Tesis defendida por

Daniel Miramontes Jaramillo

y aprobada por el siguiente comité

Dr. Vitaly Kober

Director del Comité

Dr. Hugo Homero Hidalgo Silva

Miembro del Comité

Dr. Josué Álvarez Borrego

Miembro del Comité

Dr. José Antonio García Macías

Coordinador del Programa de Posgrado en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales

Director de la Dirección de Estudios de Posgrado

20 de agosto de 2012

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR
DE ENSENADA

Programa de Posgrado en Ciencias
en Ciencias de la Computación

Estudio comparativo de métodos de correspondencia de imágenes

tesis

que para cubrir parcialmente los requisitos necesarios para obtener el grado de Maestro en Ciencias

Presenta:

Daniel Miramontes Jaramillo

Ensenada, Baja California, México,
2012.

Resumen de la tesis de Daniel Miramontes Jaramillo, presentada como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación. Ensenada, Baja California, México. Agosto de 2012.

Estudio comparativo de métodos de correspondencia de imágenes.

Resumen aprobado por:

_____

Dr. Vitaly Kober
Director de Tesis

Una de las características más apreciadas por el ser humano es su capacidad de ver, mediante ésta característica es captada la mayor cantidad de información del mundo que nos rodea, es por ello que actualmente muchas ramas de la ciencia se han enfocado a estudiar este vital recurso en diversas formas, tanto biológicas como tecnológicas.

La correspondencia de imágenes tiene como objetivo el ubicar un objeto de interés en una escena, si se determina que el objeto ha sido encontrado luego se debe determinar la posición en que fue ubicado.

A lo largo del presente trabajo de investigación se estudiaron diversos algoritmos de correspondencia de los métodos basados en área, características e híbridos; y finalmente se desarrolló un nuevo algoritmo híbrido basado en el algoritmo SICHT. Los algoritmos del método basado en características construyen espacios de escala en los cuales no solo se busca la invarianza a escala sino también la identificación de los pixeles más prominentes en la imagen mediante la Diferencia de Gaussianas o el Laplaciano de Gaussianas; de estos pixeles se extraen características de gradiente de su vecindario espacial para construir vectores a los cuales llaman *descriptores*. Por otro lado, los algoritmos del método basado en área recorren la imagen por medio de ventanas deslizantes de forma iterativa extrayendo estructuras como los histogramas, los cuales son una representación del fragmento de la imagen a evaluar, entre otro tipo de características que es posible obtener; finalmente se realiza una comparación entre las estructuras de dos imágenes para determinar qué tan parecidas son.

El algoritmo propuesto en este trabajo es un algoritmo híbrido pues toma algunas ideas de los algoritmos basados en características, haciendo uso de espacios de escala y de los algoritmos basados en área mediante el uso de histogramas de gradiente orientado, mediante un procesamiento iterativo y paralelo.

Palabras Clave: **Correspondencia, método basado en área, método basado en características, histograma de gradiente orientado**

Abstract of the thesis presented by Daniel Miramontes Jaramillo as a partial requirement to obtain the Master of Science degree in Computer Sciences. Ensenada, Baja California, México. August 2012.

Comparative study of image matching methods.

Abstract approved by:

_____

Dr. Vitaly Kober
Thesis Director

One of the most appreciated features for the human is the vision, because a large quantity of information is perceived from the world around us by means of the vision. This is why currently many research areas are focused in investigation of this vital resource using many different ways, from biological to technological ones.

The objective of image matching is to locate an object of interest in a scene. If the object is detected then the next step is to localize the object position on a scene.

In this work area-based, feature-based, and hybrid matching algorithms were studied, and a new algorithm based on the known SICHT algorithm was developed. The feature-based algorithms usually build scale spaces in which the scale invariance as well as the location of the most prominent pixels in the images through the Difference of Gaussians and the Laplacian of Gaussians is intended. From the neighborhood of these pixels gradient features are extracted to construct vectors called descriptors. The area-based method algorithms match the target image in a sliding window iteratively obtaining at each step different parameters such as statistical moments, histograms etc. These parameters are compared to estimate a similitude between the target image and a running window image. The hybrid method algorithms take advantages of the both methods in order to improve the image matching result. However, the use of parameters and features of the two approaches in an efficient way is a difficult task because they often have very different descriptions.

The algorithm proposed in this work is hybrid. It exploits some ideas from the feature-based algorithms using, for instance, scale spaces and from area-based algorithms, using the histograms of oriented gradients. The proposed algorithm is fast because all parameters can be efficiently computed through iterative and parallel calculations at high rate.

Keywords: **Matching, area-based method, features-based method, histogram of oriented gradients**

Dedicatoria

A Mi Niña Pamela,
mi fuente de inspiración,
creatividad y felicidad.

Agradecimientos

Quisiera expresar mi sincero agradecimiento:

Mi familia por su amor, esfuerzo y apoyo incondicional para hacerme quien soy ahora.

A Pamela, por su amor, paciencia, tolerancia y compañía en esta campaña.

Al Dr. Vitaly Kober, por su tiempo, enseñanzas, paciencia y consejos.

A los miembros de mi comité de tesis Dr. Hugo Homero Hidalgo Silva y Dr. Josué Álvarez Borrego por sus consejos y apoyo.

A los miembros del Departamento de Ciencias de la Computación por sus enseñanzas, tiempo y paciencia.

A todos mis compañeros por su amistad y por hacer posible las reuniones de desestrés denominadas “Casita Feliz”.

Al CICESE por darme la oportunidad de realizar mis estudios de posgrado.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo económico mediante la beca No. 242880.

Contenido

Contenido

Lista de Figuras

Lista de Figuras

Lista de Figuras

Lista de Figuras

Lista de Tablas

Capítulo I

Introducción

El ser humano se encuentra en una constante introspección estudiándose a sí mismo tratando de entender su propio cuerpo y tratando de otorgar nuestras increíbles capacidades a objetos de diversos tipos. Nuestra comunicación con el mundo se lleva a cabo por medio de cinco sentidos: tacto, gusto, olfato, audición y vista; los cuales se encuentran bajo investigación en todo el mundo por diversos tipos de científicos con una gran variedad de objetivos, desde investigación con beneficios para la salud hasta investigación para la asistencia humana por medio de desarrollos tecnológicos.

La visión se considera como uno de los sentidos más importantes, pues por medio de los ojos percibimos una gran cantidad de información que será posteriormente procesada por nuestro cerebro con distintos objetivos: ya sea localizar un objeto, reconocer una forma o silueta, encontrar a una persona en una multitud, entre muchísimos otros. Esta capacidad es muy valorada por el ser humano y por tanto, existen diversas disciplinas que la estudian, entre éstas las ciencias computacionales que, con bases y conocimientos derivados de otras ciencias y áreas tecnológicas adquieren y procesan información visual. Mientras el cuerpo humano registra la información visual por medio de los ojos y el cerebro la procesa, un sistema computacional registra la información visual por medio de aparatos fotosensibles como cámaras de fotografía o video y procesa la información en el CPU y/o GPU.

Una gran parte de la investigación en visión por computadora y procesamiento digital de imágenes se hace sobre imágenes digitales adquiridas por medio de cámaras digitales; una imagen digital es una función de dos

dimensiones $f(x, y)$, donde x e y son coordenadas espaciales y el valor de $f(x, y)$ representa el valor de la intensidad en las coordenadas (x, y). El procesamiento digital de imágenes es la rama de las Ciencias de la Computación que se encarga de analizar y procesar imágenes digitales con algún objetivo; dicho objetivo depende de la aplicación. Las aplicaciones son muy variadas, desde el arte hasta el desarrollo de avanzados sistemas de seguridad.

Una de las capacidades básicas de la visión es el poder buscar un objeto específico en el campo de visión y discernir si dicho objeto se encuentra o no, y poder especificar el lugar preciso donde se encontró el objeto en el caso de haberlo localizado. Esta es una capacidad de correspondencia entre el objeto buscado y la imagen visual que se adquiere mediante la visión; en el campo computacional existen muchos métodos para realizar la correspondencia de imágenes digitales; básicamente se dividen en tres categorías: basados en características, basados en área e híbridos. Los métodos basados en características forman vectores de puntos de interés o características en las imágenes y los comparan entre sí buscando correspondencia (Lowe, 1999; Bay *et al.* 2008). Los métodos basados en área hacen uso de los histogramas de las imágenes y niveles de saturación de brillo, entre otras, para buscar la correspondencia de las imágenes (Zalesky y Lukashevich, 2011). Los métodos híbridos toman características de los dos métodos anteriores para buscar correspondencia (Zhou y Dorrer, 2000; Yang et al., 2009).

I.1 Definición del problema

De entre las muchas ramas del Procesamiento Digital de Imágenes, la que compete al presente trabajo de investigación es la de correspondencia de imágenes, o *image matching* en inglés. El problema consiste en ubicar un *modelo* de una imagen; entiéndase por *modelo* un área, un fragmento, una parte o una sección de una imagen o un objeto que se desea ubicar, en otra imagen más

compleja que puede o no contener al modelo; luego, el resultado del procesamiento debe indicar si el modelo fue ubicado y en qué parte de la imagen.

I.2 Objetivos de la investigación

I.2.1 Objetivo general

Realizar un estudio comparativo entre los métodos de correspondencia de imágenes. Desarrollar un algoritmo rápido de correspondencia que tome ventajas de los métodos basados en área y basados en características, y funcione para el procesamiento de imágenes grandes. Este algoritmo debe ser invariante a rotación, traslación, escalamiento, cambios en iluminación y poseer tolerancia a ruido blanco aditivo.

I.2.2 Objetivos específicos

1. Estudiar las bases teóricas de los métodos de correspondencia.
2. Implementar los algoritmos del método basado en área y evaluar su desempeño con un banco de imágenes.
3. Implementar los algoritmos del método basado en características y evaluar su desempeño con un banco de imágenes.
4. Implementar los algoritmos del método híbrido y evaluar su desempeño con un banco de imágenes.
5. Realizar un estudio comparativo de los métodos, obtener conclusiones. Se utilizarán métricas como recursos utilizados, tiempo de ejecución y confiabilidad estadística de correspondencia.
6. Proponer un algoritmo rápido de correspondencia de imágenes invariante a escalamiento, rotación, traslación, cambios en iluminación y poseer

tolerancia a ruido blanco aditivo para el procesamiento de imágenes grandes.

7. Realizar pruebas con el método propuesto sobre el banco de imágenes.
8. Comparar el método propuesto y los métodos existentes previamente evaluados con respecto a invariancia a escalamiento, rotación, traslación, cambios en iluminación y tolerancia a ruido blanco aditivo. Se utilizarán métricas como recursos utilizados, tiempo de ejecución y confiabilidad estadística de correspondencia.

I.3 Limitaciones y suposiciones

En los últimos años han surgido diversos algoritmos de correspondencia, éstos se logran ejecutar en tiempos cortos para imágenes relativamente pequeñas; pero que incrementan enormemente conforme aumenta el tamaño de la imagen. Muchos de los algoritmos que se han propuesto se han desarrollado y probado en equipos computacionales científicos de gran capacidad de procesamiento; esto supone un problema pues la mayoría de las aplicaciones que requieren este tipo de algoritmos deben ejecutarse en equipos comerciales a los que tienen acceso la mayoría de las empresas y usuarios. El presente trabajo pretende realizar un estudio de estos métodos de correspondencia para finalmente desarrollar un nuevo método que sea rápido en imágenes de gran tamaño, manteniendo las propiedades de invariancia a escalamiento, rotación, traslación e iluminación.

I.4 Investigación previa

La correspondencia de imágenes se basa en la detección y localización del *modelo* dentro de una imagen; existen varios enfoques para este proceso, cuando el modelo es parte de una imagen más grande y queremos localizarlo automáticamente, por ejemplo buscar una región específica en una imagen

satelital; cuando el modelo es un objeto dentro de una escena y se debe localizar sin importar el tipo de fondo que contenga; cuando la imagen y el modelo son imágenes del mismo tamaño y se necesita ubicar las imágenes que más se parezcan entre sí. Para realizar este trabajo se cuenta con tres categorías de métodos: basados en características, basados en área e híbridos. En esta sección se presentan algoritmos de estos tres métodos.

I.4.1 Método basado en características

Harris y Stephens (1988) desarrollaron un detector de esquinas, el cual se basa en los eigenvalores del segundo momento de una matriz. Zhang et al. (1995) utilizan este método para obtener puntos característicos en imágenes para luego buscar su correspondencia solo en las regiones concentradas en las esquinas de cada imagen. Con esta idea en mente Schmid y Mohr (1997) utilizan el algoritmo de *Diferencia de Gaussianas (DoG)* para identificar características y crear vectores con éstas que representan a los modelos, utilizando estos vectores para buscar la correspondencia de estos modelos en otras imágenes. Lindeberg (1998) y Mikolajczyk y Schmid (2001) expanden estas ideas extrayendo los vectores de características en un espacio de escala con alta repetitividad. Estos avances sirvieron a investigadores como Lowe (1999) y Bay *et al.* (2008) para crear los algoritmos *Scale Invariant Feature Transform (SIFT)* y *Speeded Up Robust Features (SURF)* respectivamente, los cuales son considerados como *Estado del Arte*; estos algoritmos trabajan sobre un espacio de escala de las imágenes extrayendo sus vectores de características mediante el uso de *DoG* y la convolución de imágenes integrales y otras estructuras como los *Histogramas de Gradiente Orientado (HoG)* (Apéndice B). Actualmente existen varios algoritmos basados en *SIFT* principalmente como GLOH y PCA-SIFT (Ke *et al.*, 2004; y Foo y Sinha, 2007), SIFT-Rank (Toews y Wells, 2009).

I.4.2 Método basado en área

El algoritmo básico de este método se basa en recorrer la imagen de búsqueda mediante una ventana deslizante del mismo tamaño que la imagen modelo y, en cada paso, comparar ambas imágenes mediante una medida de similitud. Esta idea básica se lleva a cabo mediante varias técnicas utilizando diversos factores inherentes a la imagen en sí; los factores de comparación van desde los propios niveles de gris/color de la imagen, histogramas, varios valores estadísticos, entre otros. Cootes *et al.* (2001) desarrollan un algoritmo de modelos de apariencia activa, el cual busca cambios de forma y niveles de gris en la imagen para encontrar correspondencia utilizando para esto un modelo de textura y otro de forma por cada imagen. Lu *et al.* (2006) desarrollan un algoritmo de comparación de histogramas de color inspirándose en el concepto de regiones esféricas de la geometría analítica espacial buscando correspondencia entre varias esferas de color. Varios investigadores hacen uso de plantillas para buscar correspondencia entre dos imágenes, algunas técnicas utilizadas son la correlación cruzada (Tate y Northern, 2008; y Sarvaiya *et al.*, 2009) y el uso de histogramas locales de gradientes orientados como hacen Zalesky y Lukashevich (2011) con su algoritmo *Scale Invariant Compressed Histogram Transform (SICHT)*.

I.4.3 Método híbrido

El método híbrido combina características del método basado en características y del método basado en área. La dificultad de este tipo de métodos consiste en la combinación de los diferentes algoritmos y de su comportamiento específico con el tipo de información que obtienen; mientras los algoritmos basados en área toman en cuenta todo un fragmento de la imagen, los basados en características solo toman en cuenta algunos pixeles y sus vecinos; por lo tanto el combinar estas dos ideas en un solo tipo de procesamiento es una tarea difícil. Zhou y Dorrer (2000), presentan un algoritmo de escala no lineal y libre de

orientación, extrayendo las diferencias geométricas locales con un algoritmo basado en área para después utilizar la correlación entre las dos imágenes con restricciones geométricas para realizar la correspondencia. Yang *et al.* (2009) preprocesan la imagen mediante la ecualización del histograma, posteriormente utilizan un detector de esquinas de Harris para aplicar la covarianza cruzada y buscar la correspondencia entre las imágenes.

I.5 Organización de la tesis

A continuación se detalla la organización del resto del presente trabajo. En el Capítulo II se presentan los fundamentos matemáticos que se utilizan en los distintos procesos de correspondencia. Conceptos importantes como histograma, vecindario, convolución y correlación se detallan para posteriormente introducir otros conceptos como transformaciones geométricas, gradientes y espacio de escala.

En el Capítulo III se presentan los algoritmos de correspondencia que se tomaron en cuenta para el estudio comparativo. Se detallan las propiedades de los algoritmos basados en área y de los algoritmos basados en características, así como un algoritmo híbrido.

En el Capítulo IV se presenta el algoritmo de correspondencia que se desarrolló a lo largo del trabajo de tesis. Se muestran paso a paso los filtros y transformaciones por las que pasa una imagen para obtener su histograma de gradiente orientado y poder buscar su correspondencia con otra imagen.

En el Capítulo V se detallan los experimentos realizados para comparar los diversos algoritmos de correspondencia bajo varios criterios de evaluación como invarianza a rotación, escalamiento, cambios de iluminación y tolerancia a ruido blanco aditivo. Además se presentan los resultados de cada experimento destacando las cualidades de los algoritmos de acuerdo al porcentaje de aciertos, distancia entre el objeto encontrado y la posición real del objeto y tiempo de ejecución promedio por iteración.

Finalmente, en el último apartado se exponen las conclusiones a las que se llegaron en este trabajo de investigación y se comentan algunas líneas de investigación motivadas por las conclusiones obtenidas.

Capítulo II

Fundamentos

II.1 Imagen digital

Una imagen puede definirse como una función bidimensional $f(x,y)$, donde x y y son coordenadas espaciales (plano), y la amplitud de f en cualquier par de coordenadas (x,y) se le llama *intensidad* o *nivel de gris* de la imagen en ese punto. Cuando los valores de x, y y el nivel de gris son finitos, es decir valores discretos, la imagen se denota como *imagen digital*. La imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene una localización particular y valor particular; estos elementos son conocidos como *elementos de imagen*, *pels* o *pixeles*, siendo este último el más frecuentemente utilizado (Gonzalez y Woods, 2002).

Una imagen digital puede ser representada en una multitud de formas que proveen formas distintivas de discriminarla de otras imágenes digitales y a las cuales se les pueden aplicar diversos procesos o transformaciones dependiendo el objetivo para el cual se requiera la imagen. Algunas representaciones y transformaciones comunes en las imágenes digitales se describirán a continuación; se discutirán conceptos como filtros lineales, convolución, gradiente, histograma, deformaciones geométricas e imagen integral.

II.2 Histograma

En estadística un histograma es una representación gráfica que muestra una impresión visual de la distribución de los datos. Es una estimación de la distribución de probabilidad de una variable continua. El histograma consiste en la

tabulación de frecuencias mediante barras rectangulares en intervalos discretos. La altura de cada rectángulo depende de la frecuencia en el intervalo de los datos.

El histograma de una imagen digital con niveles de gris en el rango $[0, L-1]$ es una función discreta $h(r_k) = n_k$, donde r_k es el *k-ésimo* nivel de gris y n_k es el número de pixeles en la imagen que contienen el nivel de gris r_k. Es una práctica común normalizar el histograma al dividir cada uno de sus valores por el número total de pixeles en la imagen. De esta forma, un histograma normalizado está dado por $p(r_k) = n_k/n$ para $k = 0,1, \dots, L-1$; luego $p(r_k)$ es una estimación de probabilidad de ocurrencia del nivel de gris r_k (Gonzalez y Woods, 2002; Solomon y Breckon, 2011; Sonka *et al.*, 2008).

II.3 Vecindario y conectividad de pixeles

Un concepto importante en el procesamiento de imágenes es el de *conectividad.* Varias operaciones en el procesado de imágenes involucran el concepto de *vecindario local* para definir un área de influencia, relevancia o interés local. En la definición del vecindario local se muestra la noción de la conectividad de pixeles, es decir, determinar qué pixeles están conectados entre sí.

Un pixel p con coordenadas (x, y) tiene cuatro vecinos *horizontales* y *verticales* los cuales tienen las siguientes coordenadas

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1), \tag{1}$$

a este conjunto de pixeles se les llama los *4-vecinos* de p, y se denotan por $N_4(p)$. Cada pixel está a una unidad de distancia de (x, y). Los *4-vecinos diagonales* de p tienen coordenadas

$$(x+1, y-1), (x-1, y+1), (x+1, y+1), (x-1, y-1), \tag{2}$$

y se denotan por $N_D(p)$. Estos puntos junto con los *4-vecinos* se les denomina como los *8-vecinos* de p, y se denotan por $N_8(p)$.

Este concepto puede extenderse más allá del plano de la imagen, a imágenes multidimensionales, espacios de escala o a diferentes capas de una

imagen digital; teniendo por ejemplo en un espacio de escalas, 26-vecinos que serían los 8-vecinos del plano de la imagen más 9-vecinos del plano de la escala superior más 9-vecinos del plano de la escala inferior.

La conectividad de pixeles es un concepto fundamental que simplifica la definición de varios conceptos de imágenes digitales, como regiones y límites. Para establecer si dos pixeles están conectados, se debe determinar primero si son vecinos y si sus niveles de intensidad satisfacen un criterio de similitud especificado. Luego, en una imagen binaria con valores 0 y 1, dos pixeles pueden ser vecinos, pero se dice que están conectados solo si tienen el mismo valor.

Sea V el conjunto de valores de niveles de intensidad usados para definir adyacencia. En una imagen binaria, $V = \{1\}$ si se refiere a la adyacencia de pixeles con valor 1. En una imagen en escala de grises se tiene la misma idea, pero el conjunto V contiene más elementos; por ejemplo, la adyacencia de pixeles con un rango posible de niveles de gris de 0 a 255, el conjunto V puede contener cualquier subconjunto de estos 256 valores. Se consideran tres tipos de adyacencia:

Adyacencia-4. Dos pixeles p y q con valores de V son adyacentes si q está en el conjunto $N_4(p)$.

Adyacencia-8. Dos pixeles p y q con valores de V son adyacentes si q está en el conjunto $N_8(p)$.

Adyacencia-m (mixta). Dos pixeles p y q con valores de V son adyacentes:

1. q está en $N_4(p)$, o
2. q está en $N_D(p)$ y el conjunto $N_4(p) \cap N_4(q)$ no contiene pixeles cuyos valores estén en V.

Otro tipo de vecindarios son los vecindarios adaptativos. Sea $V(r)$ el r-ésimo estadístico de orden prioritario y $r(V)$ el rango de valor V. El rango y cualquier estadístico de orden prioritario se pueden calcular a partir del histograma local $\{h(q), q = 0, \dots, Q-1\}$, donde Q son los niveles de gris, sobre un vecindario centrado en un pixel de la siguiente manera

$$r(V) = \sum_{q=0}^{V(r)} h(q). \quad (3)$$

Sean $v = \{v_{n,m}\}$ un vector de pixeles de la imagen sin ruido y $\hat{v} = \{\hat{v}_{n,m}\}$ un vector de pixeles de la imagen resultante. Luego, podemos definir los siguientes vecindarios adaptativos los cuales se ilustran en la Figura 1:

Vecindario EV. Subconjunto de pixeles cuyos valores se desvían del valor del pixel central por una cantidad predeterminada. Ecuación recuperada de Kober *et al.* (2001, 972 p.).

$$EV(v_{k,l}) = \{v_{n,m} : v_{k,l} - \varepsilon_v \leq v_{n,m} \leq v_{k,l} + \varepsilon_v\}. \quad (4)$$

Vecindario KNV. Subconjunto de K número de pixeles cuyos valores son los más cercanos al pixel central. Ecuación recuperada de Kober *et al.* (2001, 972 p.).

$$KNV(v_{k,l}) = \left\{V(r) : \sum_{r=p}^{p+K-1} |v_{k,l} - V(r)| = MIN_p\right\}. \quad (5)$$

Vecindario ER. Subconjunto de pixeles cuyos rangos se desvían del rango del pixel central por una cantidad predeterminada. Ecuación recuperada de Kober *et al.* (2001, 972 p.).

$$ER(v_{k,l}) = \{v_{n,m} : r(v_{k,l}) - \varepsilon_r \leq v_{n,m} \leq r(v_{k,l}) + \varepsilon_r\}. \quad (6)$$

II.4 Transformada de Fourier

La transformada de Fourier $\mathcal{F}$ transforma una función $f(t)$ a la representación del dominio de frecuencia, $\mathcal{F}\{f(t)\} = \mathrm{F}(\omega)$. La función compleja F se denomina espectro de frecuencia, en el cual es sencillo visualizar proporciones relativas de diferentes frecuencias. La transformada de Fourier continua $\mathcal{F}$ está dada por

$$\mathcal{F}\{f(t)\} = \mathrm{F}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\omega t}dt. \quad (7)$$

Y la transformada inversa de Fourier $\mathcal{F}^{-1}$ está dada por

$$\mathcal{F}^{-1}\{\mathrm{F}(\omega)\} = f(t) = \int_{-\infty}^{\infty} \mathrm{F}(\omega)e^{2\pi i\omega t}d\omega. \quad (8)$$

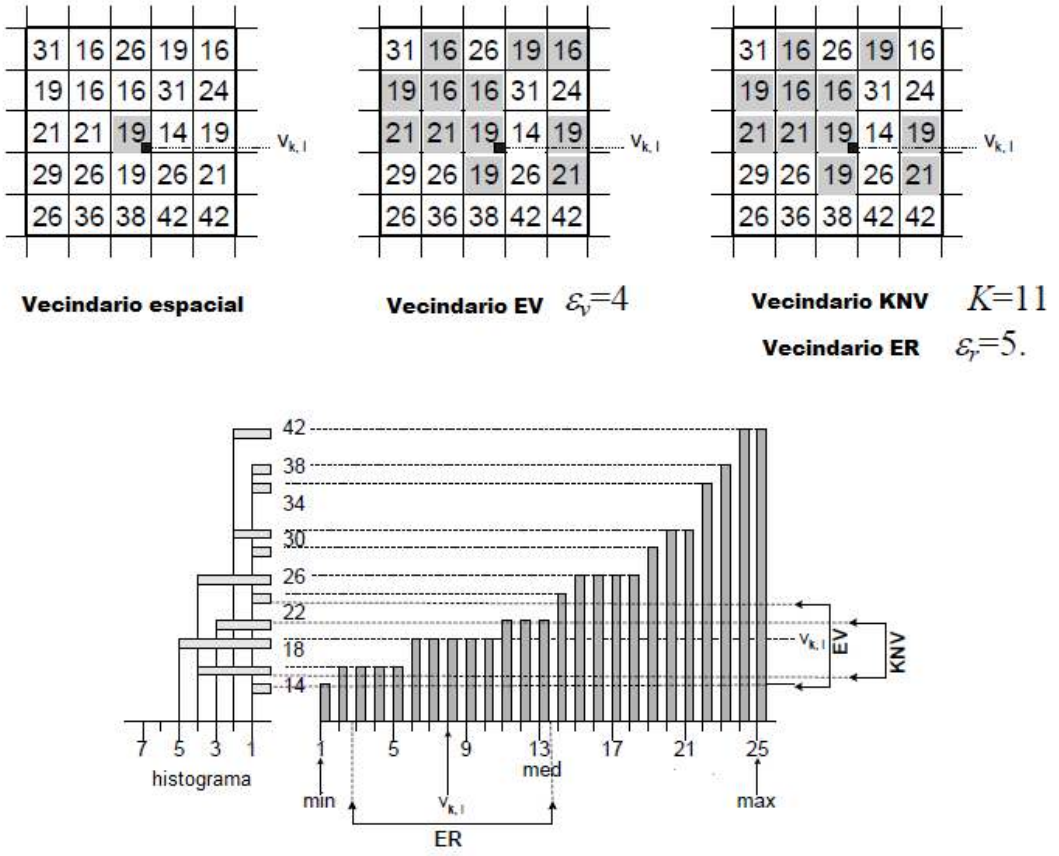


Figura 1: Vecindarios adaptativos. En la parte superior se observa la formación del vecindario en el vecindario espacial. En la parte inferior se muestran los vecindarios en el histograma. Imagen recuperada de Kober et al., 2001. 973 p.

La transformada de Fourier siempre existe para señales digitales, incluyendo imágenes, pues tienen límites y un número finito de discontinuidades.

La transformada de Fourier es fácilmente generalizada a 2D, luego la transformada de Fourier y su inversa para una imagen continua están definidas respectivamente como

$$\mathrm{F}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(xu+yv)}dxdy, \tag{9}$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{F}(u,v)e^{2\pi i(xu+yv)}dudv. \tag{10}$$

La transformada de Fourier puede usarse también en el caso de señales e imágenes discretas, en estas circunstancias las integrales se cambian por sumas en las ecuaciones respectivas. Luego, en el caso de imágenes digitales la transformada de Fourier y su inversa se definen respectivamente por (Sonka *et al.*, 2008)

$$\mathrm{F}(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-2\pi i\left({}^{mu}/_{M} + {}^{nv}/_{N}\right)}, \tag{11}$$

$$f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \mathrm{F}(u,v)e^{2\pi i\left({}^{mu}/_{M} + {}^{nv}/_{N}\right)}. \tag{12}$$

II.5 Convolución

Un impulso ideal es una señal de entrada importante; el impulso ideal en el plano de imagen se define mediante el uso de la *distribución de* Dirac $\delta(x,y)$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x,y)\,dx\,dy = 1, \tag{13}$$

y $\delta(x,y) = 0$ para todo $(x,y) \neq 0$.

La *propiedad de cernimiento* de la distribución de Dirac determina el valor de la función $f(x,y)$ en el punto (λ,μ)

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x-\lambda,y-\mu)\,dx\,dy = f(\lambda,\mu). \tag{14}$$

La *ecuación de cernimiento* (14) es usada para describir el proceso de muestreo de una función de imagen continua $f(x, y)$. Se debe expresar la función de imagen como una combinación lineal de pulsos de Dirac en los puntos a, b que cubren todo el plano de la imagen; las muestras son pesadas por la función de imagen $f(x, y)$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(a,b)\ \delta(a-x, b-y)\ da\ db = f(x,y). \tag{15}$$

La *convolución* es una operación importante en el análisis de imágenes. La convolución es una integral que expresa la cantidad de *empalme* de una función $f(t)$ mientras es desplazada sobre otra función $h(t)$. Una convolución en una dimensión $f * h$ ($*$ denota la operación de convolución) de las funciones f, h sobre un rango finito $[0, t]$ está dada por

$$(f * h)(t) \equiv \int_{0}^{t} f(t)\ h(t-\tau)\ d\tau. \tag{16}$$

Sean f, g y h funciones y a un escalar constante, la convolución satisface las siguientes propiedades:

Propiedad conmutativa

$$f * h = h * f. \tag{17}$$

Propiedad asociativa

$$f * (g * h) = (f * g) * h. \tag{18}$$

Propiedad distributiva

$$f * (g + h) = (f * g) + (f * h). \tag{19}$$

Multiplicación por un escalar

$$a(f * g) = (af) * g = f * (ag). \tag{20}$$

La convolución puede ser generalizada a mayores dimensiones. La convolución de funciones en 2D f y g es denotada por $f * g$, y se define por la integral

$$\begin{aligned}(f * h)(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(a,b)\, h(x-a, y-b)da\, db \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-a, y-b)\, h(a,b)da\, db \\ &= (h * f)(x,y).\end{aligned} \tag{21}$$

En el análisis de imágenes digitales, la *convolución discreta* se expresa mediante sumas en lugar de integrales. Una imagen digital tiene un dominio limitado en el plano de la imagen. Sin embargo el dominio limitado no previene el uso de la convolución, ya que fuera de los límites de la imagen se considera como cero.

Las operaciones lineales calculan el valor resultante en los pixeles de la imagen de salida $g(i,j)$ como una combinación lineal de las intensidades de la imagen en un vecindario local N del pixel $f(i,j)$ de la imagen de entrada. La contribución de los pixeles del vecindario N se pesan por los coeficientes h

$$f(x,y) * h(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(x-m, y-n)\, h(m,n). \tag{22}$$

La ecuación (22) es equivalente a la convolución discreta con un kernel h (Sonka *et al.*, 2008).

Teorema de convolución

El teorema de convolución establece que la relación existente entre dos funciones y sus transformadas de Fourier está dada por

$$\mathcal{F}\{f(x,y) * h(x,y)\} = F(u,v)H(u,v), \tag{23}$$

$$\mathcal{F}\{f(x,y)h(x,y)\} = F(u,v) * H(u,v), \tag{24}$$

donde $F(u,v)$ y $H(u,v)$ son las transformadas de Fourier de $f(x,y)$ y $h(x,y)$ (Gonzalez y Woods, 2002).

II.6 Correlación

La correlación de dos funciones $f(x,y)$ y $h(x,y)$ está definida como

$$(f \circ h)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^*(a,b)\, h(x+a, y+b)\, da\, db \tag{25}$$

donde $\circ$ denota la operación de correlación. Al igual que la convolución, la correlación se realiza mediante sumas en lugar de integrales en el caso discreto de la siguiente forma

$$f(x,y) \circ h(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f^*(x+m, y+n)\, h(m,n), \tag{26}$$

donde f^* denota el complejo conjugado de f. Como las imágenes digitales son funciones reales, luego $f^* = f$. La función de correlación tiene la misma forma que la función de convolución con la excepción de complejo conjugado y el cambio de signos en uno de los términos de la operación.

La correlación se usa principalmente para establecer un factor de similitud entre dos funciones, de tal forma que dos funciones están correlacionadas si al conocer algo de una función se obtiene información de la otra función.

Normalmente se utilizan dos términos cuando se usa la correlación, *correlación cruzada* se utiliza para definir que las funciones o imágenes que se están correlacionando son diferentes y *autocorrelación* se utiliza para denotar la correlación de una función o imagen con ella misma.

Teorema de correlación

El teorema de correlación establece que la relación existente entre dos funciones y sus transformadas de Fourier está dada por

$$\mathcal{F}\{f(x,y) \circ h(x,y)\} = F^*(u,v)H(u,v), \tag{27}$$

$$\mathcal{F}\{f^*(x,y)h(x,y)\} = F(u,v) \circ H(u,v), \quad (28)$$

donde $F(u,v)$ y $H(u,v)$ son las transformadas de Fourier de $f(x,y)$ y $h(x,y)$.

Teorema de autocorrelación

Derivado del teorema de correlación se puede ver que si las dos funciones son idénticas

$$\mathcal{F}\{f(x,y) \circ f(x,y)\} = |F(u,v)|^2, \quad (29)$$

$$\mathcal{F}\{|f(x,y)|^2\} = F(u,v) \circ F(u,v), \quad (30)$$

obtenemos la energía total de la función f. Al realizar esta operación en el espacio de frecuencia obtenemos lo que se llama *espectro de frecuencia* que al volver al espacio de tiempo mediante la transformada inversa de Fourier obtenemos la secuencia de autocorrelación de la señal o el mapa de autocorrelación de la imagen (Gonzalez y Woods, 2002).

II.7 Métricas de distancia

Las métricas de distancia proveen una forma de calcular la similitud entre dos imágenes. Mediante sencillas operaciones sobre la intensidad de los pixeles en las imágenes a comparar indican cuán similar es una imagen o un fragmento de una imagen con otra imagen. Existen varias métricas de distancia, de las cuales se mencionan algunas a continuación.

Distancia Euclidiana

La distancia Euclidiana es, por mucho, la métrica de distancia más comúnmente usada. Recuperada de Sonka *et al.* (2008, 17 p.) está dada por

$$D(I_1, I_2) = \sqrt{\sum_{(i,j)\in V} \big(I_1(i,j) - I_2(i,j)\big)^2}, \quad (31)$$

donde I_1 y I_2 son las imágenes a comparar, V es la área de imágenes a comparar, $I(i,j)$ es la intensidad del pixel en la imagen en las coordenadas (i,j).

Suma de diferencias absolutas (Chambon, 2003, 387 p.)

$$D(I_1, I_2) = \sum_{(i,j)\in V} |I_1(i,j) - I_2(i,j)|. \quad (32)$$

Suma de diferencias cuadradas (Chambon, 2003, 387 p.)

$$D(I_1, I_2) = \sum_{(i,j)\in V} (I_1(i,j) - I_2(i,j))^2. \quad (33)$$

Error máximo (Chambon, 2003, 387 p.)

$$D(I_1, I_2) = \max_{(i,j)\in V} (|I_1(i,j) - I_2(i,j)|). \quad (34)$$

II.8 Transformaciones geométricas

Las transformaciones geométricas son comunes en imágenes de computadora, y son frecuentemente usadas en el análisis de imágenes. Éstas permiten la eliminación de distorsiones geométricas que ocurren cuando la imagen es capturada. Si se intenta corresponder dos imágenes diferentes del mismo objeto, puede ser necesario utilizar una transformación geométrica. En el presente trabajo solo se consideran transformaciones geométricas en 2D, pues es suficiente para las imágenes digitales (Sonka *et al.*, 2008).

Al estudiar las transformaciones geométricas, debemos tener presente primeramente el concepto de *forma*. En un concepto simple, una *forma* implica la existencia de *bordes, límites o fronteras*. De forma general una *forma* es toda la información geométrica que se mantiene después de que los efectos de localización, escalamiento y rotación han sido filtrados del objeto. Luego, una *forma* es un conjunto ordenado de coordenadas, de modo que la *forma* se describe matemáticamente localizando un número finito de puntos N a lo largo de los bordes del objeto concatenándolos para obtener un vector de forma.

Existen varios tipos de transformaciones geométricas, aquí se presentan las más comunes, *traslación, rotación* y *escalamiento*. Trasladar, rotar o escalar un objeto son operaciones que cambian las coordenadas del vector de forma, pero no cambian la forma en sí, es decir, la forma del objeto es algo que está definida por sus bordes pero es invariante a traslación, rotación y escalamiento. Luego, la forma puede ser representada como un arreglo ordenado de N coordenadas Euclidianas $\{(x_1, y_1) \dots (x_N, y_N)\}$ escritas como columnas en una matriz S

$$S = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_N \\ y_1 & y_2 & y_3 & \dots & y_N \end{bmatrix}. \tag{35}$$

La ventaja de esta representación de la matriz S es que las transformaciones lineales de la forma pueden realizarse mediante una simple multiplicación. En general, el resultado de aplicar una matriz T de transformación de 2×2 a S produce un nuevo conjunto de coordenadas de forma S' dadas por

$$S' = TS, \tag{36}$$

con los pares coordenados transformándose como

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} \quad o \quad x' = Tx = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{37}$$

con los parámetros a_{ij} controlando la forma de la transformación.

Las operaciones de escalamiento en dos dimensiones y rotación sobre el origen sobre un ángulo θ pueden expresarse mediante matrices de 2×2 en la forma descrita en la ecuación (37). Explícitamente la matriz de escalamiento está dada por

$$T_E = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}, \tag{38}$$

donde α es un factor de escala. Y la matriz de rotación está dada por

$$T_R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{39}$$

La operación de traslación sin embargo, se lleva a cabo mediante una operación de suma de vectores como se muestra a continuación

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} \qquad x' = x + d, \tag{40}$$

(Solomon y Brekon, 2011).

II.9 Gradiente

El gradiente ∇f de un campo escalar f es un campo vectorial que indica en cada punto del campo escalar la dirección de máximo incremento del mismo. El gradiente se define como el campo vectorial cuyas funciones coordenadas son las derivadas parciales del campo escalar, esto es

$$\nabla f(r) = \left(\frac{\partial f(r)}{\partial x_1}, \dots, \frac{\partial f(r)}{\partial x_n}\right). \tag{41}$$

El gradiente permite calcular fácilmente las derivadas direccionales. Definiendo en primer lugar la derivada direccional según un vector

$$\frac{\partial \phi}{\partial n} \equiv \lim_{\epsilon \to 0} \frac{\phi(r - \epsilon \hat{n}) - \phi(r)}{\epsilon}. \tag{42}$$

Geométricamente el gradiente es un vector que se encuentra normal a una superficie o curva en el espacio en el cual se le está estudiando en un punto cualquiera.

Un gradiente de imagen es el cambio direccional en la intensidad o color en la imagen. El gradiente de imágenes puede usarse para extraer información de las imágenes. Matemáticamente, el gradiente de una función de dos variables es un vector en 2D en cada pixel de la imagen, con los componentes dados por las derivadas en las direcciones verticales y horizontales. En cada pixel, el vector gradiente apunta en la dirección del mayor incremento de intensidad posible y la longitud del vector gradiente corresponde con el ritmo de cambio en esa dirección. Luego, el gradiente de imagen está dado por

$$\nabla f(x,y) = G = \begin{bmatrix} g_x & g_y \end{bmatrix} = \left[\frac{\partial}{\partial x} f(x,y) \quad \frac{\partial}{\partial y} f(x,y)\right]. \tag{43}$$

El vector gradiente apunta en la dirección de variación máxima de f en el punto (x, y) por unidad de distancia, con magnitud y orientación dadas por las siguientes ecuaciones respectivamente

$$|\nabla f| = \sqrt{g_x^2 + g_y^2}, \tag{44}$$

$$\phi(\nabla f) = arctan\frac{g_y}{g_x}. \tag{45}$$

Debido al costo computacional, para el cálculo de las derivadas se suelen utilizar las diferencias de primer orden entre dos pixeles adyacentes como se muestra a continuación

$$g_x = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}, \tag{46}$$

$$g_y = \frac{f(y + \Delta y) - f(y - \Delta y)}{2\Delta y}. \tag{47}$$

Existen varios operadores de gradiente que examinan pequeños vecindarios locales que son de hecho operadores de convolución, y pueden ser expresados mediante kernels de convolución. Los operadores que pueden detectar la dirección del gradiente están representados por un conjunto de kernels, cada uno corresponde a cierta dirección. Debido al alto costo computacional que involucra el obtener el gradiente de una imagen, estos operadores son una aproximación a la derivada de la imagen. A continuación se presentan los operadores de gradiente más conocidos, sin embargo hay muchos otros que se pueden utilizar como se muestra en el siguiente capítulo (Sonka *et al.*, 2008).

Operador de Roberts

El operador de Roberts es uno de los más antiguos. Es fácil de procesar pues solamente utiliza kernels de 2×2 en el vecindario del pixel. Sus kernels de convolución como menciona Sonka *et al.* (2008, 135 p.) son

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{48}$$

La principal desventaja de este operador es su sensibilidad al ruido, pues utiliza muy pocos pixeles para aproximar el gradiente.

Operador Prewitt

El operador de Prewitt, al igual que el operador de Sobel, Kirsch, Robinson y otros más, aproxima la primera derivada. El gradiente se estima en ocho posibles direcciones, para kernels de convolución de 3×3. Es posible tener kernels más grandes. Normalmente se utilizan solo dos kernels, uno para la dirección en x y otro para la dirección en y, sin embargo los demás pueden calcularse al rotar cualquiera de estos kernels como se ve en Sonka *et al.* (2008, 136 p.)

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}. \tag{49}$$

Operador Sobel (Sonka *et al.,* 2008, 136 p.)

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}. \tag{50}$$

Operador Robinson (Sonka *et al.,* 2008, 137 p.)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}. \tag{51}$$

Operador Kirsch (Sonka *et al.,* 2008, 138 p.)

$$\begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}, \quad \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix}. \tag{52}$$

II.10 Escala y espacio de escala

Muchas técnicas de procesado de imágenes se aplican localmente, teóricamente a nivel de pixel. El problema esencial en estos procesos es la *escala*. Por ejemplo, en la detección de esquinas, éstas corresponden al gradiente de la

función de imagen, el cual es procesado mediante la diferencia entre pixeles en un vecindario, el tamaño del vecindario depende del tamaño del objeto bajo investigación. Éste se expresa a diferentes resoluciones y se forma un modelo por cada resolución, luego, el comportamiento del modelo se conoce estudiando las diferentes resoluciones. Esta metodología permite deducir información del objeto que no es posible ver en las diferentes resoluciones individualmente.

Las diferentes resoluciones pueden ser interpretadas como escalas en el dominio de las imágenes digitales. La idea de *escala* es fundamental en el algoritmo de detección de esquinas de Marr, donde se obtienen diferentes escalas mediante filtros Gaussianos de diferentes tamaños. El objetivo no es solamente eliminar el ruido en la imagen sino también separar eventos en las diferentes escalas provenientes de diversos procesos físicos (Marr, 1982).

Witkin (1983), propone el filtraje por *espacio de escalas*, en el cual trata de describir señales cualitativamente con respecto a su escala. El problema fue formulado para señales $f(x)$ en una dimensión, pero puede ser fácilmente generalizado para funciones de dos dimensiones como imágenes. La imagen original en una dimensión es suavizada por convolución con una Gaussiana de una dimensión

$$G(x,\rho) = e^{-x^2/2\rho^2}. \tag{53}$$

Si la desviación estándar varía lentamente, la función:

$$F(x,\rho) = f(x) * G(x,\rho), \tag{54}$$

representa una superficie sobre el plano (x,ρ) llamada *espacio de escala*. Los puntos de inflexión de la curva $F(x,\rho)$ para distintos valores ρ_0

$$\frac{\partial^2 F(x,\rho_0)}{\partial x^2} = 0 \quad y \quad \frac{\partial^3 F(x,\rho_0)}{\partial x^3} \neq 0, \tag{55}$$

describen la curva $f(x)$ cualitativamente. La posición de los puntos de inflexión se puede dibujar como un conjunto de curvas en coordenadas (x,ρ).

La información cualitativa contenida en el espacio de escala de la imagen puede ser transformada a un simple árbol de intervalo que expresa la estructura

de la señal $f(x)$ en todas las escalas observadas. El árbol de intervalo se construye desde la raíz que corresponde a la escala más grande $(\rho_{\max})$, y luego se busca en el espacio de escala de la imagen en dirección de ρ decreciente. (Sonka *et al.*, 2008).

El filtro Gaussiano no es el único mediante el cual puede construirse el espacio de escala, sin embargo, Lindeberg (1994) demostró que el filtro Gaussiano y sus derivadas son los únicos filtros de suavizado posibles para el análisis de espacios de escala.

II.11 Imagen integral

La imagen integral es una representación intermedia de la imagen con la cual es posible procesar de forma rápida características rectangulares. La imagen integral en la posición (x, y) contiene la suma de los pixeles a la izquierda y sobre el pixel (x, y) incluyéndolo, como se muestra en Bay *et al.* (2008, 348 p.)

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \tag{56}$$

Usando la imagen integral, cualquier suma rectangular puede procesarse mediante la suma de cuatro puntos en la imagen integral como se muestra en la Figura 2. De tal forma que es una manera más efectiva de extraer sumas en áreas específicas de la imagen de una manera eficiente (Viola y Jones, 2001).

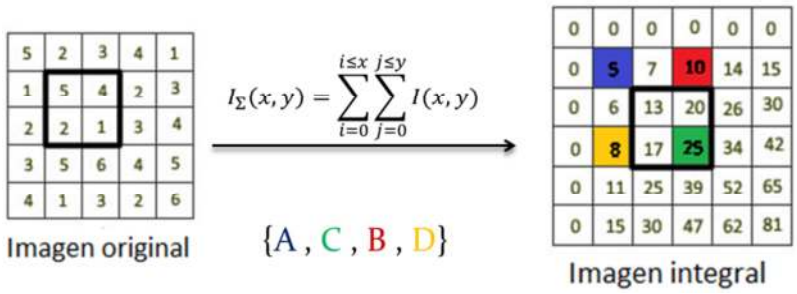


Figura 2: La suma de pixeles dentro del rectángulo marcado puede procesarse mediante la suma de cuatro puntos en la imagen integral A + C - (B + D).

Capítulo III

Algoritmos de métodos de correspondencia

III.1 Introducción

En este capítulo se presentan cinco algoritmos de los diferentes métodos de correspondencia, donde doy una concisa explicación de cómo funciona cada uno. Primeramente se introducen los algoritmos del método basado en características, los cuales funcionan de forma muy similar, extrayendo los pixeles más relevantes de la imagen sobre un espacio de escala para luego construir descriptores con información del vecindario de estos pixeles. Enseguida se detallan los algoritmos del método basado en área, los cuales escanean la imagen a través de una ventana deslizante realizando comparaciones mediante los valores de intensidad de cada pixel o construyendo histogramas. Finalmente se describe un algoritmo híbrido basado en el algoritmo de área MSER *(Maximally Stable Extremal Regions)* (Matas *et al.*, 2004) que extrae áreas de pixeles conectados y el algoritmo SIFT para construir descriptores en base al centro de las áreas conectadas.

III.2 Algoritmos del método basado en características

Esta sección presenta los dos algoritmos más ampliamente estudiados en la literatura sobre correspondencia por características. Estos algoritmos tienen la premisa de extraer vectores de características con un alto factor de repetitividad, es decir, que puedan ser extraídos en cada ocasión aún y cuando las características de la imagen varíen. Los algoritmos estudiados que se presentan a continuación son *SIFT* propuesto por Lowe (1999) y *SURF* propuesto por Bay *et*

al. (2008). Es importante mencionar que este par de algoritmos están patentados, de tal manera que está permitido su uso para cuestiones académicas pero no para aplicaciones comerciales.

III.2.1 SIFT

SIFT es un algoritmo que extrae características invariantes distintivas de imágenes que pueden ser usadas para realizar correspondencia confiable entre objetos o escenas en imágenes. Las características extraídas de las imágenes son invariantes a escala y rotación, y proveen correspondencia robusta aún en condiciones de ruido y ligeros cambios de iluminación (Lowe, 2004).

A continuación se describen la serie de pasos que sigue SIFT para extraer los descriptores de características que pueden ser usados para realizar la búsqueda de correspondencia entre dos o más imágenes.

Detección de extremas en el espacio de escala

Esta primera etapa del algoritmo trata de detectar localizaciones en la imagen que sean repetibles e invariantes a cambios de escala y bajo diferentes vistas del objeto; para esto se hace la búsqueda a través de un espacio de escala utilizando un kernel Gaussiano. El espacio de escala de la imagen se define como una función $L(x, y, \rho)$ que se produce por la convolución de un kernel Gaussiano de escala variable $G(x, y, \rho)$ y la imagen de entrada $I(x, y)$

$$L(x, y, \rho) = G(x, y, \rho) * I(x, y), \tag{57}$$

donde $*$ es la operación de convolución en x y y, y el kernel Gaussiano está definido por

$$G(x, y, \rho) = \frac{1}{2\pi\rho^2} e^{-(x^2+y^2)/2\rho^2}. \tag{58}$$

El espacio de escala en SIFT difiere del concepto en una forma particular, ya que cuenta con un conjunto de escalas y un conjunto de octavas por escala. En este algoritmo la escala es la convolución de la imagen con el kernel Gaussiano

como se mostró anteriormente, y una octava puede definirse como un muestreo de la imagen a la mitad del tamaño de la octava anterior, este concepto puede apreciarse claramente en la Figura 3.

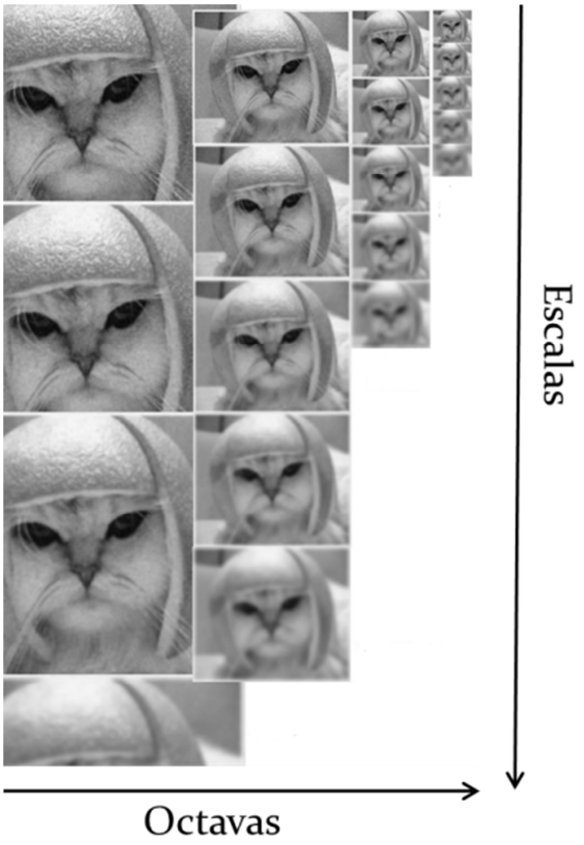


Figura 3: Espacio de escalas. Imagen recuperada de http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/.

Para la localización de características estables en el espacio de escala se realiza una Diferencia de Gaussianas (DoG) entre las escalas de la imagen, esta operación es una aproximación del Laplaciano de Gaussianas (LoG) (Apéndice A) como se muestra en la Figura 4; para después extraer los pixeles extremos, máximos y mínimos, de entre las imágenes resultantes de la DoG, esta operación se realiza al comparar cada pixel con sus 26 vecinos como se muestra en la

Figura 5, cabe destacar que esta operación no se realiza en las imágenes en los extremos de la DoG.

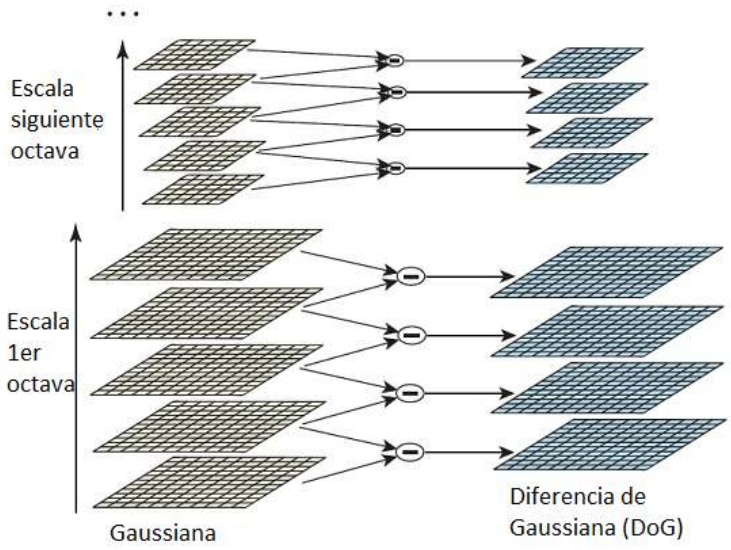


Figura 4: Diferencia de Gaussianas (DoG) sobre las escalas en cada octava de la imagen. Imagen recuperada de Lowe, 2004. 96 p.

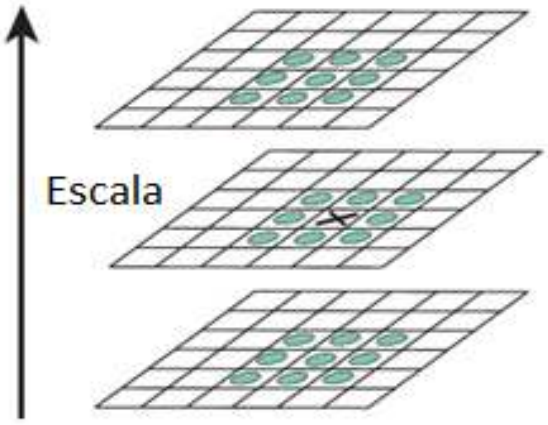


Figura 5: Localización de máximos y mínimos sobre el resultado de la DoG. Comparación del pixel X con sus 26 vecinos. Imagen recuperada de Lowe, 2004. 97 p.

Asignación de orientaciones

Por cada máxima/mínima extraída, a las cuales de aquí en adelante se les llamarán características, se extraen las magnitudes y orientaciones de todos los pixeles a su alrededor con un radio de 1.5ρ dependiendo de su escala mediante las ecuaciones (44) y (45) sobre las imágenes generadas por la DoG.

Con estos datos se forma un histograma de gradiente orientado (HoG) (Apéndice B) de 36 compartimientos (10° por compartimiento), donde cada instancia que entra en un compartimiento del histograma es pesado por su magnitud correspondiente. Del histograma resultante la característica toma la orientación del compartimiento que resulte ser el máximo del histograma, y se crean características nuevas con la orientación de los compartimientos que cuenten con al menos el 80% de la altura del compartimiento máximo como se muestra en la Figura 6, con lo cual se tienen características iguales con orientaciones diferentes.

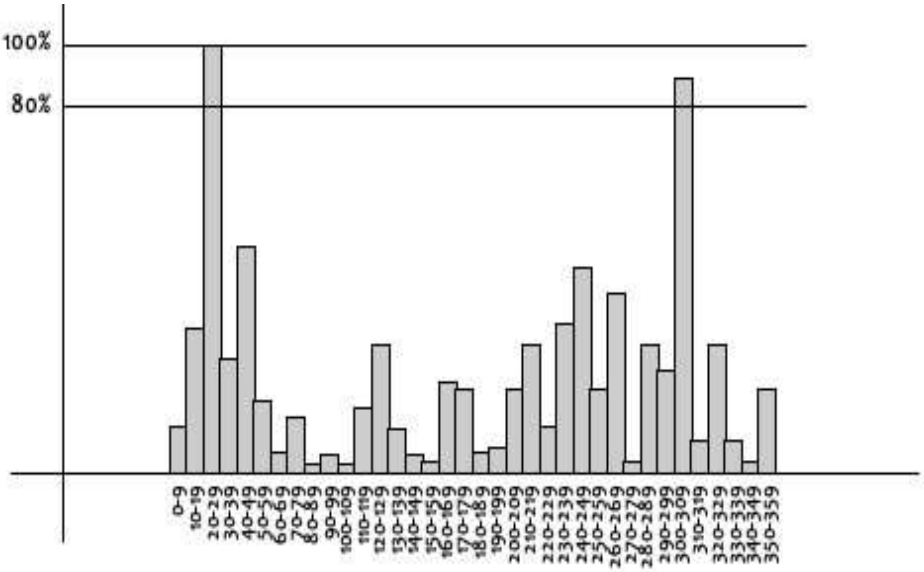


Figura 6: HoG en el cual se asigna la orientación de 20°-29° y se crea una nueva característica con orientación 300°-309°. Imagen recuperada de http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/.

Descriptor de características

Para formar los descriptores de características se sigue un procedimiento similar a la asignación de orientaciones. Por cada característica se obtienen las magnitudes y orientaciones en un vecindario de 16x16 alrededor de la característica, luego se formarán 16 HoGs de 8 compartimientos de la misma forma que se realizó anteriormente formando un vector de 128 dimensiones como se muestra en la Figura 7, a este vector se le llama *descriptor*.

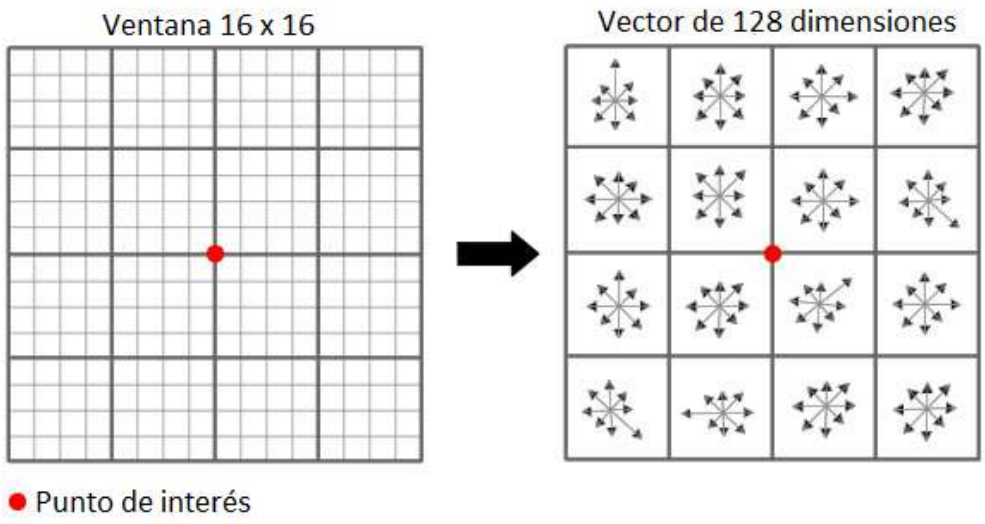


Figura 7: Descriptor de características SIFT. Imagen recuperada de http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/.

Correspondencia

Una imagen de 500x500 pixeles puede generar 2000 descriptores aproximadamente, estos descriptores se comparan uno por uno contra los descriptores de otras imágenes donde se busca algún objeto. Esta búsqueda de

correspondencia entre vectores se realiza mediante el algoritmo de vecinos cercanos. Sin embargo, los resultados pueden dar a lugar a una gran cantidad de características correspondientes, para filtrar estas correspondencias se evalúa la cercanía entre los vectores de características, sobreviviendo solo aquellas correspondencias en las que sus dos vecinos más cercanos no sobrepasen de un umbral establecido; Lowe establece este umbral en 0.6.

III.2.2 SURF

SURF es un algoritmo invariante a escala y rotación que se basa en el uso de imágenes integrales para las operaciones de convolución. Este algoritmo tiene sus bases en SIFT por lo que muchas de las operaciones que realiza son, si no idénticas, muy parecidas al algoritmo de Lowe.

Sin embargo, sacrificando en parte la precisión de SIFT, SURF opera mucho más rápido gracias al uso de imágenes integrales, a su innovativa forma de crear el espacio de escala y a su descriptor reducido de 64 dimensiones. A continuación se detalla el procedimiento que sigue SURF.

Localización de características en el espacio de escala

Para crear el espacio de escala, en lugar de filtrar la imagen una y otra vez con kernels Gaussianos y muestreando la imagen a la mitad de su tamaño original para cada octava, SURF opta por otra medida que es computacionalmente más eficiente. SURF crea una pirámide de kernels que aproximan al Laplaciano de Gaussianas incrementando cada vez el tamaño del kernel, de esta forma la convolución se lleva a cabo entre la imagen y la pirámide de kernels dando como salida el espacio de escalas. La Figura 8 muestra este concepto y los kernels resultantes de esta operación. La escala ρ se calcula mediante la siguiente fórmula

$$\rho_{aprox.} = Tamaño\ del\ kernel\ \frac{\rho\ base}{Tamaño\ del\ kernel\ base}. \tag{59}$$

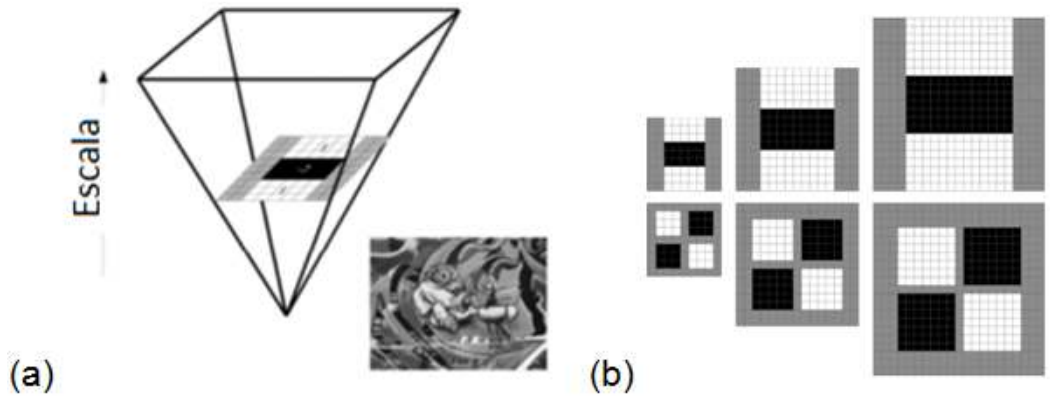


Figura 8: Espacio de escala de SURF. (a) Generación de la pirámide de kernels Gaussianos. (b) Conjunto de kernels Gaussianos que se convolucionarán con la imagen. Imagen recuperada de Bay et al., 2008. 349 p.

Una vez generado el espacio de escala cada imagen resultante se evalúa por medio de un umbral y todos aquellos puntos que no superen este umbral son eliminados dejando solo las características más fuertes; al incrementar disminuyen el número de características y viceversa. Después de este paso se realiza, al igual que en SIFT, la supresión de características no máximas comparando las características de las escalas internas con sus 26 vecinos, como muestra en la Figura 5, esto provee un mapa de características reducido.

Asignación de orientaciones

Teniendo un grupo selecto de características, el siguiente paso es asignarle una orientación que pueda ser repetible bajo distintas circunstancias, para ello se utilizan los filtros Haar (Figura 9) de tamaño 4ρ en un radio de tamaño 6ρ alrededor de la característica en la imagen integral, donde ρ es la escala en la cual la característica fue detectada. Los filtros Haar proveen los gradientes de la imagen con los cuales se extraen las respuestas al filtro en las direcciones x y y de cada pixel en la zona seleccionada. Para seleccionar la orientación dominante se rota una ventana de $\frac{\pi}{3}$ sobre el origen de la característica sumando las

respuestas al filtro y extrayendo su magnitud y orientación (Figura 10). La orientación de la característica será aquella del vector que tenga la mayor magnitud.

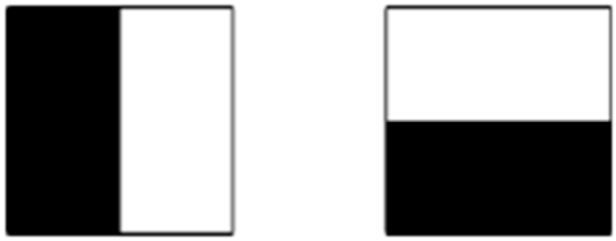


Figura 9: Filtros Haar. Izquierda, gradiente en x, derecha gradiente en y. La zona negra = 1, zona blanca = -1. Los filtros Haar en imágenes integrales requieren solo 6 operaciones. Imagen recuperada de Bay et al., 2008. 351 p.

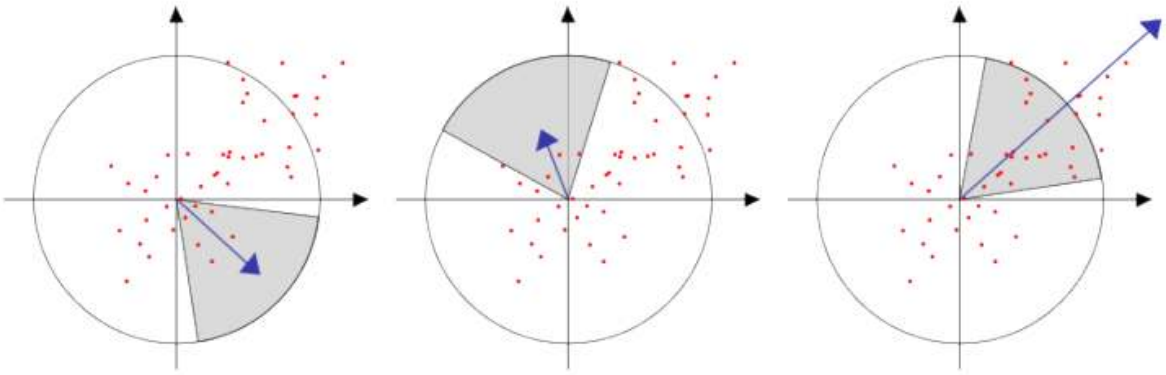


Figura 10: Asignación de orientación. La ventana rota alrededor de la característica sumando las respuestas de los filtros Haar y obteniendo un vector por zona, aquel con la mayor magnitud proporciona su orientación a la característica. Imagen recuperada de Bay et al., 2008. 351 p.

Descriptor de características

Para extraer el descriptor SURF se construye una ventana cuadrada de tamaño 20ρ (donde ρ es la escala en la cual se localizó la característica) alrededor

de la característica y orientada en la dirección calculada anteriormente. Esta ventana se divide en 16 zonas cuadradas regulares las cuales se filtran con filtros Haar de tamaño 2ρ obteniendo de esta forma los siguientes componentes por cada zona como lo definen Bay *et al.* (2008, 352 p.)

$$zona = \left\{\sum dx\,, \sum dy\,, \sum |dx|\,, \sum |dy|\right\}. \tag{60}$$

De tal manera que cada zona contribuye 4 componentes al descriptor, por lo tanto el descriptor SURF se convierte en un vector de $16x4 = 64$ dimensiones (Figura 11).

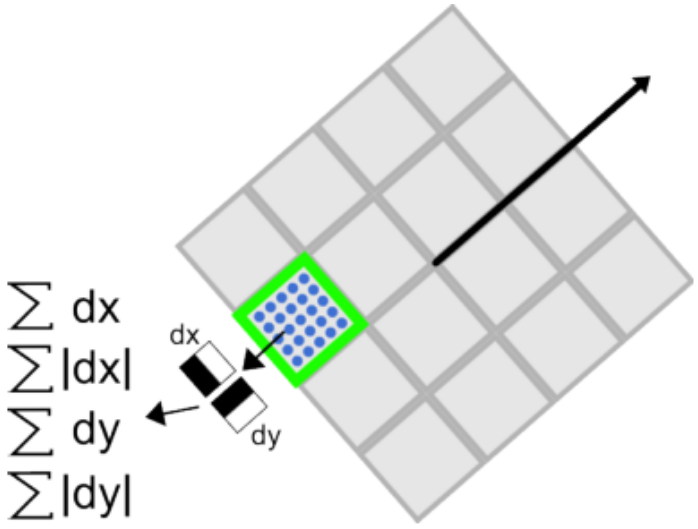


Figura 11: Componentes del descriptor. Por cada zona se calculan las respuestas a los filtros Haar obteniendo 4 componentes por zona. Imagen recuperada de Bay et al., 2008. 352p.

Correspondencia

Este paso se lleva a cabo por medio del algoritmo de vecinos cercanos tomando en cuenta el signo de la respuesta con los kernels de Gaussianas, de tal

forma que solo se comparan aquellos que tengan el mismo signo pues las características siempre se ubican en zonas claras u oscuras que darán lugar a respuestas con signos diferentes, de tal forma que al comparar de esta manera el proceso se acelera.

III.3 Algoritmos del método basado en área

Esta sección presenta el algoritmo clásico basado en área y un nuevo algoritmo basado en histogramas orientados (HoG) denominado SICHT propuesto por Zalesky y Lukashevich (2011). El principio de los algoritmos de este método es el uso de una ventana deslizante que abarca una cierta porción de la imagen en la cual se busca otra imagen modelo; por cada espacio en el cual se evalúa la ventana deslizante contra el modelo pueden realizarse diversas operaciones para otorgarle robustez al algoritmo.

La velocidad y precisión de este tipo de algoritmos depende en gran medida del ritmo de avance de la ventana deslizante; si la ventana avanza de unos pocos pixeles por cada comparación que se realiza es altamente probable que tenga buena precisión pero baja velocidad, mientras que si la ventana deslizante avanza una gran cantidad de pixeles (digamos el tamaño del modelo buscado o más) por cada comparación el algoritmo puede ejecutarse rápidamente pero con baja precisión; por lo tanto debe equilibrarse el avance de la ventana deslizante para obtener buena velocidad y precisión.

III.3.1 Plantilla

El algoritmo clásico del método basado en área. Consiste en buscar pixel por pixel con una ventana del tamaño de la imagen modelo el área más parecida a ésta en la imagen de búsqueda como se muestra en la Figura 12. Este es un proceso iterativo en el cual en cada iteración se comparan, mediante una métrica de distancia, la imagen modelo y el parche de la imagen de búsqueda.

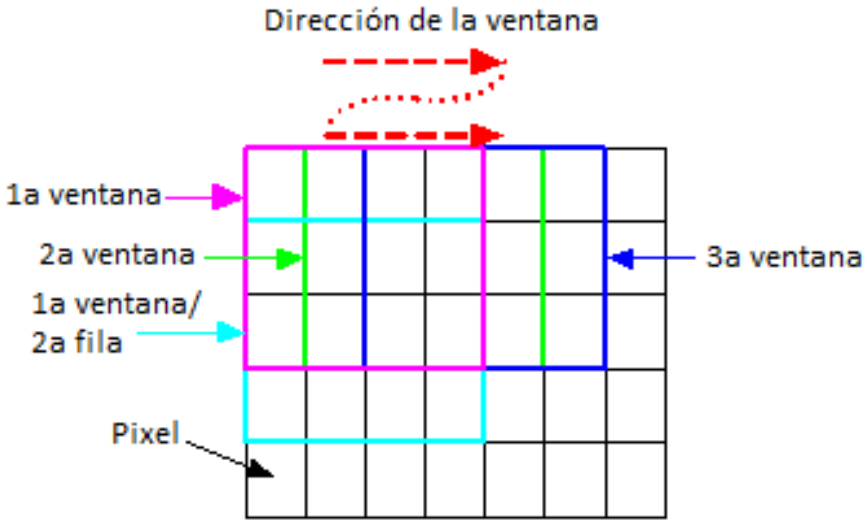


Figura 12: Ventana deslizante. La ventana avanza un número determinado de pixeles por iteración (en el ejemplo 1 pixel) hasta que termina la fila, luego baja un pixel y repite el proceso. Imagen recuperada de http://www.ece.neu.edu/groups/rcl/projects/SWO/index.htm.

Este proceso es sencillo de visualizar y programar, sin embargo es muy ineficiente pues no presenta ningún tipo de invarianza a deformaciones geométricas, cambios de iluminación y/o escala. Una mejora sencilla que se le puede hacer a este algoritmo es realizar la comparación entre la imagen modelo y el parche de la imagen de búsqueda mediante sus histogramas de intensidad.

III.3.2 SICHT

SICHT es un algoritmo que se basa fuertemente en el uso de histogramas de gradiente orientado (HoG) generando histogramas de 360 compartimientos, uno por cada grado de rotación; esto permite discriminar los histogramas mediante un amplio espectro contando efectivamente los cambios de dirección en los pixeles de la imagen.

Este algoritmo se diseñó para trabajar con imágenes aéreas y satelitales, por lo tanto no se espera que su desempeño en localización de objetos con fondo sea sobresaliente; sin embargo, este algoritmo se seleccionó para aplicarle

mejoras y tenga un mejor desempeño tanto para imágenes de objetos con fondo como para las mismas imágenes satelitales y aéreas, en el Capítulo IV se plantean las modificaciones y mejoras hechas a este algoritmo.

Selección de puntos de prueba en el espacio de escala

El primer paso que sigue este algoritmo es la generación de un espacio de escala, a diferencia de SIFT y SURF que usan convolución con kernels Gaussianos para definir las escalas y muestreo para definir las octavas, SICHT solo utiliza el muestreo, a cada imagen muestreada le llama simplemente *escala*.

Este proceso solo se lleva a cabo en la imagen de búsqueda. El espacio de escala corresponde a un conjunto de copias de la imagen de búsqueda $L = \{I(0), \dots, I(k-1)\}$, donde k es el número de escalas, con un factor de escala apropiado $0 < \rho < 1$, luego, el tamaño de cada imagen como muestra Zalesky (2011, 26 p.), está dado por

$$I(l) = \rho^l \times I, \quad l = 0, \dots, k-1. \tag{61}$$

Una vez generado el espacio de escala deben seleccionarse los puntos en los cuales se realizará el proceso de comparación, normalmente en un algoritmo basado en área se haría pixel a pixel como se muestra en la Figura 12, sin embargo para este algoritmo se decide sacrificar precisión por velocidad, de esta forma se selecciona un conjunto de puntos equidistantes en las imágenes del espacio de escala, siempre la misma distancia sin importar el factor de escala, Figura 13, y se extraen un conjunto de sub-imágenes del tamaño de la imagen modelo.

Generación de histogramas de gradiente

El siguiente procedimiento se aplica tanto a la imagen modelo como a cada parche del espacio de escala que se generó en el paso anterior. Se obtiene el gradiente de cada imagen mediante los kernels que se muestran en la Figura 14,

después se obtiene la orientación de cada pixel en la imagen a través de (45), para finalmente generar un HoG con 360 compartimientos.

Figura 13: Espacio de escala SICHT. Se selecciona un conjunto de puntos equidistantes en las imágenes del espacio de escala sin importar su factor de escala.

$$W_H = \begin{bmatrix} -1 & \cdots & & 0 & +1 & \cdots & +1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & & 0 & +1 & \cdots & +1 \end{bmatrix} \quad W_V = \begin{bmatrix} -1 & \cdots & -1 \\ \vdots & \ddots & \vdots \\ -1 & \cdots & -1 \\ 0 & \cdots & 0 \\ +1 & \cdots & +1 \\ \vdots & \ddots & \vdots \\ +1 & \cdots & +1 \end{bmatrix}$$

Figura 14: Ventanas de convolución SIFT. Izquierda gradiente en x. Derecha gradiente en y.

El siguiente paso es alinear el histograma haciendo un corrimiento cíclico para colocar el valor máximo de éste en la posición 0, lo cual proporciona en cierto grado invarianza a rotación, sin embargo, en este punto el histograma no es muy

suave lo cual suele provocar que la alineación cause un número significativo de errores en la localización de las direcciones principales del gradiente; para solucionar esto en cierto grado se suaviza el histograma al convolucionarlo con un filtro de media

$$h' = h * a, \tag{62}$$

$$a = \frac{1}{d} \quad d = tamaño\ de\ la\ ventana\ de\ convolución. \tag{63}$$

Ya suavizado el histograma se procede a alinear el histograma con el valor máximo de éste en la posición 0

$$h' = \max_{0 \le i \le 359} \{h'_i\}. \tag{64}$$

La Figura 15 muestra un histograma alineado y su respectivo parche de la imagen de búsqueda.

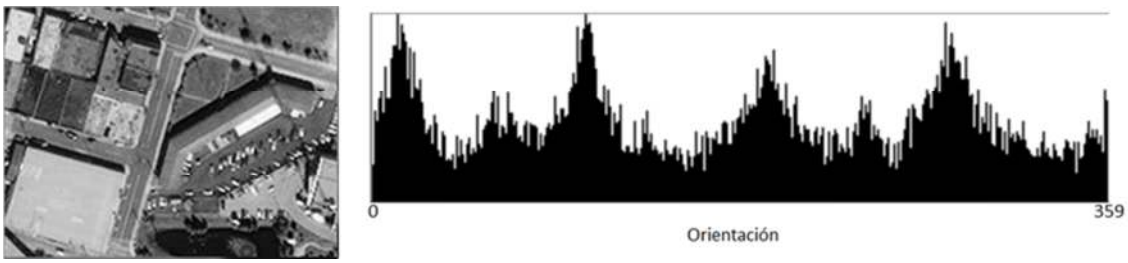


Figura 15: Histograma de gradiente orientado de 360 compartimientos de la imagen de la izquierda.

Es posible, sacrificando velocidad de procesamiento, obtener un conjunto de histogramas alineados a una cantidad finita de máximos locales $H = \{h'_{(1)}, h'_{(2)}, h'_{(3)}, \dots\}$ para aumentar la robustez del proceso.

Correspondencia

Una vez calculados todos los histogramas, éstos se comparan contra el propio histograma de la imagen modelo mediante la métrica de distancia Euclidiana, pues de acuerdo con Zalesky otras métricas no alcanzaron la eficiencia de la distancia Euclidiana. En caso de haber procesado múltiples histogramas por muestra, éstos se comparan con cada uno de los histogramas de la imagen modelo. Finalmente, las coordenadas del parche que tiene la menor distancia con la imagen modelo es la que se toma como solución a la correspondencia.

III.4 Algoritmos del método híbrido

En esta sección se presenta un algoritmo híbrido, estos algoritmos tratan de extraer y combinar las mejores características de los algoritmos de los métodos basados en área y características. La dificultad de estos algoritmos se encuentra en el grado de compatibilidad que tengan los algoritmos y estructuras de datos empleadas para su construcción; o la forma de convertir un área en una característica o viceversa.

Aún y cuando en los últimos años se ha visto un incremento en el desarrollo de estos algoritmos, la comunidad científica prefiere centrarse en el desarrollo de los métodos clásicos en lugar de intentar fusionarlos. Aunque es verdad que muchos de los algoritmos de éste método se enfocan a ciertos problemas solamente, hay algunos otros como son la simple combinación de un algoritmo de cada método, una solución viable que se muestra a continuación.

III.4.1 MSER + SIFT

MSER es un algoritmo basado en área que busca conjuntos de pixeles conectados para crear áreas de interés, de éstas áreas se buscan luego los

pixeles centrales que fungirán como las características de entrada para generar los descriptores SIFT.

Como se puede apreciar, éste algoritmo es una simple combinación entre dos algoritmos donde la entrada de uno es fácilmente convertida de la salida del otro con buenos resultados por las características de construcción de los descriptores SIFT (véase III.2.1). A continuación se presenta el procedimiento para extraer las características a partir de áreas conectadas utilizadas por MSER.

Generación de regiones máximas

Para obtener las regiones máximas se procede de la siguiente manera: primeramente se ordenan por intensidad todos los pixeles de la imagen en forma ascendente o descendente, la lista de componentes y áreas conectadas se mantiene mediante un algoritmo de búsqueda-unión con el cual pueden extraerse las zonas conectadas con cierta intensidad. Este proceso produce una estructura de datos que guarda áreas de componentes conectados como una función de intensidad, de tal forma que es posible elegir diversos umbrales en la intensidad hasta encontrar áreas estables, de tal forma que al cambiar el umbral, pequeñas áreas pueden fusionarse entre sí dependiendo del umbral seleccionado.

Los umbrales que producen las regiones estables máximas se seleccionan de las áreas conectadas en sí mismas, se toma la intensidad máxima de cada área conectada, lo cual puede ocasionar que otras pequeñas áreas adyacentes se adhieran a las más grandes.

Descriptores SIFT y correspondencia

Después de calcular las regiones máximas, el pixel central de cada una de éstas se toma como una característica y se procede a calcular los descriptores SIFT, los cuales toman la información del área conectada al pixel central del área conectada para extraer el vector de características como se muestra en III.2.1. De

tal forma que los descriptores se forman de información contenida en las regiones máximas por lo cual preservan su conectividad.

Finalmente se utiliza el mismo método de correspondencia que utiliza SIFT entre las imágenes que se están comparando.

III.5 Resumen

En este capítulo se presentaron los algoritmos de correspondencia que se evaluarán posteriormente. Como puede verse, los algoritmos del método basado en características son muy similares en su estructura básica, ambos algoritmos construyen un espacio de escala, buscan los pixeles más prominentes en la imagen y generan vectores de características llamados descriptores mediante los cuales buscan correspondencia entre diferentes imágenes. De forma similar, los algoritmos del método basado en área recorren la imagen mediante ventanas deslizantes en las cuales extraen información como gradientes, histogramas o magnitudes para luego comparar éstas entre diferentes imágenes buscando correspondencia entre fragmentos. Mientras que los algoritmos del método híbrido toman características de ambas metodologías implementando algoritmos de ambos métodos y tratando de combinarlos de la mejor manera posible, esto es una tarea difícil pues no siempre la salida de un algoritmo es afín a la entrada de otro.

Capítulo IV

Algoritmo de correspondencia propuesto

IV.1 Introducción

Como se vio en el Capítulo III, existen varios algoritmos dedicados a la correspondencia de objetos o áreas de interés, desarrollándose en dos áreas distintivas, aquellos que atacan el problema mediante la extracción de puntos característicos en la imagen y aquellos que buscan similitudes entre dos áreas completas en dos imágenes. En este capítulo se describe paso a paso el algoritmo desarrollado el cual está basado en SICHT (Zalesky y Lukashevich, 2011) y contiene las siguientes etapas: pre-filtraje, creación del espacio de escala, generación de los histogramas de gradiente orientado y correspondencia.

IV.2 Pre-filtraje

Normalmente una imagen digital contiene algún tipo de ruido que modifica los valores esperados de los pixeles en ésta, el ruido puede provenir de varias fuentes, desde el dispositivo de adquisición de la imagen, valores modificados por algún tipo de compresión de la imagen, errores de transmisión, entre muchos otros. Existen varios tipos de ruido, sin embargo en este trabajo de tesis nos concentraremos en el ruido blanco aditivo con media cero.

El ruido blanco es una señal aleatoria la cual no está correlacionada, por lo tanto, su función de autocorrelación es una delta y su espectro de potencia es una constante como se muestra en la Figura 16. Esto significa que la señal contiene todas las frecuencias y todas muestran el mismo valor en el espectro de potencia.

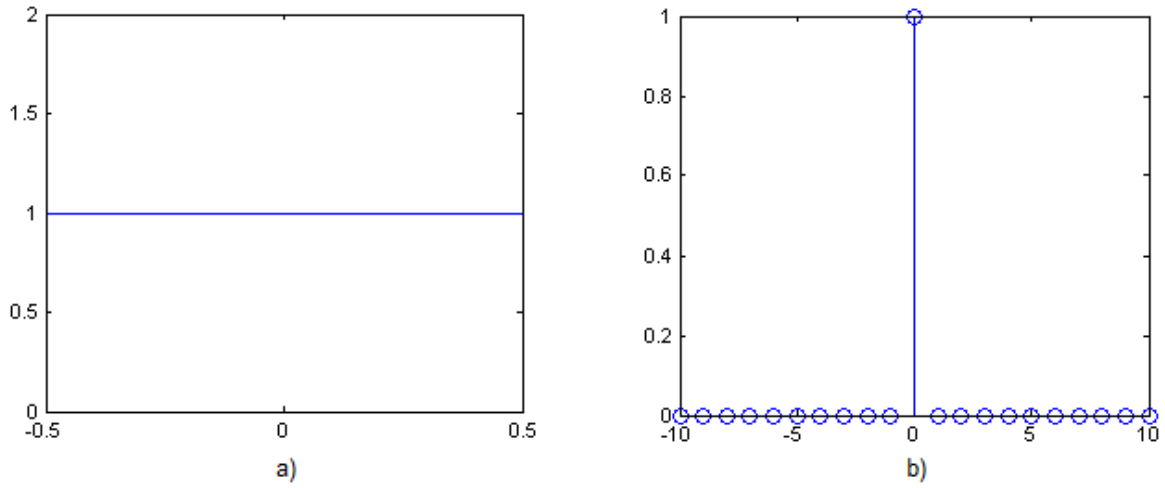


Figura 16: a) Espectro de potencia de ruido blanco. b) Autocorrelación de ruido blanco.

Esta fase del algoritmo consta de dos pasos, primero detectamos la cantidad de ruido blanco que contiene la imagen mediante la extracción de desviación estándar y después aplicamos un filtro de media con un vecindario adaptativo EV.

Detección del ruido

Como vimos anteriormente la autocorrelación del ruido blanco se concentra en el origen, además se sabe que el coeficiente de correlación de dos señales es igual a la covarianza de los datos de ésta, y si estas señales son iguales, entonces es igual a la varianza. En estadística la correlación está dada por

$$Corr(X,Y) = E[(x_i - \mu_X)(y_i - \mu_Y)] = Cov(X,Y), \tag{65}$$

luego,

$$Autocorr(X) = E[(x_i - \mu)(x_{i-k} - \mu)] = Cov(X,X) = Var(X) = \sigma^2. \tag{66}$$

Entonces, considerando el modelo de ruido aditivo como se muestra en la Figura 17

$$f'(x,y) = f(x,y) + n(x,y), \tag{67}$$

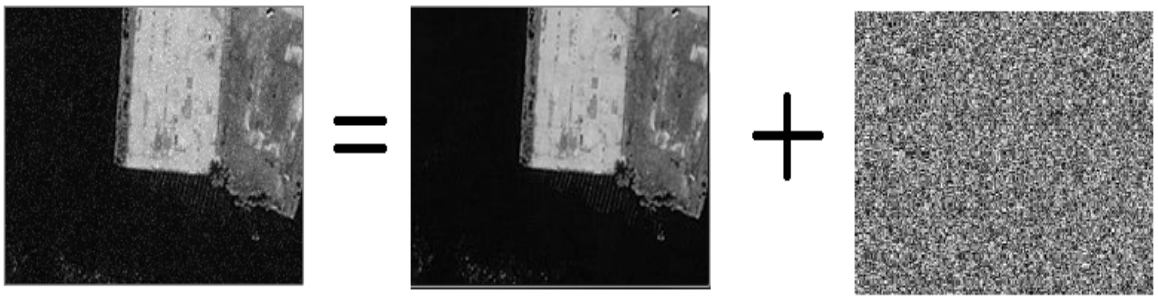


Figura 17: Modelo de ruido aditivo. La imagen observada f' resultado de la adición de ruido a la imagen real.

donde f' es la imagen observada, f es la imagen sin ruido y n es el ruido blanco que está deformando la imagen original. Considerando que la varianza del ruido blanco se suma a la varianza de la imagen; luego obteniendo la autocorrelación de la imagen ruidosa es posible estimar la varianza de la imagen no ruidosa mediante interpolación lineal con los valores contiguos y restando ésta al pico de la autocorrelación se obtiene la varianza del ruido blanco, este concepto se ilustra en la Figura 18.

El proceso de autocorrelación es computacionalmente muy costoso, luego se aplican los siguientes pasos para obtener el mapa de autocorrelación

$$\begin{aligned} \mathrm{F}(\omega) &= \mathcal{F}\{f(x)\}, \\ S(\omega) &= \mathrm{F}(\omega)\mathrm{F}^{*}(\omega), \\ A(\tau) &= \mathcal{F}^{-1}\{S(\omega)\}, \end{aligned} \tag{68}$$

donde $\mathrm{F}(\omega)$ es la transformada de Fourier de $f(x)$, $S(\omega)$ es el espectro de potencia $|F(\omega)|^2$ y $A(\tau)$ es el mapa de autocorrelación que se obtiene mediante la transformada inversa de Fourier de $S(\omega)$. Este proceso es computacionalmente más eficiente que realizar la autocorrelación directamente sobre las imágenes pues en el espacio de frecuencias la operación correlación se reduce a la multiplicación punto a punto de dos matrices.

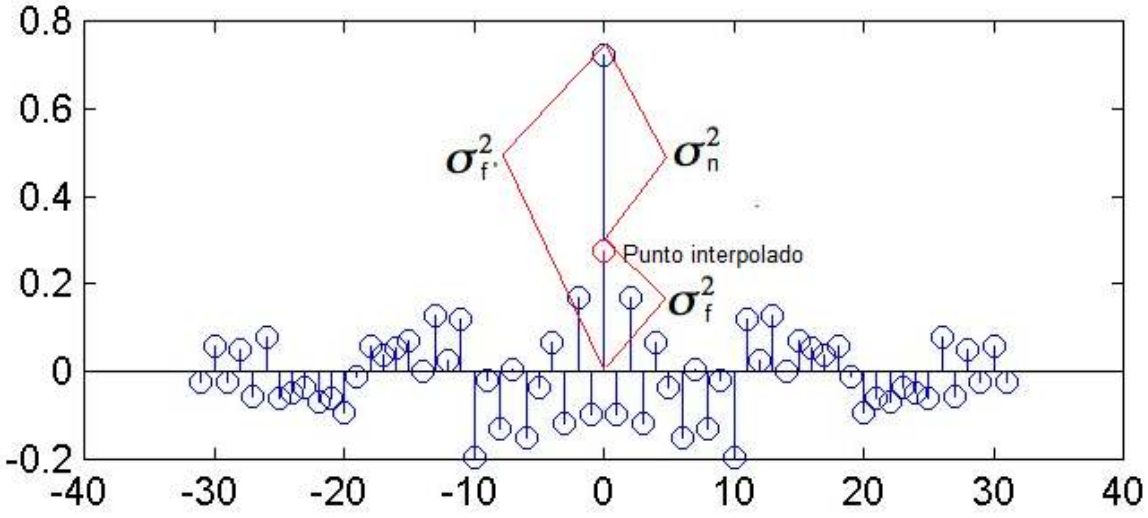


Figura 18: Autocorrelación de $f'(x)$. Estimación de la varianza de $f(x)$ en el origen del mapa de autocorrelación y de la varianza del ruido.

Habiendo obtenido el mapa de correlación, sabemos que en el origen se tiene la varianza de la imagen ruidosa, aplicamos entonces interpolación lineal con los dos puntos adyacentes para obtener la varianza de la imagen libre de ruido

$$\begin{aligned} \sigma_{f'}^2 &= A(0), \\ \sigma_f^2 &= 2A(1) - A(2), \end{aligned} \tag{69}$$

para finalmente extraer la desviación estándar del ruido mediante

$$\sigma_n = \sqrt{\sigma_{f'}^2 - \sigma_f^2}. \tag{70}$$

Supresión del ruido blanco

El siguiente paso es remover el ruido mediante un filtro. Se ha seleccionado el filtro de media Ecuación recuperada de Kober *et al.* (2001, 973 p.).

$$\hat{v}_{i,j} = MEDIA\left(EV\left\lfloor v_{i,j} \right\rfloor\right), \tag{71}$$

donde $EV(v_{i,j})$ es el vecindario adaptativo que se muestra en la ecuación (4), para el cual se requiere conocer el tamaño de ε, de acuerdo con Kober ***et al.*** (2001) el valor recomendado es $\varepsilon = 1.5\sigma$, donde σ es la desviación estándar del ruido blanco (σ_n).

Mediante este enfoque buscamos un valor para el pixel central en un vecindario espacial de NxN que esté fuera del rango del ruido blanco ya que se expande la búsqueda de pixeles con valores más grandes y más chicos que la desviación estándar del ruido.

IV.3 Espacio de escala

Uno de los aspectos importantes del algoritmo es que debe ser invariante a escala, es decir, que debe ser posible ubicar la imagen modelo dentro de la imagen base aunque la escala sea distinta; para ello se crea un espacio de escala que consta de un conjunto de imágenes muestreadas de la imagen base mediante un factor predeterminado $0 < \rho < 1$, se denominará a esta estructura como SS tal como se define en Zalesky (2011, 26 p.)

$$SS(\ell) = \rho^{\ell} \times I, \tag{72}$$

donde $\ell \in \mathbb{Z}$ es el número de escala, e I es la imagen base. Por lo tanto si ℓ toma números negativos la imagen incrementará su tamaño por un factor ρ^{ℓ}, según Lowe (1999) el muestrear la imagen al doble de tamaño original ayuda a SIFT a generar una mayor cantidad de puntos de interés, en este caso ayuda a incrementar el factor de correspondencia al resaltar estructuras en la imagen; si ℓ toma valores positivos la imagen base disminuirá su tamaño por un factor ρ^{ℓ} proporcionando varias escalas posibles en las que puede encontrarse la imagen modelo.

El hecho de muestrear una imagen no es suficiente para crear el espacio de escala, al muestrear la imagen es posible que se cree un efecto de *aliasing*[1] el cual se desea eliminar o disminuir, para ello se utiliza un filtro Gaussiano pues de acuerdo con Lindeberg (1994) estos filtros son los únicos que proporcionan un suavizado correcto para el análisis del espacio de escala. Para aplicar el filtro Gaussiano debe elegirse la cantidad de suavizado que se le aplicará a la imagen, este factor es la desviación estándar (σ) que se especifica en el filtro Gaussiano. Para la imagen original se ha seleccionado $\sigma = \sqrt{2}$ como se hace en SIFT, y para las imágenes subsecuentes en $SS\ \sigma = k\sqrt{2}$ donde $k = 1/\rho^{\ell}$.

Para construir el espacio de escala entonces se sigue el siguiente procedimiento, primero se muestrea la imagen hacia arriba y se le aplica el filtro Gaussiano, se toma como la imagen original y se comienza a muestrear hacia abajo tantos niveles como se necesiten, se aplica el filtro Gaussiano antes de muestrear hacia abajo. En este trabajo de escala se eligió un factor de escala $\rho = \frac{1}{2}$, lo cual nos proporciona una primera imagen del doble de tamaño a la original y siguientes imágenes a la mitad de la anterior, y $\ell = [-1,3]$ de tal manera que trabajamos con cinco escalas por búsqueda. El proceso de construcción del espacio de escala se resume en la Figura 19.

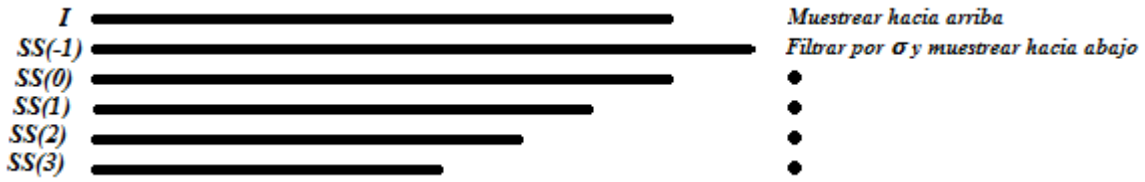


Figura 19: Proceso de creación del espacio de escala.

.

[1] Aliasing: Distorsión en la que las esquinas de los objetos aparecen cortadas de forma irregular. Este fenómeno se debe a la naturaleza cuadrada de los pixeles utilizados para construir la imagen.

IV.4 Generación de los histogramas de gradiente orientado

Una vez filtrado el ruido en las imágenes a comparar y generado el espacio de escala de la imagen base se procede a generar el conjunto de histogramas de gradiente orientado de ambas imágenes para comparar.

Por cada imagen en el espacio de escala se seleccionan un conjunto de puntos equidistantes a lo ancho de la imagen, la distancia entre punto y punto es independiente del nivel del espacio de escala que se vaya a trabajar. Por cada punto extraído se obtiene un *parche* de las mismas dimensiones que la imagen modelo de donde se extraerán histogramas de gradiente orientado como se muestra en la Figura 20.

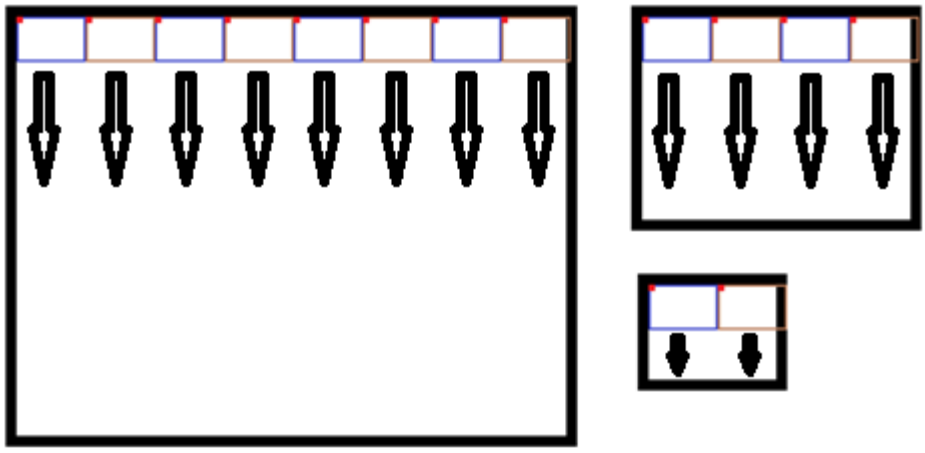


Figura 20: Selección de puntos equidistantes en el espacio de escala y extracción de parches con las dimensiones de la imagen modelo.

Es importante notar que los siguientes procesos pueden realizarse de forma paralela en varios hilos de ejecución, pues cada parche es independiente de los demás y cada uno generará un histograma de gradiente orientado que se comparará con el histograma de gradiente orientado de la imagen modelo. Otro punto a notar es que solo se extrae el parche en la *primera fila* de cada imagen,

pues después de obtener el histograma de gradiente orientado de este primer parche se realiza un proceso iterativo pixel a pixel a todo lo alto de la imagen modificando el histograma de la iteración anterior como se muestra en la Figura 21.

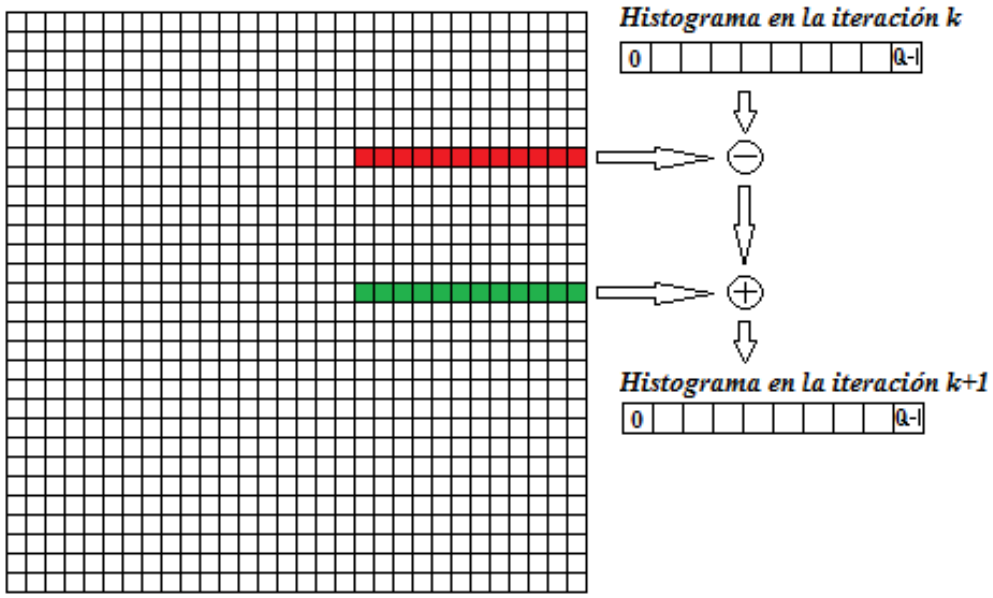


Figura 21: Generación iterativa de histogramas. La ventana deslizante baja pixel por pixel restando los valores de las orientaciones de la fila que sale de la ventana y sumando las orientaciones de la fila que entra en la ventana.

Histogramas de magnitud y de gradiente orientado

Habiendo establecido la manera en que se extraen los histogramas en cada parche y cómo se modifican iteración tras iteración procedemos a detallar cómo se forman los histogramas de gradiente orientado y sus restricciones mediante sus histogramas de magnitud correspondiente.

A cada *parche inicial* se le aplica el operador de gradiente de Sobel de la ecuación (49) sobre todos sus pixeles para obtener dos matrices de gradiente g_x y

g_y; mediante la ecuación (44) se crea la matriz de magnitud del parche, la cual contiene la magnitud de cada pixel en el parche, finalmente se genera el *histograma de magnitud* que representa al parche. Mientras más grande sea la magnitud es indicio de grandes cambios en la intensidad de los pixeles alrededor del pixel central, es decir, indica la presencia de bordes y esquinas, mientras que magnitudes bajas indican la presencia de texturas que no aportan mucha información útil al proceso.

Por lo tanto, se selecciona un umbral para tomar en cuenta solamente las magnitudes que representen bordes y esquinas en la imagen. Para ello se seleccionó la *mediana* del histograma, ya que es un estadístico robusto, es decir, valores muy altos o muy bajos en el histograma no afectan el valor de la mediana como lo puede hacer con el estadístico de la media como se muestra en la Figura 22.

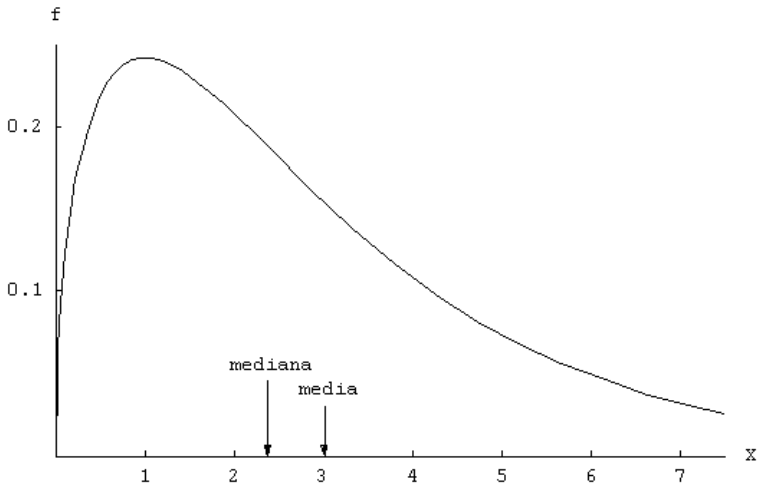


Figura 22: La mediana es seleccionada como umbral del histograma de magnitud pues no se ve afectada por valores muy altos o muy bajos como la media.

Una vez establecido el umbral de magnitud se procede a construir el histograma de gradiente orientado. Para todo pixel del parche que su magnitud

sea mayor al umbral se toma en cuenta su orientación para formar parte del histograma de gradiente orientado

$$h(ori_{i,j}) = \begin{cases} +1 & si \quad mag_{i,j} > umbral \\ 0 & de\ otra\ manera \end{cases}, \tag{73}$$

donde $ori_{i,j}$ es la orientación en la posición (i,j) del parche, $mag_{i,j}$ es la magnitud en la posición (i,j) del parche y h es el histograma de gradiente orientado.

El histograma de gradiente orientado puede construirse con un máximo de 360 *compartimientos o bins*; sin embargo, un histograma con un bin por cada grado de orientación contiene muchos detalles, los cuales en presencia de ruido pueden cambiar el ángulo real del pixel en varios grados, es por ello que se propone cuantizar el histograma de gradiente orientado a una menor cantidad de bins, acumulando en cada uno de éstos varios grados de orientación del pixel evitando de esta manera la ambigüedad que pudiera causar el ruido en la imagen como lo hacen SIFT y SURF en el proceso de construcción de sus descriptores como se muestra en la Figura 23.

Como se mencionó anteriormente, el procedimiento descrito arriba se aplica tanto a la imagen modelo para extraer su único histograma de gradiente orientado y a la primera fila de cada imagen en el espacio de escala.

Después de extraer el primer histograma de gradiente orientado se procede a modificar de manera iterativa el histograma de magnitud como se muestra en la Figura 21, donde al extraer datos de la ventana deslizante e insertar nuevos datos en ésta es posible que el valor del umbral cambie también, es por ello que se hace un cálculo para identificar si hubo un cambio en el valor del umbral, si el valor del umbral tuvo cambios se extraen o insertan los valores correspondientes al histograma de gradiente orientado. De esta manera se mantienen actualizados dependiendo de la posición de la ventana deslizante los histogramas de magnitud y de gradiente orientado así como el umbral que delimita la cantidad de información que se incluye en este último histograma.

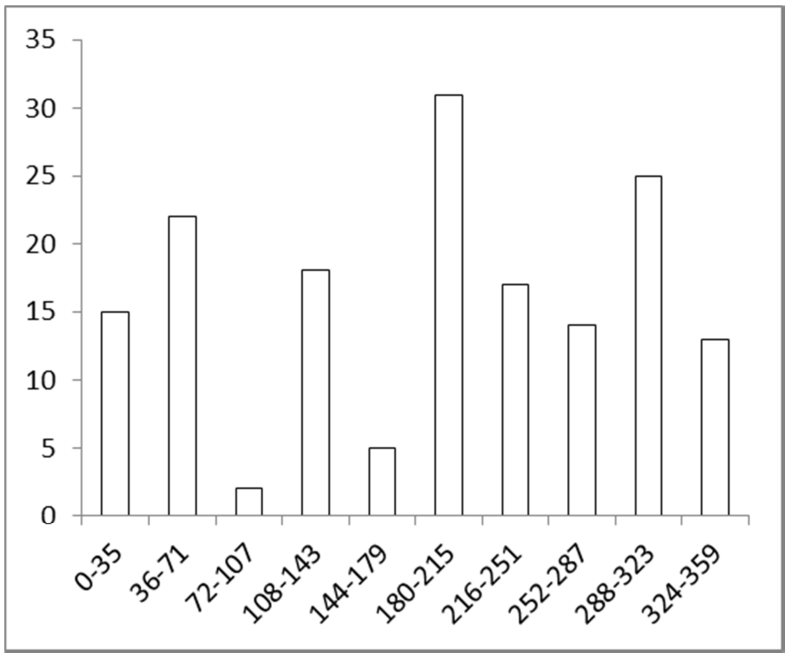


Figura 23: Histograma de gradiente orientado cuantizado a 10 bins de 36 grados de orientación cada uno.

IV.5 Correspondencia

Una vez adquiridos los histogramas de gradiente orientado de las imágenes del espacio de escala se procede a comparar cada uno de ellos con el histograma de gradiente orientado de la imagen modelo; sin embargo, antes de que esta comparación pueda darse es necesario realizar un par de procesos que ayudan a mejorar la calidad de correspondencia.

Zalesky y Lukashevish (2011) proponen convolucionar el histograma de gradiente orientado con un filtro de suavizado proponiendo el filtro de media

$$h' = h * a,$$
$$a = \frac{1}{d}, \tag{74}$$

donde d es el número de componentes a tomar en cuenta para la media. El segundo proceso que proponen es alinear el histograma haciendo un recorrido

cíclico del histograma de tal manera que el máximo de éste quede en la primera posición, este proceso le brinda al histograma invarianza a rotación

$$h'_0 = \max_{0 \le i \le bins} \{h'_i\}. \quad (75)$$

Finalmente, teniendo los histogramas suavizados y alineados el último paso es comparar todos los histogramas de gradiente orientado del espacio de escala con el de la imagen modelo, esta comparación se realiza mediante la distancia Euclidiana que se muestra en la ecuación (31), el histograma del parche con la menor distancia al histograma de la imagen modelo se toma como correspondencia, por lo tanto el parche asociado se denomina como correspondiente a la imagen modelo.

IV.6 Resumen

Este algoritmo aprovecha muchas de las ventajas de SICHT, pero también hace varias contribuciones para mejorar la correspondencia entre la imagen modelo y la imagen base. Primeramente se propone un pre-filtraje para eliminar en la medida de lo posible el ruido aditivo que pudiera existir en la imagen base, añadiéndole al algoritmo tolerancia a ruido aditivo. Adicional a esto, la cuantización del histograma de gradiente orientado como consecuencia del ruido aditivo para disminuir la ambigüedad del cambio de orientación que pudiera sufrir un pixel como consecuencia. Finalmente, el proceso iterativo y paralelo que se propone construye y evalúa de forma rápida y eficiente los histogramas de gradiente orientado de cada parche en el espacio de escala.

La Tabla I muestra los parámetros que se establecieron después de experimentos exhaustivos comparando el algoritmo con diferentes configuraciones.

Tabla I. Parámetros del algoritmo propuesto. Algunos parámetros son adaptativos y dependen de la cantidad de ruido blanco encontrado en la imagen.

Parámetro	**Valor**
ε del vecindario EV	$\begin{cases} 1.5\sigma & si\ \sigma \leq 15 \\ 2\sigma\ de\ otra\ manera \end{cases}$
Vecindario espacial $N \times N$	$\begin{cases} 7 \times 7 & si\ \sigma \leq 15 \\ 9 \times 9\ de\ otra\ manera \end{cases}$
Bins de HoG	$\begin{cases} 90 & si\ \sigma \leq 15 \\ 18\ de\ otra\ manera \end{cases}$
Factor de escala ρ	$1/2$
Distancia entre puntos equidistantes	Número de columnas de la imagen modelo

Capítulo V

Estudio comparativo y resultados

V.1 Introducción

En este capítulo se describen los experimentos para realizar el estudio comparativo, objetivo principal de esta tesis. Cada experimento está diseñado para describir el comportamiento de seis algoritmos en dos ámbitos distintos, el primero es para reconocer objetos en una imagen de escena mientras que el segundo está diseñado para ubicar fragmentos de *tierra* en imágenes aéreas y satelitales.

Los seis algoritmos que se evaluaron son aquellos descritos en esta tesis: SIFT, SURF, por plantillas, SICHT, MSER+SIFT y un nuevo algoritmo que se propone en esta tesis que ha sido basado en SICHT. Las implementaciones de estos algoritmos provienen de las siguientes fuentes: SIFT, SURF y MSER+SIFT son algoritmos implementados en la librería para C++ de OpenCV 2.3.1; el algoritmo de plantillas y SICHT son implementaciones basadas en el algoritmo general de correspondencia por plantillas y el algoritmo descrito en (Zalesky y Lukashevich, 2011) respectivamente; y el algoritmo propuesto es una implementación propia basada en SICHT.

Los criterios a evaluar son: rotación en plano, escala y tolerancia a ruido blanco aditivo en ambos experimentos; adicionalmente se evaluará rotación fuera de plano e iluminación en el primer experimento por la disposición de imágenes con estas características para este experimento. Para cada uno de estos criterios se cuenta con tres métricas con las cuales se evaluará cada criterio, las cuales son el porcentaje de aciertos, es decir si se encontró o no el objeto o fragmento en

la escena; la distancia máxima en pixeles del centro de la imagen encontrada por el algoritmo respecto al centro real de la imagen en la escena; y el tiempo de ejecución promedio del algoritmo para cada imagen.

A continuación se describen los experimentos así como el objetivo de cada uno de ellos y las características de las imágenes sobre las que se evalúan los algoritmos.

Las características del equipo de cómputo para realizar los experimentos se describen en la Tabla II:

Tabla II. Características del equipo de cómputo.

Modelo	ASUS G74SX
Procesador	Intel Core i7-2670QM 2.2 GHz
Memoria RAM	8 GB
Tarjeta gráfica	NVIDIA GeForce 560M
Sistema operativo	Windows 7 Home Premium

Finalmente se presentan los resultados de estos experimentos en cada uno de los algoritmos y se discuten las fortalezas y debilidades de cada algoritmo en cada criterio evaluado.

V.2 Experimentos

V.2.1 Experimento 1

El objetivo de este experimento es encontrar un objeto en una escena compleja que contiene un paisaje y otros objetos. Se utiliza el banco de imágenes ALOI (Amsterdam Library of Object Images), que consta de 1000 objetos de 192×144 pixeles de los cuales se cuenta con 72 imágenes de rotación fuera de plano cada 5° de 0° a 355° como se muestra en la Figura 24, y 24 imágenes de

iluminación de diferentes posiciones y distintas intensidades como se muestra en la Figura 25.

Se seleccionaron 10 imágenes de paisajes complejos de Internet y se ajustó su tamaño a 1280×1024 pixeles cada una a las cuales se les incrustaron 5 objetos diferentes del banco de imágenes ALOI para generar una escena más compleja como se muestra en la Figura 26. Se seleccionaron además otros 10 objetos diferentes los cuales serán buscados en la escena con diferentes rotaciones en plano, fuera de plano, con ruido blanco añadido, con diferentes escalas y diferentes iluminaciones.

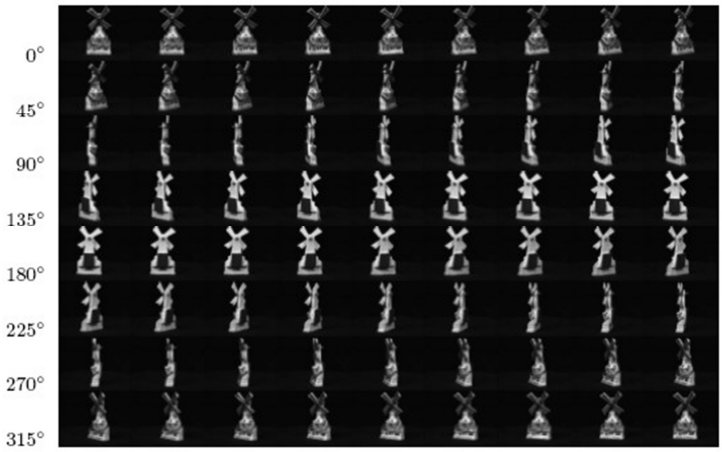


Figura 24: Ejemplo de un objeto del banco de imágenes ALOI mostrando una rotación fuera de plano de 360°. Imagen recuperada de http://staff.science.uva.nl/~aloi/.

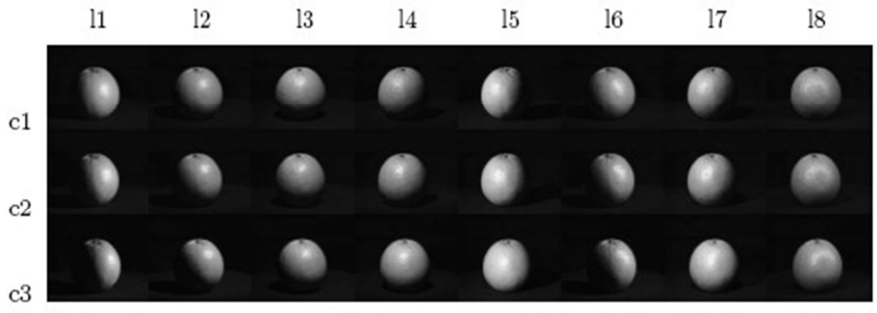


Figura 25: Ejemplo de un objeto del banco de imágenes ALOI que muestra las diferentes iluminaciones por objeto. Imagen recuperada de http://staff.science.uva.nl/~aloi/.

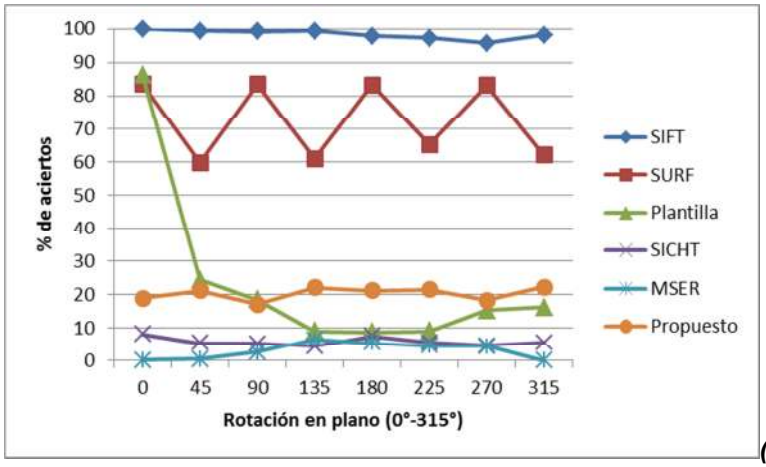
Se seleccionaron 100 posiciones distintas de forma aleatoria que estuvieran dentro del rango de la imagen de escena, luego, por cada uno de los 5 criterios mencionados anteriormente se busca el objeto en las 100 posiciones previamente mencionadas.

Figura 26: Imagen de escena en la cual se ubicará un objeto.

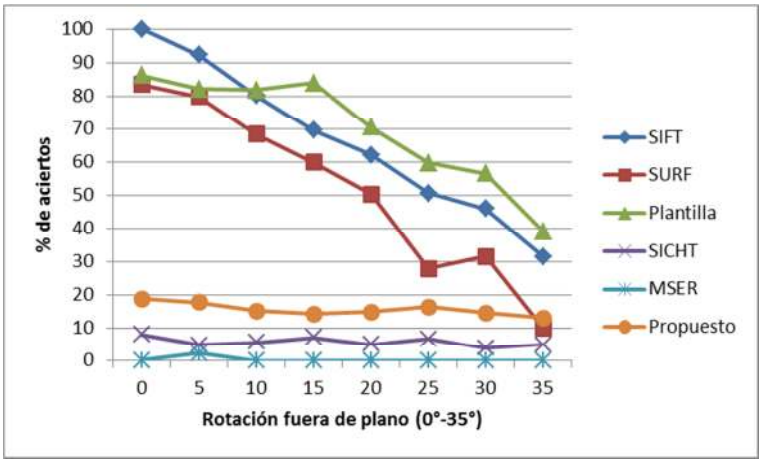
Resultados

La Figura 27 muestra el número de aciertos en puntos porcentuales de los diferentes algoritmos de correspondencia. Como puede apreciarse SIFT y SURF tienen por lo general los mayores porcentajes de aciertos en los diferentes criterios de comparación, siendo muy constantes en sus resultados teniendo desempaño similar con las diferentes variaciones de escala, rotación e iluminación y una

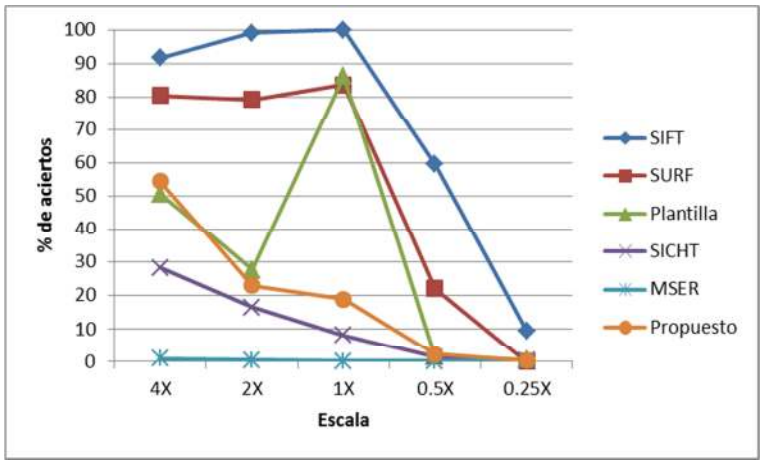
degradación constante mientras aumenta la desviación del ruido blanco. Por otro lado, es de notarse la ineficiencia de los algoritmos basados en área como el algoritmo de plantillas y SICHT, los cuales tienen muy bajo desempeño en la mayoría de los criterios, esto es debido al fondo de la imagen de escena que contribuye a aumentar la diferencia entre la ventana deslizante y la imagen de búsqueda que está segmentada. Finalmente, el algoritmo propuesto aunque no tiene un porcentaje alto de aciertos se antepone claramente a los resultados que se obtienen con SICHT, el algoritmo sobre el cual se basó; luego, puede apreciarse que este tipo de algoritmos no están diseñados para ubicar objetos.



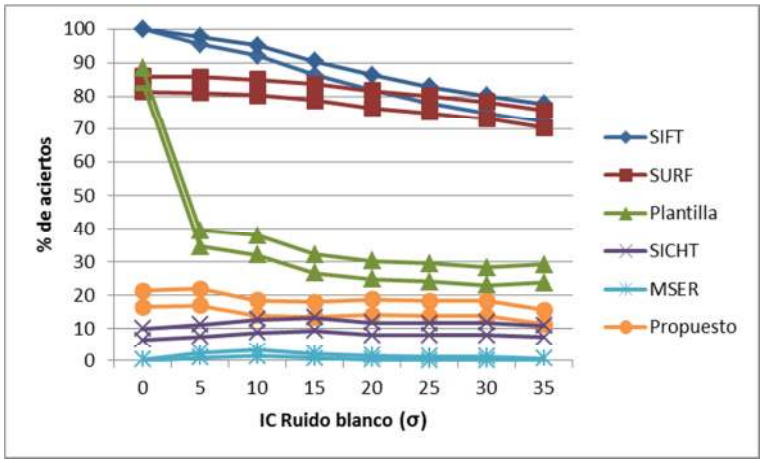
(a)



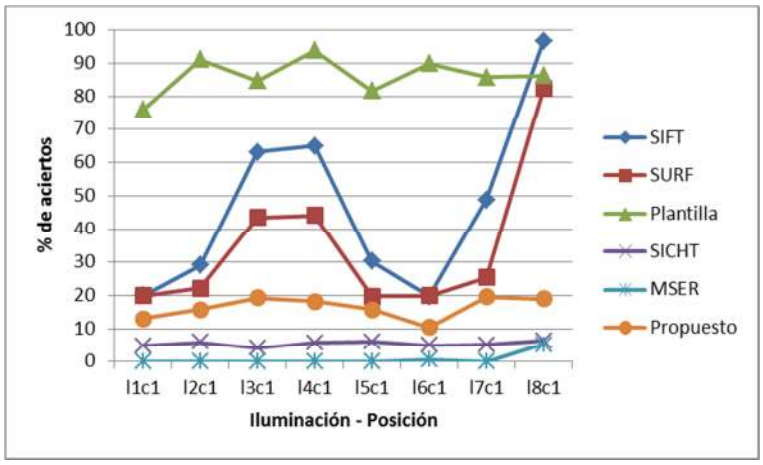
(b)



(c)



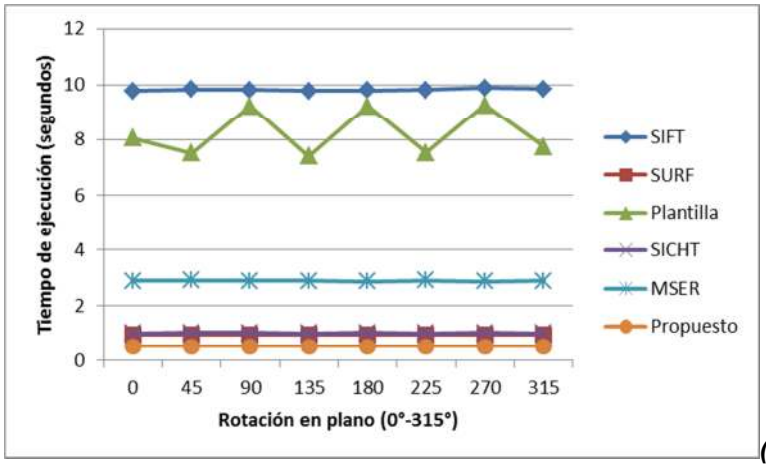
(d)



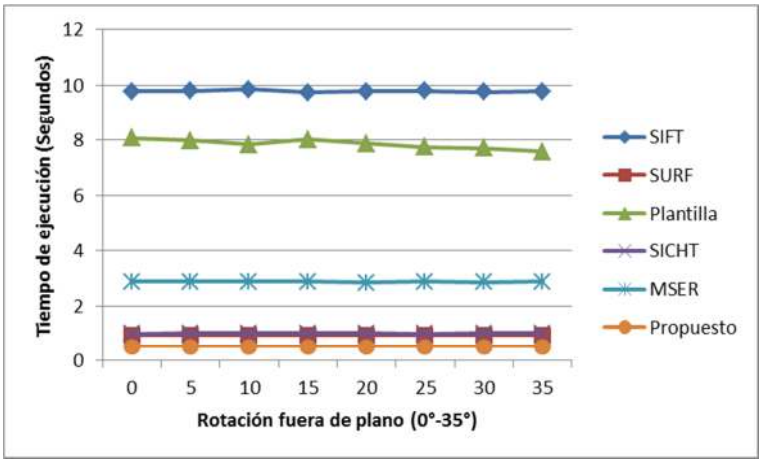
(e)

Figura 27: Desempeño de los algoritmos de correspondencia evaluados en: (a) rotación en plano, (b) rotación fuera de plano, (c) escala, (d) tolerancia a ruido aditivo blanco y (e) dirección de iluminación.

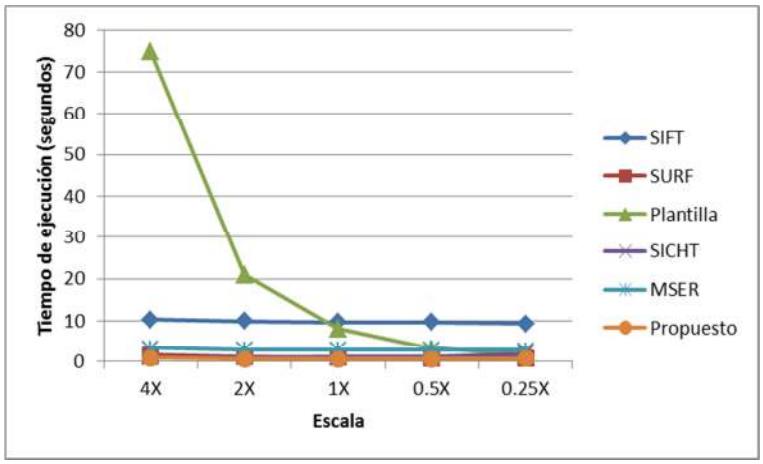
Sin embargo, el desempeño de SIFT se ve opacado por el tiempo de procesamiento para obtener resultados; es aquí donde SURF sale a relucir, pues si bien tiene un desempeño ligeramente inferior a SIFT su tiempo de respuesta es mucho menor como se muestra en la Figura 28. El algoritmo de plantillas y SIFT son claramente los más lentos de entre todos los algoritmos; y SICHT si bien es de los algoritmos más veloces, es una cualidad que no sirve de nada si no tiene contundencia a la hora de encontrar la imagen correspondiente. El algoritmo propuesto mantiene una constancia en cuanto al tiempo de procesamiento siendo el más rápido de entre los seis algoritmos implementados.



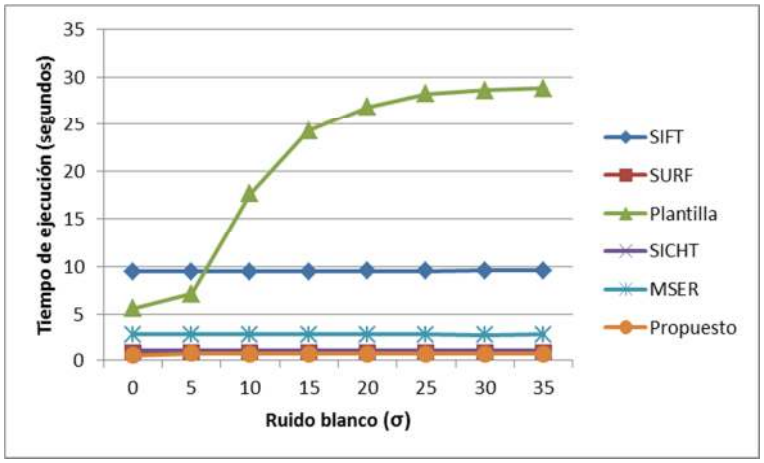
(a)



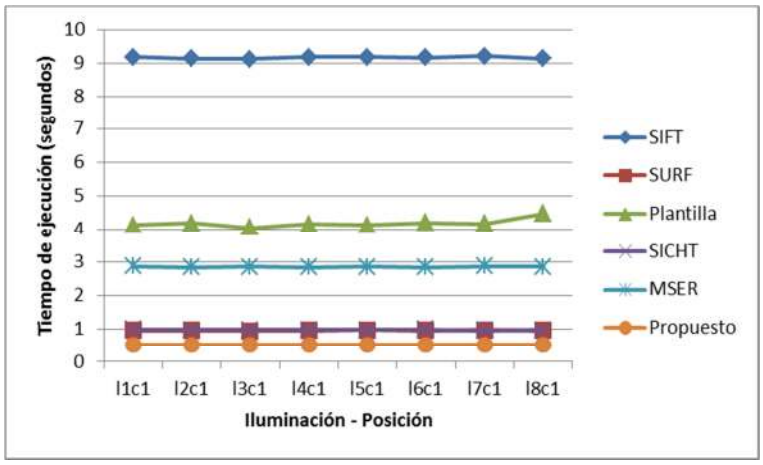
(b)



(c)



(d)

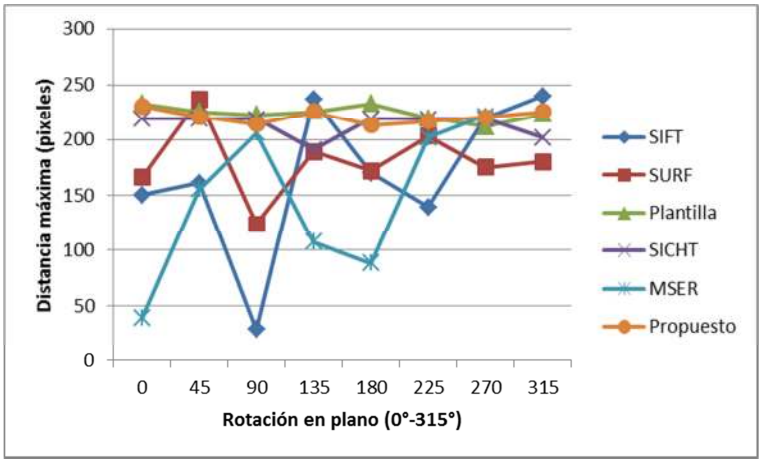


(e)

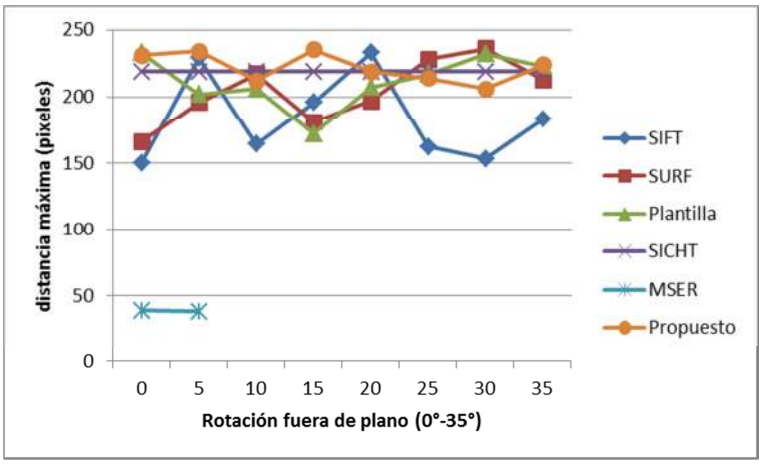
Figura 28: Tiempo de ejecución, en segundos, de los algoritmos evaluados en: (a) rotación en plano, (b) rotación fuera de plano, (c) escala, (d) tolerancia a ruido aditivo blanco y (e) dirección de iluminación.

La Figura 29 muestra la distancia máxima bajo la cual se encontró una correspondencia bajo los distintos algoritmos. En todos los casos la distancia máxima representa la correspondencia de dos imágenes en solo un fragmento o solo unos pixeles de la ventana completa, lo cual nos indica que los algoritmos son capaces de encontrar objetos con poca información. En el caso de los algoritmos basados en características, solo unos cuantos pixeles con información útil son necesarios para localizar un objeto; mientras que en los algoritmos basados en área, aunque las comparaciones no sean precisas, el objeto a buscar influye de manera significativa en la generación de histogramas. El algoritmo propuesto muestra constancia en todos los experimentos, este tipo de algoritmos mantienen un compromiso entre la velocidad de procesamiento y la precisión de localización dependiendo de cuántos pixeles de diferencia haya entre los pasos de la ventana deslizante.

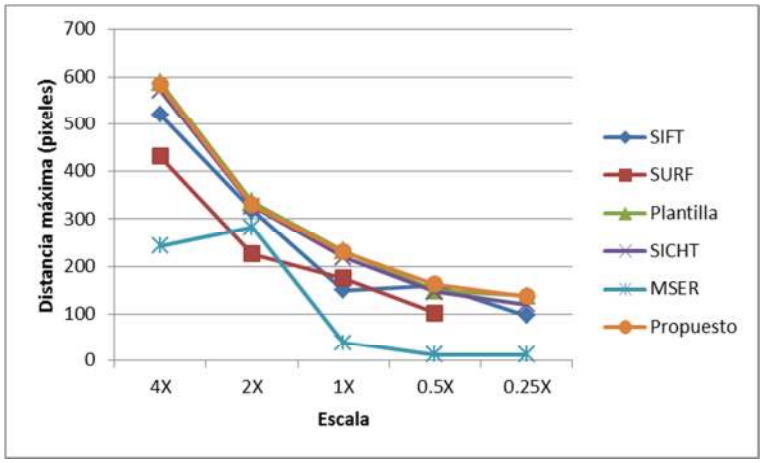
El algoritmo MSER+SIFT se muestra con muy bajo desempeño en todos los criterios evaluados, lo cual nos muestra que no es fácil combinar dos algoritmos de diferentes naturalezas para trabajar juntos. Si bien el algoritmo MSER extrae información de área, es posible que sea más eficiente tratar toda esta información en conjunto, ya que al traducir toda esta información a un solo punto central que posiblemente no sea ni siquiera un pixel con las características adecuadas no se extraen descriptores SIFT que cumplen con los parámetros de invarianza que caracterizan al algoritmo original.



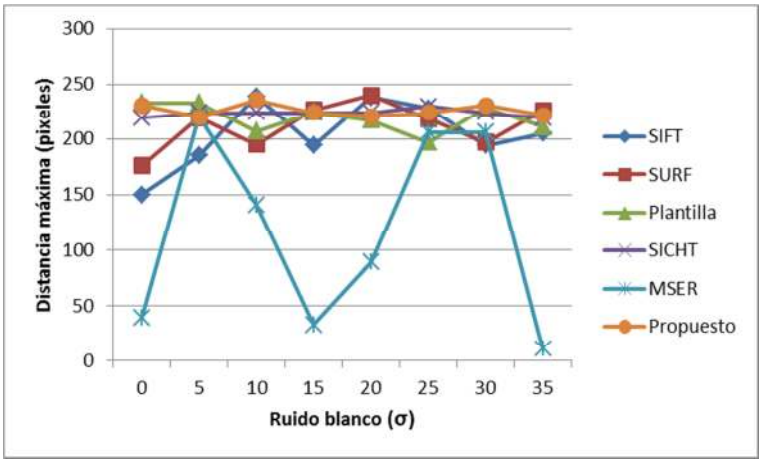
(a)


250
200
150
100
50
0
distancia máxima (pixeles)
0 5 10 15 20 25 30 35
Rotación fuera de plano (0°-35°)
SIFT
SURF
Plantilla
SICHT
MSER
Propuesto

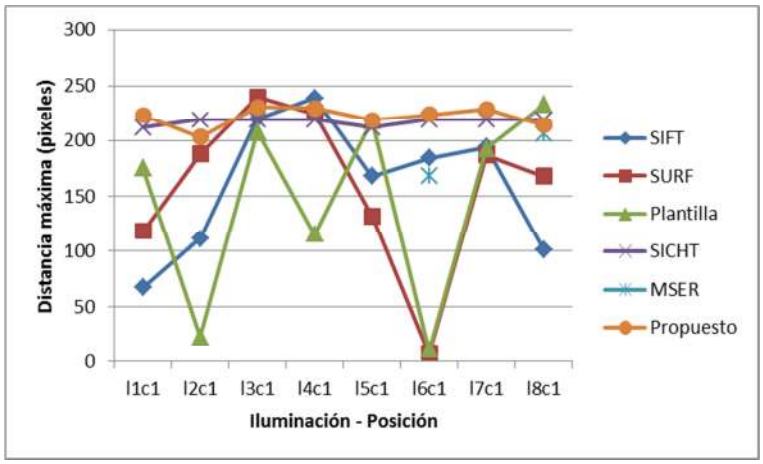
(b)


700
600
500
400
300
200
100
0
Distancia máxima (pixeles)
4X 2X 1X 0.5X 0.25X
Escala
SIFT
SURF
Plantilla
SICHT
MSER
Propuesto

(c)


300
250
200
150
100
50
0
Distancia máxima (pixeles)
0 5 10 15 20 25 30 35
Ruido blanco (σ)
SIFT
SURF
Plantilla
SICHT
MSER
Propuesto

(d)



(e)

Figura 29: Distancia máxima entre el centro real de la imagen de búsqueda y el centro de la imagen encontrada respecto a: (a) rotación en plano, (b) rotación fuera de plano, (c) escala, (d) tolerancia a ruido aditivo blanco y (e) dirección de iluminación.

V.2.2 Experimento 2

Este experimento tiene como objetivo buscar un fragmento de una escena dentro de ésta desconociendo su posición. Las escenas en este caso son imágenes satelitales de diversas fuentes como Geo-Eye1, IKONOS, QuickBird, WorldView y WorldView2, las cuales fueron recortadas a un tamaño de 1663×965 pixeles cada una; los fragmentos que se buscan en estas imágenes son de 200×150 pixeles. En la Figura 30 se muestra un ejemplo de las imágenes satelitales que se utilizan.

Se realizan 100 iteraciones por cada una de las 10 imágenes de satélite seleccionadas; se seleccionan 100 posiciones aleatoriamente de donde se extraen los fragmentos o parches de las imágenes de satélite, un ejemplo de estos parches se muestra en la Figura 31, a cada uno de éstos se les aplicará rotación en plano o se les añadirá ruido aditivo o se escalarán para posteriormente ser buscadas en la imagen de satélite. De la misma manera que el experimento anterior, al conocerse la posición real del centro del fragmento es posible calcular la distancia en pixeles entre la posición real y la posición encontrada como correspondiente por los diversos algoritmos.

Figura 30: Ejemplo de imagen satelital utilizada en el segundo experimento. Imagen recuperada de http://emap-int.com/.

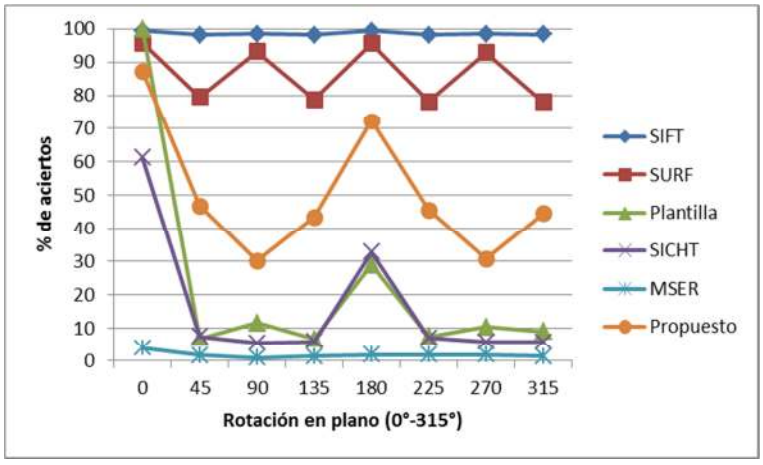
Figura 31: Ejemplo de uno de los 100 fragmentos que se extraen por imagen satelital para ser ubicados en éstas.

Resultados

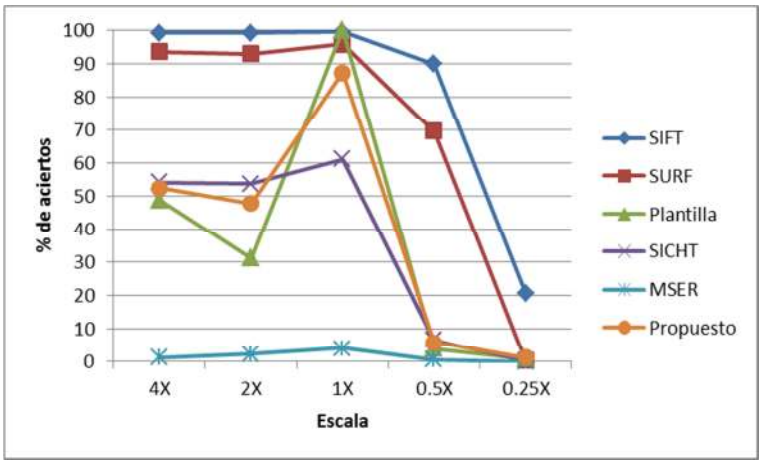
En la Figura 32 se presenta el porcentaje de aciertos de los parches en imágenes satelitales, como puede observarse SIFT y SURF continúan teniendo un

desempeño formidable en comparación a los demás algoritmos; sin embargo se aprecia de inmediato la mejoría de los algoritmos basados en área en este tipo de situaciones, especialmente en el algoritmo de plantilla en los casos donde no existe rotación, ruido aditivo ni escalamiento, donde localiza el 100% de los parches; SICHT también mejora significativamente en estas situaciones, sin embargo su desempeño baja cuando existen cambios de escala, en especial cuando se busca un parche con menores dimensiones.

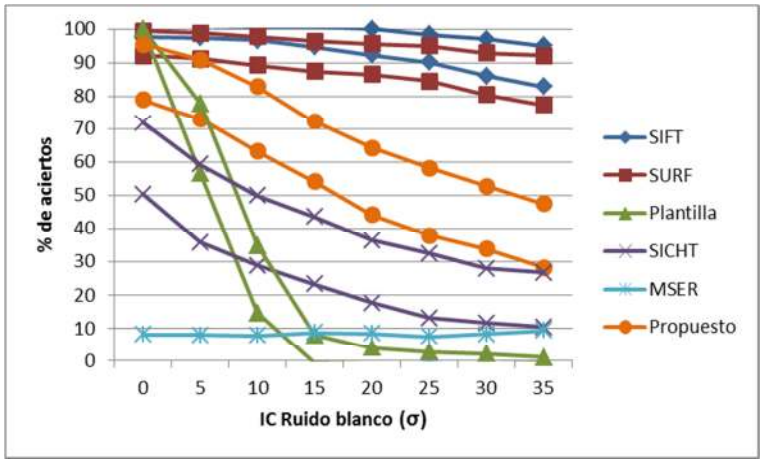
Por otro lado, el desempeño del algoritmo propuesto muestra un nivel de invarianza a rotación aceptable, incluso superior a SICHT a lo largo de las rotaciones; es de notarse que la mayor cantidad de aciertos se obtienen al no rotarse la imagen o hacerlo 180°, sin embargo, baja gradualmente obteniendo sus mínimos a los 90° y 270°, por lo tanto esta es sin duda un área de oportunidad para seguir mejorando el algoritmo. En cuanto a la invarianza a escala, el algoritmo obtiene resultados aceptables cuando la imagen es 2 ó 4 veces más grande que la muestra, pero baja para escalas pequeñas; se considera que debe estudiarse más a fondo. Por último, la tolerancia a ruido blanco aditivo es satisfactoria, el pre-filtraje por medio del vecindario adaptativo EV ayuda a reducir el ruido en la imagen brindando una muestra más cercana a la imagen real y por lo tanto con mayor probabilidad de éxito.



(a)



(b)



(c)

Figura 32: Desempeño de los algoritmos de correspondencia evaluados en: (a) rotación en plano, (b) escala y (c) tolerancia a ruido aditivo blanco.

Sin embargo, se muestra nuevamente que algoritmos como SIFT y el algoritmo de plantillas, pese a su buena respuesta en este experimento en sus casos particulares, muestran un aumento significativo en sus tiempos de ejecución como se muestra en la Figura 33. SURF y SICHT se ejecutan con mucha más rapidez que los demás algoritmos, teniendo mejores resultados el primero. Es importante notar en este punto que estos algoritmos son consistentes en sus tiempos de ejecución excepto el algoritmo de plantillas que tiene grandes variaciones en presencia de ruido blanco aditivo y más aún con cambios de escala. El algoritmo propuesto es el algoritmo más rápido entre todos los algoritmos evaluados, siendo su tiempo de ejecución constante en todas las pruebas realizadas.

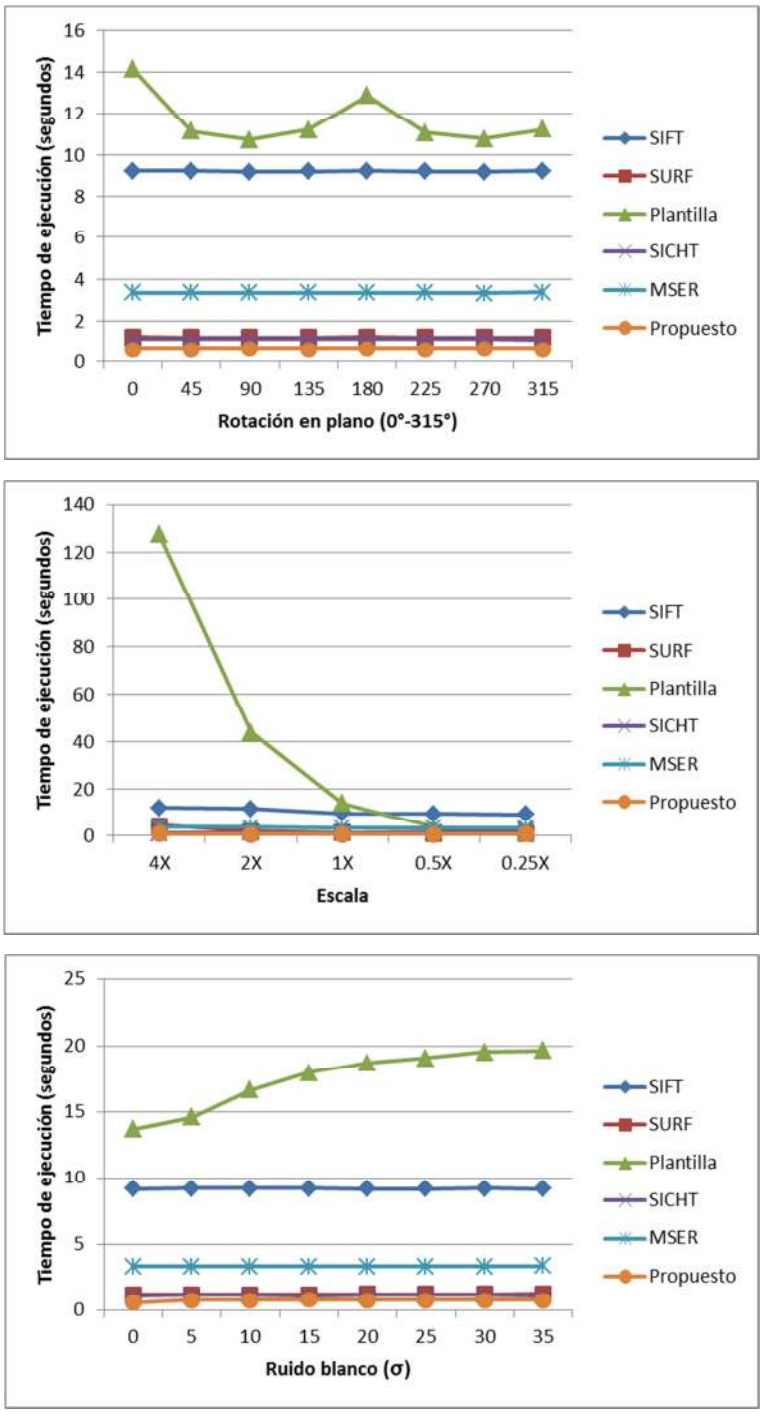
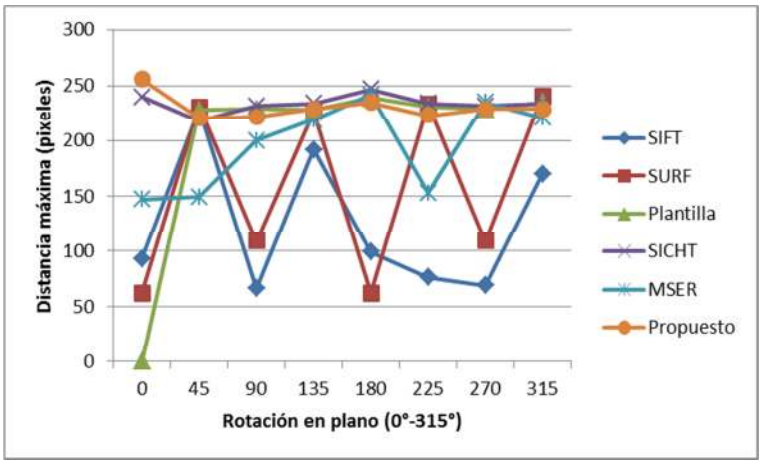
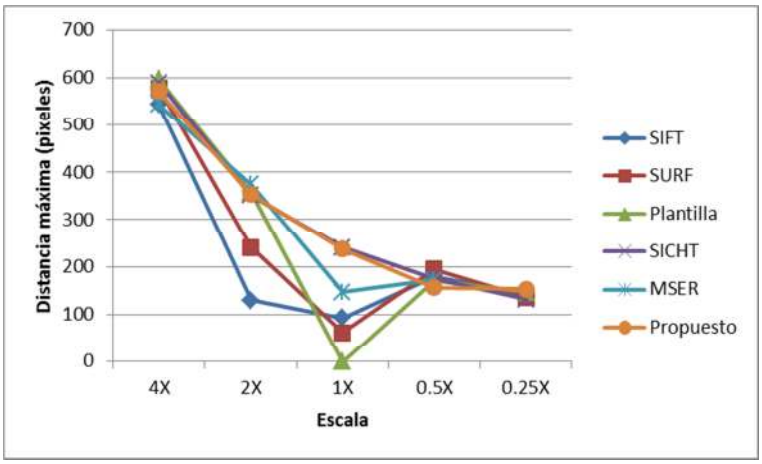


Figura 33: Tiempo de ejecución, en segundos, de los algoritmos evaluados en: (a) rotación en plano, (b) escala y (c) tolerancia a ruido blanco aditivo.

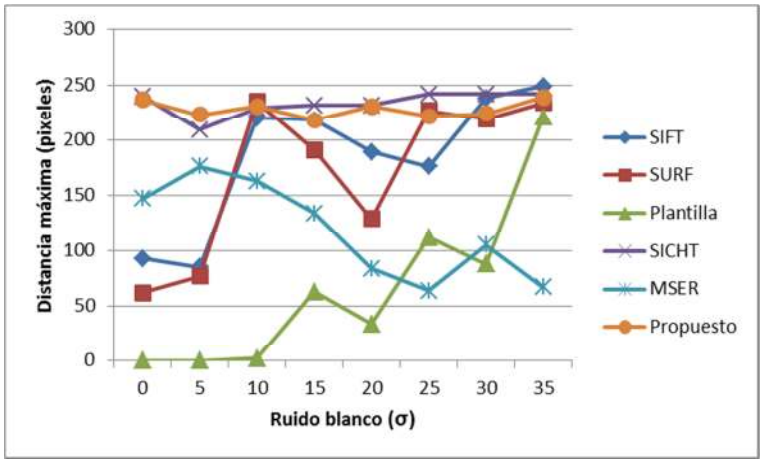
Finalmente en la Figura 34 se muestra la distancia máxima entre el centro real del parche en la imagen de escena y el centro del parche encontrado por los algoritmos. La realidad es que no es posible determinar una tendencia de distancia máxima para ningún algoritmo, solo pueden hacerse algunas observaciones. En el caso del algoritmo de plantillas se observa que tiene un registro perfecto cuando no existe rotación ni escala y es tolerante al ruido blanco cuando su desviación estándar es menor o igual a 10. Aunque no debería considerarse como una tendencia puede observarse que SIFT y SURF disminuyen su distancia máxima cuando la rotación en plano se encuentra en {0°, 90°, 180°, 270°} y aumenta en las demás rotaciones. El algoritmo propuesto muestra un comportamiento constante, aunque sus distancias máximas están entre las mayores de entre todos los algoritmos. Esto puede deberse a la forma de procesar las ventanas deslizantes en el algoritmo, donde se procesa pixel a pixel recorriendo la imagen hacia abajo pero a lo ancho se toman ventanas no traslapadas; este criterio podría reducirse al traslapar las ventanas deslizantes o al realizar escaneos locales en áreas extendidas en los puntos donde se obtiene la correspondencia.



(a)



(b)



(c)

Figura 34: Distancia máxima entre el centro real de la imagen de búsqueda y el centro de la imagen encontrada respecto a: (a) rotación en plano, (b) escala, (c) tolerancia a ruido blanco aditivo.

V.3 Algoritmo propuesto

Como se mostró en las secciones anteriores el algoritmo propuesto muestra ventajas sobre SICHT tanto en porcentaje de aciertos como en tiempo de procesamiento. El hecho de utilizar la magnitud de gradiente para la construcción del histograma de gradiente orientado permite al algoritmo delimitar la información realmente útil que pertenece a bordes y esquinas de la información que pertenece a texturas y pudiera considerarse como ruido.

El algoritmo tiene un tiempo de ejecución promedio de 0.57 segundos en imágenes de 1000×1000 pixeles aproximadamente, lo cual nos lleva a pensar que en imágenes más pequeñas su desempeño podría aumentar significativamente. Es necesario investigar las causas de los grandes valores en distancia máxima, una posible solución para reducir ese valor es realizar una búsqueda local en un área ampliada al doble del tamaño de la imagen de búsqueda.

A continuación se muestra la precisión de localización del algoritmo propuesto con una confianza del 95%, se realizaron 100 iteraciones en cada una de las 10 imágenes por experimento; como la media arroja un valor entre 0 y 1, es posible traducir este valor a porcentaje y establecer el intervalo de confianza en porcentaje de aciertos. En las Tabla III y Tabla ***IV*** se muestran los intervalos de confianza de aciertos para el experimento 1 y el experimento 2 respectivamente.

V.4 Resumen

En este capítulo se mostraron los dos experimentos que se llevaron a cabo para comparar los algoritmos estudiados y el algoritmo propuesto en esta tesis. Aunque ambos experimentos se centran en ubicar un objeto o un parche de imagen es importante recalcar que las condiciones no fueron las mismas; por un lado en el primer experimento se contó con un banco de imágenes más rico en características como la iluminación y la rotación fuera de plano, además de contar con objetos segmentados con los cuales se sabía de antemano que los algoritmos basados en área iban a tener dificultades para procesar por la adición de textura de fondo en las comparaciones o histogramas generados. Mientras que el segundo experimento se esperaba más favorecedor para los algoritmos basados en área no lo fue en realidad, solo en ciertos casos donde no existía un alto grado de deformaciones geométricas.

Aunque el algoritmo propuesto no estuvo a la altura de SIFT y de SURF, sí se cumplió el objetivo de obtener una mejora sustancial sobre el algoritmo sobre el

cual se basó su construcción, SICHT. Estas mejoras se debieron a los varios cambios y consideraciones que se hicieron sobre la construcción de los histogramas de gradiente orientado y el pre-filtraje de ruido blanco aditivo.

De entre los algoritmos estudiados y de acuerdo a los experimentos mostrados anteriormente es fácil deducir que el algoritmo más balanceado entre precisión y tiempo de procesamiento es el algoritmo denominado SURF.

Tabla III. Intervalos de confianza al 95% del porcentaje de aciertos del Experimento 1 para el algoritmo propuesto.

Rotación en plano	0° 12-25%	45° 13-28%	90° 10-23%	135° 13-30%	180° 13-29%	225° 13-29%	270° 12-24%	315° 13-31%
Rotación fuera de plano	0° 12-25%	5° 11-24%	10° 8-22%	15° 8-20%	20° 9-20%	25° 10-22%	30° 8-21%	35° 7-19%
Escala	4X 51-57%	2X 20-25%	1X 16-21%	0.5X 1-3%	0.25X 0-0.6%			
Tolerancia a ruido blanco	σ = 0 16-21%	σ = 5 17-22%	σ = 10 14-18%	σ = 15 13-18%	σ = 20 14-19%	σ = 25 14-18%	σ = 30 12-18%	σ = 35 11-15%
Cambio de iluminación	Pos(l1c1) 6-19%	Pos(l2c1) 9-22%	Pos(l3c1) 12-26%	Pos(l4c1) 12-24%	Pos(l5c1) 9-22%	Pos(l6c1) 5-15%	Pos(l7c1) 12-27%	Pos(l8c1) 12-26%

Tabla IV. Intervalos de confianza al 95% del porcentaje de aciertos del Experimento 2 para el algoritmo propuesto.

Rotación en plano	0° 79-95%	45° 37-56%	90° 21-39%	135° 34-53%	180° 64-80%	225° 35-56%	270° 22-40%	315° 34-54%
Escala	4X 39-65%	2X 36-59%	1X 79-95%	0.5X 1-10%	0.25X 0-3%			
Tolerancia a ruido blanco	σ = 0 79-95%	σ = 5 73-91%	σ = 10 63-83%	σ = 15 54-72%	σ = 20 44-64%	σ = 25 38-58%	σ = 30 34-53%	σ = 35 28-47%

Conclusiones

Durante décadas se han desarrollado algoritmos que intentan discernir entre un objeto y el fondo que lo rodea. Con esta idea en mente se construyeron algoritmos de detección de bordes y esquinas, los cuales han llegado a ser la base o a formar parte de los modernos algoritmos de correspondencia. Idealmente se espera que el objeto a localizar tenga bordes muy definidos que lo separen del fondo a su alrededor, sin embargo en la realidad los bordes no son captados como tales por el ojo humano ni por los sistemas de formación de imágenes, sino que nosotros diferenciamos dichos bordes mediante el cambio brusco de color entre un objeto y otro; en el caso de los sistemas de extracción de bordes y esquinas lo que se busca es el cambio de intensidad con respecto de uno o varios pixeles muestra.

El desarrollo de algoritmos de correspondencia en la actualidad se ha visto dividido en dos ramas principales y éstas en varias sub-ramas. Los algoritmos del método basado en área se han especializado más que nada en el procesamiento sobre imágenes satelitales y aéreas, es decir, imágenes donde se trata de ubicar toda la información concentrada en la imagen de búsqueda; por otro lado, los algoritmos del método en características se especializan en la ubicación de objetos en una escena pues construyen descriptores de características con el vecindario de pixeles sobresalientes en los objetos, lo cual les proporciona mayor flexibilidad y tolerancia al fondo de la escena. Esto no implica que los algoritmos no se puedan utilizar en todos los casos, sin embargo existe un compromiso entre la eficiencia del algoritmo ya sea en precisión, recursos utilizados o tiempo de ejecución.

En los experimentos realizados puede llegarse a la conclusión de que el algoritmo más balanceado es SURF pues su precisión y velocidad de procesamiento en cada uno de los criterios evaluados son de los mejores de entre todos los algoritmos, por lo cual se recomendaría su uso como un algoritmo de correspondencia de carácter general. Sin embargo, si el tiempo de respuesta no

es muy estricto, SIFT es el mejor algoritmo desarrollado hasta la fecha, con una precisión casi perfecta en la mayoría de los criterios evaluados.

En esta tesis también se estudió un algoritmo híbrido, el cual no presentó buenos resultados en ninguno de los criterios evaluados en ninguno de los experimentos, esto demuestra la dificultad de combinar los dos tipos de algoritmos encontrados en la literatura en la búsqueda de un algoritmo que tome las mejores características de ambos mundos. El algoritmo propuesto es híbrido pues tiene en mente algunas ideas presentes en los algoritmos basados en características así como el procesamiento por ventanas encontrado en los algoritmos basados en área.

El algoritmo propuesto aunque no resaltó con resultados espectaculares si demostró mejoras sustanciales sobre el algoritmo en el cual estuvo basado, SICHT. Debido al tamaño de las imágenes de prueba en los experimentos se debió sacrificar en cierto grado la precisión para ganar velocidad de procesamiento; creemos que con imágenes de prueba más pequeñas como se hace en varios artículos estudiados es posible aumentar el desempeño del algoritmo; sin embargo, uno de los objetivos de este trabajo de tesis era construir un algoritmo que fuese eficiente en imágenes de gran tamaño.

Así pues, al finalizar el desarrollo y evaluación del algoritmo aquí propuesto se logró obtener una implementación que rivaliza fuertemente con algoritmos basados en características como SIFT y SURF en áreas importantes como rotación, tolerancia a ruido aditivo y escalas mayores a la imagen muestra. Aunque es cierto que aún es posible hacer mejoras en todos estos criterios, se obtuvo un algoritmo lo suficientemente bueno y rápido para imágenes de gran tamaño y que se espera sea aún mejor para imágenes más pequeñas. Como se ha mencionado anteriormente, la mayor contribución de este trabajo de tesis es el haber obtenido un algoritmo que supera en todas las áreas al algoritmo SICHT en el cual se basó obteniendo mejores resultados en general en todas las áreas evaluadas y con mejor tiempo de procesamiento.

Trabajo futuro

Una vez habiendo concluido el estudio comparativo y la creación de un algoritmo de correspondencia que aprovecha cualidades tanto de los algoritmos basados en área como de los basados en características, aún existen varias líneas de investigación con las que se puede trabajar.

Una primera propuesta es continuar el trabajo sobre el algoritmo propuesto para mejorar su desempeño en áreas donde se tuvo una baja precisión como la invarianza a escala cuando se reduce mucho el área u objeto a evaluar. Es necesario continuar trabajando en el algoritmo para hacerlo más robusto en el área de localización de objetos y evaluarlo en el área de reconocimiento de rostros.

Otra línea de investigación interesante y aprovechando el trabajo derivado de esta tesis es el seguimiento de objetos en secuencias de video, aplicando el algoritmo propuesto como primer paso en un algoritmo de seguimiento lo cual se pretende sea muy rápido en secuencias de video normales y muy aceptable en secuencias de video de alta definición, pues este algoritmo fue probado en imágenes casi tan grandes como un cuadro de video de alta definición.

Referencias bibliográficas

Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008). SURF: Speeded up robust features. *Computer Vision and Image Understanding*. *110*(3), 346-359

Chambon, S. and Crouzil, A. (2002). Mesures de correlation robustes aux occultations. *Actes du Congrès Francophone de Vision par Ordinateur ORASIS*. 385-392

Cootes, T. F., Edwards, G. J. and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *23*(6), 681-685

Cyganek, B. and Siebert, J. P. (2009). An introduction to 3D computer vision techniques and algorithms. Wiley. Chichester

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*. 886-893

Foo, J. J. and Sinha, R. (2007). Pruning SIFT for scalable near-duplicate image matching. *Proceedings of Australasian Database Conference*. 63-71

Gonzalez, R. C. and Woods, R. E. (2002). Digital image processing. Prentice Hall. 2a Edición. New Jersey

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings of the Fourth Alvey Vision Conference*. 147-151

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition*. *2*(2004), 506-513

Kober, V., Mozerov, M. and Álvarez Borrego, J. (2001). Nonlinear filters with spatially connected neighborhoods. *Optical Engineering*. *40*(6), 971-983

Li, J. and Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*. *71*(10-12), 1771-1787

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysis structures at different scales. *Journal of Applied Statistics*. *21*(2), 224-270

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal on Computer Vision*. *30*(2), 79-116

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision*. *2*(1999), 1150-1157

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision. 2*(2004), 91-110

Lu, Y., Hu, J. and Huang, D. (2006). Study on a Image matching algorithm based on sphere similarity of color histogram intersection. *Worlds Congress on Intelligent Control and Automation*. 9945-9948

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information. W. H. Freeman. New York

Matas, J., Chum, O., Urban, M. and Pajdla, T. (2004). Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*. *22*(2004), 761-767

Mikolajczyk, K. and Schmid, C.. (2001). Indexing based on scale invariant interest points. *International Conference on Computer Vision*. *1*(2001), 525-531

Sarvaiya, J. N., Patnaik, S. and Bombaywala, S. (2009). Image registration by template matching using normalized cross-correlation. *Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. 819-822

Schmid, C. and Mohr, R. (1997). Local gray value invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *19*(5), 530-534

Solomon, C. and Brekon, T. (2011). Fundamentals of digital image processing: A practical aproach with examples in Matlab. Wiley Blackwell. Oxford, UK

Sonka, M., Hlavac, V. and Boyle, R. (2008). Image processing, analysis, and machine vision. Thompson. 3a Edición. London

Tate, R. and Northern, J. (2008). Fast template matching system using VHDL. *IEEE Region 5 Conference 2008*. 1-5

Toews, M. and Wells, W. (2009). SIFT-Rank: Ordinal description for invariant feature correspondence. *Computer Vision and Pattern Recognition*. *1*(2009), 172-177

Witkin, A. P. (1983). Scale-space filtering. *Proceedings of the 8th Joint Conference on Artificial Intelligence*. *1*(1983), 1019-1022

Yang, L., Wang, R., Ge, P. and Cao, F. (2009). Research on area-matching algorithm based on feature-matching constraints. *International Conference of Neural Computation*. 208-213

Zalesky, B. A. and Lukashevich, P. V. (2011). Scale invariant algorithm to match regions on aero or satellite images. *Proceadings of Pattern Recognition and Information Processing 11*. 25-30

Zhang, Z., Deriche, R., Faugeras, O. and Luong, Q. T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*. *78*, 87-119

Zhou, X. and Dorrer, E. (2000). Non-linear scale and orientation free correlation matching algorithm based on edge correspondence. *International Society for Photogrammetry and Remote Sensing*. *33*(2000), 1054-1062

Apéndice A

En este apéndice se presenta la relación existente entre el Laplaciano de Gaussianas (LoG) y la Diferencia de Gaussianas (DoG). Estos dos operadores son comúnmente utilizados como detectores de características.

El operador de Laplace se define como

$$\nabla^2 I(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}. \tag{76}$$

En el caso de una señal, su primera derivada resalta un valor extremo, mientras que su segunda derivada, como el operador de Laplace, resalta dos valores extremos con diferente signo, lo cual implica un valor cero entre éstos dos como se muestra en la Figura 35.

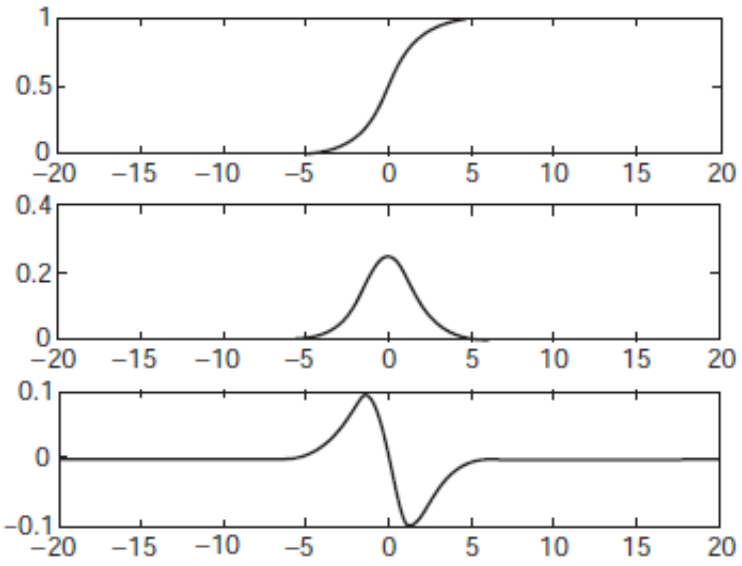


Figura 35: De arriba hacia abajo: señal de entrada, su primera derivada, su segunda derivada. Imagen recuperada de Cyganek y Siebert, 2009. 121 p.

En el caso de imágenes, la falla del operador de Laplace se da en su susceptibilidad al ruido, el cual es ubicuo en las imágenes. Luego, es necesario

limitar el nivel de ruido en la señal de entrada. Esto puede hacerse al filtrar la señal de entrada previamente con un filtro pasa-bajas como el filtro Gaussiano. La conexión entre el operador de Laplace y el filtro Gaussiano es llamado *Laplaciano de Gaussianas (LoG)*. Matemáticamente esta idea se expresa como

$$\nabla^2(G * I), \tag{77}$$

donde $G(x, y, \sigma)$ es una función Gaussiana de dos dimensiones dada por la ecuación (58). En el caso de una función continua puede expresarse de la siguiente forma

$$\nabla^2(G * I) = (\nabla^2 G) * I, \tag{78}$$

lo cual implica una conexión de la operación de suavizado del filtro Gaussiano con la diferencial de segundo orden del operador de Laplace en un solo operador; luego, el operador conjunto se aplica a la imagen de entrada. Esta conexión lleva a la siguiente expresión

$$\nabla^2 G(x, y, \sigma) = -\frac{1}{2\pi\sigma^4}\left(2 - \frac{x^2 + y^2}{\sigma^2}\right) e^{-x^2+y^2/2\sigma^2}. \tag{79}$$

Este operador en dos dimensiones puede descomponerse en una combinación de dos operadores de una dimensión de la siguiente manera

$$g(x, y) = g_1(x)g_2(y) + g_2(x)g_1(y), \tag{80}$$

donde

$$g_1(t) = -\frac{1}{2\pi\sigma^4}\left(1 - \frac{t^2}{\sigma^2}\right) e^{-t^2/2\sigma^2}, \quad g_2(t) = e^{-t^2/2\sigma^2}. \tag{81}$$

El Laplaciano de Gaussianas es un punto importante en la teoría del espacio de escalas como fue demostrado por Lindeberg (1994), probando que el LoG normalizado por σ^2 *(ssLoG)* dado por

$$\sigma^2\nabla^2 G(x, y, \sigma) = -\frac{1}{2\pi\sigma^2}\left(2 - \frac{x^2 + y^2}{\sigma^2}\right) e^{-x^2+y^2/2\sigma^2}, \tag{82}$$

es un requisito para obtener una verdadera invarianza a escala. Puede mostrarse que ssLoG está estrictamente relacionado con la *Diferencia de Gaussianas (DoG)*;

como fue demostrado por Lowe (2004), tomando la ecuación de difusión de calor e intercambiando el parámetro de tiempo t por σ, obtenemos

$$\frac{\partial G(x,y,\sigma)}{\partial \sigma} = \sigma \nabla^2 G(x,y,\sigma), \tag{83}$$

en donde el lado izquierdo de la ecuación puede ser aproximado por

$$\frac{\partial G(x,y,\sigma)}{\partial \sigma} = \lim_{\Delta\sigma \to 0} \frac{G(x,y,\sigma+\Delta\sigma) - G(x,y,\sigma)}{\Delta\sigma}. \tag{84}$$

Remplazando $\Delta\sigma = \sigma(k-1)$ en la ecuación anterior obtenemos

$$\frac{\partial G(x,y,\sigma)}{\partial \sigma} = \lim_{k \to 1} \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{\sigma(k-1)}. \tag{85}$$

Luego, puede aproximarse como

$$(k-1)\sigma^2 \nabla^2 G(x,y,\sigma) \approx G(x,y,k\sigma) - G(x,y,\sigma), \tag{86}$$

donde el lado derecho de la ecuación es la Diferencia de Gaussianas (DoG) definida como

$$D(x,y,\sigma) = G(x,y,k\sigma) - G(x,y,\sigma). \tag{87}$$

Luego, puede verse que $\sigma^2 \nabla^2 G(x,y,\sigma) \sim D(x,y,\sigma)$ para k suficientemente cerca de 1. (Cyganek y Siebert, 2009, 120-126 p.).

Apéndice B

Los histogramas de gradiente orientado (HoG[2]) (Dalal y Triggs, 2005) son descriptores utilizados para la detección de objetos en procesamiento de imágenes y visión por computadora. Esta técnica cuenta las ocurrencias de orientaciones de gradiente en porciones localizadas de la imagen.

La idea detrás de los histogramas de gradiente orientado es que la apariencia local y forma de los objetos en una imagen pueden ser descritas mediante la distribución de la dirección de los gradientes. La implementación de este descriptor consiste en dividir la imagen en varias ventanas o celdas, y por cada celda calcular un histograma en el cual cada compartimiento o *bin* es un ángulo de orientación de gradiente, la combinación de varios de estos histogramas forman el descriptor. Para aumentar la efectividad del descriptor los histogramas pueden ser normalizados.

La generación de los histogramas de gradiente orientado sigue los siguientes pasos:

1. **Calcular el gradiente de la imagen.** El procedimiento más simple para procesar los gradientes es utilizar un filtro de derivación sencillo de una dimensión vertical y horizontalmente que se muestra en la ecuación (88). Sin embargo es posible utilizar otros operadores de gradientes más complejos como el operador de Roberts, el operador de Prewitt o el operador de Sobel.

$$[-1, 0, 1] \ y \ [-1, 0, 1]^T. \tag{88}$$

2. **Acumulación de orientaciones.** Se genera un histograma de 0 a 180 ó de 0 a 360 compartimientos en los cuales cada pixel *emite su voto* de acuerdo a la orientación derivada del cálculo del gradiente (Figura 36).

[2] Del inglés Histogram of Oriented Gradient.

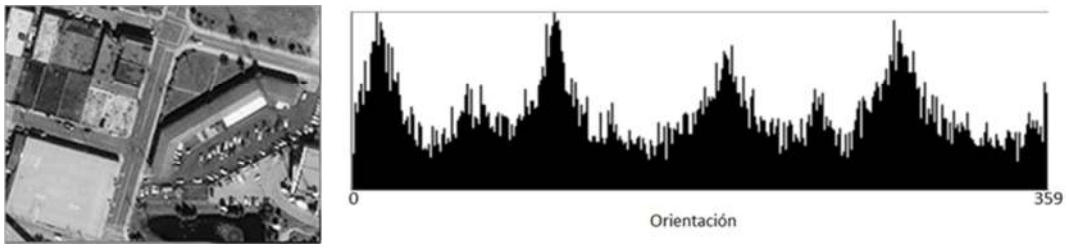


Figura 36: Histograma de gradiente orientado de 360 compartimientos.

3. **Creación de bloques.** Es posible crear descriptores más complejos al crear vectores de histogramas de gradiente orientado de celdas contiguas. Existen dos tipos de descriptores de este tipo, los HoG rectangulares (R-HoG) y los HoG circulares (C-HoG) (Figura 37).

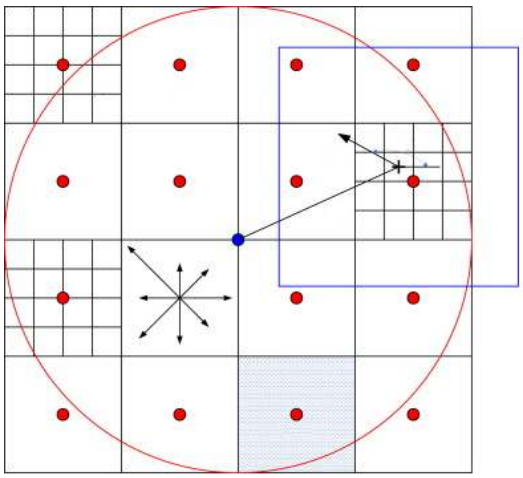


Figura 37. Azul ejemplo de R-HoG, rojo ejemplo de C-HoG. Imagen recuperada de Li y Allinson, 2008. 1780 p.

4. **Normalización de histogramas.** Al normalizar el histograma de gradiente orientado se llega a tener un mejor desempeño a la hora de realizar las comparaciones de histogramas.