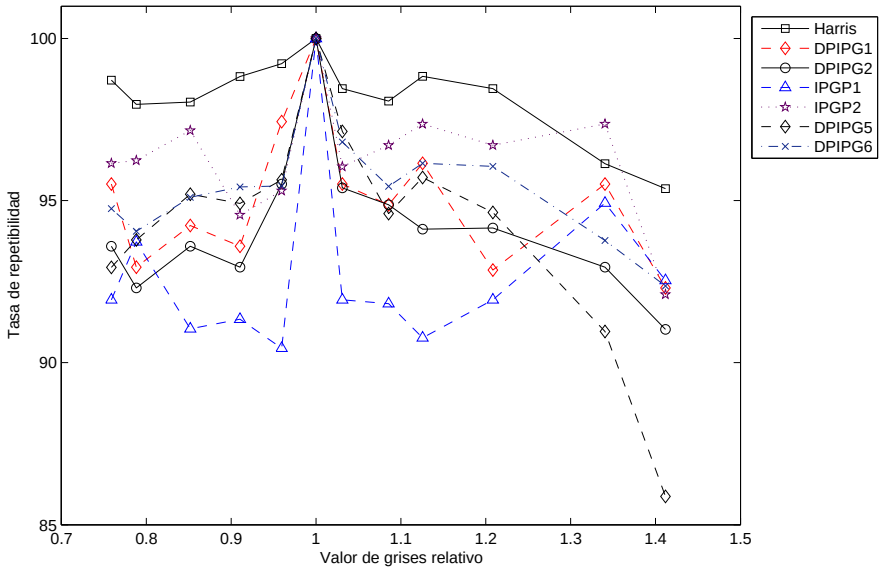


Figura 47: Medidas de desempeño: Tasa de repetibilidad graficada contra el valor relativo de grises de las 12 imágenes de la escena Graph.

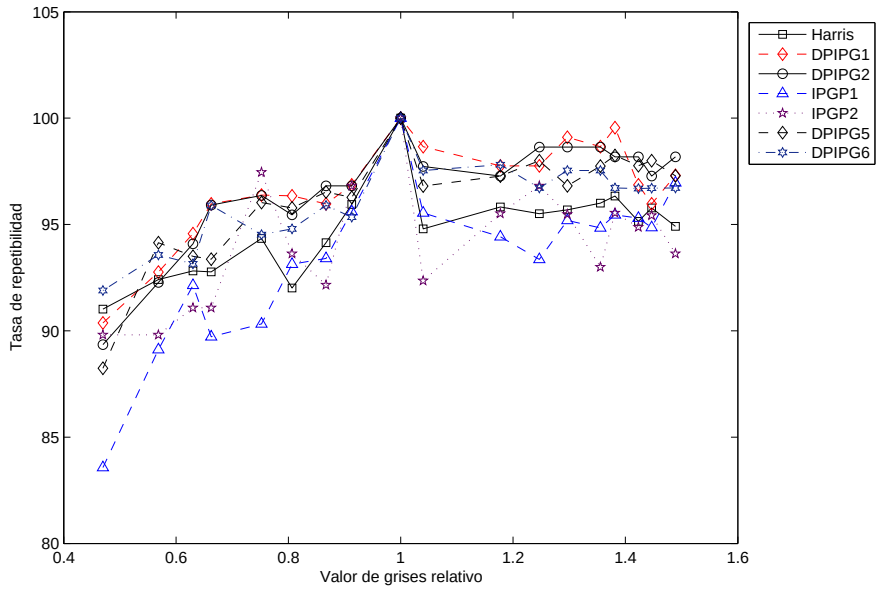


Figura 48: Medidas de desempeño: Tasa de repetibilidad graficada contra el valor relativo de grises de las 18 imágenes de la escena Mosaic.

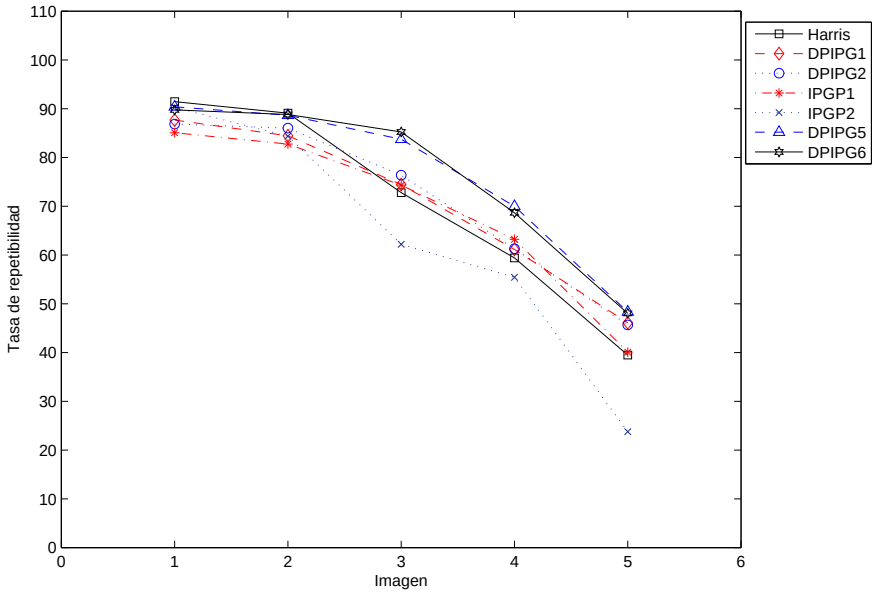


Figura 49: Medidas de desempeño: Tasa de repetibilidad graficada contra el valor relativo de grises para las 6 imágenes de la escena Leuven.

Tabla V: Tasa de Repetibilidad media obtenida con los conjuntos de imágenes de entrenamiento y prueba.

	Tasa de repetibilidad media						
Detector	VanGogh	Monet	Mars	New York	Graph	Mosaic	Leuven
IPGP1	96.4144	85.1765	91.4207	88.4192	92.0404	93.1133	69.0689
IPGP2	93.7467	94.8041	86.2660	83.4790	95.9757	93.7923	63.1896
Harris	90.7172	92.9669	89.9034	89.7559	98.0091	94.4375	70.4231
DPIPG1	98.3378	86.7308	95.5344	94.2436	94.6304	96.5193	70.7416
DPIPG2	97.7527	89.4079	95.7237	94.6484	93.6785	96.4592	71.2243
DPIPG5	96.4974	86.6968	96.0086	95.3701	93.7633	95.9872	76.2103
DPIPG6	95.9042	82.7285	95.8533	96.5738	95.0355	95.8226	76.0811

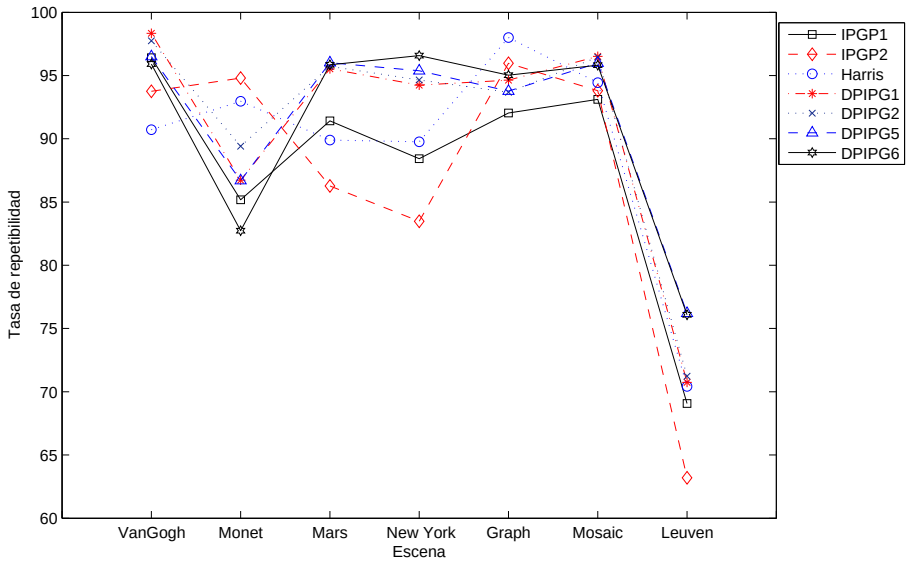
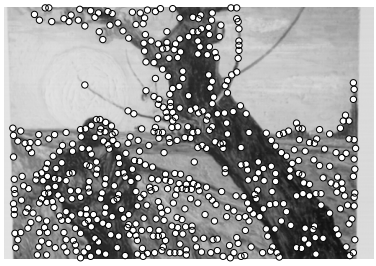
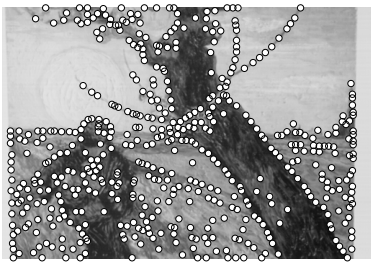


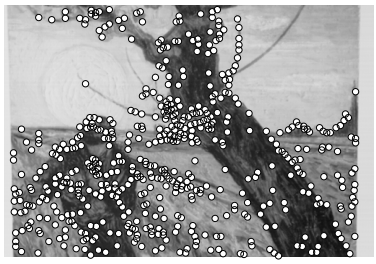
Figura 50: Medidas de desempeño: Tasa de repetibilidad obtenidad por cada detector en cada una de las 7 escenas utilizadas.



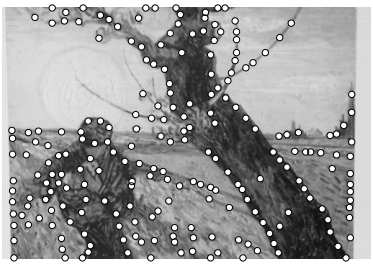
(a) Harris



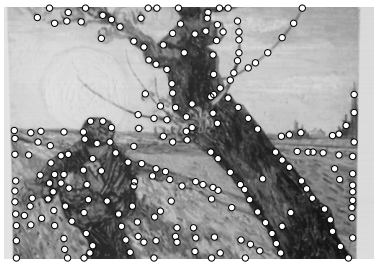
(b) IPGP1



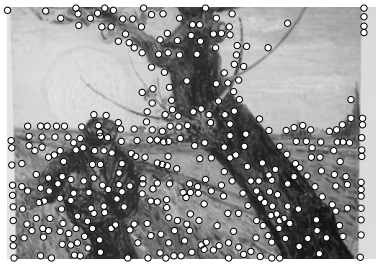
(c) IPGP2



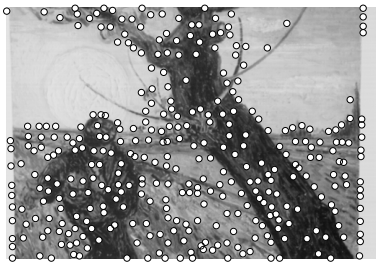
(d) DPIPG1



(e) DPIPG2

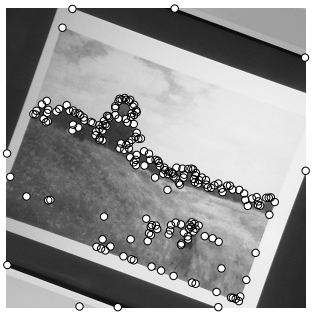


(f) DPIPG5

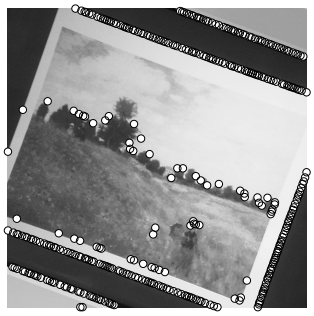


(g) DPIPG6

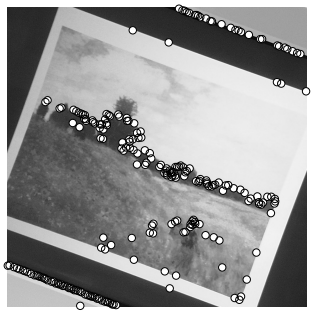
Figura 51: Imagen muestra de VanGogh, con los puntos de interés extraídos por cada detector.



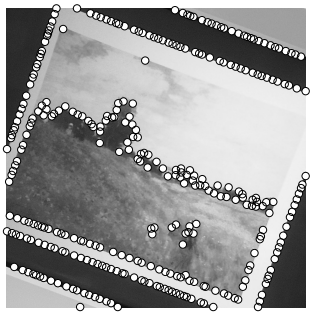
(a) Harris



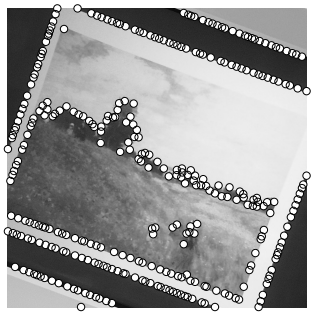
(b) IPGP1



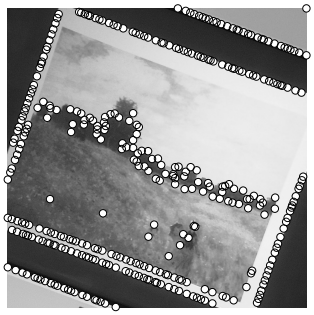
(c) IPGP2



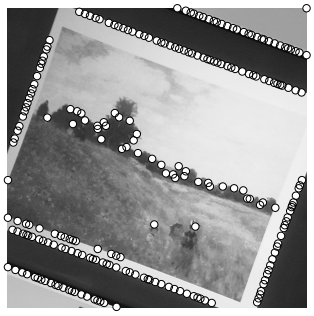
(d) DPIPG1



(e) DPIPG2

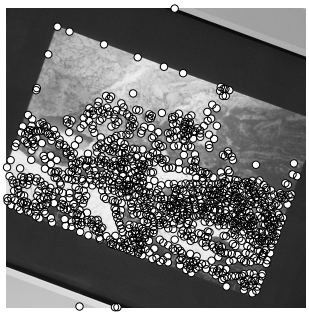


(f) DPIPG5

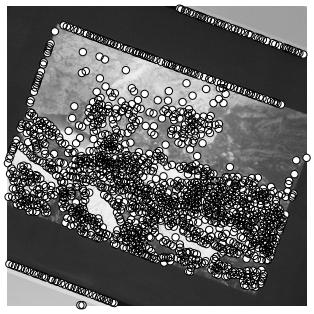


(g) DPIPG6

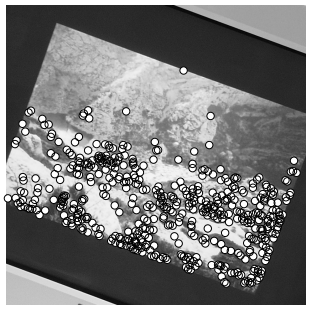
Figura 52: Imagen muestra de Monet, con los puntos de interés extraídos por cada detector.



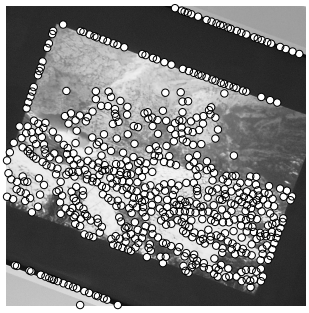
(a) Harris



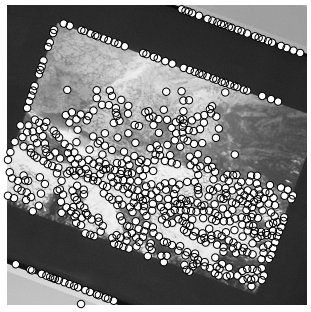
(b) IPGP1



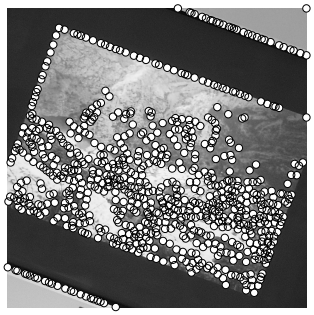
(c) IPGP2



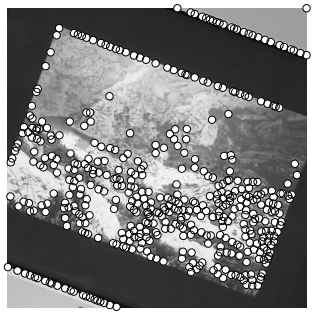
(d) DPIPG1



(e) DPIPG2



(f) DPIPG5



(g) DPIPG6

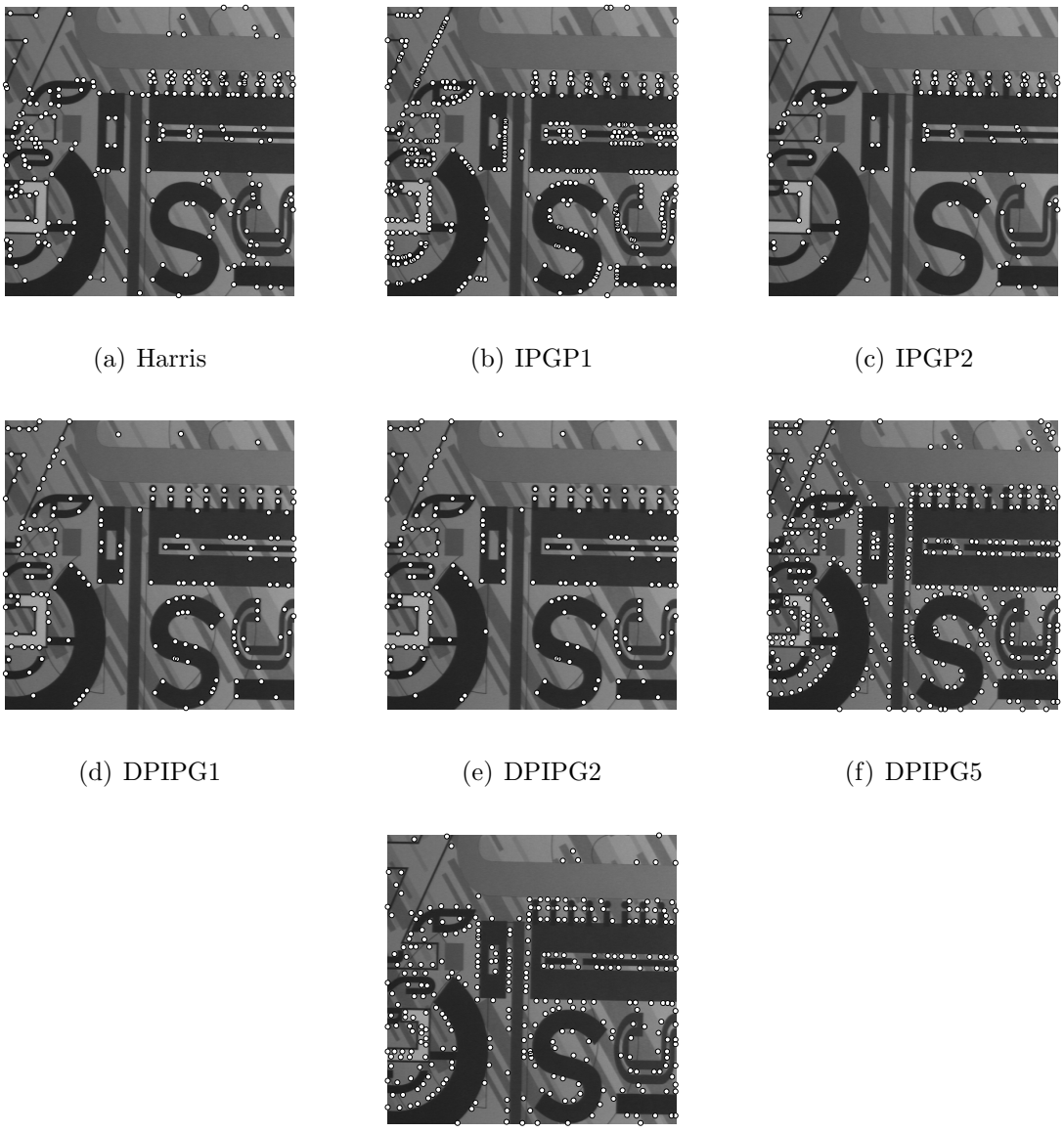
Figura 53: Imagen muestra de Mars, con los puntos de interés extraídos por cada detector.

(a) Harris (b) IPGP1 (c) IPGP2

(d) DPIPG1 (e) DPIPG2 (f) DPIPG5

(g) DPIPG6

Figura 54: Imagen muestra de New York, con los puntos de interés extraídos por cada detector.



(a) Harris (b) IPGP1 (c) IPGP2

(d) DPIPG1 (e) DPIPG2 (f) DPIPG5

(g) DPIPG6

Figura 55: Imagen muestra de Graph, con los puntos de interés extraídos por cada detector.

(a) Harris (b) IPGP1 (c) IPGP2

(d) DPIPG1 (e) DPIPG2 (f) DPIPG5

(g) DPIPG6

Figura 56: Imagen muestra de Mosaic, con los puntos de interés extraídos por cada detector.

(a) Harris

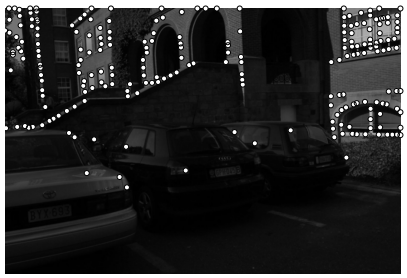
(b) IPGP1

(c) IPGP2

(d) DPIPG1

(e) DPIPG2

(f) DPIPG5



(g) DPIPG6

Figura 57: Imagen muestra de Leuven, con los puntos de interés extraídos por cada detector.

Capítulo VI

Conclusiones y Trabajo Futuro

En este trabajo se presentó un enfoque que aborda el problema de detección de características como un problema de optimización. Dicho enfoque, genera detectores de características de bajo nivel, específicamente, puntos de interés, para aplicaciones de visión por computadora, mediante el aprendizaje de máquina. El aprendizaje se llevo a cabo por medio de la programación genética. La evaluación de aptitud promueve el desempeño optimizado de acuerdo a la tasa de repetibilidad y separación global de los puntos extraídos.

Los resultados experimentales, demuestran que el enfoque planteado es capaz de generar detectores confiables, compactos y competitivos con los detectores más utilizados en la visión por computadora, utilizando funciones aritméticas simples y derivadas de la imagen como conjuntos de funciones y terminales respectivamente.

Del conjunto de funciones utilizadas, podemos concluir que las más apropiadas para la detección de puntos de interés son las derivadas y las funciones de suavizado Gaussiano.

El utilizar más de una imágen de entrenamiento hace al detector menos estable, sin embargo, se obtienen buenos resultados al utilizarlo con imágenes que presentan características similares a las del entrenamiento.

Se presentaron cuatro detectores de puntos de interés generados por nuestro enfoque, DPIPG1, DPIPG2, DPIPG5 y DPIPG6, los cuales mostraron resultados competitivos con los detectores que existen actualmente, incluyendo el detector más popular dentro de la visión por computadora propuesto por Harris y Stephens. Con el detector DPIPG1 se demostró la habilidad de la programación genética para "descubrir" operaciones que el ser humano ya ha descubierto y demostrado su utilidad. Esto, porque no sólo encontró el Laplaciano sino que también eliminó su desventaja principal. No se pretende afirmar que los detectores generados con nuestro enfoque sean los mejores. Ya que ha quedado demostrado, con base en las pruebas realizadas, que no existe un detector que obtenga el mejor desempeño con todas las imágenes, sino que depende de las características de la imágen y del detector. Sin embargo, podemos concluir que sí es posible sintetizar detectores de puntos de interés confiables por medio de técnicas de aprendizaje como lo es la programación genética.

En el presente trabajo, debido a los criterios que la función aptitud evaluó, los detectores encontrados con nuestra metodología son aptos para ciertas tareas de la visión por computadora. Creemos, que utilizando una metodología similar, es posible diseñar algoritmos de aprendizaje que generen detectores de puntos de interés apropiados a diferentes tipos de tareas de la visión por computadora, a diferentes transformaciones de la imagen o a diferentes tipos de imágenes.

El estado actual de nuestro trabajo, deja algunas preguntas abiertas:

- ¿Podría diseñarse una función aptitud que evite el uso de parámetros?
- ¿El conjunto de funciones y terminales es suficiente o necesario?

El trabajo futuro que se podría realizar con el presente trabajo es extenso. Entre lo que encuentra lo siguiente:

- Obtener detectores de puntos de interés que trabajen tanto con imágenes a color como con imágenes en niveles de grises.
- Obtener detectores de puntos de interés que sean optimizados para un cierto tipo de imágenes, como imágenes de exteriores o interiores.
- Lograr que los detectores sean invariantes a otros tipos de transformaciones como cambios de escala.
- Agregar un termino a la función aptitud para medir el contenido de información alrededor de un vecindario, utilizando descriptores, con el objetivo de estimar la utilidad del punto detectado.
- Utilizar optimización multiobjetivo, ya que, como lo demuestran los resultados obtenidos, no existe una solución única y dominante al problema de detección de puntos de interés. La optimización multiobjetivo es deseable, porque de esta forma en lugar de obtener una sola solución, podríamos encontrar un conjunto de detectores óptimos en el sentido Pareto.
- Extender este trabajo para extraer regiones de interés.

Bibliografía

Baker, S. y S. Nayar 1998. "Parametric feature detection". Intenational Journal of Computer Vision, 27(1):27-50 p.

Banzhaf, W., P. Nordin, R. Keller, y F. Francone 1998. "Genetic programming – an introduction; on the automatic evolution of computer programs and its applications". Morgan Kauffman. 470 pp.

Beaudet, P. 1978. "Rotationally invariant image operators.". En: "Proc. Of the International Conference on Pattern Recognition.". Noviembre 7-10. Kyoto, Japón, 579-583 p.

Canny, J. 1983. "Finding edges and lines in images". Reporte técnico , MIT AI Lab. Tech Report, TR-720. 145 pp.

Deriche, R. 1993. "Recursively implementing the gaussian and its derivatives". Reporte técnico , Institud National de Recherche en Informatique et Automatique. Tech Report, TR-1893. 25 pp.

Deriche, R. y T. Blazka 1993. "Recovering and characterizing image features using an efficient model based approach.". En: "IEEE Proc. of the Conference on Computer Vision and Pattern Recognition.". Junio 15-17. New York, EUA, 530-535 p.

Deriche, R. y G. Giraudon 1993. "A computational approach for corner and vertex detection.". International Journal of Computer Vision., 2(10):101-124 p.

Dreschler, L. y H. Nagel 1982. "On the selection of critical points and local curvature extrema of region bounderaries for interframe matching". En: "On the proceedings of the International Conference on Pattern Recognition". Octubre 19-22. Munich, Alemania, 542-544 p.

Ebner, M. 1998. "On the evolution of interest operators using genetic programming". En: "Late Breaking Papers at EuroGP98: The First European Workshop on Genetic Programming". Abril 14-15. Paris, Francia, 6-10 p.

Ebner, M. y A. Zell 1999. "Evolving a task specific image operator". En: "Joint Proceedings of the First European Workshops on Evolutionary Image Analysis (EvoIASP99)". Mayo 28. Göteborg, Suecia, 74-89 p.

Fischler, M. y R. Bolles 1981. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". Comm. Assoc. Comp. Mach., 6(24):381-395 p.

Fleming, P. y C. Fonseca 1993. "Genetic algorithms in control systems engineering". Reporte técnico , The University of Sheffield. Tech Report, TR-470. 254 pp.

Fogel, D. 1994. "An introduction to simulated evolutionary optimization". IEEE Transactions on neural networks, 5(1):3-14 p.

Forrest, S. 1990. "Emergent computation: self organizing, collective, and a cooperative phenomena in natural and artificial computing networks". Emergent Computation, MIT Press, 1:1-11 p.

Forrest, S. 1993. "Genetic algorithms: Principles of natural selection applied to computation". Science, 261(5123):872-878 p.

Forrest, S., B. Javornik, R. Smith, y A. Perelson 1993. "Using genetic algorithms to explore pattern recognition in the immune system". Evolutionary Computation, 1(3):191-211 p.

Förstner, W. 1994. "A framework for low level feature extraction". En: "European Conference on Computer Vision (ECCV94)". Mayo 2-6. Estocolmo, Suecia, 383-394 p.

Förstner, W. y E. Gülch 1987. "A fast operator for detection and precise location of distinct points, corners and centres of circular features.". En: "Proceedings ISPRS

Intercommision Conference on fast procesing of photogrammetric data". Junio 2-4. Interlaken, Suiza, 281-305 p.

Goldberg, D. 1993. "Genetic and evolutionary algorithms come of age". Communications of the ACM, 37(3):113-119 p.

Gonzalez, R. y R. Woods 2002. "Digital image processing". Prentice Hall, Inc., Saddle River, New Jersey, second edition. 793 pp.

Harris, C. J. y M. Stephens 1988. "A combined corner and edge detector". En: "4th Alvey Vision Conference". Agosto 31-Septiembre 2. Manchester, GB, 147-151 p.

Hartley, R. I. y A. Zisserman 2004. "Multiple view geometry in computer vision". Cambridge University Press, second edition. 655 pp.

Hernández, B. y G. Olague 2001. "Un detector sub-pixel paramétrico de esquina múltiples". En: "Memorias del VI Taller Ibereamericano de Reconocimiento de Patrones". Noviembre 11-12. México, D.F., 37-49 p.

Holland, J. 1992. "Adaptation in natural and artificial systems.". MIT Press. 211 pp.

Horaud, R., T. Skordas, y F. Veillon 1990. "Finding geometric and relational structures in an image". Proceedings of the 1st European Conference on Computer Vision, 427:374-384 p. Abril 23-27. Antibes, Francia.

Kitchen, L. y A. Rosenfeld 1982. "Gray level corner detection". Pattern Recognition Letters, 1(2):85-102 p.

Koza, J. R. 1992. "Genetic programing: On the programming of computers by means of natural selection". MIT press, Cambridge, MA. USA. 840 pp.

Lindeberg, T. 1993. "Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction". Journal of Mathematical Imaging and Vision, 3(4):349-376 p.

McGlone, J., E. Mikhail, y J. Bethel 2004. "Manual of photogrammetry". American Society of Photogrammetry and Remote Sensing., quinta edition. 1151 pp.

Medioni, G. y Y. Yasumoto 1987. "Corner detection and curve representation using cubic b-spline". Computer Vision, Graphics and Image Processing, 39:267-278 p.

Moravec, H. 1977. "Towards automatic visual obstacle avoidance". Proceedings of the 5th International Joint Conference on Artificial Intelligence, 2:584 pp.

Olague, G. y B. Hernández 2005. "A new accurate and flexible model based multi-corner detector for measurement and recognition". Pattern Recognition Letters, 26(1):27-41 p.

Rohr, K. 1992. "Recognizing corners by fitting parametric models.". International Journal of Computer Vision, 9(3):213-230 p.

Schmid, C., R. Mohr, y C.Bauckhage 2000. "Evaluation of interest point detectors". International Journal of Computer Vision, 2(37):151-172 p.

Shi, J. y C. Tomasi 1994. "Good features to track". En: sponsored by the IEEE Computer Society, editor, "Conference on Computer Vision and Pattern Recognition (CVPR94)". Junio 21-23. Seattle, EUA, 593-600 p.

Sohn, K., J. Kim, y W. Alexander 1998. "A mean field annealing aproach to robust corner detection". IEEE Transactions on System, Man, and Cybernetics-Part B, 28(1):82-90 p.

Sutherland, I. 1963. "Sketchpad: A man-machine graphical communications system". Reporte técnico , MIT Lincoln Laboratories. Tech Report, TR-296. 92 pp.

Trujillo, L. y G. Olague 2006. "Synthesis of interest point detectors through genetic programming". En: "Genetic and Evolutionary Computation Conference (GECCO)", volume 1. Julio 8-12. Seattle, EUA, 887-894 p.

Tsai, D., H. Hou, y H. Su 1999. "Boundary-base corner detection using eigenvalues of covariance matrices". Pattern Recognition Letters, 20:31-40 p.

van Vliet, L., I. Young, y P. W. Verbeek 1998. "Recursive gaussian derivative filters". En: Jain, A., S. Venkatesh, y B. Lovell, editores, "International Conference on Pattern Recognition (ICPR98)". IEEE Computer Society Press. Agosto 16-20. Brisbane, Australia, 509-514 p.

Wang, H. y M. Brady 1995. "Real-time corner detection algorithms for motion estimation". Image and Vision Computing, 13(9):695-703 p.

Yokono, J. y T. Poggio 2004. "Oriented filters for object recognition: an empirical study". En: "International Conference on Automatic Face and Gesture Recognition (FGR2004)". IEEE. Mayo 17-19. Seúl, Korea, 755-760 p.

Zheng, Z., H. Wang, y E. Teoh 1999. "Analysis of gray level corner detection". Pattern Recognition Letters, 20:149-162 p.