



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“CORRESPONDENCIA MULTIESPECTRAL EN EL ESPACIO DE HOUGH.”

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS MULTIMEDIA

Presentado por:

GONZALO ANDRÉS ANDRADE MOREIRA

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

Agradezco a Dios por brindarme salud y bienestar. A mi familia por su apoyo incondicional en todo momento. A mis profesores por otorgarme conocimiento y motivación. A la ESPOL por brindarme las herramientas y oportunidades. A mis amigos por los momentos compartidos.

DEDICATORIA

A Dios. A mi familia, que siempre me ha ayudado a cumplir mis objetivos. A mis profesores, que me han ayudado a mejorar gracias a su tiempo y dedicación.

TRIBUNAL DE SUSTENTACIÓN



Ph. D. Angel D. Sappa

PROFESOR DEL SEMINARIO




MSc. Gonzalo Luzardo Morocho

PROFESOR DELEGADO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL)



Gonzalo Andrés Andrade Moreira

RESUMEN

El presente trabajo ha sido desarrollado como un proyecto final de graduación para la carrera de grado en Ciencias Computacionales de la Facultad de Ingeniería en Electricidad y Computación – FIEC de la ESPOL.

El objetivo del proyecto es desarrollar una aplicación que pueda establecer o describir una correspondencia de características (rectas o círculos) entre imágenes del espectro visible y del espectro infrarrojo con el fin de poder ser utilizados en aplicaciones futuras.

En el proyecto se pondrán en práctica algoritmos ya desarrollados en los cuales se evaluarán los resultados obtenidos con diferentes variables o procedimientos. Para esto, se desarrollará una interfaz gráfica donde se podrán verificar los procedimientos que se realizaron durante la evaluación.

A lo largo del documento se realizará una descripción más detallada de cada una de las partes del desarrollo del proyecto.

ÍNDICE GENERAL

RESUMEN	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS.....	X
INTRODUCCIÓN	XII
CAPÍTULO 1	
DESCRIPCIÓN DEL PROYECTO	1
1.1. DESCRIPCIÓN DEL PROBLEMA	1
1.2. SOLUCIÓN PROPUESTA	2
1.3. OBJETIVOS.....	3
1.3.1. Objetivo General.....	3
1.3.2. Objetivos Específicos	3
1.4. INDICACIONES.....	4
CAPÍTULO 2	
REVISIÓN BIBLIOGRÁFICA	5
2.1. DETECCIÓN DE PUNTOS DE INTERÉS.	5
2.1.1. Detectores de puntos de interés basados en la derivada de intensidad.	6
2.1.1.1. Moravec	7
2.1.1.2. Harris.....	8

2.1.2. Detectores de puntos basados en el valor de la intensidad.....	11
2.1.2.1. SUSAN.....	11
2.1.2.2. Basados en AST	12
2.1.2.3. FAST	13
2.1.2.4. FAST-ER.....	14
2.2. DESCRIPCIÓN DE PUNTOS DE INTERÉS.....	15
2.2.1. SIFT.....	17
2.2.2. SURF.....	20
2.2.3. BRISK.....	21
2.3. DESCRIPCIÓN DE PUNTOS DE INTERÉS EN IMÁGENES MULTIESPECTRALES: VS-NIR, VS-LWIR.....	24
2.3.1. GOM-SIFT	26
2.3.2. OR-SIFT	27
2.3.3. EOH-SIFT.....	28
CAPÍTULO 3	
DESARROLLO	35
3.1. DETECCIÓN DE CONTORNOS	36
3.1.1. Suavizado de la imagen	37
3.1.2. Cálculo de la magnitud y ángulo del gradiente	37
3.1.3. Supresión de no-máximos.....	38

3.1.4. Aplicación de umbral con histéresis	39
3.2. TRANSFORMADA DE HOUGH	40
3.2.1. Transformada de Hough para curvas paramétricas	41
3.2.2. Detección de Líneas y Círculos	42
3.3. DESCRIPCIÓN DE CARACTERÍSTICAS	44
3.4. MATCHING.....	48
3.5. INTERFAZ GRÁFICA DESARROLLADA	49
CAPÍTULO 4	
EVALUACIÓN Y ANÁLISIS DE RESULTADOS	51
4.1. RESULTADOS EXPERIMENTALES	51
4.2. PROPUESTA DE MODIFICACIONES.....	54
4.2.1. Rectificación manual de imágenes	54
4.2.2. Modificación del umbral de Canny para imágenes LWIR	55
4.2.3. Análisis espacio-escala	56
CONCLUSIONES	58
RECOMENDACIONES.....	60
BIBLIOGRAFÍA.....	61

ÍNDICE DE FIGURAS

Figura 1.1. A la izquierda: visible, a la derecha: infrarrojo lejano.	2
Figura 2.1. Cambio de intensidad para el desplazamiento $[u, v]$	8
Figura 2.2. Clasificación de los puntos de imagen según los valores propios de $C_{x,y}$	10
Figura 2.3. Circulo de Bresenham	13
Figura 2.4. Patrón circular aplicado sobre el pixel p	14
Figura 2.5. Descriptores invariantes a las trasformaciones: traslación, rotación, escalamiento, brillo.	16
Figura 2.6. La magnitud y orientación del gradiente local son agrupadas para crear un descriptor de puntos característicos.	20
Figura 2.7. Detector espacio-escala.	22
Figura 2.8. El patrón de muestreo BRISK con $N = 60$ puntos.	23
Figura 2.9. Ejemplo de matching con BRISK. Se utiliza un umbral de detección de 70 y un umbral de distancia de Hamming de 90.	24
Figura 2.10. Contenedor de orientaciones SIFT a la izquierda. Contenedor de orientaciones modificadas a la derecha.	28
Figura 2.11. Diagrama de flujos del método EOH-SIFT.	29
Figura 2.12. Descriptor basado en EOH.	31
Figura 2.13. Pasos del método de descripción de puntos característicos para pares de imágenes VS-LWIR.	34
Figura 3.1. Diagrama de flujo del método propuesto.	35

Figura 3.2. Ilustración de una supresión de no-máximos.....	39
Figura 3.3. Transformada de Hough.	41
Figura 3.4. Detección de líneas utilizando la Transformada de Hough.....	43
Figura 3.5. El set de prueba. Imagen de la izquierda: visible. Imagen de la derecha: infrarrojo.....	44
Figura 3.6. Canny de imagen VS.....	58
Figura 3.7. Canny de imagen LWIR.....	58
Figura 3.8. Resultado del cálculo de la transformada de Hough.....	59
Figura 3.9. Detección de líneas y círculos en el par de imágenes.	48
Figura 3.10. Correspondencia de líneas y círculos en el par de imágenes...	48
Figura 3.11. Interfaz gráfica desarrollada.	49
Figura 4.1. Rectificación manual de la imagen.	55
Figura 4.2. Modificación de umbral en la imagen LWIR.....	56
Figura 4.3. Diagrama del flujo del método propuesto.	57

INTRODUCCIÓN

En el área de visión por computador existen muchas aplicaciones que sirven para resolver problemas de interés, ya sea para mejorar procesos u obtener resultados a tareas específicas, entre estas aplicaciones podemos nombrar procedimientos de realzado, compresión, restauración, detección, correspondencia y reconocimiento de objetos.

La correspondencia entre dos objetos se define como una función f , que determina alguna relación binaria entre algún elemento del primer conjunto X con algún elemento del segundo conjunto Y . Podremos decir que esa función f definirá una correspondencia [1].

La correspondencia entre imágenes se basa en la búsqueda y detección de entidades como puntos, líneas y áreas entre dos o más fotogramas [2], las cuales son luego utilizadas como referencias durante el proceso de correspondencia.

El propósito de este proyecto es realizar un análisis entre dos imágenes de diferentes espectros utilizando la transformada de Hough, el cual nos permite detectar correspondencias a través de figuras tales como rectas o circunferencias.

Para el estudio mencionado se utilizarán imágenes infrarrojas (LWIR) y del espectro visible (VS). Dichas imágenes fueron proporcionadas por un par estereoscópico multiespectral (VS-LWIR) del Centro de Visión por Computador de la Universidad Autónoma de Barcelona, España.

El documento se encuentra organizado de la siguiente manera:

Capítulo 1 – Descripción del proyecto: El primer capítulo es introductorio, el cual constará de la descripción del problema y la solución propuesta.

Capítulo 2 – Revisión de Literatura: En este capítulo se describirán los diversos algoritmos relacionados al proyecto, se ha dividido este capítulo en 3 apartados: detectores y descripción de características en la imagen; descripción de puntos de interés y descripción de puntos de interés en imágenes multiespectrales.

Capítulo 3 – Desarrollo: En este capítulo se detallará los procedimientos realizados para poder obtener la correspondencia entre los pares de imágenes del espectro visible e infrarrojo lejano. También se describe detalladamente la interfaz gráfica que permitirá modificar las variables necesarias para poder obtener resultados óptimos en la imagen infrarroja.

Capítulo 4 – Análisis y evaluación: En este capítulo se presentarán los resultados de las pruebas con diferentes variables en los algoritmos para poder identificar en qué condiciones se produce una correspondencia multiespectral robusta.

CAPÍTULO 1

DESCRIPCIÓN DEL PROYECTO

1.1. DESCRIPCIÓN DEL PROBLEMA

El principal problema es encontrar correspondencia entre los elementos de las imágenes multiespectrales, los cuales no son obtenibles mediante técnicas clásicas de correspondencia como SURF y SIFT o similares, debido a que ambas imágenes tienen información fotométrica diferente.

Podemos observar en la imagen a continuación, que el ojo humano detecta correspondencia, sin embargo, ambas imágenes poseen información diferente, en la imagen de la izquierda (espectro visible) tenemos información fotométrica, mientras que en la imagen de la derecha (infrarrojo lejano) tenemos información térmica.

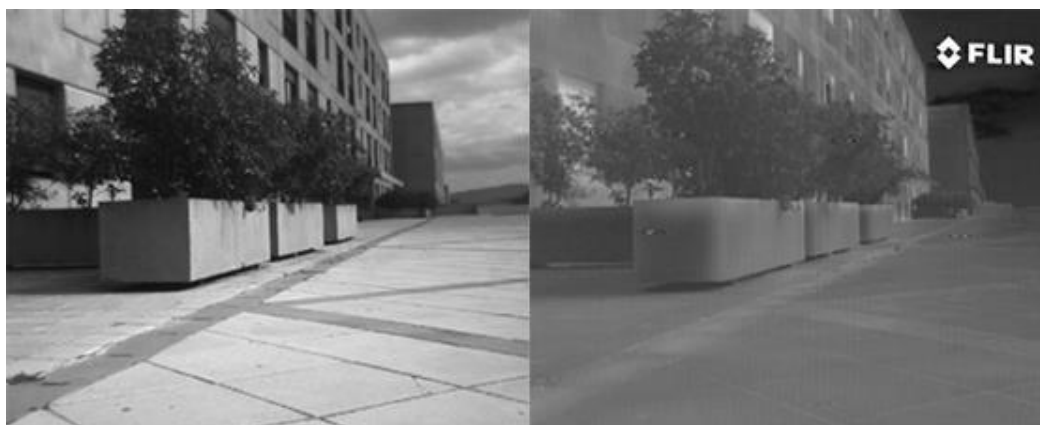


Figura 1.1 A la izquierda: visible, a la derecha: infrarrojo lejano.

1.2. SOLUCIÓN PROPUESTA

Debido a que el tipo de información de las imágenes son diferentes, se deberá transformar los patrones a un espacio en común, en nuestro caso utilizando la transformada de Hough.

Se desarrollará una interfaz gráfica (GUI), en donde se pueda manipular con facilidad las variables de los algoritmos de detección de características y descriptores, así como también poder observar los resultados obtenidos.

También se desarrollarán los algoritmos de detección de características y correspondencia para finalmente analizar los resultados obtenidos.

1.3. OBJETIVOS

1.3.1. Objetivo General

El objetivo del proyecto es poder establecer o describir una correspondencia de característicos (rectas o círculos) entre imágenes del espectro visible y del espectro infrarrojo para que puedan ser utilizados en aplicaciones futuras.

1.3.2. Objetivos Específicos

- Estudio del estado del arte en técnicas de detección y descripción características en imágenes (puntos, líneas, regiones) del espectro visible.
- Estudiar las distintas técnicas para encontrar correspondencias en el espacio de características.
- Adaptar los conceptos anteriores al dominio multiespectral (detección, descripción y puesta en correspondencia)
- Evaluación de las distintas técnicas para poder poner en correspondencia imágenes obtenidas en distintas bandas del espectro (visible e infrarrojo).
- Fortalecer los conocimientos y habilidades en el ámbito de procesamiento de análisis de imágenes a través del estudio de la teoría y la implementación de proyectos con enfoque de aplicación real.

1.4. INDICACIONES

El proyecto será desarrollado sobre la plataforma Qt para el entorno gráfico, para las funcionalidades del sistema se utilizará C++ en conjunto con la librería gráfica OpenCV.

El hardware y software requerido para que el proyecto funcione correctamente se detalla a continuación:

- Sistema operativo Windows y Linux.
- Procesador de 32 bit (x86) o 64 bit (x64).
- Procesador dual-core 2.66-GHz o posterior.
- 2 GB RAM o superior.
- OpenCV 2.4.6 o superior.
- Framework Qt 5.1.0 o superior.

CAPÍTULO 2

REVISIÓN BIBLIOGRÁFICA

En este capítulo se realiza una revisión bibliográfica de los distintos temas abordados en el presente trabajo. En el apartado 2.1 se describen las técnicas de detección de puntos de interés y su evolución en los últimos años. Seguidamente, en la sección 2.2 se detallan las técnicas más representativas de descripción de características, en particular para la descripción de los puntos de interés. Finalmente, en la sección 2.3 se presentan los distintos algoritmos que se han propuesto recientemente para abordar el caso multiespectral.

2.1. DETECCIÓN DE PUNTOS DE INTERÉS.

La detección de puntos de interés es un área en la visión por computadora que tiene como objetivo detectar puntos de importancia o interés para un futuro procesamiento. Un punto de interés será claro y se encuentra bien definido cuando la imagen de cara a garantizar una cierta repetitividad. Los puntos de interés son utilizados para la

detección, tracking y reconocimiento de objetos, generalmente en dos o tres dimensiones.

Se puede clasificar los tipos de detección de puntos de interés en dos grupos:

- Basados en la derivada de la intensidad.
- Basados en el valor de la intensidad.

Una vez que las los puntos de interés han sido detectadas, se puede describir sus características mediante procesamiento de imágenes, las técnicas para esto son conocidas como algoritmos de descripción, los cuales se detallarán en el apartado 2.2.

2.1.1. Detectores de puntos de interés basados en la derivada de intensidad.

Una esquina puede ser definida como una intersección entre dos bordes, donde un borde es un cambio brusco en el valor de la intensidad de la imagen. Normalmente una esquina es considerada un punto de interés vital ya que es utilizado en el área de visión por computador para obtener diferentes tipos de características de una imagen dada. La primera persona en utilizar el término “puntos de interés” de una imagen, los cuales se usan para encontrar regiones coincidentes en diferentes

imágenes fue Hans P. Moravec en 1977. El operador de Moravec es considerado como un detector de esquinas ya que define puntos de interés como puntos donde hay una gran intensidad de variaciones en todas las direcciones.

Para el ojo humano es sencillo identificar una esquina, sin embargo, los algoritmos utilizados por una computadora necesitan de cálculos matemáticos. Chris Harris y Mike Stephens realizaron mejoras al detector de Moravec en 1988, tomando en cuenta el diferencial de la esquina respecto a la dirección. A continuación se detalla las características del método de Moravec y de Harris.

2.1.1.1. Moravec

El algoritmo de Moravec es uno de los primeros algoritmos para la detección de esquinas [3]. El objetivo del algoritmo es probar cada pixel en una imagen mediante una pequeña ventana y buscar si hay alguna esquina presente (o un gran cambio en intensidad).

El cambio de intensidad para el desplazamiento $[u, v]$ está dado por:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

Función Ventana $w(x, y) =$ 1 si esta en la ventana, 0 si esta afuera

Figura 2.1. Cambio de intensidad para el desplazamiento $[u, v]$.

Fuente: [<http://www.cse.psu.edu/~rcollins/CSE486/>]

Entre los problemas que tiene el detector de Moravec podemos mencionar la respuesta ruidosa dada por la función de ventana, la consideración de un set de desplazamientos cada 45 grados, y que solo el valor mínimo de “E” es tomado en cuenta. Estos problemas son resueltos por el detector de esquinas de Harris.

2.1.1.2. Harris

El detector de esquinas de Harris [4] se encuentra basado en una función de autocorrelación local de una señal que mide los cambios locales de la señal con ventanas desplazadas por un pequeño número de direcciones diferentes. Dado un desplazamiento (x, y) y un punto en la función de autocorrelación, se define como:

$$c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \quad (1)$$

Donde $I = (x_i, y_i)$ representa la función de la imagen y (x_i, y_i) son los puntos en la ventana W centrado en (x, y) la imagen desplazada se aproxima por una expansión de Taylor de primer orden.

$$I(x_i + \Delta x, y_i + \Delta y) \approx \left[I(x_i, y_i) + \begin{bmatrix} I_x(x_i, y_i) & I_y(x_i, y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right] \quad (2)$$

Donde $I_x(x_i, y_i)$ y $I_y(x_i, y_i)$ indican las derivadas parciales de x y y respectivamente. Con un filtro $[-1, 0, 1]$ y $[-1, 0, 1]^T$, las derivadas parciales pueden ser calculadas sustituyendo las ecuaciones (2) en (1).

$$c(x, y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$C(x, y)$ es la matriz de autocorrelación, la cual captura la estructura de intensidad de los puntos cercanos. Sea α_1 y α_2 los valores propios de $C(x, y)$ entonces existen 3 casos que considerar:

- Ambos valores propios pequeños: región uniforme o intensidad constante.

- Ambos valores propios grandes: significa que existe una esquina.
- Un valor propio grande: significa que hay un contorno o borde.

Para encontrar los puntos de interés, se caracteriza una función $H(x, y)$ por los valores propios de $C(x, y)$. Se dice que $C(x, y)$ es simétrica y positiva si α_1 y α_2 son mayores a cero. Harris sugirió que $H(x, y) = \alpha_1 \alpha_2 - 0.04(\alpha_1 + \alpha_2)^2$ donde las esquinas pueden ser encontradas en los máximos locales de $H(x, y)$.

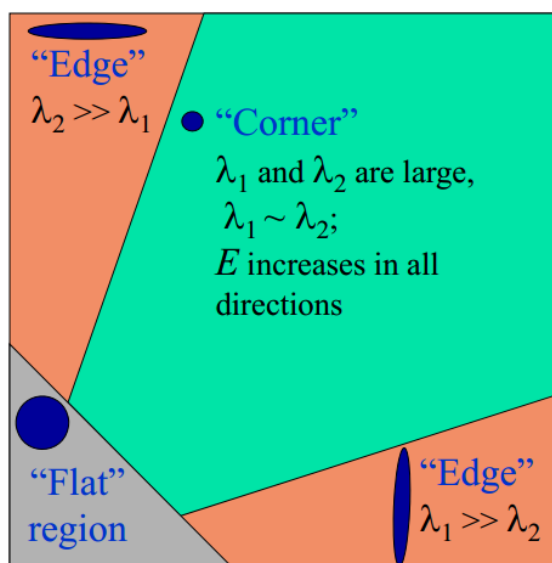


Figura 2.2. Clasificación de los puntos de imagen según los valores propios de $C(x, y)$.

Fuente: [<http://crcv.ucf.edu/courses/CAP5415/Fall2012>]

2.1.2. Detectores de puntos de interés basados en el valor de la intensidad.

Existe otra familia de detectores los cuales se basan en el valor de la intensidad de los pixeles de la imagen, en lugar de su derivada. A continuación se describen brevemente los más representativos.

2.1.2.1. SUSAN

El detector de esquinas SUSAN, acrónimo de *smallest univalue segment assimilating nucleus*, utiliza una máscara circular sobre el pixel de prueba denominada “núcleo” para detectar características [5]. La región de la máscara es representado por M , y el pixel de esta máscara es representada por $\vec{m} \in M$. Cada pixel es comparado con el núcleo \vec{m}_0 usando la siguiente función de comparación:

$$c(\vec{m}) = e^{-\left(\frac{I(\vec{m}) - I(\vec{m}_0)}{t}\right)^6}$$

Donde t determina el radio e I es el brillo del pixel. El área de SUSAN está dado por:

$$n(M) = \sum_{\vec{m} \in M} c(\vec{m})$$

Si c es una función rectangular entonces n es el número de píxeles de la máscara que se encuentran entre el t del núcleo. La respuesta del operador SUSAN está dada por:

$$R(M) = \begin{cases} g - n(M) & \text{if } n(M) < g \\ 0 & \text{otherwise,} \end{cases}$$

Donde a g se le denomina “Umbral Geométrico”. Es decir, el operador de SUSAN solo tiene un resultado positivo si el área es lo suficientemente pequeña. Para la detección de esquinas se requieren dos pasos más, primero el centroide de SUSAN debe ser encontrado, una esquina tendrá el centroide lejos del núcleo. El segundo paso consiste en que todos los puntos de una línea desde el núcleo al centroide hacia el borde de la máscara se encuentren en el SUSAN.

2.1.2.2. Basados en AST

La detección basada en AST (*accelerated segment test*) utiliza solamente los píxeles en una circunferencia de radio r alrededor del punto candidato a ser utilizado. Esta circunferencia se conoce como el círculo de Bresenham, el cual es un algoritmo utilizado para

determinar los puntos necesarios para dibujar un círculo en gráficos por computadora. El primer detector de esquinas basado en AST es FAST [6].

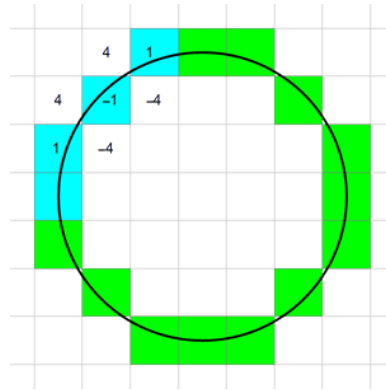


Figura 2.3. Círculo de Bresenham

Fuente: [<http://members.chello.at/easyfilter/bresenham.html>]

2.1.2.3. FAST

FAST (*features from accelerated segment test*) considera un círculo de 16 píxeles alrededor del candidato de esquina p . Denominaremos n a los píxeles continuos en el círculo que tengan mayor intensidad de brillo que el pixel candidato I_p . En el ejemplo a continuación se ha escogido $n = 12$ ya que permite una prueba rápida que excluye un gran número de píxeles no esquina, esta prueba solo examina los píxeles 1, 5, 9 y 13 (los cuatro puntos cardinales). Si p es una

esquina entonces al menos 3 de estos puntos deberán ser más brillantes que $I_p + t$ o más oscuros que $I_p - t$. Si en cualquiera de los dos ocurre lo antes mencionado, entonces p no puede ser una esquina. El detector FAST presenta un buen rendimiento, sin embargo primero requiere de entrenamiento para cada escenario.

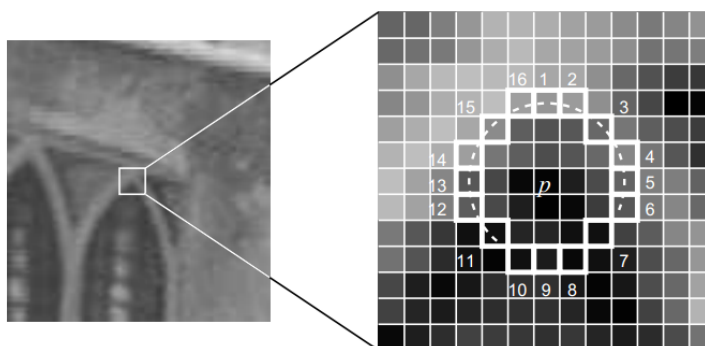


Figura 2.4. Patrón circular aplicado sobre el píxel p .

Fuente: [Computer Vision–ECCV 2006, 2006 - Springer]

2.1.2.4. FAST-ER

El detector FAST-ER (*Feature Accelerated Segment Test-Enhanced Repeatability*) es una mejora del detector FAST, utiliza el algoritmo meta-heurístico simulated annealing (algoritmo de recocido simulado) para optimizar la estructura del árbol de decisión y sea más fácil tomar cantidades mayores de iteraciones para encontrar una solución cercana a la real.

2.1.2.5. AGAST

El detector AGAST (*Adaptive and Generic Accelerated Segment Test*), se encuentra basado en el mismo criterio de detección de esquinas de FAST, sin embargo, AGAST no necesita ser entrenado para la detección de puntos de interés en una determinada escena, ya que el algoritmo elige entre dos árboles de decisiones, uno en regiones homogéneas y otro en heterogéneas.

2.2. DESCRIPCIÓN DE PUNTOS DE INTERÉS

Una vez detectado un punto de interés debe ser descrito usando características específicas de la imagen, de manera tal que independientemente del punto de vista e iluminación se lo pueda identificar de manera unívoca. Existen distintos algoritmos para describir puntos característicos, desde el más sencillo que podría ser utilizar un valor booleano que describa si el punto característico es una esquina de un objeto de la imagen, hasta usar un vector de 128 dimensiones que caracterice el punto de interés usando información de sus vecinos.

Una característica importante de los puntos de interés es que las posiciones relativas entre ellos y la escena original no deberían cambiar de una imagen a otra. Por ejemplo, cuatro esquinas de una puerta son utilizadas como puntos de interés, estos cuatro puntos pueden ser

encontrados sin importar la posición de la puerta, sin embargo, el reconocimiento podrá fallar si la puerta se encuentra abierta o cerrada. Los puntos de interés localizados en objetos flexibles podrían no funcionar si ocurre algún cambio en su geometría interna durante el procesamiento. En la práctica, para reducir errores en el reconocimiento de objetos se describen muchos puntos característicos de cara a minimizar los errores causados por las variaciones locales. Seguidamente se presentan las características principales de los descriptores más utilizados en la literatura.

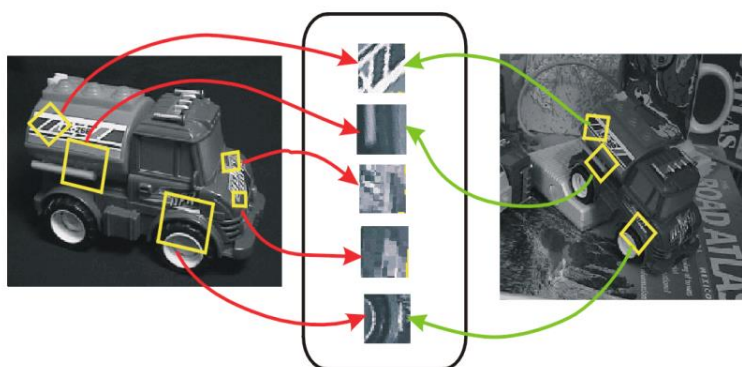


Figura 2.5. Descriptores invariantes a las transformaciones: traslación, rotación, escalamiento, brillo.

Fuente: [www.imvc.co.il/Portals/10/MosheShahar.pdf]

2.2.1. SIFT

SIFT (*Scale-invariant feature transform*) es un algoritmo para detectar y describir puntos característicos en imágenes. Los puntos característicos (también llamados puntos clave) de SIFT son extraídos de un conjunto de imágenes de referencia y guardados en una base de datos [7]. Un objeto se reconoce en una imagen nueva mediante la comparación individual de sus características con la base de datos y encontrar candidatos basados en la distancia euclidiana de sus vectores característicos. SIFT se divide en cuatro etapas: detección de extremos en una representación espacio-escala, mediante funciones de diferencia de Gaussianas; localización de puntos clave; asignación de orientación y descripción de puntos clave.

- **Detección de extremos en espacio-escala:** En esta primera etapa se pretende identificar las localidades y tamaños que son identificables desde diferentes puntos de vista del mismo objeto. Esto se puede lograr utilizando una función “espacio-escala” definida por:

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

Donde * es el operador convolución, $G(x, y, \sigma)$ es un Gaussiano e $I(x, y)$ es la imagen de entrada. Existen varias formas de detectar

puntos en espacio-escala estables, por ejemplo la diferencia de Gaussianos, encontrando extremos espacio-escala $D(x, y, \sigma)$ calculando la diferencia entre las dos imágenes, una con escala k veces el otro. $D(x, y, \sigma)$ está dado por:

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

Para detectar el máximo y mínimo locales de $D(x, y, \sigma)$ cada punto debe ser comparado con sus 8 vecinos en es la misma escala, y sus 9 vecinos arriba y abajo por una escala. Si este valor es el mínimo o máximo de todos estos puntos, entonces es considerado un extremo.

- **Localización de puntos clave:** Esta etapa pretende eliminar más puntos de la lista de puntos clave encontrando aquellos que tienen el menor contraste o son pobremente localizados en un borde. Esto se logra calculando el Laplaciano.
- **Asignación de orientación:** Esta etapa consiste en asignar una orientación consistente a los puntos clave basados en las propiedades locales de la imagen. El descriptor de puntos clave, puede ser representado relativo a esta orientación, logrando una invariancia a la rotación.

Primero usamos los puntos clave para seleccionar la imagen L (previamente filtrado por un desenfoque Gaussiano), luego calculamos la magnitud del gradiente m :

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

Calculamos la orientación θ :

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

Formamos un histograma de orientaciones de gradientes de puntos de muestra, luego localizamos el pico más alto en el histograma. Usamos este pico y cualquier otro pico que tenga el 80% de altura del primer pico para crear un punto clave con esa orientación. Finalmente ajustamos los tres valores cercanos a cada pico en una parábola para que se interpole con la posición del pico.

- **Descripción de puntos clave:** La información de gradiente es rotada para alinearse con la orientación del punto clave y luego es pesada por un Gaussiano con varianza de $1.5 * \text{escala del punto clave}$. Esta información luego es usada para crear un conjunto de histogramas sobre una ventana centrada en el punto clave.

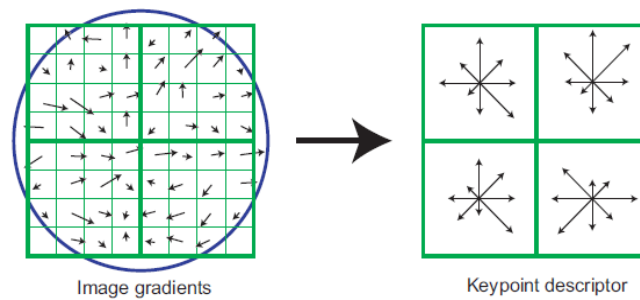


Figura 2.6. La magnitud y orientación del gradiente local son agrupadas para crear un descriptor de puntos característicos.

Fuente: [<http://correll.cs.colorado.edu/?p=2048>]

2.2.2. SURF

SURF (*Speed Up Robust Feature*) consta de cuatro etapas similares al algoritmo de SIFT. La versión estándar de SURF es muchas veces más rápida que SIFT, y según sus autores, más robusto que SIFT en cuanto a diferentes transformaciones de imágenes se refiere [8]. Las etapas del algoritmo SURF son las siguientes: generación de imagen integral para realizar cálculos rápidos; creación del espacio-escala utilizando aproximaciones a la segunda derivada de la Gaussiana; localización de puntos clave; asignación de orientación (utilizando wavelets Haar) y descripción de puntos clave.

2.2.3. BRISK

BRISK (*Binary Robust Invariant Scalable Keypoints*) es un método rápido y de bajo costo computacional para la detección de puntos clave, descripción y correspondencia. Su velocidad reside en la aplicación de un detector basado en FAST en el espacio-escala, en combinación con un descriptor cuyo vector de características es una cadena de bits obtenida a partir de comparaciones de intensidad de cada punto clave de los vecinos [9].

- **Detector de espacio-escala:** El objetivo es buscar el máximo no solo en el plano de la imagen, sino también en el espacio-escala utilizando resultados de FAST (s) como forma de medición. Se utiliza el detector FAST 9-16 en cada octava y entre sus octavas próximas arriba y abajo, como se muestra en la Figura 2.7. En las tres capas de interés, la prominencia local máxima es refinada por un sub-píxel ante una parábola ajustada a lo largo de la escala-eje para determinar la escala real del punto clave. La ubicación del punto clave se re-interpola entre el parche máximo más cercano a la escala determinada.

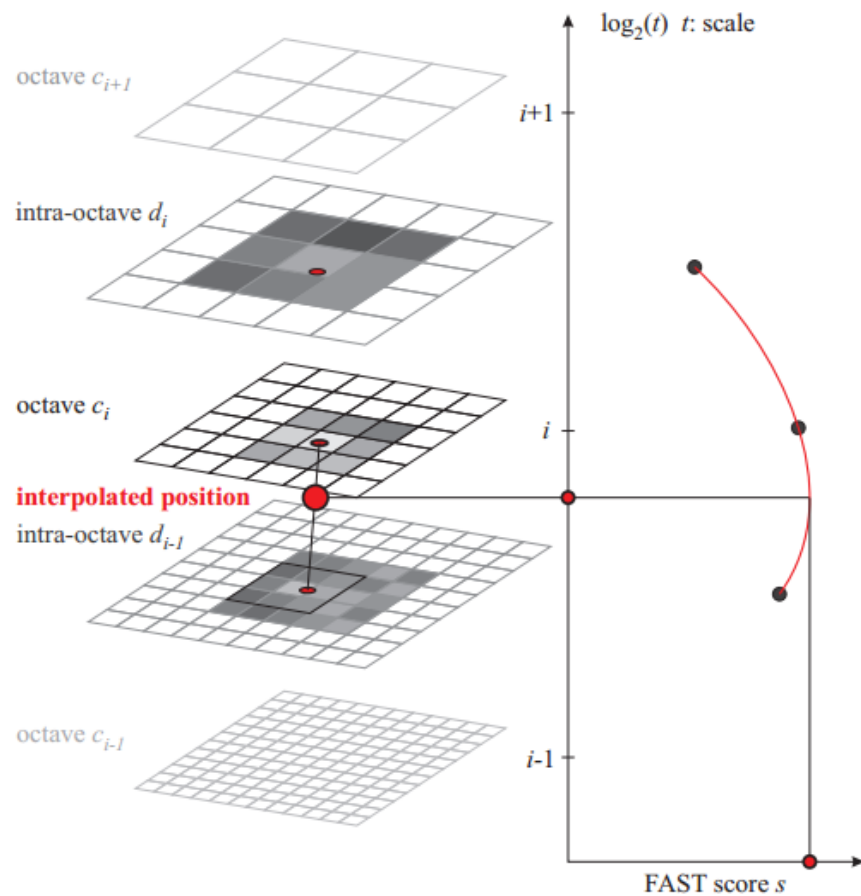


Figura 2.7. Detector espacio-escala.

Fuente: [<http://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf>]

- **Descriptor de puntos característicos:** Dado un conjunto de puntos clave, el descriptor BRISK se compone de una cadena binaria, concatenando los resultados de simples pruebas de comparación de niveles de intensidad. En BRISK, se identifica la dirección característica de cada punto clave para permitir descriptores normalizados por su orientación y por lo tanto, lograr

una invariancia la cual es la clave de una robustez general. También se debe seleccionar cuidadosamente las comparaciones de niveles de intensidad para maximizar la descripción.

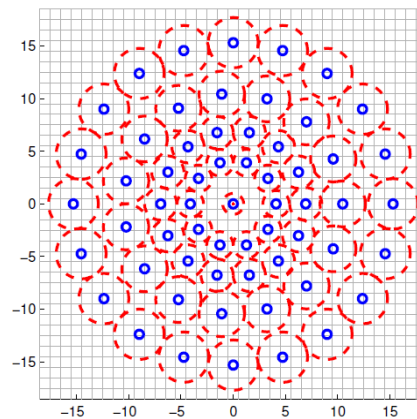


Figura 2.8. El patrón de muestreo BRISK con $N = 60$ puntos.

Fuente: [<http://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf>]

- **Correspondencia de descriptores:** Encontrar la correspondencia entre dos descriptores BRISK es un simple cálculo de la distancia de Hamming como se lo hace en BRIEF [10]: el número de bits diferentes en dos descriptores es una medida de disimilitud.

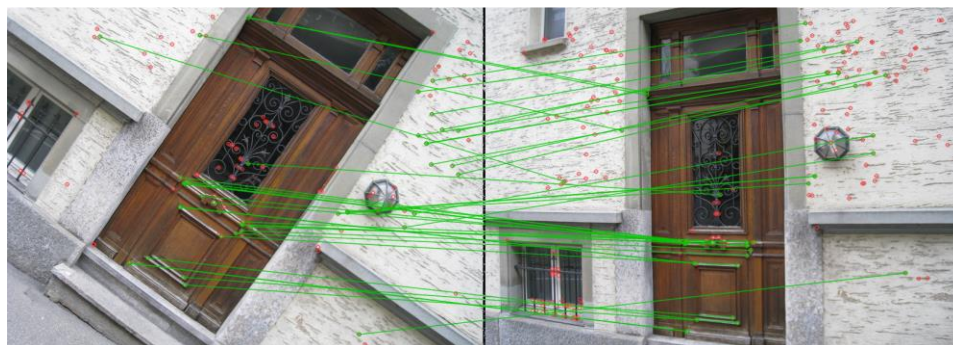


Figura 2.9. Ejemplo de matching con BRISK. Se utiliza un umbral de detección de 70 y un umbral de distancia de Hamming de 90.

Fuente: [<http://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf>]

2.3. DESCRIPCIÓN DE PUNTOS DE INTERÉS EN IMÁGENES MULTIESPECTRALES: VS-NIR, VS-LWIR

El análisis multiespectral ha sido objeto de atención durante los últimos años en el área de procesamiento de imágenes y videos [11]. La información que proveen las imágenes de diferentes bandas espectrales puede ser directa o suplementaria a la información proporcionada por el espectro visible, lo que puede ayudar a resolver problemas más eficientemente. El procesamiento de imágenes de bandas espectrales fuera del espectro visible requiere el desarrollo de nuevas herramientas o adaptación de los métodos ya existentes. El análisis multiespectral ha sido ampliamente estudiado en el área de sensores remotos como lo son los satélites, sin embargo, gracias los avances en la tecnología, ahora las imágenes multiespectrales se

pueden utilizar en aplicaciones de cámaras de seguridad o asistencia a la conducción de vehículos.

Los puntos característicos son la base de diferentes problemas de visión por computadora y generalmente son estudiados en tres niveles: detección, descripción y correspondencia. Uno de los enfoques más sobresalientes es el algoritmo SIFT [11]. Durante la última década, se han realizado varias modificaciones al algoritmo original con el objetivo de mejorar las debilidades de SIFT. En el caso en particular en las que las imágenes correspondan a distintas bandas del espectro, descriptores clásicos como SIFT y SURF requieren ser adaptados [12].

La dificultad de hallar correspondencias entre puntos característicos de imágenes VS-LWIR se haya en la relación no lineal entre las intensidades de sus píxeles. En las imágenes LWIR la intensidad de brillo se encuentra relacionada a las variaciones de temperatura de los objetos, mientras que en VS el color del objeto proviene del reflejo de luz. Esto produce una falta de correlación en los resultados entre sus respectivos gradientes. Otra característica de las imágenes LWIR es que poseen pérdida de detalle y textura [13]. También hay que tomar en cuenta que las imágenes LWIR poseen otro tipo de información que no se encuentra presente en la visible, por ejemplo, cuando no hay suficiente iluminación en una escena, como en la oscuridad de la noche.

A continuación se exponen diferentes descriptores para la búsqueda de correspondencias entre imágenes de distintas bandas del espectro.

2.3.1. GOM-SIFT

GOM-SIFT (*Gradient Orientation Modification*) modifica la orientación de cada punto característico detectado con SIFT para obtener un valor de orientación entre 0° y 180° con el fin de mejorar el rendimiento ligeramente [14].

$$\beta(\alpha) = \begin{cases} \alpha & \text{si } \alpha \in [0,180] \\ 360 - \alpha & \text{si } \alpha \in (180,360) \end{cases}$$

Donde α es el gradiente de orientación calculado por SIFT y β es el resultado de la modificación propuesta.

Para eliminar la cantidad de correspondencias encontradas incorrectamente, se plantea un criterio de restricción durante el proceso de búsqueda de correspondencias. Dado un par de puntos característicos $P_1(x_1, y_1, \sigma_1, \theta_1)$ y $P_2(x_2, y_2, \sigma_2, \theta_2)$, donde (x_i, y_i) representa la ubicación, σ_i la escala y θ_i la orientación. Se le denomina diferencia de escala (SD) como:

$$SD(P_1, P_2) = \sigma_1 - \sigma_2$$

Asumiendo que la transformación espacial entre un par de imágenes es la misma, las SDs de correspondencias correctas deberían converger a un valor constante, lo que significa que la desviación estándar de las SDs de correspondencias correctas es mucho menor que las incorrectas. Planteándose la siguiente restricción:

$$a < SD(P_1, P_2) < b$$

Solo los puntos característicos cuyas diferencias de escala (SD) se encuentren acotadas entre dos valores (a, b) serán considerados como correspondencias correctas.

2.3.2. OR-SIFT

OR-SIFT (*Orientation Restricted SIFT*) combina los elementos del vector característico del descriptor SIFT en direcciones opuestas, de tal manera que consigue un vector de descripción más corto, y por lo tanto, menor será la complejidad en el proceso de búsqueda de correspondencias [15].

Debido a la modificación hecha por SIFT, el número de correspondencias encontradas en la rotación se reduce, lo cual significa que no es invariante a la rotación, por lo que se propone modificar el proceso de obtención del vector característico. Cada uno de los ocho

contenedores de orientación de SIFT es combinado con el contenedor que difiere en 180° , de tal manera que, el contenedor con orientaciones de $45^\circ - 89^\circ$ es combinado con el que posee orientaciones $225^\circ - 270^\circ$ para formar un único contenedor, por lo que el tamaño del vector característico se reduce a la mitad.

1	0-45°	1	0-45° and 180-225°
2	45-90°	2	45-90° and 225-270°
3	90-135°	3	90-135° and 270-315°
4	145-180°	4	145-180° and 315-0°
5	180-225°		
6	225-270°		
7	270-315°		
8	315-0°		

Figura 2.10. Contenedor de orientaciones SIFT a la izquierda. Contenedor de orientaciones modificadas a la derecha.

Fuente: [<http://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf>]

2.3.3. EOH-SIFT

EOH-SIFT (*Edge Oriented Histogram*) es un descriptor que permite encontrar correspondencias de puntos de interés en imágenes de la misma escena pero de diferentes bandas espectrales [15]. En este método, los puntos de interés son detectados en ambas imágenes a través de representaciones espacio-escala basadas en SIFT. Luego, estos puntos son caracterizados usando un descriptor EOH (*Edge Oriented Histogram*). Finalmente, los puntos de interés de la imagen

multiespectral se emparejan encontrando las parejas más cercanas usando la información del descriptor.

El esquema consiste en el uso de una representación en espacio-escala piramidal, usando características invariantes pero modificando el vector de características de tal manera que incorpore información espacial de los contornos de cada punto clave sin usar la información de gradiente. Esto permite generar un parámetro correlacionado en ambas imágenes. En la Figura 2.11 se muestra un diagrama de flujos del método.

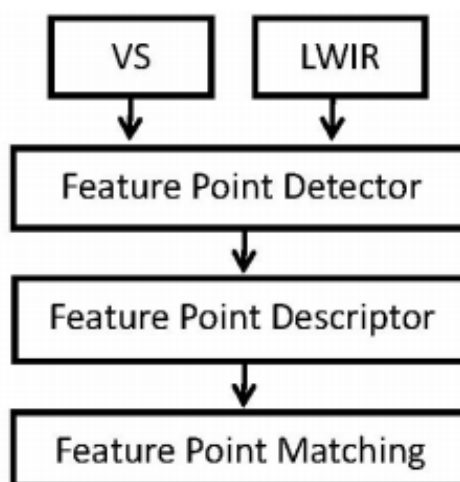


Figura 2.11. Diagrama de flujos del método EOH-SIFT

Fuente: [15]

Detección de puntos característicos: Los puntos característicos son detectados mediante el uso de una representación piramidal de

espacio-escala. El resultado de esta etapa es un conjunto de puntos clave, similares a los que se obtienen por medio de SIFT, incluyendo su invariancia. En esta implementación los puntos clave potenciales se obtienen ajustando los siguientes parámetros: $\sigma = 1.2$ y umbral = 40. Los mismos parámetros son usados también en las imágenes multiespectrales resultando en un conjunto de puntos clave P_{VS} y P_{LWIR} . Cada punto clave es denotado por un vector (x_i, y_i, σ_i) donde (x_i, y_i) corresponde a la ubicación y (σ_i) es la escala de la representación piramidal donde aparece el punto clave.

Descripción de puntos característicos: Los puntos característicos detectados son descritos a través del uso de un histograma de orientación de borde (EOH). Este histograma incorpora información espacial de los bordes en el vecindario de cada punto característico, el cual describe la forma y contornos de ambas imágenes VS y LWIR. La idea de este descriptor es motivado por la relación no lineal presente entre las intensidades de las imágenes multiespectrales, por lo tanto el descriptor debe basarse en la información de los contornos, mas no de la intensidad de brillo. Inicialmente, los bordes son extraídos utilizando

el algoritmo de Canny en ambas imágenes, luego los puntos característicos son detectados como se muestra en la Figura 2.12.

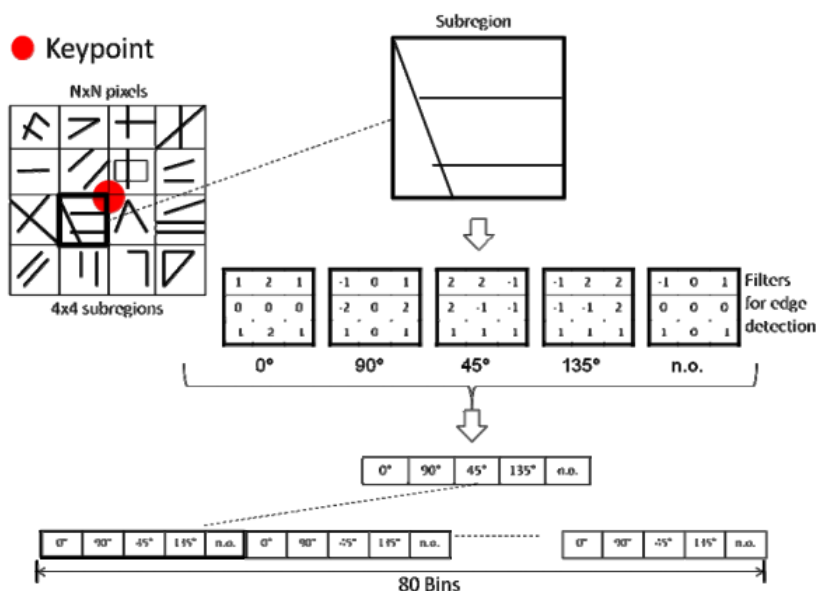


Figura 2.12. Descriptor basado en EOH

Fuente: [15]

Primero se obtiene una región de $N \times N$ píxeles, centrada en el punto clave dado, luego esta región se divide en $4 \times 4 = 16$ subregiones. Finalmente cada uno de esas subregiones se representa por un histograma de contornos. Este histograma representa la distribución espacial de cuatro bordes direccionales y de un borde no direccional, siendo en total cinco; estos contenedores corresponden a las orientaciones del contorno de 0, 45, 90 y 135 grados; también se

considera un quinto contenedor sin orientación, que corresponde a las áreas donde no se encuentren contornos.

Cada pixel de cada subregión contribuye a un contenedor del histograma de acuerdo con los cinco filtros, de 3 x 3 píxeles, que se muestran en la Figura 2.12. Después de procesar todos los elementos de las 16 subregiones, se obtiene un vector con 80 elementos ($16 \times 5 = 80$). Este vector se normaliza y es denominado vector de características, el cual también será referido como vector descriptor para cada punto clave.

La selección del tamaño de ventana correcto (N) es un factor importante para este esquema. Una ventana muy pequeña o muy grande podría incrementar el número de correspondencias erróneas de los puntos clave.

Correspondencia de puntos característicos: En esta etapa se encuentran los puntos clave más cercanos de diferentes imágenes espectrales, filtrando vectores de características que tengan pocos elementos descriptivos. Este proceso se basa en la distancia Euclidiana entre vectores descriptores correspondientes. Al igual que en el algoritmo SIFT, para incrementar la robustez de la correspondencia, dos puntos son asociados si la relación entre la distancia del primer y

segundo punto más cercano es inferior a un cierto valor. Adicionalmente se descartan puntos que no contengan información en sus subregiones. Finalmente, se impone una restricción en la diferencia de escala (SD) en los puntos, el cual consiste en descartar coincidencias erróneas en el par de puntos clave dados.

Los puntos son descartados si no satisfacen el siguiente criterio de restricción:

$$a < SD < b$$

Donde a y b se obtienen calculando un histograma de SD s de todas las coincidencias, luego se definen nuevos valores utilizando el histograma \overline{SD} , que contiene el pico en el histograma SD .

$$a = \overline{SD} - 0.9 \quad b = \overline{SD} + 0.9$$

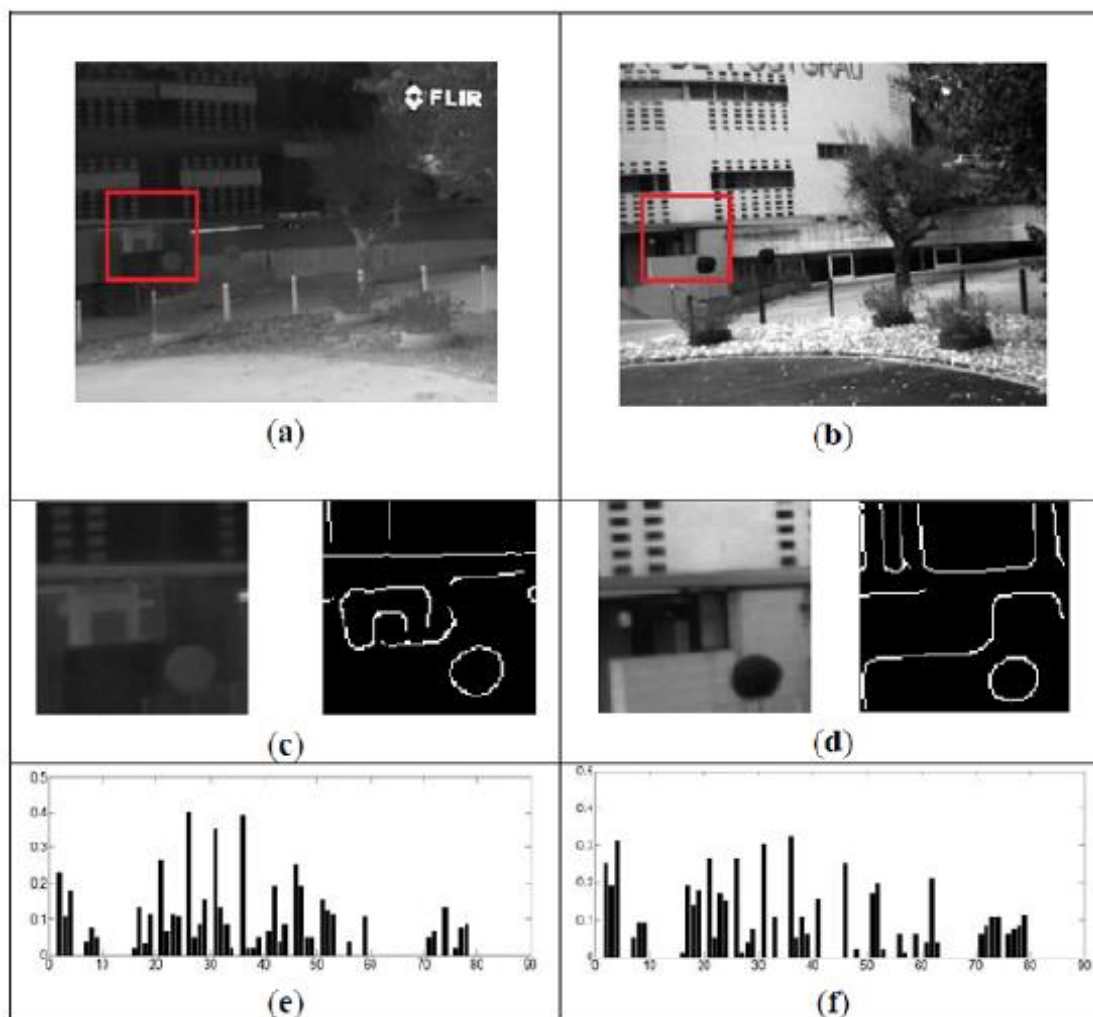


Figura 2.13. Pasos del método de descripción de puntos característicos para pares de imágenes VS-LWIR: (a) y (b) par de puntos clave en las imágenes LWIR y VS junto con su vecindad correspondiente; (c) y (d) imágenes de contornos de los puntos detectados; (e) y (f) histogramas utilizados como vectores de descripción.

Fuente: [15]

CAPÍTULO 3

DESARROLLO

Basados en la literatura mencionada en el capítulo anterior, se propone un nuevo método para la búsqueda de correspondencias para líneas y círculos entre pares multispectrales VS-LWIR, a continuación se presenta el diagrama de flujo del método propuesto.

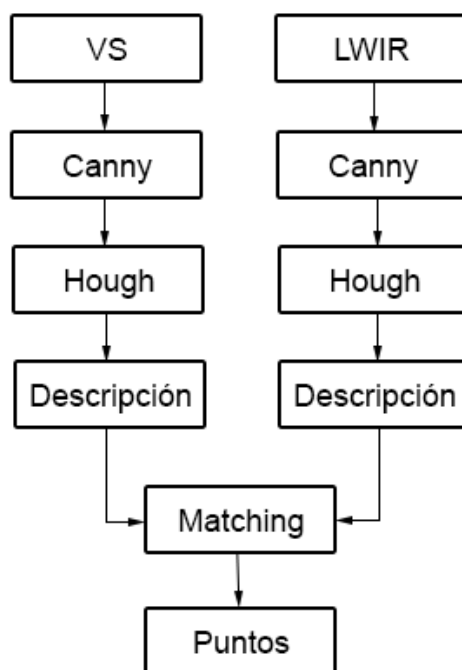


Figura 3.1. Diagrama de flujo del método propuesto.

El método consiste en analizar las imágenes VS-LWIR por separado, utilizando el método de Canny para detección de bordes [17], con el fin de suavizar las imágenes. Debido a que una de las imágenes es de espectro diferente, no se podrá utilizar métodos clásicos a niveles de puntos; para esto utilizamos la transformada de Hough con el objetivo de llevar ambas imágenes a un dominio común donde ambos están representados. Finalmente se realizará el matching para encontrar correspondencia de líneas y círculos en el par de imágenes.

3.1. DETECCIÓN DE CONTORNOS

El detector de contornos Canny es uno de los métodos más utilizados en la actualidad, detectando los bordes de una manera robusta [18]. La detección de contornos de Canny es un proceso de cuatro pasos:

- Suavizado de la imagen
- Cálculo de la magnitud y ángulo del gradiente
- Supresión de no-máximos
- Aplicación de umbral con histéresis

3.1.1. Suavizado de la imagen

Se aplica un filtro Gaussiano $G(x, y)$ para suavizar y eliminar el ruido existente en la imagen, se recomienda que se utilice una imagen en la escala de grises para mejores resultados.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Donde σ determina el nivel de suavizado, mientras más alto el valor, mayor inmunidad al ruido.

3.1.2. Cálculo de la magnitud y ángulo del gradiente

Se calculan las derivadas en las direcciones x y y ($D_x(x, y), D_y(x, y)$) de cada pixel de la imagen suavizada, con el fin de obtener la magnitud $D(y, x)$ y ángulo de los gradientes $\theta(x, y)$:

$$D(y, x) = \sqrt{D_x^2(x, y) + D_y^2(x, y)}$$

$$\theta(x, y) = \arctan\left(\frac{D_x(x, y)}{D_y(x, y)}\right)$$

Luego se obtiene $\tilde{\theta}(x, y)$ redondeando los valores de $\theta(x, y)$ en cuatro direcciones: $0^\circ, 45^\circ, 90^\circ, 135^\circ$.

3.1.3. Supresión de no-máximos

En este paso se convierten los bordes borrosos en bordes más marcados. Esto se logra preservando todos los máximos locales en la imagen gradiente y eliminando el resto. Se redondea la dirección del gradiente θ a los 45° más cercanos, para esto se utiliza la información proporcionada por $D(y, x)$ y $\tilde{\theta}(x, y)$ de la siguiente manera:

- Si $\tilde{\theta}(x, y) = 0$ los pixeles $(x + 1, y)$, (x, y) y $(x - 1, y)$ son examinados.
- Si $\tilde{\theta}(x, y) = 45$ los pixeles $(x, y + 1)$, (x, y) y $(x, y - 1)$ son examinados.
- Si $\tilde{\theta}(x, y) = 90$ los pixeles $(x + 1, y + 1)$, (x, y) y $(x - 1, y - 1)$ son examinados.
- Si $\tilde{\theta}(x, y) = 135$ los pixeles $(x + 1, y - 1)$, (x, y) y $(x - 1, y + 1)$ son examinados.

Si el pixel (x, y) tiene un gradiente de magnitud más alto que el resto, se preserva el gradiente de magnitud, caso contrario se remueve el valor.

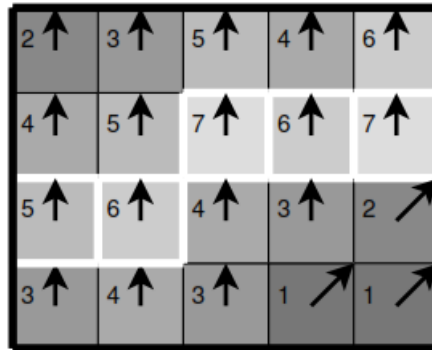


Figura 3.2. Ilustración de una supresión de no-máximos. El grosor de los bordes se encuentra indicado con un color y un número, mientras que la dirección se muestra como flechas. Los píxeles del borde resultante tienen los bordes blancos.

Fuente: [<http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf>]

3.1.4. Aplicación de umbral con histéresis

El umbral utilizado en el operador Canny utiliza un método llamado "Histéresis". La mayoría de umbrales usan un solo límite, lo que significa que si el valor del borde oscila por encima y por debajo del valor, la línea se verá entrecortada. La histéresis se encarga de este problema, estableciendo un límite superior t_{high} e inferior t_{low} . Si consideramos un segmento de línea, si un valor se encuentra por encima de límite del superior, se acepta inmediatamente. Si el valor se encuentra por debajo del límite inferior, se rechaza automáticamente. Si el valor se encuentra

entre los dos límites, será aceptado solo si está conectado a los pixeles que presentan una respuesta robusta (fuerte).

3.2. TRANSFORMADA DE HOUGH

La transformada de Hough es una técnica para la extracción de características, utilizada en el análisis de imágenes, visión por computadora y procesamiento digital de imágenes. El algoritmo fue presentado por Paul Hough en 1962 [19] para la detección de líneas en una imagen, pero después fue extendida para la detección de formas arbitrarias, como círculos o elipses.

La transformada de Hough está definida para una función $A(x, y)$ como se indica a continuación:

$$H(\theta, \rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(x, y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy$$

Donde A es una imagen binaria, cada pixel en (x, y) es transformado en una onda senoidal $\rho = x \cos \theta - y \sin \theta$.

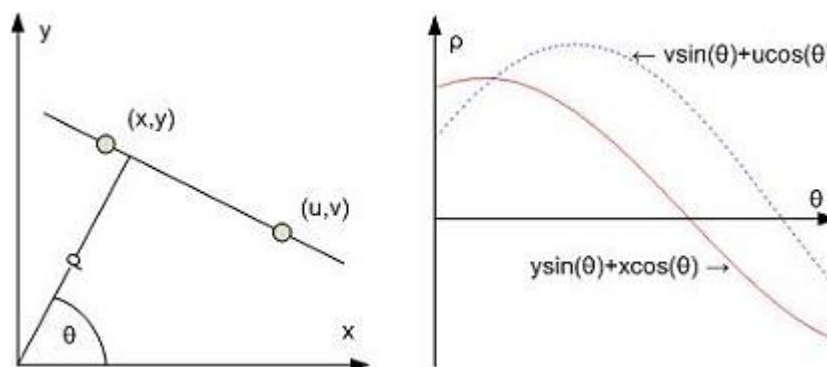


Figura 3.3. Transformada de Hough.

Fuente: [<http://campar.in.tum.de/Students/DaPentnerieder>]

Cada punto del plano $x - y$ es representado por una curva sinusoidal en el plano $\rho - \theta$. Los puntos colineales se mapean en curvas que tienen un punto de intersección común. Por lo tanto las líneas en el plano $x - y$ pueden ser detectadas contando el número de curvas que se intersecan en cada punto del plano $\rho - \theta$. Esto se logra con un arreglo acumulador.

3.2.1. Transformada de Hough para curvas paramétricas

Como habíamos mencionado, la transformada de Hough no solo se encuentra restringida para la detección de líneas rectas, también puede ser utilizada para detectar otro tipo de curvas en la imagen. Por ejemplo para encontrar círculos, utilizamos la ecuación:

$$(x - a)^2 + (y - b)^2 = r^2$$

Donde el círculo se define con estos tres parámetros: la x e y son las coordenadas de su centro y r su radio. Para encontrar círculos utilizando la transformada de Hough, se necesita un arreglo acumulador de tres dimensiones.

3.2.2. Detección de Líneas y Círculos

Las líneas y círculos son detectados por la interpretación del acumulador cuando todos los puntos hayan sido transformados. Un ejemplo de este proceso se muestra en la Figura 3.4. La manera más básica de detectar líneas o círculos es estableciendo un umbral para el acumulador, e interpretar todos los valores por encima de este umbral.

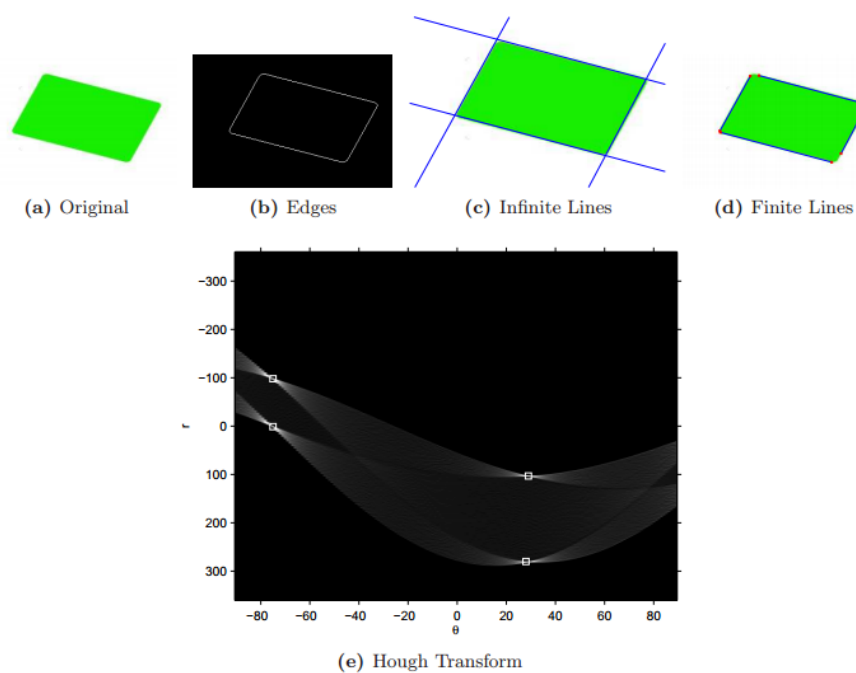


Figura 3.4. Detección de líneas utilizando la Transformada de Hough.

Fuente:

[http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/HoughTrans_lines_09.pdf]

Este umbral deberá tener un valor adecuado para una correcta detección, el cual será variable dependiendo del tamaño y contenido de la imagen. Por lo tanto dos imágenes de igual tamaño pero con diferente composición morfológica y niveles de ruido necesitaran distintos valores de umbral.

También hay que tomar en cuenta que los valores máximos globales no son detectados, ya que tener un umbral fijo genera

una serie de máximos locales que implicaría la detección de líneas y círculos erróneos.

3.3. DESCRIPCIÓN DE CARACTERÍSTICAS

En el siguiente apartado se detallarán los resultados obtenidos por los detectores y descriptores mencionados anteriormente. El análisis se realizó en varios pares de imágenes VS-LWIR de dos tipos de sets diferentes, imágenes que contienen líneas y círculos respectivamente.



Figura 3.5. El set de prueba. Imagen de la izquierda: visible.
Imagen de la derecha: infrarrojo.

Para la detección de bordes Canny se utilizó la función CV:Canny presente en las librerías de OpenCV, el cual utiliza dos umbrales; un umbral inferior σ_{low} y uno superior σ_{high} , donde σ_{high} se sugiere que sea un valor 3 veces más que el de σ_{low} (siguiendo la recomendación

de Canny). Por esto la interfaz tendrá por defecto $\sigma_{low} = 50$, con la opción de asignarle un valor entre 0 y 100, mientras que σ_{high} será un valor constante de 150 (3 veces el valor de σ_{low}).



Figura 3.6. Canny de imagen VS.



Figura 3.7. Canny de imagen LWIR.

En la Figura 3.8 se puede apreciar que en la imagen LWIR se pierde mucha más información en comparación de la imagen VS, debido al aspecto suavizado que esta posee. Por consecuencia de esto, muchos bordes no son considerados como tales.

Un acumulador es utilizado en el espacio de Hough para determinar las áreas donde haya una mayor cantidad de intersecciones. Cada vez que un punto es transformado, el contenedor del acumulador se incrementa por todas las líneas que puedan pasar por ese punto. La resolución del acumulador determina la precisión con la que las líneas o círculos puedan ser detectadas.

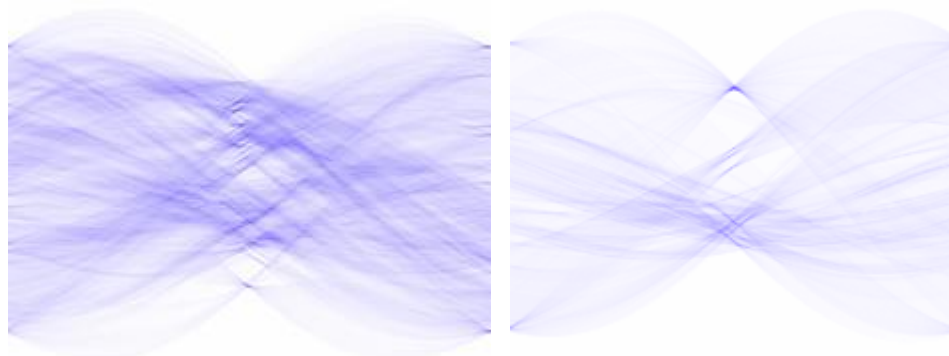


Figura 3.8. Resultado del cálculo de la transformada de Hough (correspondiente a las Figuras 3.6 y 3.7) para detección de líneas en la imagen VS (izquierda) y LWIR (derecha).

Posteriormente las líneas y círculos son detectados en el par de imágenes como se muestra en la Figura 3.9. Hay que tomar en cuenta que para obtener una detección robusta se deberá manipular los valores de los umbrales, tanto el de Canny como de la transformada de Hough.



Figura 3.9. Detección de líneas y círculos en el par de imágenes.

3.4. MATCHING

La búsqueda de correspondencias se realiza comparando cada punto del acumulador en ambos espacios de Hough (VS y LWIR) hasta encontrar puntos de intersección iguales. Si existe coincidencia, solo se mostrará las líneas o círculos que compartan las mismas características. Para realizar un matching robusto se necesitará modificar los valores de los diferentes umbrales, el análisis y evaluación de dichas pruebas se encuentran en el capítulo 4 del presente trabajo.



Figura 3.10. Correspondencia de líneas y círculos en el par de imágenes.

3.5. INTERFAZ GRÁFICA DESARROLLADA

En la Figura 3.11. podemos observar la interfaz gráfica diseñada para el presente proyecto, el cual fue desarrollado utilizando el framework gráfico Qt. El resto de algoritmos y funcionalidades utilizados en el proyecto fueron implementados en C++. La descripción de los diferentes elementos presentes en la interfaz se detalla a continuación.



Figura 3.11. Interfaz gráfica desarrollada.

- (1) y (2): Se muestran las líneas o círculos presentes luego de aplicar la transformada de Hough a los contornos extraídos

utilizando el algoritmo de Canny a las imágenes VS (3) y LWIR (4) respectivamente.

- (3) y (4): Contornos de las imágenes VS y LWIR respectivamente cargadas.
- (5) y (7): Selección del tipo de Transformada de Hough a ser procesada, puede ser para detectar líneas, círculos o ambos.
- (6) y (8): Barra de desplazamiento que permite modificar el valor del umbral del acumulador de la transformada de Hough, el botón “show” nos muestra una imagen de como se ve el acumulador.
- (9) y (11): Selección de la vista previa de la imagen, puede ser la original o en escala de grises.
- (10) y (12): Barra de desplazamiento que permite modificar el valor del umbral de Canny.
- (13) y (14): Botón que permite ampliar la imagen y verla en sus píxeles reales.
- (15): Permite encontrar líneas y círculos en ambas imágenes. (Matching) manteniendo sus valores de umbrales individuales.
- (16): Barra de estado, muestra información relevante sobre lo que ocurre en las diferentes etapas del proceso.
- (17): Menú con las siguientes opciones: cargar imagen VS, cargar imagen LWIR, salir.

CAPÍTULO 4

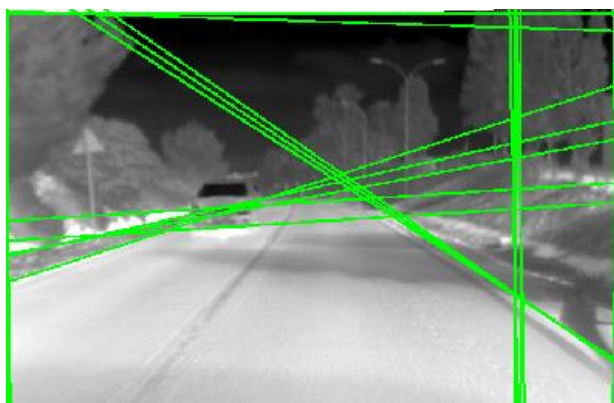
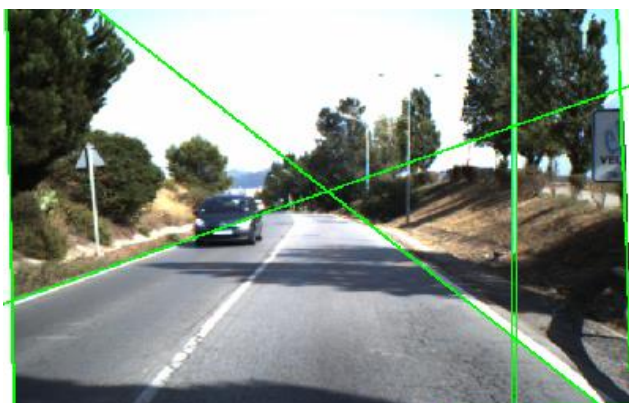
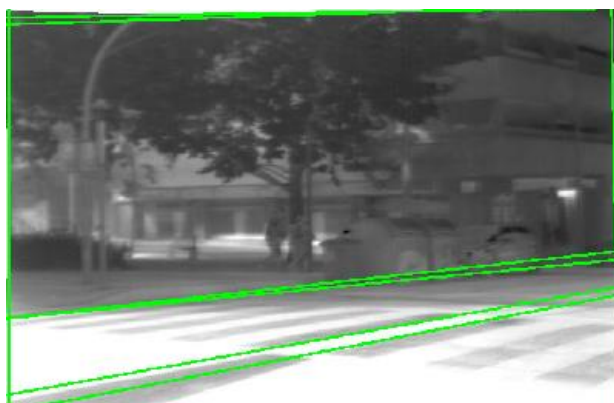
EVALUACIÓN Y ANÁLISIS DE RESULTADOS

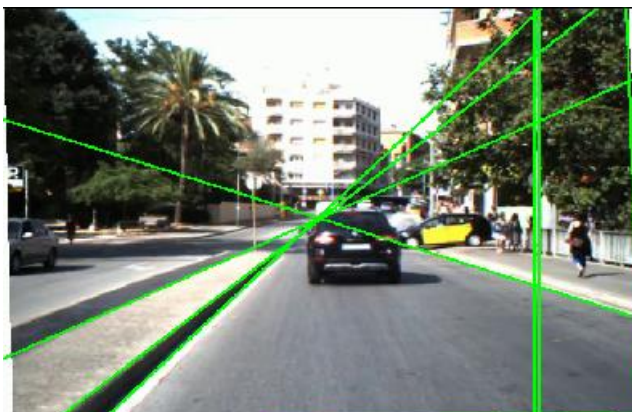
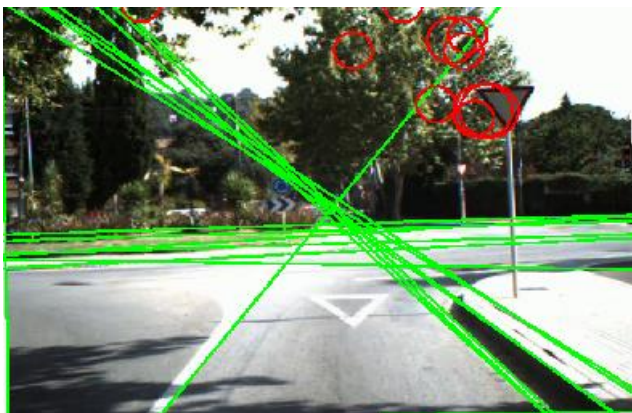
En este capítulo se presenta una evaluación de los resultados obtenidos con la técnica propuesta y un análisis de los mismos. Del análisis mencionado anteriormente se proponen mejores que son detalladas al final de este capítulo.

4.1. RESULTADOS EXPERIMENTALES

Seguidamente se presentan los resultados experimentales obtenidos de usar la técnica propuesta en la base de datos del Centro de Visión por Computador¹. En todos los casos se ha utilizado el mismo conjunto de parámetros para los umbrales descriptos en la Figura 3.11.

¹ <http://www.cvc.uab.es/adas/projects/simeve/>





4.2. PROPUESTA DE MODIFICACIONES

A partir del análisis realizado en las imágenes de prueba, alguna de las cuales fueron mostradas en la sección anterior, se identificaron los siguientes problemas:

- Reducido número de correspondencias.
- Detección de líneas y círculos no presentes en el par de imágenes (falsos negativos).

Para atenuarlas, se proponen las siguientes correcciones:

4.2.1. Rectificación manual de imágenes

Las imágenes proporcionadas por el par estereoscópico multiespectral (VS-LWIR) son imágenes de una misma toma, sin embargo no son exactamente iguales ya que fueron tomadas por cámaras que tenían una pequeña separación entre sus ejes ópticos. Esto significa que el par de imágenes de prueba, a pesar de que fue rectificado utilizando los parámetros obtenidos al inicio del proceso de adquisición, podrá tener un pequeño desfase de píxeles como se puede observar en la Figura 4.1. Para resolver este problema, se sobreponen las imágenes con menor opacidad y se cuadran hasta ser lo más precisos posibles. Este proceso manual se realiza solo a los efectos de obtener pares de imágenes perfectamente rectificadas que será usada como referencia.

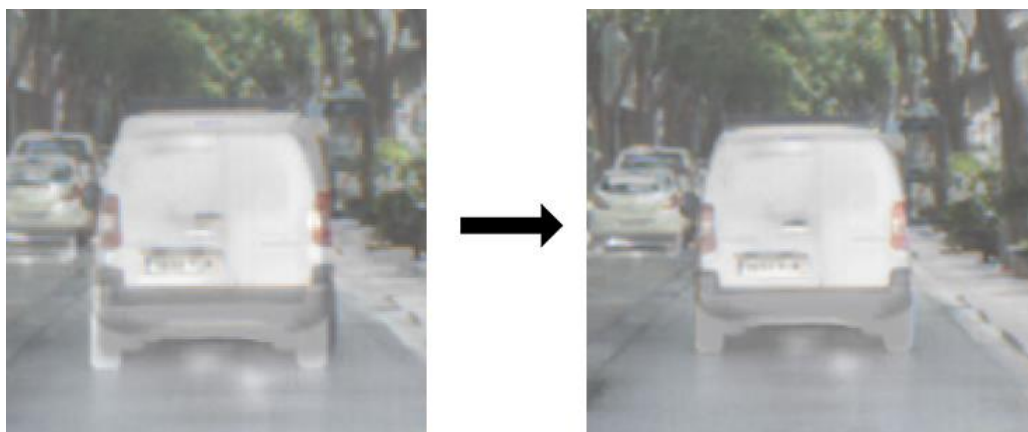


Figura 4.1. Rectificación manual de la imagen.

4.2.2. Modificación del umbral de Canny para imágenes LWIR

El detector Canny es comúnmente utilizado en imágenes de espectro visibles dando muy buenos resultados, sin embargo, debido a la naturaleza de la imagen LWIR (el suavizado, por ejemplo) mucha información se pierde utilizando los valores de umbrales recomendados por defecto en el algoritmo Canny. Para resolver este problema, se recomienda utilizar valores de umbrales diferentes en las imágenes LWIR. En la Figura 4.2. se puede observar la diferencia entre detectores de Canny con diferentes valores de umbral máximos σ_{high} , se puede observar que con un menor σ_{high} aparecerán líneas que antes no eran visibles.

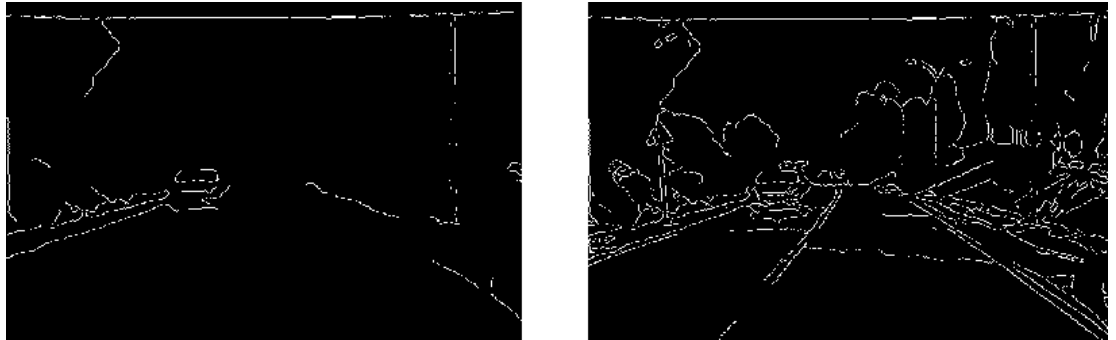


Figura 4.2. Modificación de umbral en la imagen LWIR. En la izquierda el umbral se encuentra en sus valores recomendados: $\sigma_{high} = 150$ (3 veces σ_{low}). En la derecha se encuentra el umbral personalizado:

$$\sigma_{high} = 50.$$

Es muy importante que la detección de bordes sea óptima en ambas imágenes ya que impacta directamente sobre la transformada de Hough y las correspondencias encontradas. Mientras más robusta sea la detección de bordes, existirá menor cantidad de falsos positivos.

4.2.3. Análisis espacio-escala

Con el fin de mejorar los resultados se ha utilizado una representación espacio-escala, tanto a la hora de obtener los contornos utilizando el detector Canny como a la hora de extraer las líneas y círculos más representativos del espacio de Hough. A continuación se detalla el enfoque que se evaluó para obtener una mayor cantidad de correspondencias:

Para mejorar los resultados obtenidos, se propone modificar el diagrama de flujo del método propuesto en el capítulo 3. El objetivo es utilizar diferentes valores de umbral en el detector de Canny para obtener diferentes contornos de la imagen LWIR. Luego se obtiene la transformada de Hough para la imagen LWIR y se busca correspondencia con la imagen VS. Las correspondencias obtenidas con los distintos valores de umbral son acumuladas en una única representación. En la Figura 4.3. se muestra el diagrama de flujo del método propuesto.

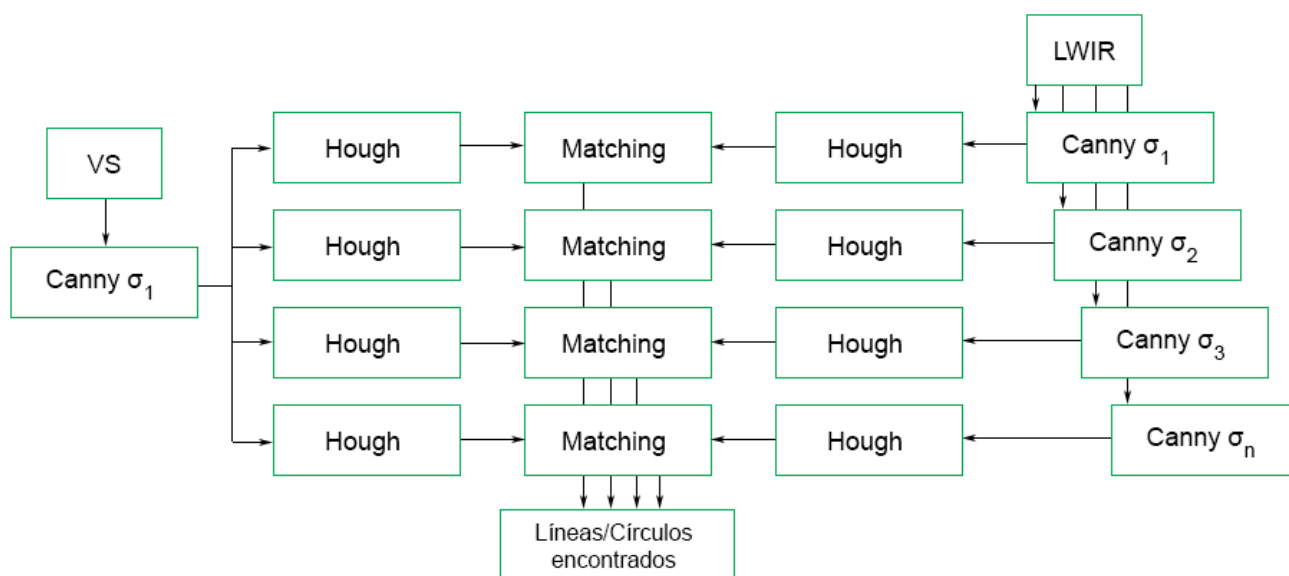


Figura 4.3. Diagrama del flujo del método propuesto.

CONCLUSIONES

A partir del análisis empírico sobre el par de imágenes expuesto en el presente trabajo se llegó a las siguientes conclusiones:

1. Utilizar valores no convencionales en los umbrales en la imagen LWIR permite obtener una detección de bordes más robusta.
2. Utilizando el valor recomendado para el umbral de la transformada de Hough (umbral de 50, en una escala de 0 a 100) se obtendrá una mejor detección tanto en líneas como en círculos.
3. La modificación manual en el proceso de rectificación de imágenes mejoró notoriamente la cantidad de correspondencias encontradas.
4. Mientras más grande sea el valor de los umbrales de Canny, la cantidad de correspondencias disminuye.
5. La transformada de Hough para círculos no es tan robusta en comparación a la de líneas debido a que es más probable que las formas circulares en una imagen sean descontinuadas por objetos presentes en la escena (oclusiones), mientras que las líneas son mucho más fáciles de detectar.
6. Los resultados obtenidos utilizando procedimientos generales no resultaron óptimos al momento de trabajar con imágenes LWIR debido a su naturaleza.

7. Luego de analizar los problemas que ocasionaban el bajo rendimiento de la búsqueda de correspondencias, se recomendaron sus posibles soluciones, las cuales fueron puestas a prueba en el enfoque presentado en el capítulo 4, obteniéndose muy buenos resultados.
8. El método propuesto es una solución robusta para la detección de correspondencia en pares de imágenes multiespectrales. Sin embargo, estos resultados todavía pueden ser mejorados utilizando alguna técnica de optimización que permita encontrar los mejores parámetros para los algoritmos utilizados en el proceso (umbrales de Canny y Hough).

RECOMENDACIONES

A continuación se presentan las recomendaciones que a nuestro entender serían las líneas de trabajo futuro que permitirían mejorar los resultados aquí presentados.

1. El algoritmo de Canny, a pesar de ser una de las mejores opciones al momento de detectar contornos, sigue dependiendo del ajuste de umbrales y máscaras para su optimización, este algoritmo puede ser mejorado si aplicáramos un pre procesamiento al par de imágenes LWIR-VS con el fin de obtener líneas y círculos más pronunciados. De ese modo, la transformada de Hough dará mejores resultados.
2. Se recomienda dividir la imagen en diferentes regiones, y en cada una de ellas utilizar diferentes valores de umbrales dependiendo de su composición morfológica y niveles de ruido presentes. Finalmente, se deberán unir los resultados obtenidos para proceder a aplicar la transformada de Hough.
3. Para mejorar la cantidad de correspondencias encontradas se propuso mejorar la rectificación de imágenes utilizando un proceso manual de imágenes, esta etapa podría ser mejorada con la ayuda de algoritmos de rectificación que simplifiquen el problema al momento de encontrar correspondencias.

BIBLIOGRAFÍA

- [1] Hurtado, F., Atlas de matemáticas (1 edición). 1997. Idea Books, S.A, página 8
- [2] Zitová B., Flusser J., Image registration methods: a survey, Image Vision Comput, páginas 977–1000, 2003
- [3] Moravec, H., Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute, 1980
- [4] Harris, C., Stephens, M., A Combined Corner and Edge Detector, A Proceedings of 4th Alvey Vision Conference, 1988
- [5] Smith, S. M., Brady, J. M., SUSAN – a new approach to low level image processing, International Journal of Computer Vision 23, páginas 45–78, doi:10.1023/A:1007963824710, 1997
- [6] Rosten, E., Drummond, T., Machine learning for high-speed corner detection, European Conference on Computer Vision, 2006
- [7] Lowe, David G., Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision 2, páginas 1150–1157, doi:10.1109/ICCV.1999.790410, 1999
- [8] Bay, H., et al., Speed-Up Robust Features (SURF), Computer Vision and Image Understanding, vol 110, 2008, páginas 346–359 (ISSN: 1077-3142), 2008

- [9] Leutenegger, S., Chli, M., Roland, Y., BRISK: Binary Robust Invariant Scalable Keypoints, Autonomous Systems Lab, ETH Zurich.
- [10] Calonder, M., Lepetit, V., Strecha, C., Fua, P., BRIEF: Binary Robust Independent Elementary Features, In Proceedings of the European Conference on Computer Vision (ECCV), 2010
- [11] Lowe D.G., Local Feature, View Clustering for 3D Object Recognition, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition; Kauai, HI, USA, páginas 8–14, 2001
- [12] Lee J., Kim Y., Lee D., Kang D., Ra J., Robust CCD and IR image registration using gradient-based statistical information, IEEE Signal Process; páginas 347–350
- [13] Zhang L., Wu B., Nevatia R., Pedestrian Detection in Infrared Images Based on Local Shape Features, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition; Minneapolis, MN, USA, páginas 1–8, 2007
- [14] Yi, Z., Zhiguo, C., Yang, X., Multi-Spectral Remote Image Registration Based on SIFT, Electronics Letters, vol. 44, no. 2, páginas 107-108, 2008
- [15] Aguilera, C., Barrera, F., Lumbreras, F., Sappa, A. D., Toledo, R., Multispectral Image Feature Points, 2012
- [16] Aguilera, C., Barrera, F., Sappa, A. D., Toledo, R., A Novel SIFT-Like-Based Approach for FIR-VS Images Registration, 2012

- [17] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6) páginas 679–698, 1986
- [18] Mai, F., Hung, Y., Zhong, H., Sze, W., A hierarchical approach for fast and robust ellipse extraction, Pattern Recognition, 41(8) páginas 2512–2524, 2008
- [19] Hough, P., Method and Means for recognizing complex Patterns, U.S. Patent 3,069,654