



## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**“DETECCIÓN Y DESCRIPCIÓN DE PUNTOS CARACTERÍSTICOS EN IMÁGENES  
MULTIESPECTRALES UTILIZANDO ESQUEMAS CLÁSICOS”**

### **TESINA DE SEMINARIO**

**Previo a la obtención del título de:**

**INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS  
MULTIMEDIA**

**PRESENTADO POR:**

**WILFRIDO ROLANDO GUERRERO ALBÁN**

**GUAYAQUIL – ECUADOR**

**2014**

## **AGRADECIMIENTO**

A Dios por darme las fuerzas, la sabiduría y la vida para poder llevar adelante este proyecto.

A mis padres, hermano y todas aquellas personas que siempre confían en mí y en mi trabajo.

Al PHD Ángel D. Sappa director del presente proyecto, por su invaluable ayuda y colaboración en el desarrollo de este trabajo.

## **DEDICATORIA**

A mis padres, por todo el apoyo  
brindado durante toda mi vida.

A mi hermano por ser el ejemplo  
de vida en lo personal y  
profesional.

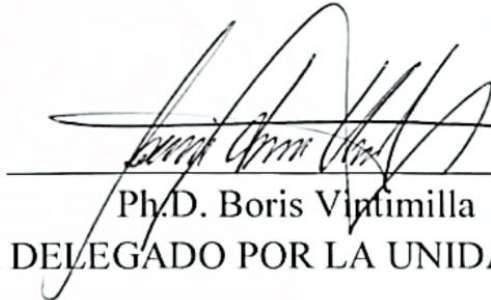
A mi novia quien en todo momento  
ha sabido darme fuerzas para  
seguir adelante.

**TRIBUNAL DE SUSTENTACIÓN**



---

Ph.D. Angel D. Sappa  
PROFESOR DEL SEMINARIO DE GRADUACIÓN



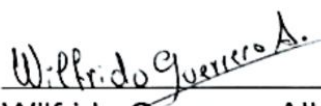
---

Ph.D. Boris Vintimilla  
PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ECUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL)

  
\_\_\_\_\_  
Wilfrido Guerrero Albán

## RESUMEN

En la actualidad existen algoritmos que pueden detectar y describir puntos característicos en imágenes visibles, permitiendo así detectar un objeto específico dentro de una escena. Para esto existen los denominados algoritmos clásicos como el SIFT, SURF, FAST, etc.

Estos algoritmos proporcionan un resultado muy favorable al momento de ser ejecutados en imágenes del espectro visible, pero que sucede si estos algoritmos son aplicados a imágenes de tipo infrarroja?

En este documento se investigará el comportamiento de estos esquemas clásicos al usarlos en imágenes infrarrojas, de tal manera que se pueda obtener un resultado, también, favorable para imágenes de este tipo.

Para esto se procederá a buscar con criterio las variantes que se puedan ejecutar a los algoritmos anteriormente mencionados, para así obtener un resultado, que nos permita concluir de manera favorable el uso de estos esquemas clásicos, para este tipo de imágenes. Otro de los aspectos en que se trabajará, es en el de pre-procesar la imagen antes de aplicar cualquiera de estos algoritmos, esperando así, que esto ayude de mejor manera a la detección y descripción de puntos característicos en imágenes infrarrojas.

# ÍNDICE GENERAL

	Pág.
RESUMEN.....	III
ÍNDICE GENERAL.....	IV
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	VIII
INTRODUCCIÓN.....	IX
CAPÍTULO 1	
ANÁLISIS DEL CASO	
1.1. Análisis del problema.....	1
1.2. Análisis de la solución.....	4
1.3. Objetivos generales.....	4
1.4. Objetivos específicos.....	5
1.5. Justificación.....	5

## CAPÍTULO 2

### REVISIÓN BIBLIOGRÁFICA

2.1. Detección de puntos de interés.....	7
2.2. Descripción de puntos de interés.....	10
2.3. Esquemas clásicos de detección y descripción.....	10
2.3.1.SIFT como detector.....	11
2.3.2. SIFT como descriptor.....	17
2.3.3. SURF como detector.....	19
2.3.4.SURF como descriptor.....	25

## CAPÍTULO 3

### INSTALACIÓN Y CONFIGURACIÓN DEL EQUIPO

3.1. Características del equipo.....	27
3.2. Herramientas de desarrollo.....	28
3.3. Instalación y configuración.....	29

## CAPÍTULO 4

### DISEÑO DEL SISTEMA EXPERIMENTAL



4.1. Interfaz gráfica del sistema.....	35
4.2. Manual de usuario.....	37

## CAPÍTULO 5

### ANÁLISIS EN IMÁGNES MULTIESPECTRALES

5.1. Comportamiento del algoritmo SIFT y SURF.....	42
5.2. Algoritmo SIFT.....	43
5.2.1 Detección de puntos de interés.....	44
5.2.2 Descripción de puntos.....	48
5.2.3 Variación en parámetro en algoritmo.....	52
5.3. Algoritmo SURF.....	59
5.3.1 Detección de puntos de interés.....	60
5.3.2 Descripción de puntos.....	64
5.3.3 Variación en parámetro en algoritmo.....	66

### CONCLUSIONES Y RECOMENDACIONES

### BIBLIOGRAFÍA

## **ABREVIATURAS**

VS	Imagen de espectro visible
IR	Imagen de espectro infrarrojo
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
FAST	Features from Accelerated Segment Test

## SIMBOLOGÍA

$L(x, y, \sigma)$	Función espacio-escala de una imagen
$G(x,y,\sigma)$	Función gaussiana de escala variable
$I(x,y)$	Imagen de entrada
$D(x, y, \sigma)$	Función diferencia de gaussianas
$H$	Matriz hessiana
$m(x,y)$	Magnitud de gradiente
$\theta(x, y)$	Orientación
$L_{xx}(\rho, \sigma)$	Convolución de la derivada parcial de segundo orden

## ÍNDICE DE FIGURAS

	Pág.
Figura 1.1	Imágenes en el espectro visible.....2
Figura 1.2	Imágenes en diferentes espectros VS-IR.....2
Figura 1.3	Detección de puntos en imágenes de espectro diferente.....3
Figura 1.4	Correspondencia de puntos entre dos imágenes.....4
Figura 1.5	Aplicaciones de sistemas multispectrales.....6
Figura 2.1	Esquema de un sistema de alineación.....8
Figura 2.2	Diferentes puntos detectados.....9
Figura 2.3	Espacio-escala propuesta por Lowe.....13
Figura 2.4	Proceso de comparación entre diferentes escalas.....14
Figura 2.5	Descripción de puntos de interés.....18
Figura 2.6	Representación de la intensidad de una región respecto de la imagen integral.....21
Figura 2.7	Derivadas parciales de segundo orden de un filtro y su aproximación.....22
Figura 2.8	Filtros de Haar.....24
Figura 2.9	Asignación de la orientación de cada sector.....25
Figura 2.10	Respuestas de Haar en las sub-regiones alrededor del punto de interés.....26
Figura 3.1	Configuración de las variables de entorno.....32

Figura 3.2	Carpeta Libraries de nuestro proyecto.....	33
Figura 3.3	Archivos agregados a la carpeta Libraries.....	34
Figura 4.1	Interfaz gráfica del sistema experimental.....	36
Figura 4.2	Pestañas con los algoritmos a seleccionar.....	37
Figura 4.3	Combo box con una lista de imágenes.....	38
Figura 4.4	Panel con las imágenes originales.....	38
Figura 4.5	Parámetros del algoritmo.....	39
Figura 4.6	Puntos detectados en ambas imágenes.....	39
Figura 4.7	Puntos de interés detectados.....	40
Figura 4.8	Imágenes con puntos similares.....	40
Figura 5.1	Ilustración de puntos encontrados.....	47
Figura 5.2	Vectores características para imagen VS.....	49
Figura 5.3	Vectores características para imagen IR.....	49
Figura 5.4	Matriz de distancia de los descriptores.....	50
Figura 5.5	Comparación de imágenes con puntos de correspondencia.....	52
Figura 5.6	Incremento de puntos de interés en imagen IR.....	56
Figura 5.7	Correspondencia de puntos para un valor del contrastThreshold menor a 0,04.....	58
Figura 5.8	Comparación de imágenes con sus puntos de interés.....	63
Figura 5.9	Comparación de imágenes con sus puntos de correspondencia.....	65
Figura 5.10	Detección de puntos de interés para un valor hessianThreshold menor a 500.....	71
Figura 5.11	Correspondencia de puntos para valores de contastThreshold menor a 500.....	74

## ÍNDICE DE TABLAS

	Pág.
TABLA 1	DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA UNA IMAGEN VS (SIFT) ..... 45
TABLA 2	DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA UNA IMAGEN IR (SIFT)..... 46
TABLA 3	DIFERENCIA EN LA DETECCIÓN DE PUNTOS DE INTERÉS ENTRE IMÁGENES VS-IR (SIFT)..... 47
TABLA 4	CORRESPONDENCIA DETECTADAS Y CORRESPONDENCIAS CORRECTAS USANDO LOS PARÁMETROS INICIALES (SIFT)... 51
TABLA 5	DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN VS CON VALOR CONTRASTTHRESHOLD= 0.04..... 53
TABLA 6	DETECCIÓN DE PUNTOS DE INTERÉS PARA UN AIMAGEN IR CON VALOR CONTRASTTHRESHOLD= 0.03 ..... 54
TABLA 7	DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR CON VALOR CONTRASTTHRESHOLD= 0.02 ..... 55
TABLA 8	DIFERENCIA ENTRE LA CANTIDAD DE PUNTOS DETECTADOS ENTRE LAS IMÁGENES VS-IR (SIFT) ..... 56
TABLA 9	CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS CORRECTAS ENTRE IMÁGENES VS-IR CON VALOR DE CONTRASTTHRESHOLD= 0.03..... 57
TABLA 10	CORRESPONDENCIAS DETECTADAS ENTRE IMÁGENES VS-IR CON VALOR DE CONTRASTTHRESHOLD= 0.02 ..... 58
TABLA 11	DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA UNA IMAGEN VR (SURF) ..... 61
TABLA 12	DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA UNA IMAGEN IR (SURF)..... 62

TABLA 13	DIFERENCIA EN LA DETECCIÓN DE PUNTOS DE INTERÉS ENTRE IMÁGENES VS-IR (SURF).....	62
TABLA 14	CORRESPONDENCIAS DETECTADAS ENTRE LAS IMÁGENES VS-IR (SURF).....	65
TABLA 15	DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA UNA IMAGEN VR (SURF) .....	67
TABLA 16	DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=300 .....	68
TABLA 17	DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=200 .....	69
TABLA 18	DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=200 .....	70
TABLA 19	DIFERENCIA EN LA DETECCIÓN DE PUNTOS ENTRE LAS IMÁGENES VS-IR (SURF) .....	71
TABLA 20	CORRESPONDENCIAS DETECTADAS ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=300 .....	72
TABLA 21	CORRESPONDENCIAS DETECTADAS ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=200 .....	73
TABLA 22	CORRESPONDENCIAS DETECTADAS ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD=100 .....	74

# INTRODUCCIÓN

Sin duda alguna, uno de los aspectos más estudiados en aplicaciones de Visión por Computador es la correspondencia entre dos imágenes, como por ejemplo el reconocimiento de objetos, patrones, seguimiento de objetos, entre otras.

El establecimiento de correspondencias se lo realiza en tres etapas: detección de puntos de interés, descripción de los mismos y puesta en correspondencia. Cabe recalcar que el presente trabajo de investigación se enfocará en las dos primeras etapas, Detección y Descripción.

Cuando se busca establecer correspondencias entre dos imágenes de distinta banda espectral los métodos convencionales dejan de ser confiables a medida que dichas bandas se alejan.

En este documento se expondrá la posibilidad de utilizar esquemas clásicos de Detección y Descripción, para obtener la correspondencia entre dos imágenes, variando con criterio los parámetros de cada uno de estos algoritmos, realizando así, una evaluación de cuan confiable puede resultar.



# **CAPÍTULO 1**

## **ANÁLISIS DEL CASO**

### **1.1. Análisis del problema**

Como hemos mencionado anteriormente existen algoritmos que permiten la detección y descripción de puntos entre imágenes en el espectro visible, como se muestra en la Figura 1.1; el problema que se nos presenta ahora, es evaluar la robustez cuando aplicamos estos algoritmos entre una imagen en el espectro visible y una imagen en el espectro infrarrojo como se muestra en la Figura 1.2.



Figura 1.1 Imágenes en el espectro visible



Figura 1.2 Imágenes en diferentes espectros VS-IR

Al aplicar los algoritmos de detección con sus parámetros iniciales, nos podremos dar cuenta que la cantidad de puntos de interés, o también llamados puntos característicos, en la imagen de espectro visible supera a la cantidad puntos característicos en la imagen de espectro infrarrojo como se muestra en la Figura 1.3, presentándose así nuestro primer problema,

hay mucha diferencia entre las cantidades de puntos característicos detectados.



Figura 1.3 Detección de puntos en imágenes de espectro diferente

Al momento de aplicar un algoritmo para describir los puntos de interés detectados anteriormente, y mediante una matriz de comparación de puntos entre las dos imágenes, observamos que pocos o ninguno de los puntos concuerdan entre sí, como se muestra en la Figura 1.4.

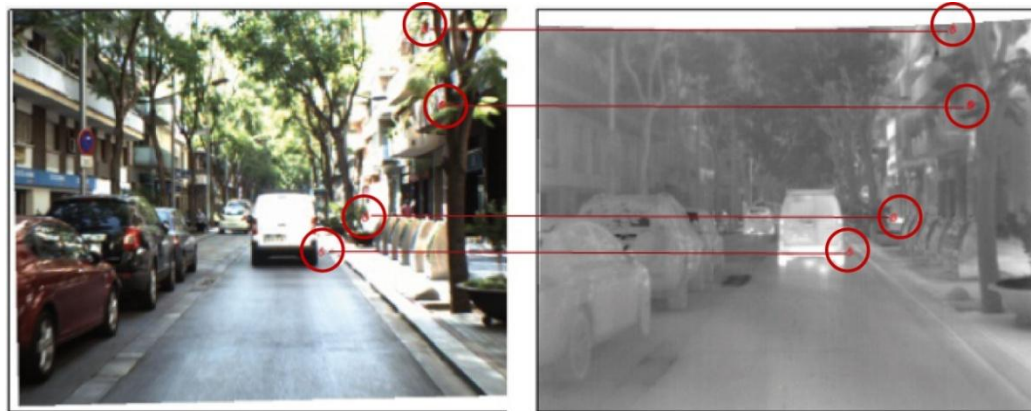


Figura 1.4 Correspondencia de puntos entre dos imágenes

## 1.2. Análisis de la solución

Como posible solución ante este problema se propone variar con criterio ciertos parámetros iniciales de los algoritmos en forma independiente en cada espectro, esperando así obtener menor diferencia entre el número de puntos característicos encontrados y a su vez mayor puntos de concordancia entre sí.

## 1.3. Objetivos generales

Evaluar el comportamiento de ciertos esquemas clásicos de detección y descripción de puntos de interés entre imágenes en el espectro visible e imágenes en el espectro infrarrojo.

#### **1.4. Objetivos específicos**

Determinar si es posible usar estos esquemas clásicos entre imágenes de espectros diferentes, y en tal caso determinar la metodología a seguir para obtener soluciones robustas.

#### **1.5. Justificación**

En la actualidad hay muchos campos donde se puede aplicar la visión por computador, y estos algoritmos nos ayudan al análisis para el reconocimiento de un objeto o de una escena. Uno de los principales motivos para el desarrollo de este documento es tratar de usar estos algoritmos para detectar y describir puntos de interés en imágenes de diferentes espectros, para así ayudar en el campo investigativo y seguir desarrollando aplicaciones que permitan un análisis multiespectral.

Cabe destacar que la coexistencia de cámaras trabajando en distintas bandas del espectro (VS-IR) está aumentando, pero en la mayoría de los casos, las imágenes multiespectrales se analizan y se utilizan de forma independiente. En la Figura 1.5 podemos ver un par de ilustraciones donde se muestran aplicaciones basadas en el procesamiento de imágenes multiespectrales.

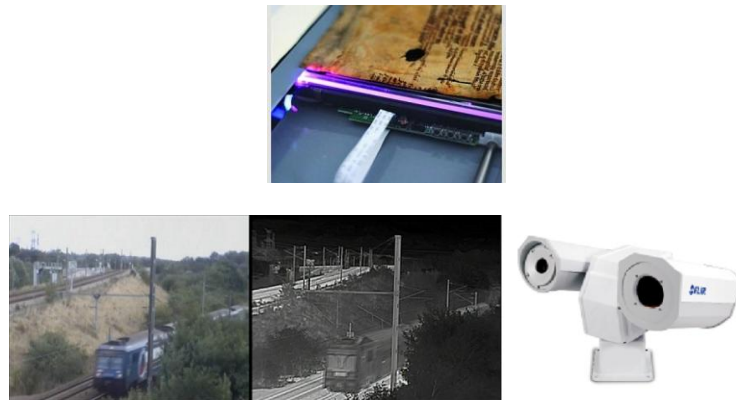


Figura 1.5 Aplicaciones de sistemas multispectrales: (superior) Escaner para análisis de documentos antiguos; (inferior) Sistema de videovigilancia.

# **CAPÍTULO 2**

## **REVISIÓN BIBLIOGRÁFICA**

En la actualidad existen muchos algoritmos, que ayudan a la detección y descripción de puntos de interés de una imagen, los mismos que han ido evolucionando a lo largo de la historia, haciéndose más robustos a diversas variantes, como la iluminación, rotación, escalabilidad.

### **2.1. Detectores de puntos de interés**

La detección de puntos de interés corresponde a la primera etapa durante el procesamiento de una imagen, que implica la individualización de

puntos, que por sus características ayudan en su conjunto a definir el objeto contenido en una imagen.

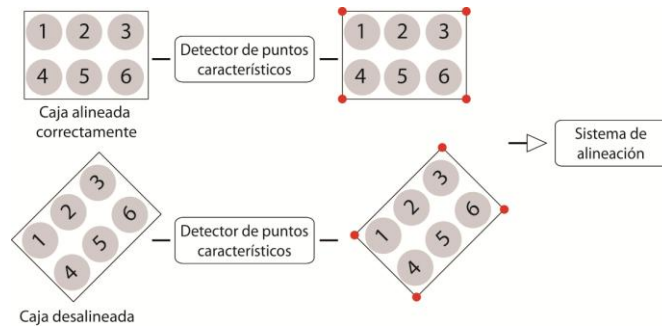


Figura 2.1. Esquema de un sistema de alineación.

En la Figura 2.1 podemos ver como la detección de estos puntos puede ser de utilidad para algún sistema automatizado de alineación, en el cual, luego de la detección es comparada con una imagen almacenada alineada correctamente.

Se pretende que los detectores sean eficientes respecto de los recursos computacionales que utilizan, esto influye directamente en la cantidad de cuadros por segundo que se puedan procesar y en las plataformas de hardware sobre las cuales se puede implementar el procesamiento en tiempo real en caso de ser necesario.



Es de gran importancia que posea un alto grado de repetitividad, es decir, un alto porcentaje de puntos de interés detectados que se repiten entre dos imágenes de un mismo objeto que haya sufrido algún tipo de variación, como puede ser una rotación, como se muestra en la Figura 2.2.

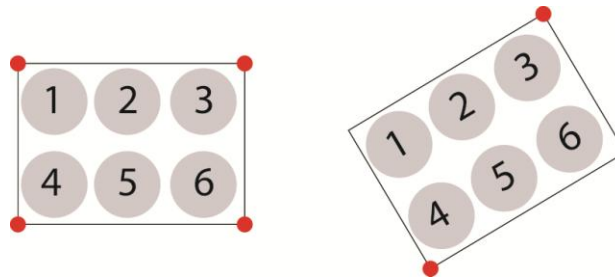


Figura 2.2. Diferentes puntos detectados.

Los métodos de detección de puntos de interés pueden clasificarse en tres grupos:

**Basado en bordes:** a partir del cambio en la dirección de los bordes.

**Basados en la derivada del nivel de grises:** se detectan los puntos ante las variaciones de su derivada en todas las direcciones.

**Basado en el nivel de grises:** en base al análisis de los niveles de grises dentro de una ventana centrada en el punto de interés, en el análisis se determina si es un punto de interés [7].

## **2.2. Descripción de puntos de interés**

Un detector ideal de puntos de interés localiza dichos puntos repetidamente a pesar del cambio de punto de vista siendo fiable ante transformaciones de la imagen.

Un descriptor ideal obtiene de cada punto de interés información importante y distintiva de la región que lo rodea, de manera que la misma estructura pueda ser reconocida si es encontrada en otra imagen. Implica un verdadero desafío describir de manera eficiente puntos de interés en forma estable y compacta, con representaciones robustas invariantes a escala, rotación, diversas transformaciones y el ruido.

## **2.3. Esquemas clásicos de detección y descripción**

Existen los denominados esquemas clásicos, que permiten llevar a cabo un eficiente procesamiento en la detección y descripción de puntos característicos, tales como los algoritmos SIFT, SURF, FAST entre otros. Cabe destacar que estos han sido propuestos para el caso de imágenes en el espectro visible (VS).

### **2.3.1. SIFT como detector**

Este detector fue presentado por el investigador y profesor de la Universidad British Columbia, David G. Lowe, en el año 1999 [1]. Los puntos de interés encontrados con este método se caracterizan por permanecer invariantes ante los posibles cambios producidos en los siguientes parámetros del sistema:

1. Iluminación
2. Ruido de la imagen
3. Rotación
4. Escala
5. Cambio de orientación

Este descriptor se caracteriza principalmente por detectar y describir una gran cantidad de puntos, muy estables, y por presentar un buen rendimiento en relación a la velocidad de cálculo y precisión.

Este algoritmo se divide en las siguientes etapas:

1. Detección de máximos y mínimos en el espacio escala
2. Localización de puntos de interés

3. Asignación de orientación
4. Generación de los descriptores de los puntos de interés

### **Detección de máximos y mínimos en el espacio escala**

La función espacio-escala de una imagen se encuentra dada por  $L(x,y,\sigma)$ , y es producida por la convolución de una función gaussiana de escala variable  $G(x,y,\sigma)$  y la imagen de entrada  $I(x,y)$ , siendo  $x$  e  $y$  la posición del punto en la imagen correspondiendo a las filas y columnas respectivamente:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

Donde  $*$  es el operador convolución en  $x$  e  $y$ , con  $G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

Para el cálculo eficiente de la localización de los puntos de interés, tanto en el espacio como en escala, se propone usar los máximos y los mínimos de la función  $D(x,y,\sigma)$  que es la diferencia de funciones gaussianas convolucionadas con la imagen (ver Figura 2.3). La diferencia de gaussianas se puede obtener de dos escalas próximas separadas por una constante multiplicativa  $k$ :

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma)$$

Una forma eficiente para la construcción de la función  $D(x,y,\sigma)$  es la que se describe en la Figura 2.3. La imagen inicial es de forma incremental convolucionada con gaussianas para producir imágenes separadas un factor  $k$  constante en la escala, que se muestran apiladas en la columna izquierda de la Figura 2.3. Cada imagen que forma la imagen inicial se denomina octava. Se divide cada octava de la escala para un número entero  $s$  de intervalos, ya que  $k=2^{1/s}$ . Después se tiene que producir  $k + 3$  imágenes por cada octava en el grupo de imágenes borrosas por efecto de la función gaussiana, de modo que la detección de máximos y mínimos final cubre una octava completa [6]. Finalmente las gaussianas adyacentes se restan para producir  $D(x, y, \sigma)$ , como se muestran apiladas en la columna de la derecha de la Figura 2.3.

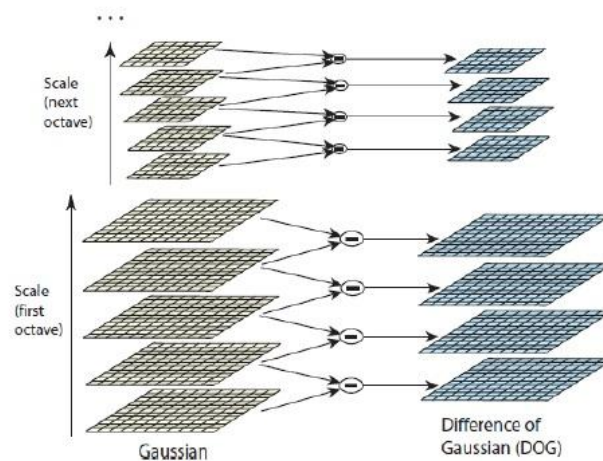


Figura 2.3 Espacio – escala propuesto por Lowe [1].

Para detectar los máximos y mínimos locales de  $D(x, y, \sigma)$ , cada punto es comparado con sus 8 – vecinos de su actual imagen y con sus 9 – vecinos de las imágenes de escalas anteriores y posteriores, como se puede ver en la Figura 2.4, siendo seleccionado solo si su valor es superior o inferior al de todos estos vecinos.

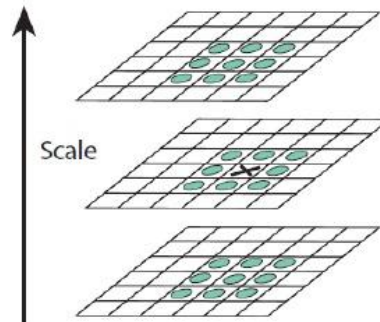


Figura 2.4 Proceso de comparación entre diferentes escalas.

### Localización de puntos de interés

Después de haber calculado los puntos de interés (también llamados puntos característicos) más apropiados, se realiza un estudio de estabilidad. Los puntos que no se encuentran bien situados sobre los bordes o con bajo contraste son vulnerables al ruido y difícilmente podrán

ser detectados bajo cambios de iluminación o variación del punto de vista de la imagen. Para excluirlos, Lowe utiliza los siguientes criterios:

1. Para eliminar los puntos con bajo contraste, se aplica un proceso de umbralización por el cual los puntos cuyo valor sea menor que dicho umbral  $D$  serán excluidos, por no considerarse estables.
  
2. Los puntos situados sobre bordes de manera difusa, conllevan un alto grado de inestabilidad, incluso ante pequeños ruidos. Para eliminar estos puntos, hacemos uso de la propiedad de la función  $DoG$  atendiendo a la gran curvatura que presenta en la dirección paralela al borde y la pequeña curvatura que se observa en la dirección perpendicular. Para esto calculamos la matriz Hessiana sobre la localización y escala del punto en estudio:

$$H = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix}$$

Las derivadas se calculan mediante la resta del valor de los puntos vecinos.

### Asignación de la orientación

La característica principal de los puntos detectados por SIFT es que estos son invariantes a una serie de transformaciones sobre las imágenes.

Mediante la asignación de una orientación para cada punto de interés basada en las propiedades locales de la imagen, estos pueden ser representados con respecto a su orientación y por lo tanto lograr invariancia a la rotación de imagen.

La escala del punto de interés se utiliza para seleccionar la imagen suavizada gaussiana  $L$  con la escala más cercana. Para cada punto de la imagen  $L(x, y)$ , se calcula la magnitud del gradiente,  $m(x, y)$ , y la orientación,  $\theta(x, y)$ , de la siguiente manera:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$$



Se utiliza un histograma de orientación, formado a partir de las orientaciones de gradiente  $\theta(x, y)$  de los puntos que se encuentran dentro de una región alrededor del punto de interés en análisis. El histograma de orientación tiene 36 contenedores que cubren 360 grados en total.

Cada valor añadido al histograma es ponderado por su magnitud del gradiente  $m(x, y)$  y por una ventana circular gaussiana con un  $\sigma$  de 1,5 veces la escala del punto de interés.

Los picos en el histograma de orientación corresponden a direcciones dominantes de los gradientes. Se detecta el pico más alto en el histograma, y luego cualquier otro pico local que se encuentre dentro del 80% del valor del pico más alto se utiliza para crear también un punto de interés con esa orientación. Por lo tanto, para las ubicaciones con múltiples picos de magnitud similar, habrá múltiples puntos de interés creados en la misma ubicación y escala, pero diferentes orientaciones.

### **2.3.2. SIFT como descriptor**

En las operaciones anteriores se ha asignado una ubicación en la imagen, escala y orientación a cada punto de interés. Estos parámetros imponen un sistema local de coordenadas 2D en el que se describe la

región local de la imagen, proporcionando invariancia a estos parámetros. El siguiente paso es calcular un descriptor para la región local de la imagen que sea altamente distintivo, y tan invariante como sea posible a variaciones como el cambio en la iluminación o el punto de vista.

La Figura 2.5 ilustra el cálculo del descriptor sobre un punto de interés, se puede observar una matriz 2 x 2 de histogramas de orientación, en el algoritmo final se utiliza una matriz 4 x 4 de histogramas, con 8 contenedores de orientación en cada uno.

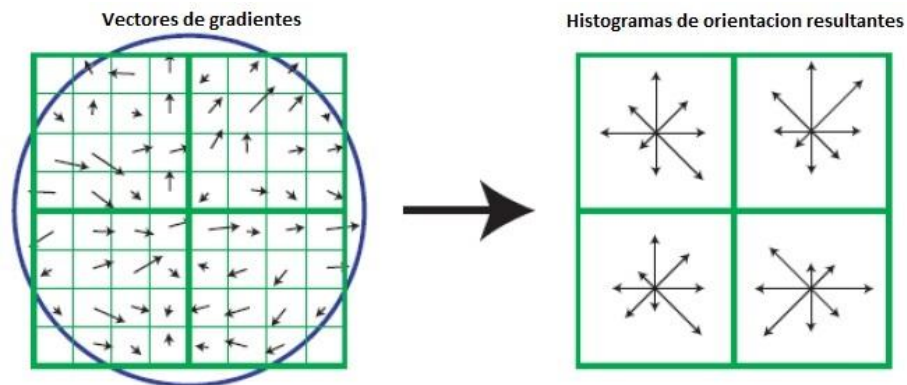


Figura 2.5 Descripción de puntos de interés.

Las magnitudes de los gráficos  $m(x, y)$  y orientaciones  $\theta(x, y)$  son muestreadas alrededor de la ubicación del punto de interés, utilizando la escala del punto de interés para calcular el  $\sigma$  de la ventana circular

gaussiana a aplicar sobre la imagen. Con el fin de lograr la invariancia a la rotación, las coordenadas del descriptor son giradas con relación a la orientación del punto de interés.

Por lo tanto, el descriptor SIFT utiliza un vector de características de longitud  $4 \times 4 \times 8 = 128$  para cada punto de interés seleccionado en la imagen.

### **2.3.3. SURF como detector**

Este detector fue desarrollado por Herbert Bay [2], guarda cierta similitud con la filosofía que usa SIFT. Sin embargo los autores afirman que este detector y descriptor presenta 2 mejoras:

1. Velocidad de cálculo considerablemente superior, sin ocasionar pérdida del rendimiento.
2. Mayor robustez ante posibles transformaciones de la imagen.

### **Detección de puntos de interés**

Para la localización y la escala de los puntos, se hace uso de la matriz Hessiana, más concretamente, del valor del determinante de la matriz. Lo más novedoso de este detector es que no utiliza diferentes medidas para

el cálculo de la posición y la escala de los puntos de interés individualmente, sino que utiliza el valor del determinante de la matriz Hessiana en ambos casos. Así, dado un punto  $p = (x, y)$  de la imagen  $I$ , la matriz Hessiana  $H(p, \sigma)$  del punto  $p$  perteneciente a la escala  $\sigma$  se define como:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix}$$

Donde  $L_{xx}(p, \sigma)$  representa la convolución de la derivada parcial de segundo orden de la Gaussiana  $\frac{\partial^2}{\partial x^2}g(\sigma)$  con la imagen  $I$  en el punto  $p$ . De manera análoga ocurre con los términos  $L_{xy}(p, \sigma)$ ,  $L_{yy}(p, \sigma)$ , de la matriz.

A pesar de que los filtros gaussianos son óptimos para el análisis del espacio – escala, se ha implementado una alternativa a los filtros gaussianos en este detector debido a una serie de limitaciones de estos filtros. Estos nuevos filtros aproximan las derivadas parciales de segundo orden de las gaussianas y pueden ser evaluados de manera muy rápida usando imágenes integrales, independientemente del tamaño de estas. Las imágenes integrales son calculadas mediante la siguiente fórmula:

$$I_{i\Sigma}(x, y) = \sum_{i=1}^{i \leq x} \sum_{j=1}^{j \leq y} I(i, j)$$

Donde  $(x, y)$  representan la posición del punto en la imagen de entrada como ya se definió en la sección 2.3.1 y  $I_i(x, y)$  representa la intensidad de la imagen en el punto.

Una vez la imagen integral ha sido creada, se puede calcular la suma de las intensidades de una región mediante una simple operación, como se puede observar en la Figura 2.6.

$$\sum R = I_{iD} + I_{iA} - I_{iB} - I_{iC}$$

De esta manera, el tiempo necesario para el cálculo de las operaciones de convolución es independiente del tamaño de la imagen.

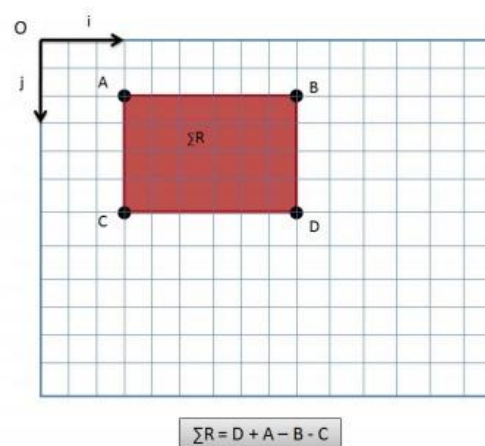


Figura 2.6 Representación de la intensidad de una región respecto de la imagen integral.

Las aproximaciones de las derivadas parciales se denotan como  $D_{xx}$ ,  $D_{xy}$ , y  $D_{yy}$ . En cuanto al determinante de la matriz Hessiana, este queda definido de la siguiente manera:

$$\det(H_{approx.}) = D_{xx}D_{yy} - (0,9D_{xy})^2$$

Donde el valor de 0,9 está relacionado con la aproximación del filtro gaussiano.

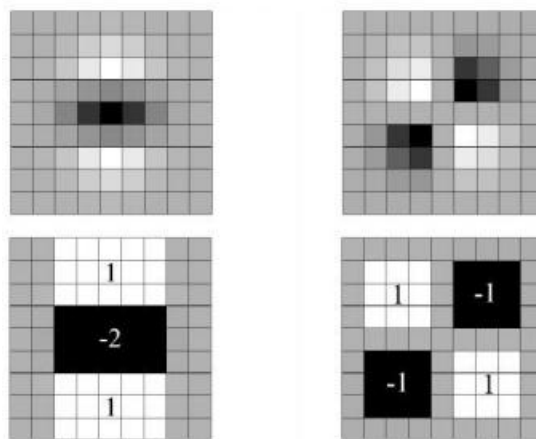


Figura 2.7 Derivadas parciales de segundo orden de un filtro y su aproximación.

La imagen de salida obtenida tras la convolución de la imagen original con un filtro de dimensiones 9 x 9, que corresponde a la derivada parcial de segundo orden de una gaussiana con  $\sigma = 1,2$ , como se muestra en la

Figura 2.7, es considerada como la escala inicial o también como la máxima resolución.

Las capas sucesivas se obtienen mediante la aplicación gradual de filtros de mayores dimensiones, evitando así los efectos de aliasing en la imagen.

Finalmente para calcular la localización de todos los puntos de interés en todas las escalas, se procede mediante la eliminación de los puntos que no cumplan la condición de máximo en un vecindario de  $3 \times 3 \times 3$ . De esta manera, el máximo determinante de la matriz Hessiana es interpolado en la escala y posición de la imagen. En este punto se da por concluida la etapa de detección.

### **Asignación de la orientación**

El primer paso para otorgar la mencionada orientación consiste en el cálculo de la respuesta de Haar en ambas direcciones  $x$  e  $y$  mediante las funciones representadas en la Figura 2.8. El área de interés para el cálculo es el área circular centrada en el punto de interés y de radio  $6s$ , siendo  $s$  la escala en la que el punto de interés ha sido detectado.

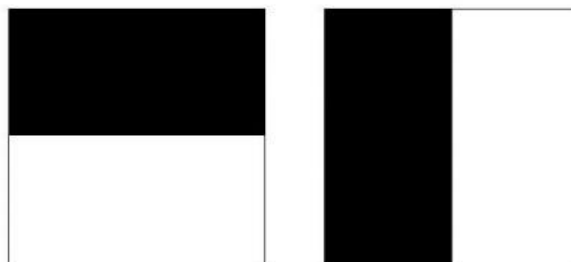


Figura 2.8 Filtros de Haar

Tras haber realizado todos estos cálculos, se utilizan imágenes integrales nuevamente para proceder al filtrado mediante las máscaras de Haar y obtener así las respuestas en ambas direcciones.

Las respuestas son representadas como vectores en el espacio colocando la respuesta horizontal y vertical en el eje de abscisas y ordenadas respectivamente. Finalmente, se obtiene una orientación dominante por cada sector mediante la suma de todas las respuestas dentro de una ventana de orientación móvil cubriendo un ángulo de  $\pi/3$  siguiendo las especificaciones recomendadas por el autor del algoritmo *H. Bay* [2], esta representación se puede observar en la Figura 2.9.



La orientación final del punto de interés será finalmente aquella cuyo vector sea el más grande dentro de los 6 sectores en los que ha sido dividida el área circular alrededor del punto de interés.

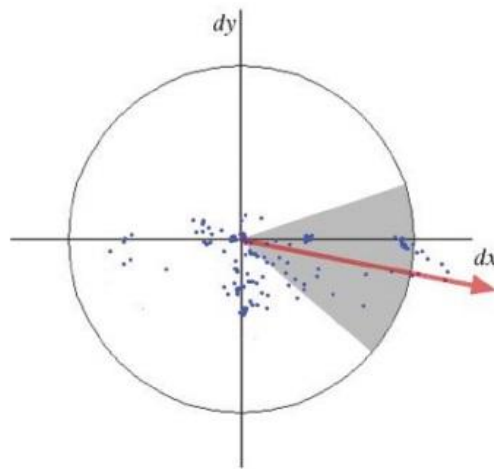


Figura 2.9 Asignación de la orientación de cada sector

#### 2.3.4. SURF como descriptor

SURF como descriptor construye una región cuadrada de tamaño  $20s$  alrededor del punto de interés y orientada en relación a la orientación calculada en la etapa de detección. Esta región es dividida en  $4 \times 4$  sub-regiones dentro de cada una de las cuales se calculan las respuestas de Haar de puntos con una separación de  $5 \times 5$  en ambas direcciones.

Consideramos  $d_x$  y  $d_y$  las respuestas de Haar en las direcciones horizontal y vertical respectivamente relativas a la orientación del punto de interés. En la Figura 2.10 se representan las respuestas de Haar en cada sub – región como las componentes  $d_x$  y  $d_y$  .

Cada una de las sub – regiones queda representada por un vector  $v$  de componentes:

$$v = ( \sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| )$$

Así, englobando las 4 x 4 sub – regiones resulta un descriptor con una longitud de 64 valores para cada uno de los puntos de interés identificados.

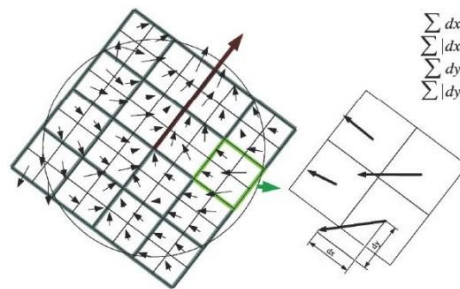


Figura 2.10 Respuestas de Haar en las sub–regiones alrededor del punto de interés.

# CAPÍTULO 3

## INSTALACIÓN Y CONFIGURACIÓN DE EQUIPO

### 3.1. Características del equipo

Para este proyecto se usará una laptop HP con las siguientes características:

**Procesador:** Intel Core i3 - 2.27 GHz

**Memoria:** 3 GB

**Tipo de sistema:** 32 bits

El sistema operativo bajo el cual funciona este proyecto es Windows 7 Home Basic - Service pack 1.

### 3.2. Herramientas de desarrollo

A continuación se describe en que lenguaje se desarrollará el proyecto y las herramientas de desarrollo que usaremos:

**Lenguaje:** Java

**IDE:** Netbeans6.1

**Librerías:** OpenCV 2.4.5 – JavaCV0.5-bin

#### **Java**

Lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases, diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

#### **Netbeans**

Es un entorno de desarrollo integrado libre, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación.

## **OpenCV**

Es una librería de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión en enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos.

## **JavaCV**

JavaCV es una implementación en java de la misma API (OpenCV), es decir está diseñado de manera que la mayoría de las funciones de OpenCV se puedan utilizar en JavaCV de la misma manera pero para lenguaje Java [3].

### **3.3. Instalación y configuración**

#### **Instalación**

Procedemos a descargar las herramientas desde los siguientes sitios recomendados:

Netbeans:

<https://netbeans.org/downloads/>

JDK-7u13-windows-i586:

<http://www.oracle.com/index.html>

OpenCV-2.4.5:

<http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.0/>

JavaCV0.5-bin

<https://code.google.com/p/javacv/downloads/detail?name=javacv-0.2-bin.zip>

vcredist\_x86:

<http://www.microsoft.com/en-us/download/details.aspx?id=5555>

## **Primero**

Instalamos JDK.

## **Segundo**

Instalamos el IDE Netbeans

### **Tercero**

Instalamos el archivo vcredist\_x86:

### **Cuarto**

Instalamos OpenCv 2.4.5, nos aseguramos que se encuentre en el directorio c:\

### **Quinto**

Instalamos javacv-0.5-bin, para esto extraemos la carpeta que se encuentra dentro del .zip y la pegamos en el c:\

### **Sexto**

Vamos a configurar las variables de entorno, para esto nos vamos a *Sistema - Configuración avanzada del sistema – Opciones avanzadas – Variables de entorno*. En Variables de usuario damos click en *Nueva* e ingresamos los siguientes datos:

**Nombre de variable:**

CLASSPATH

**Valor de la variable:**

C:\opencv\build\x86\vc10\lib;C:\opencv\build\common\tbb\ia32\vc10;C:\opencv\build\x86\vc10\staticlib;%CLASSPATH%

Nuevamente damos click en el botón *Nueva* e ingresamos los siguientes datos:

**Nombre de variable:**

PATH

**Valor de la variable:**

C:\opencv\build\common\tbb\ia32\vc10;C:\opencv\build\x86\vc10\bin;

El resultado de la configuración se muestra en la Figura 3.1

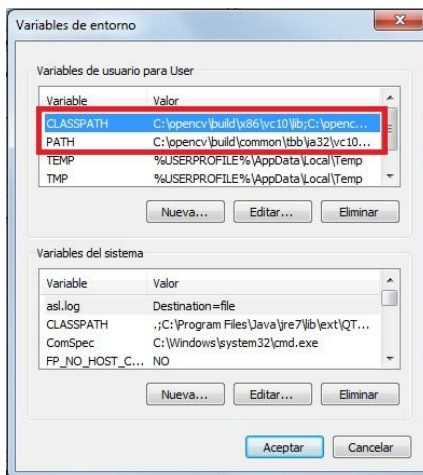


Figura 3.1 Configuración de las variables de entorno



Una vez configurado las variables de entorno del sistema procedemos a cargar las librerías en nuestro proyecto.

En nuestro proyecto nos ubicamos en la carpeta *libraries*, como se muestra en la Figura 3.2 damos click derecho y escogemos la opción *Add JAR/folder* y en la ruta *C:\javacv-bin* agregamos los archivos: *javacpp.jar*, *avacv.jar*, *javacv-windows-x86.jar*.

Los archivos quedan agregados a las librerías de nuestro proyecto como se muestra en la Figura 3.3.

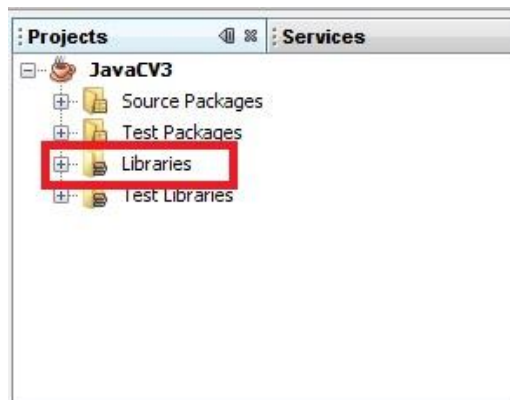


Figura 3.2 Carpeta Libraries de nuestro proyecto

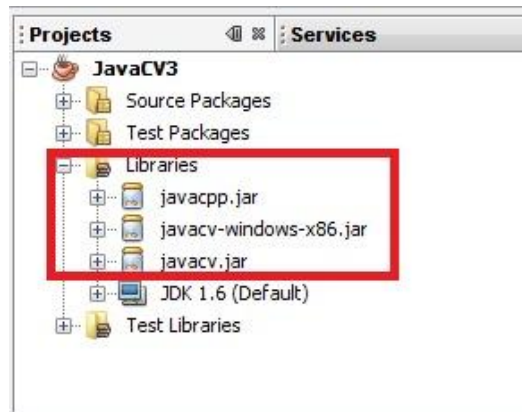


Figura 3.3 Archivos agregados a la carpeta Libraries

De esta manera nuestro proyecto en Netbeans queda configurado, y ya podemos hacer uso de las funciones de OpenCV para Java mediante JavaCV

# **CAPÍTULO 4**

## **DISEÑO DEL SISTEMA EXPERIMENTAL**

### **4.1. Interfaz gráfica del sistema**

Una vez configurado nuestro equipo con las herramientas necesarias para diseñar nuestro sistema experimental, la interfaz quedaría como se

muestra en la Figura 4.1

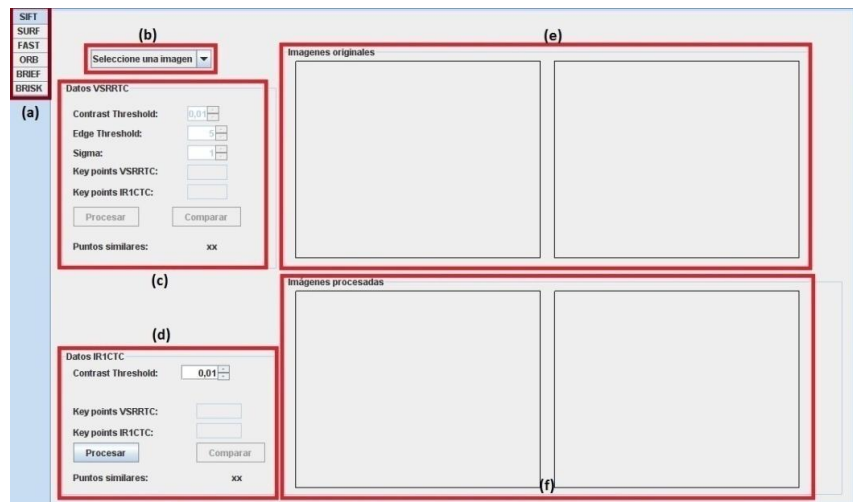


Figura 4.1 Interfaz gráfica del sistema experimental

(a) Pestañas que me permiten escoger el algoritmo:

SIFT - SURF – FAST – ORB – BRIEF – BRISK

(b) Combo box que me permite seleccionar las imágenes.

(c) Panel que contiene los parámetros iniciales de los algoritmos con sus respectivos botones para procesar y comparar imágenes originales.

(d) Panel que contiene la variación de parámetros del algoritmo con sus respectivos botones de procesar y comparar.

(e) Panel donde se cargan inicialmente las imágenes y también luego del proceso.

(f) Panel donde se cargan las imágenes como resultado experimental.

## 4.2. Manual de usuario

A continuación se muestra un pequeño manual de usuario que nos ayudará a comprender el funcionamiento del diseño experimental, en base a lo aprendido en el punto anterior.

**Primer paso:** Ejecutamos el programa, como resultado se podrá visualizar la ventana principal del proyecto, como se muestra en la Figura 4.1.

**Segundo paso:** Seleccionamos en la pestaña de algoritmos el que se requiera, para este ejemplo usaremos la pestaña SIFT, como se muestra en la Figura 4.2.

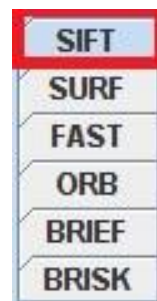


Figura 4.2 Pestañas con los algoritmos a seleccionar.

**Tercer Paso:** Una vez seleccionado el algoritmo, vamos al combo box y escogemos la imagen que queremos analizar, extrayéndolas de un directorio dentro del proyecto, como se muestra en la Figura 4.3.

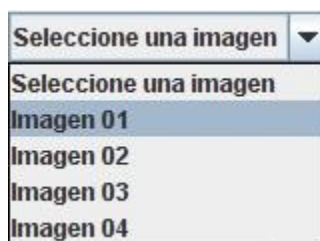


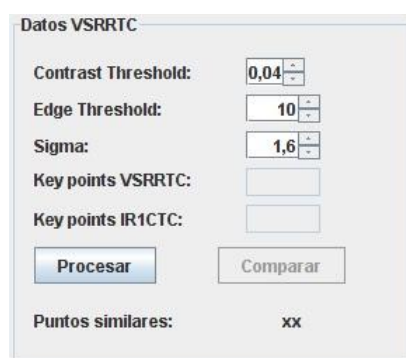
Figura 4.3 Combo box con una lista de imágenes.

Al seleccionar la imagen del combo box, se nos mostrarán las imágenes respectivas en el primer panel de imágenes, como se muestra en la Figura 4.4.



Figura 4.4 Panel con las imágenes originales

**Cuarto paso:** Ingresamos los parámetros para el algoritmo, como se muestra en la Figura 4.5.



The image shows a software dialog box titled "Datos VSRRTC". It contains several input fields and buttons. The "Contrast Threshold" is set to 0,04. The "Edge Threshold" is set to 10. The "Sigma" is set to 1,6. There are empty input fields for "Key points VSRRTC" and "Key points IR1CTC". Below these are two buttons: "Procesar" and "Comparar". At the bottom, there is a label "Puntos similares:" followed by the text "xx".

Figura 4.5 Parámetros del algoritmo

**Quinto paso:** Damos click en el botón procesar, y en el primer panel de imágenes se mostrarán las imágenes con los puntos detectados por el algoritmo, como se muestra en la Figura 4.6.



Figura 4.6 Puntos detectados en ambas imágenes

Se nos mostrará la cantidad de puntos de interés para cada imagen, como se muestra en la Figura 4.7.

Key points VSRRTC:	3337
Key points IR1CTC:	370

Figura 4.7 Puntos de interés detectados

**Sexto paso:** Damos click en el botón comparar, y se nos mostrará en el panel de imágenes los puntos similares entre las dos imágenes, como se muestra en la Figura 4.8.

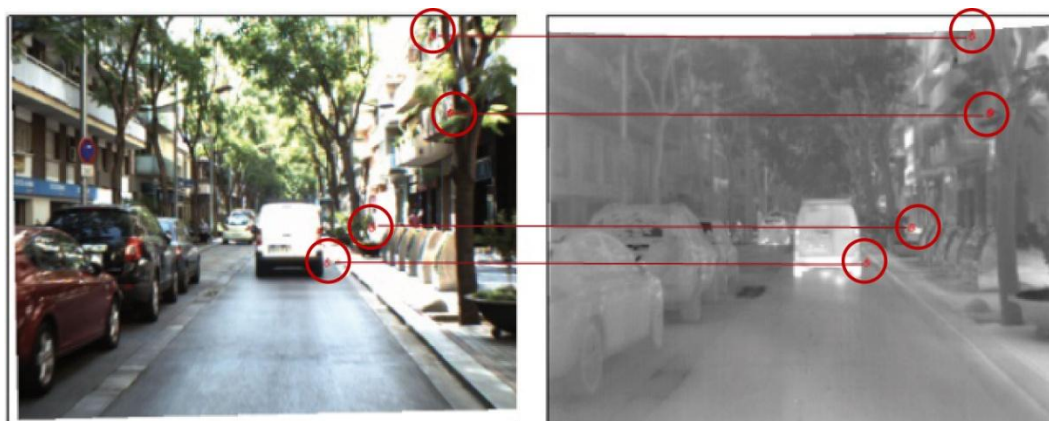


Figura 4.8 Imágenes con puntos similares



De esta manera se ha presentado el funcionamiento del sistema experimental, en el siguiente capítulo se hará un análisis más detallado acerca del uso de los algoritmos implementados para procesar las imágenes de diferentes espectros.

# **CAPÍTULO 5**

## **ANÁLISIS EN IMÁGENES MULTIESPECTRALES**

### **5.1. Comportamiento del algoritmo SIFT y SURF**

En este capítulo se analiza como caso particular el comportamiento de dos de los algoritmos presentados anteriormente. Estos algoritmos son evaluados usando imágenes de diferente espectro en el sistema experimental diseñado actualmente, y que hemos revisado en capítulos anteriores.

Primero haremos una evaluación de los algoritmos como detectores de puntos de interés, haremos las comparaciones correspondientes entre los resultados de ambas imágenes de diferentes espectros.

Luego haremos la evaluación de los algoritmos como descriptores, y haremos las comparaciones correspondientes entre los resultados obtenidos de ambos espectros.

## 5.2. Algoritmo SIFT

Este algoritmo está representado por la siguiente función, con sus respectivos parámetros:

```
SIFT(int nfeatures, int nOctaveLayers, double contrastThreshold,  
double edgeThreshold, double sigma)
```

**nfeatures**: Número de las mejores características para almacenar.

**nOctaveLayers**: Número de capas en cada octava.

**contrastThreshold**: Umbral del contraste.

**edgeThreshold**: Umbral para filtrar características de borde similar.

**Sigma:** Constante gaussiana.

Los parámetros iniciales [5] con que ejecutaremos el algoritmo son los siguientes:

**nfeatures = 0**

**nOctaveLayers = 3**

**contrastThreshold = 0.04**

**edgeThreshold = 10**

**Sigma = 1.6**

### **5.2.1. Detección de puntos de interés**

Al ejecutar el algoritmo con los parámetros anteriormente mencionados en la imagen de espectro visible obtenemos los resultados que se muestran en la TABLA 1.

**TABLA 1**  
**DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA**  
**UNA IMAGEN VS (SIFT)**

<b>Imagen VS</b>	<b>Puntos de interés</b>
<b>000026_VSRRTC</b>	3337
<b>000032_VSRRTC</b>	3425
<b>000059_VSRRTC</b>	3273
<b>000098_VSRRTC</b>	3310

Sacamos un promedio entre los puntos detectados en diferentes imágenes y obtenemos como resultado 3336 puntos de interés detectados.

Hacemos el mismo procedimiento para la imagen de espectro infrarrojo y obtenemos los resultados que se muestran en la TABLA 2.

**TABLA 2**  
**DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES PARA**  
**UNA IMAGEN IR (SIFT)**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>0000026_IR1CTC</b>	370
<b>0000032_IR1CTC</b>	382
<b>0000059_IR1CTC</b>	381
<b>0000098_IR1CTC</b>	398

Sacamos un promedio entre los puntos detectados en diferentes imágenes y obtenemos como resultado 382 puntos de interés detectados

Como podemos notar el algoritmo SIFT se comporta de manera diferente en una imagen visible que en una imagen infrarroja, obteniendo mayor cantidad de puntos característicos en la imagen visible, como se muestra en la TABLA 3 y como se puede ilustrar en la Figura 5.1.

**TABLA 3**  
**DIFERENCIA EN LA DETECCIÓN DE PUNTOS DE INTERÉS ENTRE**  
**IMÁGENES VS-IR (SIFT)**

	Imagen VS	Imagen IR	Diferencia
<b>Promedio de puntos de interés</b>	3336	382	2954



Figura. 5.1 Ilustración de puntos encontrados

Cabe destacar que las imágenes del espectro infrarrojo corresponden al mismo escenario que las imágenes evaluadas en el espectro visible. Por otra parte ambas imágenes fueron tomadas en el mismo instante usando

un trigger externo que sincroniza la adquisición de las mismas. Ambas imágenes tienen el mismo tamaño (640 x 480) y las lentes de las cámaras fueron dispuestas de manera tal que los sensores adquieren la misma información. Por otra parte, las imágenes a procesar ya se encuentran rectificadas<sup>1</sup> lo cual garantiza que un punto de interés detectado en la imagen visible se encuentra en la misma fila en la imagen infrarroja (ver detalles en la parte inferior y derecha de la imagen visible de la Figura 4.8 o detalle en la parte superior de la imagen infrarroja)

### 5.2.2. Descripción de puntos

Al ejecutar el algoritmo SIFT como descriptor analizaremos si los puntos detectados sobre ambas imágenes concuerdan en sus coordenadas  $y$ , como se comentó anteriormente, como las imágenes están rectificadas los puntos deberán pertenecer a la misma fila.

Para esto, el descriptor utiliza un vector de características de longitud 128 para cada punto de interés detectados en ambas imágenes, como se muestra en la Figura 5.2 y 5.3.

---

<sup>1</sup>[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)



	1	2	3	4	5	6	.	.	.	.	.	.	.	.	128	
Vs1	0	150	95	135	15	32	180	-	-	-	-	-	-	-	-	99
Vs2	55	10	145	5	44	31	175	-	-	-	-	-	-	-	-	123
Vs3	76	17	97	13	87	71	34	-	-	-	-	-	-	-	-	9
.																
.																
.																
Vsn	45	150	89	135	165	89	179	-	-	-	-	-	-	-	-	38

Figura 5.2 Vectores característicos para una imagen VS

	1	2	3	4	5	6	.	.	.	.	.	.	.	.	128	
Ir1	12	60	87	135	3	198	18	-	-	-	-	-	-	-	-	49
Ir2	98	32	12	56	98	43	69	-	-	-	-	-	-	-	-	12
Ir3	66	34	154	78	32	89	2	-	-	-	-	-	-	-	-	89
.																
.																
.																
Irn	56	183	67	198	15	99	189	-	-	-	-	-	-	-	-	68

Figura 5.3 Vectores característicos para una imagen IR

Para poder obtener la correspondencia de puntos entre ambas imágenes crearemos una matriz que contenga la distancia euclidiana entre cada descriptor, mediante la siguiente ecuación.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \dots}$$

La matriz distancia de los descriptores de ambas imágenes quedaría de la siguiente manera, como se muestra en la Figura 5.4

	Vs1	Vs2	Vs3	Vs4	Vs5	Vs6	.	.	.	.	.	.	VSn	
lr1	35	12	5	140	54	23	67	-	-	-	-	-	-	56
lr2	56	21	67	69	78	34	76	-	-	-	-	-	-	87
lr3	44	13	68	123	76	98	34	-	-	-	-	-	-	98
.														
.														
.														
lrn	87	34	156	143	78	92	143	-	-	-	-	-	-	12

Figura 5.4 Matriz de distancia de los descriptores

De esa matriz escogemos la menor distancia y buscamos sus correspondientes coordenadas y las comparamos, dando como resultado la información mostrada en la segunda columna de la TABLA 4. Se puede observar que no todas las correspondencias detectadas son correctas. Como se mencionó anteriormente, las parejas de correspondencias correctas se obtienen filtrando aquellas parejas con distintas coordenadas en  $y$ .

**TABLA 4**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS**  
**CORRECTAS USANDO LOS PARAMETROS INICIALES (SIFT)**

<b>Imágenes VS - IR</b>	<b>Correspondencias Detectadas</b>	<b>Correspondencias Correctas</b>
<b>0000026</b>	5	4
<b>0000032</b>	5	3
<b>0000059</b>	10	2
<b>0000098</b>	10	3

Como resultado de las imágenes analizadas se obtiene que en promedio se detectan unas 7 correspondencias. Sin embargo de ese promedio solo unas 3 correspondencias de puntos son correctas; un ejemplo se muestra en la Figura 5.5.



Figura 5.5 Comparación de imágenes con sus puntos de correspondencia (con un círculo se indican los puntos detectados y correctamente correspondidos).

### 5.2.3. Variación en parámetro en algoritmo

En base al estudio presentado y a los resultados mostrados en la sección anterior se propone un método para estudiar cómo se puede mejorar el número de correspondencias de puntos; este método

consiste en variar uno de los parámetros del algoritmo SIFT en la imagen infrarroja y analizar el número de correspondencias correctas.

Haremos el mismo proceso anteriormente descrito, pero variando el parámetro **contrastThreshold** a un valor menor a 0.04.

### **Detección de puntos de interés**

En el caso de las imágenes del espectro visible se ejecuta el algoritmo con los valores por defecto ( $\text{contrastThreshold} = 0.04$ ) y se obtienen los resultados mostrados en la TABLA 5.

**TABLA 5**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN VS CON VALOR ContrastTHRESHOLD= 0.04 (SIFT)**

<b>Imagen VS</b>	<b>Puntos de interés</b>
<b>000026_VSRRTC</b>	3337
<b>000032_VSRRTC</b>	3425
<b>000059_VSRRTC</b>	3273
<b>000098_VSRRTC</b>	3310

En el caso de las imágenes IR se adopta  $\text{contrastThreshold} = 0.03$ , y se obtienen los resultados mostrados en la TABLA 6.

**TABLA 6**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR CON VALOR ContrastTHRESHOLD= 0.03 (SIFT)**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>000026_IR1CTC</b>	580
<b>000032_IR1CTC</b>	596
<b>000059_IR1CTC</b>	596
<b>000098_IR1CTC</b>	626

Seguidamente se presentan los resultados en la TABLA 7 en el caso de que  $\text{contrastThreshold} = 0.02$ :

**TABLA 7**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR CON**  
**VALOR ContrastTHRESHOLD= 0.02 (SIFT)**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>000026_IR1CTC</b>	1079
<b>000032_IR1CTC</b>	1127
<b>000059_IR1CTC</b>	1055
<b>000098_IR1CTC</b>	1120

Como podemos observar al variar el parámetro del algoritmo para la imagen infrarroja, existe un incremento en la cantidad de puntos de interés detectados y una diferencia menor a lo anteriormente calculado, esto lo podemos visualizar en la TABLA 8 y de manera ilustrativa en la Figura 5.6.

**TABLA 8**  
**DIFERENCIA ENTRE LA CANTIDAD DE PUNTOS DETECTADOS**  
**ENTRE LAS IMAGNES VS-IR (SIFT)**

	Imagen VS	Imagen IR	Diferencia
<b>Promedio de puntos de interés</b>	3336	382	1095



Figura 5.6 Incremento de puntos de interés en imagen IR

Como conclusión a esto podemos decir que mientras menor sea el valor de ContrastThreshold mayor es el número de puntos de interés detectados. Pero esto no nos garantiza una buena correspondencia de puntos como se presentará en el siguiente análisis.



### Descripción de puntos

Para un valor de ContrastTHRESHOLD = 0.03 obtenemos los siguientes resultados mostrados en la TABLA 9:

**TABLA 9**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS CORRECTAS ENTRE IMÁGENES VS-IR CON VALOR DE ContrastTHRESHOLD= 0.03 (SIFT)**

<b>Imágenes VS - IR</b>	<b>Correspondencias Detectadas</b>	<b>Correspondencias Correctas</b>
<b>0000026</b>	9	5
<b>0000032</b>	5	4
<b>0000059</b>	3	2
<b>0000098</b>	14	5

Para un valor de 0.02 obtenemos los siguientes resultados mostrados en la TABLA 10.

**TABLA 10**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS**  
**CORRECTAS ENTRE IMÁGENES VS-IR CON VALOR DE**  
**ContrastTHRESHOLD= 0.02 (SIFT)**

<b>Imágenes VS - IR</b>	<b>Correspondencias Detectadas</b>	<b>Correspondencias Correctas</b>
<b>000026</b>	12	3
<b>000032</b>	3	2
<b>000059</b>	8	5
<b>000098</b>	11	4



Figura 5.7 Correspondencia de puntos para un valor del  
ContrastThreshold= 0.02

Como resultado de las imágenes analizadas se obtiene que en promedio se detectan unas 8 correspondencias, de las cuales solo unas 3 son correctas; en la Figura 5.7 se muestra un ejemplo ilustrativo.

### 5.3. Algoritmo SURF

Este algoritmo está representado por la siguiente función, con sus respectivos parámetros:

SURF(double **hessianThreshold**, int **nOctaves**, int **OctavesLayers**,  
bool **extendet**, bool **upright**)

**hessianThreshold:** Umbral de la constante Hessiana para la detección de puntos.

**nOctaves:** Número de pirámides.

**OctavesLayers:** Número de capas dentro de cada octava.

**extendet:** Bandera extendida del descriptor.

**upright:** Bandera de rotación.

Los parámetros iniciales [5] con que ejecutaremos el algoritmo son los siguientes:

**hessianThreshold = 500**

**nOctaves = 4**

**octavesLayers = 2**

**extendet = true**

**upright = false**

### **5.3.1. Detección de puntos de interés**

Al ejecutar el algoritmo con los parámetros anteriormente mencionados en la imagen de espectro visible obtenemos los resultados que se muestran en la TABLA 11.

**TABLA 11**  
**DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES**  
**PARA UNA IMAGEN VR (SURF)**

<b>Imagen VS</b>	<b>Puntos de interés</b>
<b>000026_VSRRTC</b>	1649
<b>000032_VSRRTC</b>	1637
<b>000059_VSRRTC</b>	1531
<b>000098_VSRRTC</b>	1508

Sacamos un promedio entre los puntos detectados en diferentes imágenes y obtenemos como resultado 1581 puntos de interés detectados.

Hacemos el mismo procedimiento para las imágenes de espectro infrarrojo y obtenemos los resultados que se muestran en la TABLA 12.

**TABLA 12**  
**DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES**  
**PARA UNA IMAGEN IR (SURF)**

Imagen IR	Puntos de interés
000026_IR1CTC	231
000032_IR1CTC	247
000059_IR1CTC	264
000098_IR1CTC	282

Sacamos un promedio entre los puntos detectados en diferentes imágenes y obtenemos como resultado 256 puntos de interés, mostrados en la TABLA 13 e ilustrado en la Figura 5.8.

**TABLA 13**  
**DIFERENCIA EN LA DETECCIÓN DE PUNTOS DE INTERÉS ENTRE**  
**IMÁGENES VS-IR (SURF)**

	Imagen VS	Imagen IR	Diferencia
<b>Promedio de puntos de interés</b>	1581	256	1325

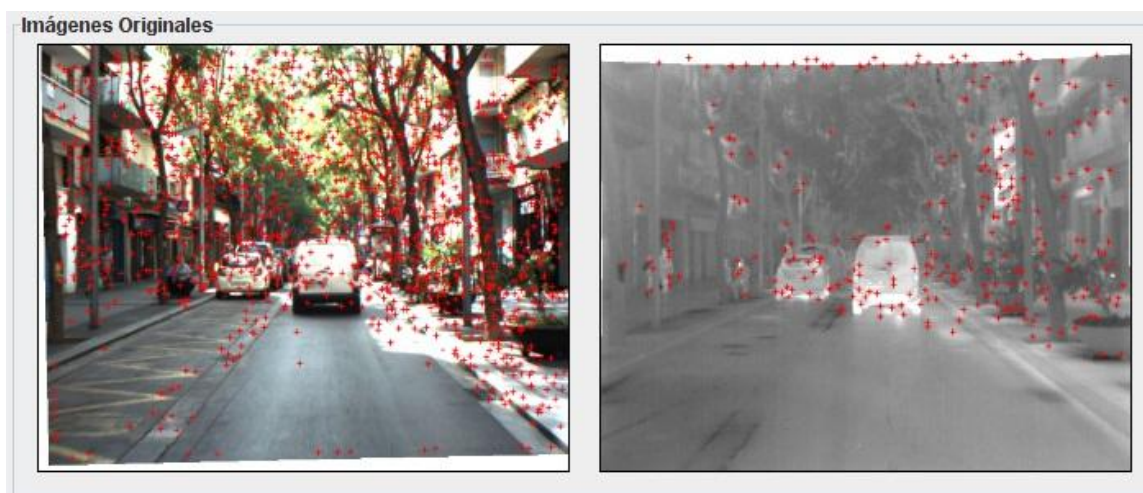


Figura 5.8 Comparación de imágenes con sus puntos de interés.

Como podemos notar el algoritmo SURF se comporta de manera diferente en una imagen visible que en una imagen infrarroja, obteniendo mayor cantidad de puntos característicos en la imagen visible, como se muestra en la siguiente tabla y como se puede apreciar en la ilustración de la Figura 5.8. Como conclusión preliminar podemos decir que SURF detecta más puntos que SIFT en los experimentos realizados en el dominio multiespectral. Sin embargo, al igual que en el caso anterior, la clave es ver si los puntos detectados en ambos espectros se corresponden.

### 5.3.2. Descripción de puntos

Al ejecutar el algoritmo SURF como descriptor analizaremos si los puntos detectados sobre ambas imágenes concuerdan en sus coordenadas  $y$ .

Para esto, al igual que en la sección 5.2.2, el descriptor utiliza un vector de características de longitud 128 para cada punto de interés detectados en ambas imágenes, como se muestra en la Figura 5.2 y 5.3 de la sección 5.2.2.

Para poder obtener la correspondencia de los puntos de ambas imágenes crearemos nuevamente una matriz que contengan la distancia euclidiana entre cada descriptor, mediante la ecuación:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \dots \dots}$$

La matriz distancia de los descriptores de ambas imágenes es similar a la ilustrada en la Figura 5.4 de la sección 5.2.2.



Una vez realizado este procedimiento obtenemos los resultados mostrados en la TABLA 14

**TABLA 14**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS CORRECTAS ENTRE LAS IMÁGENES VS-IR (SURF)**

Imágenes VS - IR	Correspondencias Detectadas	Correspondencias Correctas
0000026	3	1
0000032	6	1
0000059	2	1
0000098	8	1



Figura 5.9 Comparación de imágenes con sus puntos de correspondencia (con un círculo se indican los puntos detectados y correctamente correspondidos).

Como resultado de las imágenes analizadas se obtiene que en promedio se detectan unas 4 correspondencias, de las cuales, en promedio, solo 1 es correcta; la Figura 5.9 muestra una ilustración de un ejemplo en el cual solo 1 correspondencia es correcta.

### 5.3.3. Variación en parámetro de algoritmo

En base al estudio presentado y a los resultados mostrados en la sección anterior se propone un método para estudiar cómo se puede mejorar el número de correspondencias correctas; este método consiste en variar uno de los parámetros del algoritmo SURF en la imagen infrarroja y analizar el número de correspondencias correctas.

Haremos el mismo proceso anteriormente descrito, pero en este caso variando el parámetro **hessianThreshold** a un valor menor a 500.

### Detección de puntos de interés

Al igual que en el caso anterior (sección 5.2.3), en la imagen visible los puntos de interés son detectados usando los valores por defecto

presentados al inicio de la sección 5.3; los resultados obtenidos se presentan en la TABLA 15.

**TABLA 15**  
**DETECCIÓN DE PUNTOS DE INTERÉS CON VALORES INICIALES**  
**PARA UNA IMAGEN VR (SURF)**

<b>Imagen VS</b>	<b>Puntos de interés</b>
<b>000026_VSRRTC</b>	1649
<b>000032_VSRRTC</b>	1637
<b>000059_VSRRTC</b>	1531
<b>000098_VSRRTC</b>	1508

En el caso de la imagen infrarroja se varía el siguiente parámetro `hessianThreshold = 300`; los resultados son presentados en la TABLA

**TABLA 16**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR**  
**(SURF) PARA UN VALOR DE HESSIANTHRESHOLD= 300**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>000026_IR1CTC</b>	365
<b>000032_IR1CTC</b>	396
<b>000059_IR1CTC</b>	405
<b>000098_IR1CTC</b>	414

El experimento se vuelve a repetir pero con hessianThreshold = 200 y se muestran los resultados en la TABLA 17.

**TABLA 17**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR**  
**(SURF) PARA UN VALOR DE HESSIANTHRESHOLD= 200**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>0000026_IR1CTC</b>	489
<b>0000032_IR1CTC</b>	516
<b>0000059_IR1CTC</b>	548
<b>0000098_IR1CTC</b>	548

Finalmente se repite el experimento pero con hessianThreshold = 100;  
se muestran los resultados en la TABLA 18.

**TABLA 18**  
**DETECCIÓN DE PUNTOS DE INTERÉS PARA UNA IMAGEN IR**  
**(SURF) PARA UN VALOR DE HESSIANTHRESHOLD= 100**

<b>Imagen IR</b>	<b>Puntos de interés</b>
<b>000026_IR1CTC</b>	747
<b>000032_IR1CTC</b>	797
<b>000059_IR1CTC</b>	811
<b>000098_IR1CTC</b>	839

Como podemos observar al cambiar los valores de hessianThreshold del algoritmo en el caso de la imagen infrarroja, existe un incremento en la cantidad de puntos de interés detectados y una diferencia menor a lo anteriormente calculado en promedio, esto lo podemos visualizar en la TABLA 19 y en la ilustración de la Figura 5.8.

**TABLA 19**  
**DIFERENCIA EN LA DETECCIÓN DE PUNTOS ENTRE LAS**  
**IMÁGENES VS-IR (SURF)**

	Imagen VS	Imagen IR	Diferencia
Promedio de puntos de interés	1581	798	1095



Figura 5.10 Detección de puntos de interés para un valor de `hessianThreshold= 200`

En resumen, podemos decir que mientras menor sea `hessianThreshold`, mayor cantidad de puntos de interés se van a obtener. Pero así mismo

esto no nos garantiza que exista una buena correspondencia de puntos, como se verá en el siguiente apartado.

### Descripción de puntos

En la TABLA 20 se presentan los resultados obtenidos para un valor de `hessianThreshold = 300`.

**TABLA 20**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS CORRECTAS ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN VALOR DE HESSIANTHRESHOLD= 300**

<b>Imágenes VS - IR</b>	<b>Correspondencias Detectadas</b>	<b>Correspondencias Correctas</b>
<b>0000026</b>	3	1
<b>0000032</b>	6	2
<b>0000059</b>	4	1
<b>0000098</b>	8	1

Para un valor de `hessianThreshold = 200` se obtienen los resultados presentados en la TABLA 21.



**TABLA 21**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS**  
**CORRECTAS ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN**  
**VALOR DE HESSIAN THRESHOLD= 200**

<b>Imágenes VS - IR</b>	<b>Correspondencias Detectadas</b>	<b>Correspondencias Correctas</b>
<b>0000026</b>	3	1
<b>0000032</b>	2	2
<b>0000059</b>	1	1
<b>0000098</b>	8	1

Finalmente, para un valor de `hessianThreshold = 100`, se obtienen los resultados presentados en la TABLA 22:

**TABLA 22**  
**CORRESPONDENCIAS DETECTADAS Y CORRESPONDENCIAS**  
**ENTRE LAS IMÁGENES VS-IR (SURF) PARA UN VALOR DE**  
**HESSIANTHRESHOLD= 100**

Imágenes VS - IR	Correspondencias Detectadas	Correspondencias Correctas
0000026	9	1
0000032	5	2
0000059	3	1
0000098	8	1



Figura 5.11 Correspondencia de puntos para valores de  
hessianThreshold= 200

Como resultado en la similitud de puntos entre ambas imágenes, independientemente de los valores adoptados para la variable `hessianThreshold` (100, 200, 300 o 500), tenemos siempre un promedio de 6 puntos detectados, pero en promedio, de estos 6 puntos detectados solo 1 es correcto; la Figura 5.11 presenta una ilustración en la que se observa que finalmente solo una correspondencia es correcta.

## **CONCLUSIONES**

A lo largo de este trabajo se evaluaron distintos algoritmos de detección y descripción para analizar si estos nos sirven para el caso de imágenes de espectro infrarrojo y su correspondiente uso en el caso multiespectral (infrarrojo y visible). Se escogió el camino de variar los parámetros iniciales de los algoritmos de detección y descripción, para las imágenes en el espectro infrarrojo, de cara a favorecer el incremento del número final de correspondencias. Como conclusión a este trabajo podemos decir que, si bien al variar ciertos parámetros obtenemos un incremento en la detección de puntos de interés en cada imagen, al momento de hacer la correspondencia, pocos son los puntos que coinciden. Cabe aclarar que este es un estudio preliminar y para obtener conclusiones concluyentes se debería explorar como afectan a los resultados variaciones de otros parámetros; o más interesante aún variaciones acopladas de varios parámetros.

## RECOMENDACIONES

Como se comentó en el apartado anterior, existen otras posibilidades a explorar, ya que como se pudo observar anteriormente se logró encontrar mayor cantidad de puntos en imágenes de espectro infrarrojo, el objetivo sería lograr que esos puntos correspondan con los detectados en el espectro visible. Otra recomendación se puede dar como fruto de este trabajo es el hecho de añadir algún filtro a la imagen original, como un ecualizador de histograma que genere una imagen más llana y separada posible, esto haría que los detalles que no son tan evidentes resaltarán más y así ayude en las futuras etapas del algoritmo no solo a la detección sino también a la descripción, y lograr así aumentar el número de correspondencias entre las imágenes de diferentes espectros.

## BIBLIOGRAFÍA

[1] David G. Lowe. "Object recognition from local scale invariant features". Proceedings of the IEEE International Conference on Computer Vision, Kerkyra. Greece, September 20-27, pages 1150 – 1157.

[2] H. Bay, T. Tuytelaars and L. van Gool, "SURF: Speeded Up Robust Features" Proceedings of the 9<sup>th</sup> European Conference on Computer Vision, Graz, Austria, May 7 – 13, pages 404 – 417.

[3] Javacv. "Java interface to OpenCV and more"  
[Online] <https://code.google.com/p/javacv/> [2013]

[4] OpenCV. "Basic C structures and operations"  
[Online] [http://docs.opencv.org/modules/core/doc/old\\_basic\\_structures.html](http://docs.opencv.org/modules/core/doc/old_basic_structures.html)  
[2013]

[5] OpenCV. "Feature detection and description"  
[Online] [http://docs.opencv.org/modules/nonfree/doc/feature\\_detection.html](http://docs.opencv.org/modules/nonfree/doc/feature_detection.html)  
[2013]

[6] Abel Alguacil Gómez. "Aplicaciones del operador SIFT al reconocimiento de objetos". Capítulo 2, página 10.  
[Online] <http://e-archivo.uc3m.es/bitstream/handle/10016/9929/PFC%20Abel%20Alguacil.pdf?sequence=1>[2009]

[7] Marcelo Daniel Pistarelli. Detección y correspondencia de características en imágenes multispectrales. Proyecto final de carrera Ingeniería en Informática, Universidad Autónoma de Barcelona, 4 de Febrero de 2013, Capítulo 2, páginas 8 - 9.