

## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**“CORRESPONDENCIA DE CARACTERÍSTICAS UTILIZANDO ESQUEMAS  
CLÁSICOS EN EL ESPECTRO VISIBLE”**

### **TESINA DE SEMINARIO**

**Previo a la obtención de los títulos de:**

**INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS  
MULTIMEDIA**

**INGENIERA EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS  
MULTIMEDIA**

**PRESENTADA POR:**

Pablo Andrés Ricaurte Arias

Carmen Isabel Chilán Hidalgo

**GUAYAQUIL – ECUADOR**

**2014**

## AGRADECIMIENTO

*A mi familia por su apoyo incondicional,  
Su constante motivación y la oportunidad  
De estudiar una carrera universitaria.  
A la ESPOL por brindarme las herramientas,  
El conocimiento y la oportunidad de ser útil  
Para la sociedad.  
A mis profesores en su afán por hacer  
De nosotros personas admirables.  
A mis amigos por brindarme su apoyo,  
Su ánimo de trabajar y espíritu de competitividad.  
Pablo Andrés Ricaurte Arias.*

*Agradezco a Dios por la paciencia y salud, que nos  
Ha dado durante todo este tiempo.  
A mis padres quienes han sido y son  
Un pilar fundamental en mi vida.  
A nuestros profesores por todos los consejos,  
Y conocimientos transmitidos.  
Carmen Isabel Chilán Hidalgo.*

## DEDICATORIA

*A mi familia a quienes  
Le debo todo.  
A mi madre que siempre  
Ha estado pendiente de mí.  
A mi querida ESPOL y los  
Profesores que la componen.  
Pablo Andrés Ricaurte Arias.*

*A Dios  
A mi madre, quien me apoyo  
durante todos mis años de  
estudio, y le debo todo lo que  
soy.  
Isabel Chilán Hidalgo.*

## **TRIBUNAL DE SUSTENTACIÓN**

---

Dr. Ángel Sappa.  
**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

---

Dr. Boris Vintimilla  
**PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD**

## **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".

---

**Pablo Andrés Ricaurte Arias**

---

**Carmen Isabel Chilán Hidalgo**

## RESUMEN

El propósito de este proyecto es evaluar el rendimiento de las diferentes combinaciones detector/descriptor de puntos característicos para diversos problemas que se presentan en aplicaciones que utilizan correspondencia de características, tales problemas pueden considerarse como emborronamientos, cambios de iluminación, rotaciones en el plano, cambios de escala, ruido o combinaciones de las anteriores.

Los resultados de la evaluación permiten deducir tendencias, el mejor y el peor de las diferentes combinaciones detector/descriptor en cada transformación. En el resto de este informe nos referiremos a las combinaciones detector/descriptor como algoritmos y a estos problemas típicos como transformaciones.

Actualmente ya existen evaluaciones de este mismo tipo en el campo visible, por ello nuestro estudio también se extiende al campo infrarrojo específicamente LWIR (Long Wave Infra Red), permitiendo comparar comportamientos entre ambos dominios.

## ÍNDICE GENERAL

### CONTENIDO

<b>AGRADECIMIENTO</b> .....	II
<b>DEDICATORIA</b> .....	III
<b>TRIBUNAL DE SUSTENTACIÓN</b> .....	IV
<b>DECLARACIÓN EXPRESA</b> .....	V
<b>RESUMEN</b> .....	VI
<b>INTRODUCCIÓN</b> .....	1
<b>1. PLANTEAMIENTO DEL PROBLEMA</b> .....	2
1.1. Análisis del problema y diseño de la solución .....	2
1.1.1. Análisis del problema .....	2
1.1.2. Diseño de la solución .....	3
1.2. Objetivos del proyecto .....	3
1.2.1. General .....	3
1.2.2. Específicos .....	3
1.2.3. Justificación .....	4
1.2.4. Organización de la tesis .....	4
<b>2. REVISIÓN BIBLIOGRÁFICA</b> .....	6
2.1. Puntos característicos, definición e importancia .....	6
2.2. Detectores de puntos característicos .....	7
2.2.1. Detector Moravec .....	13
2.2.2. Detector Harris .....	17
2.3. Descriptores de puntos característicos .....	20
2.4. Algoritmo detector/descriptor .....	21
2.4.1. SIFT .....	21
2.4.2. SURF .....	23
2.4.3. BRIEF .....	26
2.4.4. ORB .....	28
2.4.5. BRISK .....	32

2.4.6.	FREAK.....	37
2.5.	Estudios anteriores.....	38
2.6.	Aplicaciones que utilizan correspondencia de características.....	42
2.7.	Naturaleza del campo visible e infrarrojo.....	48
2.7.1.	Espectro visible.....	48
2.7.2.	Infrarrojo.....	50
2.7.3.	Aplicaciones multiespectrales.....	52
3.	ESQUEMA DE EVALUACIÓN.....	61
3.1.	Funcionamiento.....	61
3.2.	Transformaciones.....	62
3.3.	Cálculo de matches correctos.....	65
4.	RESULTADOS EXPERIMENTALES.....	67
4.1.	Definición del dataset.....	67
4.2.	Definición de métricas de evaluación.....	68
4.3.	Contraste de resultados.....	71
4.3.1.	Resultados de Performance.....	71
4.3.2.	Resultados PDR.....	79
4.3.3.	Tiempos de ejecución.....	79
	<b>Bibliografía.....</b>	<b>86</b>

## ÍNDICE DE FIGURAS

Figura 2.1 - Localización de puntos detectados.....	8
Figura 2.2 - Radio de repetitividad.....	9
Figura 2.3 - Cálculo operador Moravec.....	14
Figura 2.4 - Mapa con medidas de esquina .....	15
Figura 2.5 - Moravec no es invariante a rotación.....	17
Figura 2.6 - Representación gráfica para obtener la orientación canónica ...	23
Figura 2.7 - Parche de pruebas binarias.....	28
Figura 2.8 – FAST: Detección del punto de interés en el espacio escala .....	35
Figura 2.9 – Patrón de muestreo del Brisk tomando 60 puntos .....	36
Figura 2.10 - Detectando un objeto específico en una escena .....	43
Figura 2.11 - Algoritmo de búsqueda de imágenes similares en una base de datos .....	44
Figura 2.12 - Descripción gráfica del algoritmo.....	45
Figura 2.13 - Detección de puntos claves en la escena capturada.....	46
Figura 2.14 - Registrado de una escena a partir de dos imágenes.....	47
Figura 2.15 - Aplicación móvil (CityCam) para determinar la ubicación del lugar.....	48
Figura 2.16 - Espectro electromagnético .....	49
Figura 2.17 - Descomposición de un haz de luz visible .....	50
Figura 2.18 - Ejemplo de una imagen en campo infrarrojo .....	52
Figura 2.19 – Compresión de imágenes multiespectrales .....	53
Figura 2.20 - Análisis de heridas.....	54
Figura 2.21 - Imágen multiespectrales de obra "Madonna and child" .....	54
Figura 2.22 - Configuración de cámara para toma de imágenes multiespectrales .....	55
Figura 2.23 - Agricultura de precisión: imagen multiespectral .....	58
Figura 2.24 - Filtrado de aire acondicionado por una puerta .....	59
Figura 2.25 - Imagen en pseudo-color de un tren, las partes calientes toman un color amarillo con la subida de temperatura .....	59

Figura 2.26 - Imagen de resonancia magnética de una rodilla .....	60
Figura 3.1 - Ilustración de un par de imágenes pertenecientes al dataset ((arriba) LWIR y (abajo) VS) con sus correspondientes transformaciones: (a) imagen original; (b) rotación; (c) escala; (d) blur; (e) ruido. ....	64
Figura 4.1 - Par de imágenes en ambos espectros ((izq.) visible (VS), (der.) infrarrojo (LWIR)) contenidas en el dataset evaluado .....	68
Figura 4.2 - Performance en el caso de rotación: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR). ....	73
Figura 4.3 - Performance para cambios en escala: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR). ....	75
Figura 4.4 - Performance promedio ante la degradación de blur: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR).....	76
Figura 4.5 - Caso de estudio para el ruido: (arriba) espectro visible (VS), (abajo) campo infrarrojo (LWIR).....	78

## ÍNDICE DE TABLAS

Características de evaluación (Mikolajczyk & Schmid, 2005) .....	38
Características de evaluación (Miksik & Mikolajczyk, 2012) .....	40
Características de evaluación (Schmid, Mohr, & and Bauckhage, 2000) .....	41
Algoritmo detector/descriptor y matchers.....	62
Resultados PDR para cada algoritmo.....	79
Performance para el campo visible.....	80
Performance para el campo infrarrojo.....	80

# INTRODUCCIÓN

Hoy en día son cada vez más las aplicaciones que utilizan correspondencia de características, estas aplicaciones van desde problemas simples como la creación de una imagen panorámica a más complejas como reconocimiento de patrones.

El presente documento busca evaluar los detectores y descriptores de puntos característicos en el campo visible e infrarrojo, evaluando el mejor de ellos en cada campo y encontrando características comunes entre campos.

Los algoritmos detectores y descriptores han sido por mayor tiempo evaluados y optimizados sobre el campo visible es por ello que son más eficientes que en el campo infrarrojo. Para corroborar lo anterior haremos uso de pruebas iterativas, métodos estadísticos y observación de los resultados.

En este estudio se describirá cada algoritmo, se definirá métricas generales para medir el rendimiento, se definirá un conjunto de datos de pruebas, se creará una estructura de evaluación que evaluará las métricas fijadas y se finalmente se contrastará los resultados.

# **CAPÍTULO 1**

## **1. PLANTEAMIENTO DEL PROBLEMA**

### **1.1. Análisis del problema y diseño de la solución**

#### **1.1.1. Análisis del problema**

El problema es la falta de conocimiento del algoritmo más efectivo para las diferentes transformaciones que afrontan las aplicaciones que utilizan correspondencia de puntos característicos tales como: emborronamiento, rotación en el plano, cambios de iluminación, escalamiento y ruido. Consideremos el caso de un robot móvil a base de ruedas en el cual al estar en movimiento una imagen obtenida por la cámara es igual a la anterior más un aparente escalamiento, por otro lado si parte del robot pasara por un desnivel se apreciará en el video una rotación y translación en el plano, además el robot podría pasar por zonas de diferente iluminación, entre otros casos.

### **1.1.2. Diseño de la solución**

Se realizará una evaluación de los algoritmos para cada uno de los problemas típicos de estas aplicaciones, simulando tales problemas mediante transformaciones de la imagen original. La evaluación consiste en medir la efectividad del algoritmo al realizar la correspondencia entre la imagen original y la transformada. Definiendo la efectividad a través de la medición del *performance*, *PDR* y *el tiempo de procesamiento*.

## **1.2. Objetivos del proyecto**

### **1.2.1. General**

Realizar una evaluación sobre el rendimiento de los algoritmos detector/descriptor en las diferentes afecciones típicas en las aplicaciones que utilizan correspondencia de características, además se extiende la evaluación al campo infrarrojo lejano (LWIR) dado que el seminario de graduación fue orientado al campo multiespectral.

### **1.2.2. Específicos**

En nuestra evaluación tendremos 4 objetivos específicos:

1. Definir un esquema de evaluación para correspondencia de características.

2. Definir un conjunto de algoritmos a evaluar.
3. Definir un conjunto de prueba.
4. Probar y contrastar resultados obtenidos en las distintas bandas del espectro (visible e infrarrojo).

### **1.2.3. Justificación**

A pesar de que existen numerosas evaluaciones similares en el campo visible, nuestra evaluación es importante debido a que se actualizarán la literatura de trabajos existente tomando en consideración algoritmos recientemente publicados. Además el estudio se extiende al campo infrarrojo para poder comparar tendencias entre ambos dominios.

### **1.2.4. Organización de la tesis**

En el primer capítulo se dará una introducción acerca del proyecto analizando el problema y diseñando la solución con sus respectivos objetivos y justificaciones.

En el segundo capítulo encontramos la revisión bibliográfica, en la cual se definen conceptos tales como puntos característicos, detectores, descriptores, combinaciones de ambos y referencias a estudios previos.

En el capítulo 3 se explica el funcionamiento del marco de trabajo para

la evaluación, cuáles son las transformaciones y como se implementan así como también la lógica usada para definir las correspondencias correctas.

En el capítulo 4 se definen las métricas a usar y porque fueron escogidas, se mostrarán los resultados, se contrastará lo definido en la revisión bibliográfica contra los resultados obtenidos y por último se resaltarán las similitudes y diferencias de tendencias de los algoritmos en ambos campos.

## **CAPÍTULO 2**

### **2. REVISIÓN BIBLIOGRÁFICA**

#### **2.1. Puntos característicos, definición e importancia**

Hoy en día, muchas aplicaciones requieren relacionar dos o más imágenes extrayendo información representativa sobre sus características. Por ejemplo, determinar si dos frames pertenecen a una misma secuencia de video capturado mientras la cámara estuvo en movimiento. Para tal caso, el método de fuerza bruta de comparar cada par de píxeles entre ambas imágenes requerirá un costo computacional muy alto. Una manera de evitarlo se logra a partir de la detección de puntos claves que permita relacionar localidades relevantes entre ambas imágenes.

Se definen como puntos claves aquellos que son invariantes en una imagen; es decir, puntos que difieren considerablemente de sus

vecinos a pesar de producirse cambios en alguna propiedad de la imagen, tales como cambios de iluminación, cambios de escala, cambios de orientación o emborronamiento.

## **2.2. Detectores de puntos característicos**

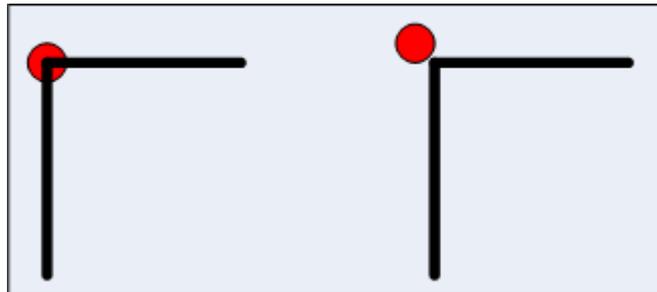
Los detectores son un conjunto de pasos que se aplican sobre los píxeles de una imagen para obtener los puntos de interés, diversos algoritmos para detectar los puntos de interés han surgido, en general deben cumplir con lo siguiente:

- Todos los puntos de interés correctos deben ser detectados
- No deben ser detectados falsos puntos de interés
- Los puntos de interés deben poder ser localizados
- El detector debe tener un alto ratio de repetitividad
- El detector debe ser robusto ante el ruido
- El detector deber ser computacionalmente eficiente

Para el caso de detectores de esquinas por ejemplo, una esquina no está bien definida en una imagen de escala de grises, es por ello que no existen detectores de esquinas perfectos, habrá puntos detectados que sí serán esquinas y aquellos que no lo sean, llamados falsos. Por

tanto es deseable que detecte la mayor cantidad de puntos verdaderos y lo menos posible falsos.

Que un punto de interés sea localizado quiere decir que la coordenada que representa ese punto de interés este exactamente encima del punto de interés, mientras más alejado entonces tendrá una mala localización. Podemos apreciar para el caso de un detector de esquinas, la imagen siguiente:

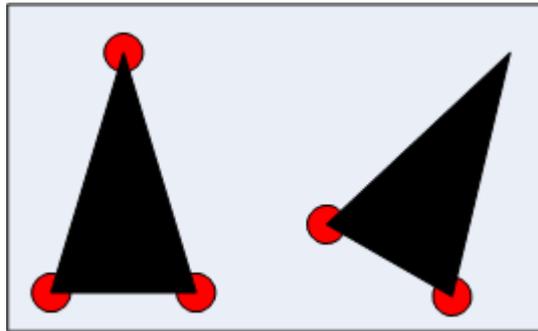


**Figura 2.1 - Localización de puntos detectados**

Fuente: <http://kiwi.cs.dal.ca/~dparks/CornerDetection/intro.htm>

Para la repetitividad consideremos dos imágenes adyacentes extraídas de una secuencia de video, en ambas existe el mismo objeto y como son adyacentes son muy parecidas por tanto los puntos de interés detectados deberían ser los mismos, pero por razones diversas ya sea pequeños desplazamientos de la cámara o pequeños cambios de luz entre otras cosas, afectan el desempeño del detector y por ello no siempre son los mismos puntos detectados, entonces se define como ratio de repetitividad a la razón entre puntos correctamente

detectados en otra imagen (en este caso la segunda) sobre el número total de puntos que debieron ser detectados de la primera, para el ejemplo siguiente el ratio de repetitividad es  $2/3$ .



**Figura 2.2 - Radio de repetitividad**

**Fuente:** <http://kiwi.cs.dal.ca/~dparks/CornerDetection/intro.htm>

El ruido en una imagen es inevitable en la mayoría de casos, por diversas razones, entre ellas la naturaleza de la misma cámara, por ello el detector debe ser robusto ante la presencia de ruido, es decir, el ruido no debe afectarle.

El detector como todo algoritmo computacional consume recursos, y en visión por computadora la detección es el primer paso en un sin número de decisiones o trabajos que se pueden realizar con la información obtenida, es por ello que el detector debe consumir la menor cantidad de recursos.

Los detectores se enfrentan a variaciones naturales como cambios de escala, rotación, etc. descritos en la sección 3.2, aun no se ha creado

un detector universal que sea invariante a todas estas variaciones naturales o que cumpla en su totalidad los requerimientos antes descritos, cada detector tiene sus características propias, ventajas y desventajas. Escogeremos el detector dependiendo de la aplicación, por ejemplo: si se desea reconocer personas y el software va a ejecutarse en un dispositivo móvil entonces la característica primordial será que el detector a usar consumirá la menor cantidad de recursos posibles, si deseamos un detector para una aplicación de reconocimiento de objetos deseamos que exista un alto ratio de repetitividad, si el detector es para una aplicación de un robot móvil entonces debe ser invariante ante escala e iluminación y así muchos otros casos.

Los detectores se pueden clasificar de acuerdo al enfoque que usan para la detección de puntos característicos, para el caso de puntos característicos que representan esquinas se los puede clasificar en tres grupos descritos a continuación, cabe recalcar que algunos detectores pueden caer en más de una categoría y existen algunos detectores que no entran en ninguno de estos grupos, algunos ejemplos son presentados al final:

### **Métodos de relación de esquinas**

Se realiza una detección de esquinas usando operadores geométricos

diferenciales, estos operadores ofrecen una medida de esquina basado en el cambio de la dirección del gradiente a lo largo de un contorno enfatizado por la intensidad del gradiente. Es decir se aplica este operador sobre la imagen y para cada pixel se obtiene una medida de esquina, donde se identifica como esquina aquella que sea máxima localmente, la superficie alrededor del punto examinado que se considera local es configurable, puede ser una matriz de 5x5, 7x7 o como se desee. Debido a que se usa el gradiente (derivada de segundo orden) lo hace sensible al ruido y ha probado tener baja repetitividad y localización. Algunos ejemplos citados son [16] y [17].

### **Métodos de topología**

Hacen uso del determinante de la matriz Hessiana para obtener una medida de esquina invariante a rotación. La matriz Hessiana hace uso de derivadas de segundo orden por tanto es sensible al ruido. Ejemplos de estos citamos [18].

### **Métodos de auto correlación**

Hacen uso de un operador que mide el cambio de intensidad, para cada pixel a evaluar se considera una región, digamos 5x5, centrada en el pixel y sus valores de intensidad son comparados con los de una ventana de igual tamaño pero desplazada una unidad en alguna dirección de las 8 posibles, la comparación consiste en una resta de

intensidades y la suma de esas restas representa el cambio de intensidad para la dirección dada, el menor cambio de intensidad de entre todas las direcciones representara el cambio de intensidad del pixel examinado. Como una esquina representa un fuerte cambio de intensidad en varias direcciones entonces este operador encuentra esquinas. Obtenido el cambio de intensidad en cada pixel si tomamos los puntos que sean máximos locales, estos máximos muy probablemente sean esquinas. Un ejemplo de esto es Moravec [20] que se desarrolla en las 4 direcciones principales y Harris [21] que soluciona el gran consumo de recursos computacionales que conlleva Moravec.

Algunos detectores que no caen dentro de estas categorías son por ejemplo SUSAN (Smallest Univalued Segment Assimilating Nucleus) [22] que usan comparaciones de intensidad de pixeles, este algoritmo realiza comparaciones entre la intensidad del pixel a evaluar con los pixeles que están dentro de una máscara circular creando el USAN (Univalued Segment Assimilating Nucleus) del punto a evaluar, para pixeles de un mismo objeto o sección se espera que tengan casi la misma intensidad de pixel por tanto si el punto a evaluar es una esquina se espera que el USAN sea un valor máximo porque dentro del área circular habrá una mayor cantidad de puntos con intensidades de pixeles diferentes al centro, si el punto a evaluar pertenece a un

borde entonces su USAN será como mucho la mitad del máximo posible.

Los detectores usados en este estudio forman parte de un algoritmo que también contiene su propio descriptor y son descritos dentro de la sección 2.4, a continuación mostraremos el algoritmo Moravec y luego Harris. Moravec fue el primer detector creado y la mayoría de los detectores posteriores son mejoras de este, este detector es simple y sirve como introducción a Harris. Harris es un detector que mejora el rendimiento de Moravec, ambos ejemplifican muy bien la evolución de los detectores.

### **2.2.1. Detector Moravec**

Creado por Hans P. Moravec en 1977 [20], define como punto característico aquel punto que presente una alta variación en la intensidad de pixel en todas las direcciones, donde es el caso de las esquinas, por tanto este operador es un detector de esquinas sin querer serlo.

El operador funciona de la siguiente forma, considérese una ventana de tamaño fijo 3x3 centrada en el punto de observación y otra ventana del mismo tamaño pero desplazada un pixel en cualquier dirección, la variación de intensidad de pixel en esa dirección se calcula como la sumatoria del cuadrado de la diferencia entre las intensidades entre

pixeles correspondientes de ambas ventanas, luego se calcula la variación para las 7 direcciones restantes y finalmente se define la variación de intensidad de pixel para el pixel en observación como la mínima variación de las 8 variaciones calculadas. En la Fig. 2.3 podemos ver el cálculo de variación de intensidad de pixel en una sola dirección para un punto aislado y para una esquina ideal (Para el ejemplo el objeto de importancia se representa de color negro y el fondo blanco).

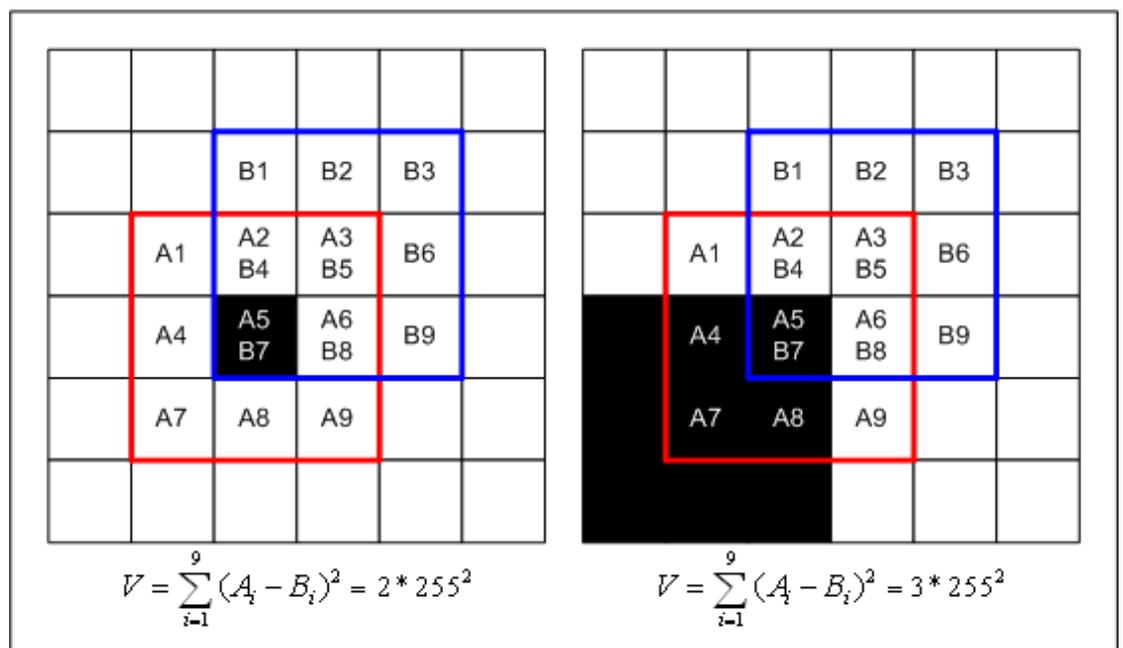


Figura 2.3 - Cálculo operador Moravec

Fuente: <http://kiwi.cs.dal.ca/~dparks/CornerDetection/moravec.htm>

Si evaluamos el operador Moravec y para cada valor obtenido de este operador lo usamos para crear una segunda imagen, recordando que cada pixel de esta segunda imagen representa una medida de esquina

para cada pixel de la imagen original, como se puede observar en la Fig. 2.4.

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figura 2.4 - Mapa con medidas de esquina

Fuente: <http://kiwi.cs.dal.ca/~dparks/CornerDetection/moravec.htm>

Entonces podemos definir como esquina al máximo valor de esquina local, sin embargo para imágenes en escala de grises puede haber máximos locales que no representen una esquina, es por ello que aplicamos a esos máximos locales un umbral, aquellos máximos locales con valor de esquina menor al umbral son llenados con cero, esto es comúnmente llamado como supresión de no máximos. Tome en cuenta que este valor de umbral debe ser lo suficientemente alto como para evitar esquinas falsas y lo suficientemente bajo para dejar pasar la mayor cantidad de esquinas verdaderas.

También podemos observar que para un punto aislado, este también es un máximo local sin ser una esquina y que no necesariamente puede ser eliminado mediante el umbral antes descrito, este punto aislado puede ser ruido, es por ello que Moravec es muy sensible al ruido. Si aumentamos el tamaño de la ventana hacemos al operador más robusto ante el ruido porque una esquina verdadera tiene una variación de intensidad de pixel mayor que el pixel aislado, pero este pixel aislado es aún un máximo local.

### **Operador Anisotrópico**

Se dice que es un operado anisotrópico porque solo se calcula en las 8 direcciones principales, en la Fig. 2.5 podemos ver que detecta como esquinas a puntos de bordes que no encajan en una de estas 8 direcciones, de igual forma podemos ver que si se rota la imagen 30 grados el efecto es aún peor, disminuye la repetitividad, es por ello que Moravec no es invariante a rotación y es considerado obsoleto.

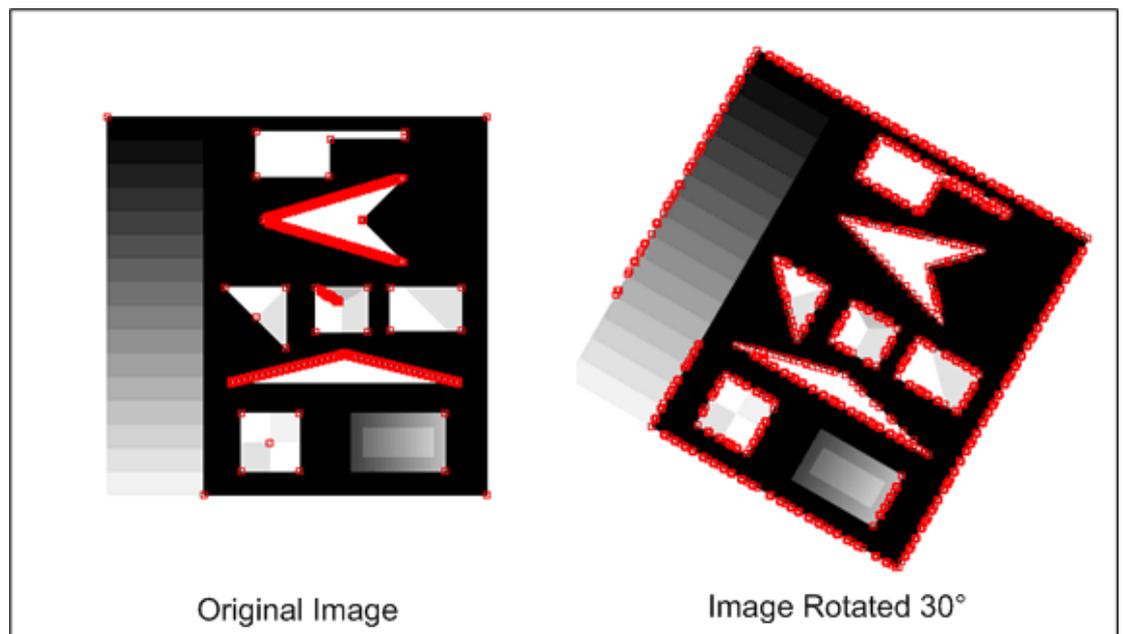


Figura 2.5 - Moravec no es invariante a rotación

Fuente: <http://kiwi.cs.dal.ca/~dparks/CornerDetection/moravec.htm>

Podemos concluir que Moravec es muy sensible al ruido, a bordes cuya inclinación este fuera de las 8 direcciones de cálculo, a rotación, y no es muy preciso al medir las variaciones de intensidad de pixel porque usa una ventana binaria cuadrada, este último es descrito en detalle en el detector Harris.

### 2.2.2. Detector Harris

Creado en 1988 por Chris Harris y Mike Stephens, comúnmente llamado como Harris o Plessey [21] es más una mejora de Moravec que una implementación nueva. Si recordamos a Moravec como intenta medir el cambio de intensidad en una dirección de las 8

posibles direcciones Harris implementa un método matemático para medir el cambio de intensidad en cualquier dirección. Definiremos al cambio de intensidad mediante la siguiente fórmula:

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

Si observamos, esta ecuación implementa al detector Moravec cuando  $\Delta x$  y  $\Delta y$  representa uno de los 8 movimientos posibles antes descritos. Realizando una expansión de Taylor a la parte de la ecuación que representa la ventana desplazada y truncando hasta las derivadas de primer orden tenemos:

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

Reemplazando esto en la ecuación inicial y factorando:

$$\begin{aligned} &= \sum_W \left( I(x_i, y_i) - I(x_i, y_i) - [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_W \left( -[I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_W \left( [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \end{aligned}$$

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

Debido a que  $[\Delta x \ \Delta y]$  representan la dirección de cambio entonces  $C(x, y)$  representa los cambios de intensidad en cualquier dirección para la vecindad local. Los auto-valores de esta matriz  $C$  serían un buen descriptor invariante a rotación, los valores que puede tomar estos auto-valores permiten concluir lo siguiente:

1. Si ambos son valores bajos, quiere decir que existe un cambio de intensidad mínimo en cualquier dirección y por tanto el punto donde estoy evaluando Harris pertenece a una zona plana.
2. Si uno es alto y el otro bajo quiere decir que el punto que estoy evaluando pertenece a un borde.
3. Si ambos son altos quiere decir que el punto que estoy evaluando pertenece a una esquina, debido a que una esquina tiene un alto valor de cambio en la intensidad de los pixeles.

Como vemos tiene invariancia a rotación, cierta invariancia ante escala, cambios de iluminación y ruido.

### **2.3. Descriptores de puntos característicos**

Los descriptores pueden ser agrupados en 3 categorías definidos por la forma en que describen:

#### **Descripción basada en distribuciones**

Usadas para representar diferentes características de apariencia o forma. Un descriptor simple de este tipo podría ser un histograma de distribución de pixeles, ejemplo de estos son las imágenes spin [23] (que es un histograma con las posiciones de los puntos de la vecindad con respecto al punto 3d de interés). Otro ejemplo es el presentado en [1], el cual se basa en histogramas de ordenamiento y relaciones recíprocas entre intensidades de pixel. Otro ejemplo es SIFT que es descrito en detalle en la sección SIFT pag.21.

#### **Técnicas espacio-frecuencia**

Usadas para describir el contenido de frecuencia de una imagen, ejemplo de esto son los filtros de Gabor y wavelets que son usadas en el ámbito de clasificación de texturas.

#### **Descriptores diferenciales**

Se usan derivadas de la imagen para representar un vecindario, se usan las propiedades obtenidas de estas derivadas, mayormente

permite obtener invariancia a rotación. Un ejemplo de esto es Steer-BRIEF [11] descrito en más detalle en la sección BRIEF pág.26

## **2.4. Algoritmo detector/descriptor**

### **2.4.1. SIFT**

El detector de SIFT obtiene los puntos clave escogiendo los máximos o los mínimos (refiriéndose a valor de intensidad, máximo será entonces aquel pixel con mayor nivel de intensidad en su localidad) en la diferencia de Gaussianas aplicadas en un espacio escala también llamado pirámide Laplaciana porque es equivalente a calcular el Laplaciano. Obtendremos varios puntos claves por cada nivel de esta pirámide Laplaciana, la aplicación de un espacio escala permite obtener puntos claves invariantes a escala. Específicamente, a la imagen original se le aplica un filtro gaussiano con  $\sigma=\sqrt{2}$ , llamemos a esta imagen "A", a esta imagen se le vuelve a aplicar el mismo filtro gaussiano, llamémosla "B", equivalente a aplicar un filtro gaussiano con  $\sigma=2$  (es decir  $\sqrt{2} * \sqrt{2}$ ) a la imagen original y obtenemos la diferencia entre "A" y "B" que vendría a ser el primer nivel de la pirámide Laplaciana, de la misma forma se procede con el 2 nivel donde la imagen "A" es igual a aplicar un filtro gaussiano con  $\sigma=\sqrt{2}$  a la imagen "B" del nivel anterior pero ampliada (escalada) o equivalente a aplicar un filtro gaussiano con  $\sigma=3*\sqrt{2}$  a la imagen original también

escalada, volvemos a aplicar el mismo filtro a esta imagen "A" para obtener "B" y finalmente la diferencia que vendría a ser el segundo nivel, de igual forma para los niveles siguientes.

Para la obtención del descriptor del punto clave, obtenemos la magnitud y la dirección del gradiente de cada punto clave en cada imagen "A" de cada nivel de la pirámide, la magnitud y la dirección se calculan mediante diferencia de pixeles, con las siguientes formulas:

$$M_{ij} = \sqrt{(A_{ij} - A_{i+1,j})^2 + (A_{ij} - A_{i,j+1})^2}$$

$$R_{ij} = \text{atan2}(A_{ij} - A_{i+1,j}, A_{i,j+1} - A_{ij})$$

Para obtener robustez ante cambios de iluminación se filtra el conjunto de puntos claves hasta ahora obtenidos conservando los puntos cuya magnitud sea mayor a 0.1 veces la mayor posible. Además de la orientación del gradiente obtenemos una orientación canónica que será usada para ofrecer cierta invariancia a rotación, para cada punto clave de cada imagen "A" de cada nivel de la pirámide y una vecindad  $3\sigma$  mayor al  $\sigma$  usado para obtener esa imagen "A" obtenemos un histograma de orientación de gradientes (HOG) donde agruparemos la orientación de gradientes de cada punto de esa vecindad en grupos, los grupos serán definidos cada 10 grados, es decir habrán 36 grupos en total para completar los 360 grados, el valor mayor en este

histograma representa la orientación canónica; en la Fig. 2.6 se presenta una representación gráfica de este proceso.

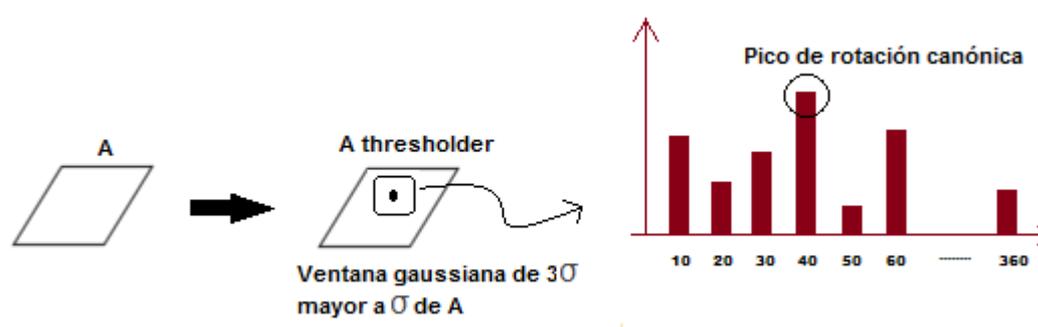


Figura 2.6 - Representación gráfica para obtener la orientación canónica

Finalmente se crean 8 planos de orientación para cada punto clave tomando los píxeles que rodean al punto clave en un radio de 8, en cada plano se consideran los píxeles cuya orientación del gradiente está dentro de su rango, los rangos van cada 45 grados, es decir el segundo plano considera a los píxeles cuyo ángulo este entre 45 y 90 grados. En cada plano se obtiene una orientación relativa a la del punto clave mediante la resta de entre la orientación del plano y la orientación del punto clave. El descriptor estará compuesto por 160 elementos para cada punto clave.

Podemos resumir que este algoritmo es altamente invariante a escala, iluminación y distorsiones locales afines.

## 2.4.2. SURF

Su detector está basado en la matriz Hessiana debido a su eficiencia en tiempo computacional y certeza, particularmente en el determinante, donde para un punto  $\mathbf{x}=(x,y)$  de la imagen y una escala  $\sigma$ , la Hessiana se define como sigue:

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

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2.$$

Creamos una pirámide gaussiana (permite invariancia en escala) donde en cada nivel se aplica un filtro gaussiano con  $\sigma=1.2$  y escala  $=1.2$  mayor al nivel anterior, en cada nivel ubicamos puntos de interés definidos como aquel que tiene el determinante mayor para un conjunto de puntos dentro de una ventana, para el primer nivel de la pirámide  $\sigma=1.2$ , escala= $1.2$  y ventana de  $9 \times 9$ , los valores de ventana toman valores de  $9 \times 9$ ,  $15 \times 15$ ,  $21 \times 21$  es decir en incrementos de 3. Una vez detectados estos puntos de interés se procede a hacer la supresión de no máximos, donde el punto es aceptado si es que su valor de gradiente es mayor o menor al pixel correspondiente en el nivel superior e inferior.

Para la descripción del punto de interés, para cada punto detectado y una vecindad dentro de un radio  $6s$ , donde  $s$  es la escala donde fue

detectado el punto, se calcula la respuesta en  $X$  e  $Y$  de aplicar filtros HAAR-Wavelets, usando estas respuestas en  $X$  e  $Y$  de cada punto de la ventana se crean vectores, el ángulo del vector resultante de la suma de aquellos vectores que formen un ángulo con respecto a las abscisas menor a 60 grados vendrá a ser el ángulo relacionado a ese punto característico. El proceso recién descrito le otorga cierta invariancia a rotación, este proceso no es realizado por una variante de SURF llamado U-SURF que no es invariante a rotación pero efectivamente más rápido.

Calculado este ángulo, se obtiene una ventana cuadrada alrededor del punto característico orientada mediante el ángulo antes calculado, esta ventana se divide constantemente en secciones de 4x4 y en cada una de estas secciones se busca por puntos característicos simples mediante una máscara de 5x5, para cada punto característico simple se obtiene la derivada parcial en  $X$  e  $Y$  denominado como  $d_x$  y  $d_y$  respectivamente, finalmente para cada sección obtenemos el siguiente vector compuesto por sumatorias de estas derivadas de puntos característicos.

$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

Para poder dar información sobre la polaridad de los cambios de intensidad es que extraemos la suma de los valores absolutos de las

respuestas  $dx$  y  $dy$ . Debido a que el punto es originalmente dividido en secciones de  $4 \times 4$  entonces tenemos que para cada punto el vector descriptor estará formado por  $(4 \times 4) \times 4$  datos por región = 64 datos. Existen diferentes variantes de SURF, si hablamos de SURF-36 es porque las divisiones fueron de  $3 \times 3$  en vez de  $4 \times 4$ , es decir  $(3 \times 3) \times 4$  datos por región = 36. Este algoritmo permite invariancia en rotación, escala y es robusto al ruido.

### 2.4.3. BRIEF

Consiste solo de descriptor, su creación es importante debido a que su descriptor es binario lo que permite una comparación rápida y necesita poco espacio de almacenamiento, esta notable mejoría fue motivo para muchos algoritmos futuros el usar un descriptor binario. Inclusive la búsqueda de correspondencias es más rápida porque permite usar la distancia de Hamming que es mucho más rápida que NORM\_L2.

Consideremos un parche de tamaño  $S \times S$  llamado  $\mathbf{p}$  cuyo centro será el punto a describir y la prueba binaria sobre ese parche descrito como sigue:

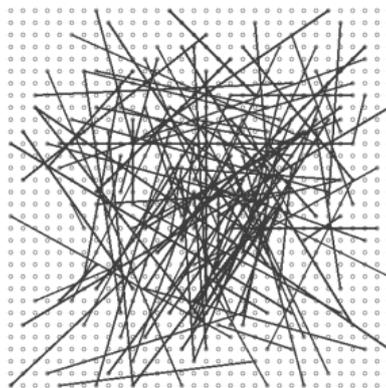
$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

Donde  $p(x)$  es el valor de intensidad en el punto  $X$  del parche, cabe

mencionar que antes de hacer la prueba, la cual como vemos es una diferencia de píxeles, hay que aplicar un filtro de suavizado debido a que la diferencia de píxeles lo hace sensible al ruido. El descriptor final es un vector binario de tamaño  $n$  creado a partir de  $n$  pruebas binarias gracias a un conjunto  $X$  e  $Y$  de puntos también de tamaño  $n$ , el descriptor es creado de la siguiente manera:

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$

Valores de  $n$  pueden ser 128, 256 y 512 debido a que los resultados experimentales muestran un buen compromiso entre velocidad, eficiencia de almacenamiento y radio de reconocimiento. Las únicas opciones que pueden ser hechas son los kernels que usaremos para suavizar la imagen y la selección de estos conjuntos de puntos  $X$  e  $Y$ . Los kernels a usar son filtros gaussianos con  $\sigma=2$  y tamaño de ventana 9x9, las pruebas se hicieron con  $\sigma$  de 0 a 3 y probaron no varía mucho, con respecto a la distribución espacial de los conjuntos  $X$  e  $Y$  se postularon varios ejemplos mostrando una pequeña mejoría en la siguiente distribución donde cada línea representa un par  $(X_i, Y_i)$ :



**Figura 2.7 - Parche de pruebas binarias**

**Fuente: [12]**

El algoritmo promete ser rápido pero no invariante a rotación en el plano, incluye robustez ante cambios de iluminación, blur y distorsión de la perspectiva.

#### **2.4.4. ORB**

Este algoritmo es invariante a rotación y robusto al ruido, es 2 órdenes de magnitud más rápido que SIFT y por ello es posible usarlo en aplicación de tiempo real. Se construye usando una variación del detector FAST y una variación del descriptor BRIEF de allí que sus siglas sean ORB (Oriented FAST and Rotated BRIEF), un beneficio adicional de ORB es que está libre de las restricciones de licencia que tienen SIFT y SURF.

Como se dijo su detector es una variación de FAST llamada oFAST

con el prefijo 'o' de 'oriented' porque entrega información de la orientación del punto característico, el proceso se describe a continuación; se crea un espacio escala, a cada nivel de ese espacio escala se aplica el detector FAST-9 que significa aplicar el algoritmo FAST con un círculo de radio igual a 16 y que comprueba cada 9 puntos consecutivos para encontrar los puntos de interés, esto nos da una gran cantidad de puntos, conservaremos solo los mejores de esos puntos, se define como mejor aquel punto que tenga mayor calidad de ser una esquina.

Se medirá la calidad de esquina mediante el detector Harris sobre los puntos ya detectados y conservaremos los n mejores puntos. Finalmente obtenemos la orientación del punto mediante un centroide de intensidades [1], este centroide asume que la intensidad de una esquina esta desplazada con respecto a su centro, este punto desplazado y el centro forman un vector que puede ser usado para definir una orientación. Rosin [11] define el momento de un parche de imagen como:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y)$$

y calculamos el centroide como:

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

El ángulo formado por el vector entre ambos puntos se calcula como:

$$\theta = \text{atan2}(m_{01}, m_{10})$$

donde  $\text{atan2}$  es la versión del arco tangente que observa en que cuadrante se realiza el cálculo, independiente de si la esquina es clara u oscura puede definir un mismo ángulo, para añadir una cierta invariancia a rotación los momentos son calculados dentro de una región circular de radio  $r$ .

Como se dijo su descriptor se basa en una variación de BRIEF llamada rBRIEF cuyo prefijo 'r' es para 'rotation-aware', recordemos que BRIEF es negativamente muy sensible a cambios de rotación, por tanto se pretende sobrellevar este problema con rBRIEF.

Primero unas definiciones, llamemos 'característica' al vector que representa el conjunto de pruebas binarias sobre el conjunto  $S$  de puntos, antes mencionado en el descriptor BRIEF:

$$f_n(\mathbf{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{x}_1, \dots, \mathbf{x}_n \\ \mathbf{y}_1, \dots, \mathbf{y}_n \end{pmatrix}$$

Observemos que la característica depende totalmente de este conjunto de puntos seleccionado, una variación de BRIEF que solventa el problema de invariancia a rotación sería obtener varios descriptores BRIEF para un mismo punto que se calcularían obteniendo el descriptor BRIEF para una misma característica de  $\mathbf{S}$  pero cada descripción sería usando una rotación de  $\mathbf{S}$ , solución que sería poco eficiente. Una solución intermedia llamada 'steer BRIEF' consiste en calcular del descriptor BRIEF para una misma característica  $\mathbf{S}$  rotada un ángulo  $\theta$ , ángulo obtenido del punto característico.

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S}$$

$$g_n(\mathbf{p}, \theta) := f_n(\mathbf{p}) | (\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{S}_\theta$$

Este conjunto  $\mathbf{S}$  que representa una característica debe tener una alta variancia para que el descriptor sea muy discriminante y las pruebas sobre  $\mathbf{S}$  deben estar poco correlacionadas unas con otras para que sean eficientes, por tanto la elección de  $\mathbf{S}$  es importante. Se sugiere un método de aprendizaje para obtener una buena característica, y es esto lo que forma el denominado rBRIEF; a continuación se presenta

este método de aprendizaje:

Se toma un conjunto de imágenes para realizar el aprendizaje, donde todas las pruebas binarias se realizan sobre un parche de 31x31.

1. Corremos cada prueba sobre todos los parches de entrenamiento.
2. Ordenamos las pruebas por su distancia a una media de 0.5, formando el vector T.
3. Realizamos una búsqueda codiciosa:
  - a. Poner la primera prueba en un vector resultado R y removerlo de T.
  - b. Tomar la siguiente prueba de T, y compararla contra todas en R, si su correlación absoluta es mayor que un umbral, eliminarla, caso contrario añadirla a R.
  - c. Repetir el paso previo hasta que R alcance las 256 pruebas, aumentar el umbral e intentarlo de nuevo.

#### **2.4.5. BRISK**

Es invariante a rotación y a escala, su eficiencia es comparable a SIFT pero con menor costo computacional, su detector se basa en FAST y

su descriptor es binario por la eficiencia observada desde su aparición en BRIEF.

Para la detección de puntos, primero se construye un espacio escala que consiste de  $n$  niveles que definen como octavas  $C_i$  estos niveles son más relevantes que otros intermedios a los que definen como intra-octavas  $D_i$ . Las octavas son creadas mediante el muestreo sucesivo a la mitad (por ejemplo ampliando la imagen al doble) de la imagen original que representa la octava  $C_0$  cada intra-octava  $D_i$  está ubicada entre las octavas  $C_i$  y  $C_{i+1}$ , donde  $D_0$  se obtiene por re-muestreo de  $C_i$  en un factor de 1.5, las siguientes intra-octavas se obtienen muestreando a la mitad la intra-octava anterior. Es decir si  $t$  representara la escala entonces  $t(C_i) = 2^i$  y  $t(D_i) = 2^i * 1.5$ .

Para cada nivel de esta pirámide se detectan puntos característicos mediante FAST 9-16 (punto aceptado si su intensidad es mayor o menor a la de 9 puntos consecutivos dentro de un círculo de radio 16) usando un mismo umbral. Para cada punto detectado se le aplica supresión de no-máximos donde un punto es descartado si su cualidad de esquina no es mayor a sus 8 puntos circundantes, cumplida esta restricción luego se compara con el nivel superior e inferior tomando el mismo tamaño de parche cuadrado, es decir comparamos con los 9 píxeles de la capa superior e inferior. Se

definen como cualidad de esquina 's' al máximo valor de umbral para el detector FAST con que un punto se sigue considerando como punto característico, la supresión de máximos es un poco diferente para el nivel 0.

Considerando la cualidad de esquina como un valor continuo no solo en la imagen sino también entre escalas, para los puntos restantes a esta supresión de no-máximos, calculamos la escala real en la que ocurre este máximo, recordemos que con el método antes descrito solo encontramos máximos para escalas discretas, entonces hacemos uso de 3 puntos: la cualidad  $s$  de la capa superior, la cualidad  $s$  del máximo y la cualidad  $s$  de la capa inferior, ajustamos la ecuación de una parábola que calce en estos puntos y calculamos el máximo de esta ecuación para así poder encontrar la escala real, podemos observar esto en la siguiente grafica extraída del artículo original [3]:

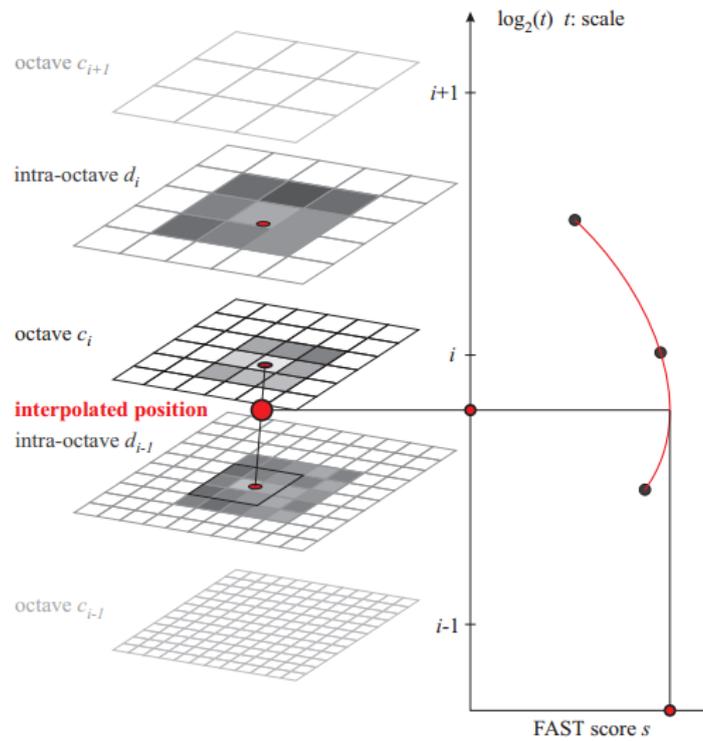


Figura 2.8 – FAST: Detección del punto de interés en el espacio escala

Fuente: [3]

Para la descripción de un punto característico, BRISK usa un patrón para pruebas binarias tal como se muestra en Fig. 2.9, se definen  $N$  ubicaciones igualmente espaciadas, concéntricas con el punto característico.

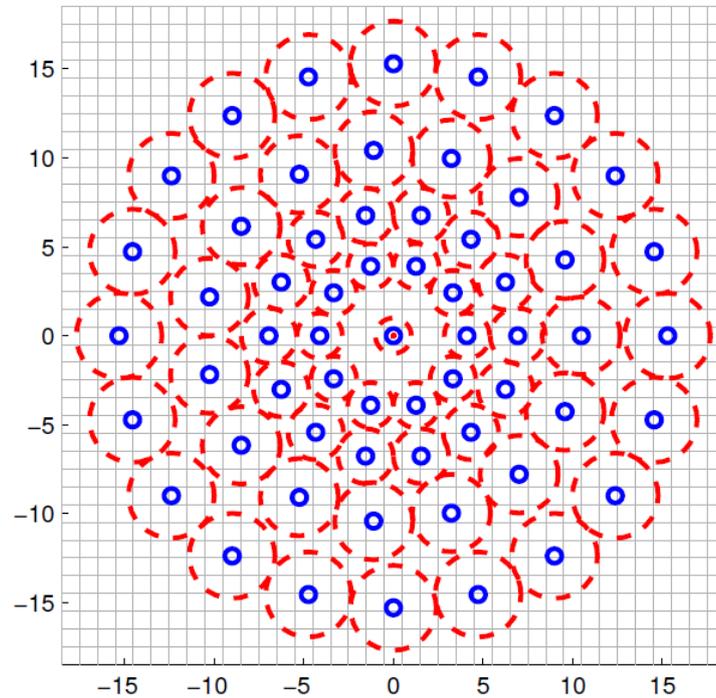


Figura 2.9 – Patrón de muestreo del Brisk tomando 60 puntos

Fuente: [3]

Para evitar efectos de alisado cuando se muestrea la intensidad en un punto  $p_i$  del patrón, aplicamos un filtro gaussianos con sigma proporcional a la distancia entre los puntos en el respectivo círculo, es decir para una prueba con puntos  $(p_i, p_j)$  tomamos las intensidades  $I(p_i, \sigma_i)$  y  $I(p_j, \sigma_j)$  para así calcular un gradiente local, definido como:

$$\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I(\mathbf{p}_j, \sigma_j) - I(\mathbf{p}_i, \sigma_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}$$

De todo el conjunto de puntos para pruebas A, tomamos el conjunto L para calcular el ángulo del punto  $\mathbf{g}$ :

$$\begin{aligned}\mathcal{A} &= \{(\mathbf{p}_i, \mathbf{p}_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\} \\ \mathcal{L} &= \{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \|\mathbf{p}_j - \mathbf{p}_i\| > \delta_{min}\} \subseteq \mathcal{A} \\ \mathbf{g} &= \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{L}} \mathbf{g}(\mathbf{p}_i, \mathbf{p}_j)\end{aligned}$$

Finalmente hacemos el descriptor invariante a rotación rotando el parche de pruebas  $\alpha$ -arctan2( $g_y, g_x$ ) grados, entonces una prueba queda definida como:

$$b = \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases}$$

$$\forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$$

BRISK conlleva algunas mejoras con respecto a BRIEF a pesar de que ambos se basan en comparaciones de píxeles, aparte de lo obvio como un pre-escalado y pre-rotación, BRISK usa un patrón de muestreo determinístico resultando en un muestreo uniforme de la densidad del punto característico, la trama de bits que describe al punto es de 512 bits.

#### 2.4.6. FREAK

Consiste de solo descriptor, se basa en el sistema de visión humano, específicamente la retina, sus siglas significan Fast Retina Keypoint, Un conjunto de cadena de bits es calculado de forma eficiente mediante la comparación de intensidades sobre un parche de

muestreo parecido al usado por la retina. Este algoritmo promete ser más rápido de procesar, con menor carga de memoria y también más robusto que SIFT, SURF o BRISK, especialmente para uso en aplicaciones embebidas. El sistema de visión humano usa una mayor cantidad de recursos en el centro de la retina, el centro de los que está mirando, análogamente se podría decir que se da una mayor resolución al centro de la imagen.

## 2.5. Estudios anteriores

Para una mejor visualización de las diferencias y ventajas de esta evaluación con respecto a anteriores mostraremos 3 evaluaciones de gran importancia y que han sido de inspiración para esta evaluación en forma de tabla denotando por filas las características puntuales.

Tabla 1 - Características de evaluación (Mikolajczyk & Schmid, 2005)

<b>Objetivo:</b>	Tratar de responder que descriptor es el más apropiado para caracterizar regiones de interés (diferente a puntos de interés) y si la elección del detector afecta el desempeño del descriptor.
<b>Transformaciones</b>	Rotación, escala, cambio de punto de vista, blur, compresión jpeg e iluminación. Creadas de

<b>Evaluadas:</b>	forma mecánica es decir cambio de escala mediante el zoom de la cámara y blur mediante cambio de foco.
<b>Dataset:</b>	Las imágenes se pueden categorizar en 2: Estructuradas (presencia de regiones homogéneas y bordes distintivos) y no estructuradas (texturas repetidas en formas diferentes).
<b>Definición de matches correctos:</b>	Dada una región A (región detectada en imagen original) y una región B (región que consideramos la misma detectada en imagen transformada), H (homografía que representa la transformación ideal preconcebida) y A' (la transformación de B hacia A usando H) es un match correcto si el área de error en la intersección A con A' es menor al 50% de la unión de sus áreas.
<b>Campo:</b>	Visible
<b>Algoritmos evaluados:</b>	Contexto de la figura, filtros dirigidos , PCA-SIFT, invariantes diferenciales, imágenes spin, SIFT,

	filtros complejos, invariantes de momento y relación cruzada para diferentes tipos de regiones de interés.
<b>Métricas de evaluación:</b>	recall con respecto a precisión

Tabla 2 - Características de evaluación (Miksik & Mikolajczyk, 2012)

<b>Objetivo:</b>	Comparar algoritmos considerando la ganancia de precisión contra la pérdida de velocidad y viceversa para extractores de puntos característicos recientes.
<b>Transformaciones Evaluadas:</b>	Transformaciones geométricas y fotométricas: Rotación, escala, cambio de punto de vista, blur, compresión jpeg e iluminación.
<b>Dataset:</b>	Imágenes no estructuradas.
<b>Campo:</b>	Visible
<b>Algoritmos evaluados:</b>	BRIEF, BRISK, ORB, MRRID, MROGH y LIOP

<b>Métricas de evaluación:</b>	repetitividad, precisión vs recall, factor de velocidad ( $St = Ts/Te$ , donde $Ts$ es tiempo para búsqueda secuencial y $Te$ tiempo para la búsqueda optimizada)
--------------------------------	---

Tabla 3 - Características de evaluación (Schmid, Mohr, & and Bauckhage, 2000)

<b>Objetivo:</b>	Evaluación de detectores, cual tiene mayor repetitividad.
<b>Transformaciones Evaluadas:</b>	Transformaciones geométricas y fotométricas: Rotación, escala, cambio de punto de vista, blur e iluminación.
<b>Dataset:</b>	Son fotos tomadas de cuadros, imágenes no estructuradas.
<b>Campo:</b>	Visible
<b>Algoritmos evaluados:</b>	Versión mejorada de Harris, Heitger, Horaud, Cottier, Forstner.
<b>Métricas de evaluación:</b>	Repetitividad y contenido de la información (mide la distintividad de los puntos característicos)

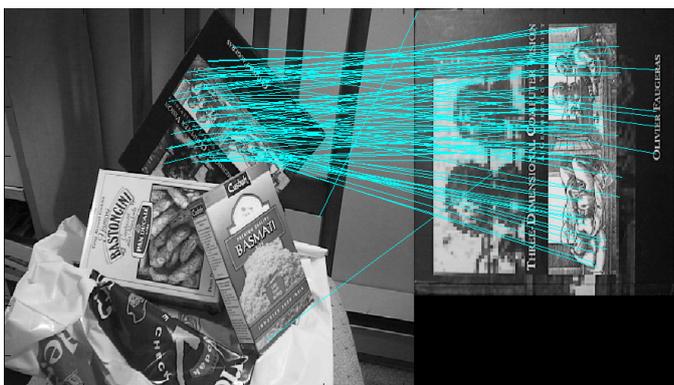
	detectados).
--	--------------

## 2.6. Aplicaciones que utilizan correspondencia de características

El primer paso a tomar en un problema de procesamiento de imágenes es el de decidir cuál será la representación de la información de la imagen. En la mayoría de los casos no es conveniente trabajar píxel a píxel de la imagen, sino que se utilizan descriptores los cuales se encargan de capturar las partes más representativas de la imagen. Por ello, en el campo de la visión artificial el uso de detectores-descriptores abarca un extenso uso en aplicaciones tales como:

- Reconocimiento de objetos

En este tipo de aplicaciones lo que se desea es detectar un objeto específico en una escena, basándose en encontrar características similares entre ambas imágenes a través de técnicas de procesamiento tales como realzado, contraste, obtención de puntos clave, etc. Generalmente, el reconocimiento se debe realizar enfrentándose a problemas tales como rotación, oclusión y escala. La Fig. 2.10 presenta una ilustración del reconocimiento de objetos usando correspondencias de características.



**Figura 2.10 - Detectando un objeto específico en una escena**

**Fuente:** <http://e-archivo.uc3m.es/bitstream/handle/10016/9929/PFC%20Abel%20Alguacil.pdf>

En este caso tenemos una aplicación para determinar si una imagen pertenece a una base de datos de imágenes previamente guardadas. En este caso, la imagen de prueba es capturada desde un punto de vista diferente de la imagen original, la aplicación deberá aprobar si la imagen pertenece o no. Se trabaja en dos fases, la fase de entrenamiento y la fase de consulta, en la fase de entrenamiento se obtienen los aspectos más representativos del conjunto de imágenes mediante el uso de descriptores y en la fase de consulta se obtienen puntos representativos de la imagen de prueba, un algoritmo de búsqueda deberá realizar correspondencia de características entre la imagen de prueba y las imágenes en la base de datos y finalmente el algoritmo de decisión determina si la imagen de prueba es aprobada (ver Fig. 2.11).

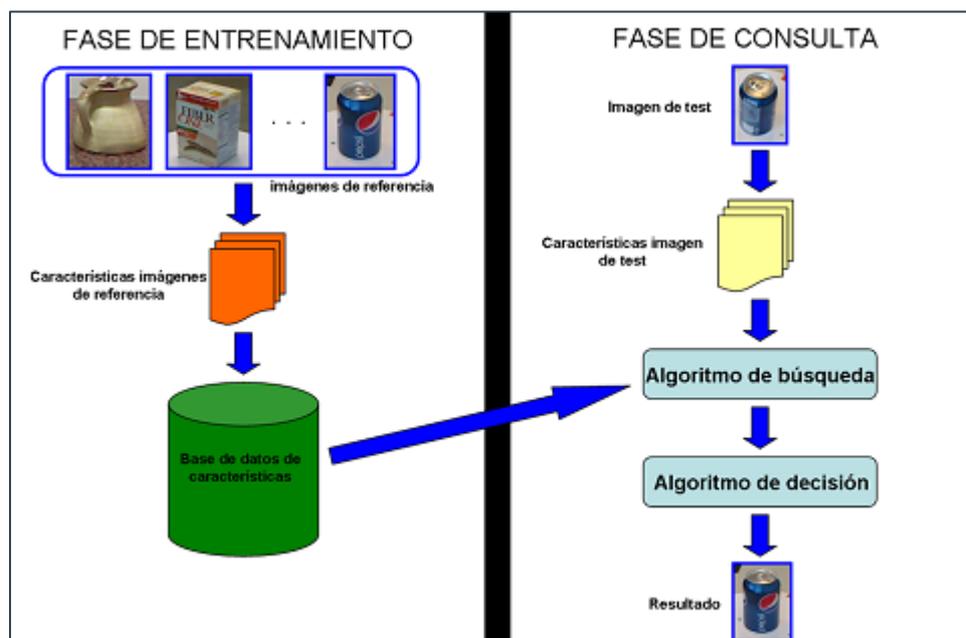


Figura 2.11 - Algoritmo de búsqueda de imágenes similares en una base de datos

Fuente: [http://robots.unizar.es/data/documentos/memoriaPFC\\_DavidBueno\\_mayo2012.pdf](http://robots.unizar.es/data/documentos/memoriaPFC_DavidBueno_mayo2012.pdf)

- Reconstrucciones 3D

Este tipo de aplicaciones es muy utilizado en el área de realidad aumentada, en la que se desea reconstruir una escena a partir de la imagen fuente. Se parte de una base de datos de imágenes con características representativas y de donde se obtendrá información relevante para la imagen fuente. En este caso particular, se desea realizar un montaje de la imagen fuente, a continuación se muestra la siguiente imagen donde se realiza una breve descripción del algoritmo a seguir (Fig. 2.12).

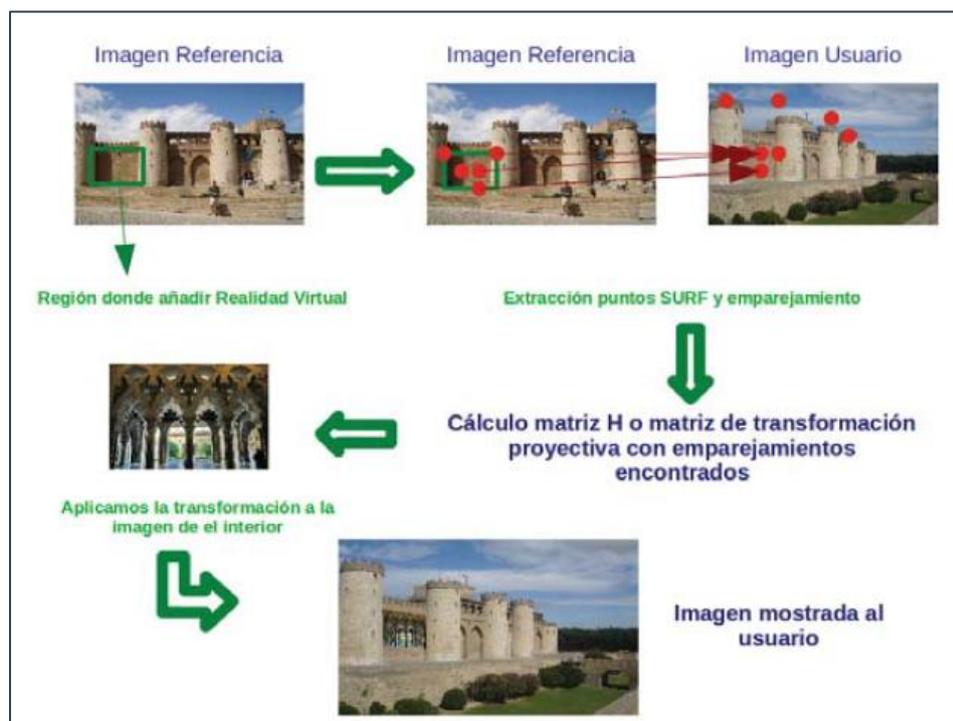


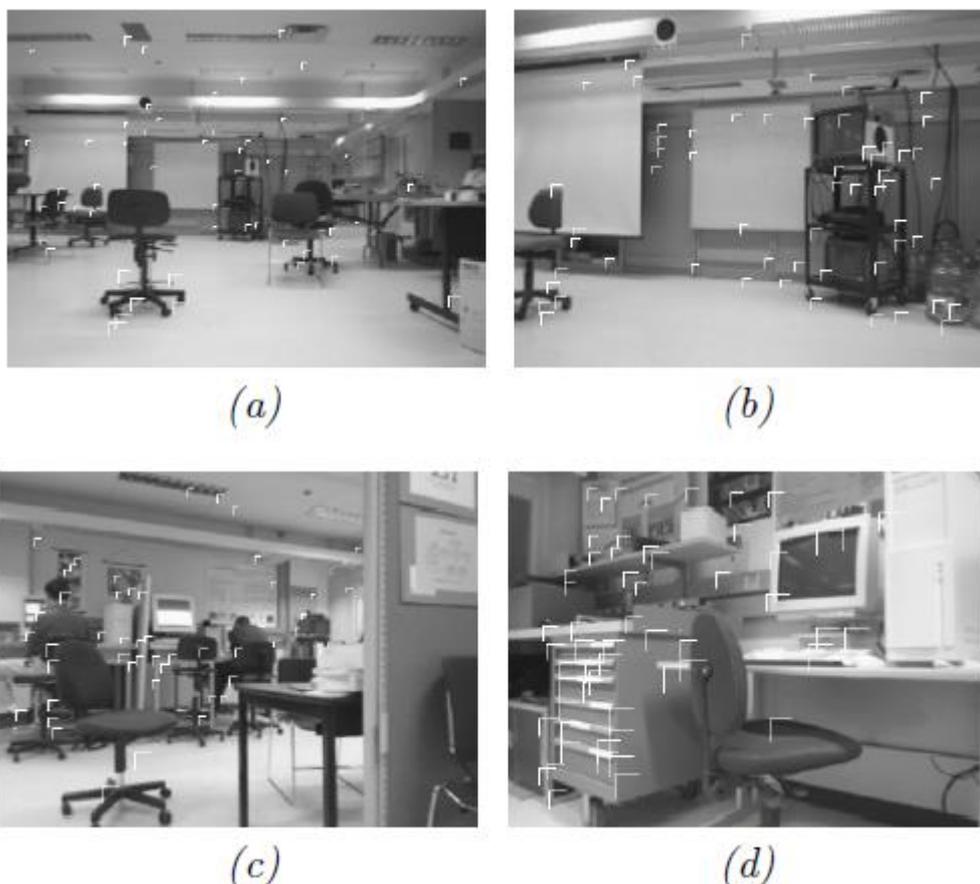
Figura 2.12 - Descripción gráfica del algoritmo

Fuente: <http://zaguan.unizar.es/TAZ/EINA/2012/6873/TAZ-PFC-2012-038.pdf>

- Robótica

Un papel importante de un sistema de robot móvil es la capacidad de localizarse con precisión y construir un mapa del medio ambiente simultáneamente. Esto proporciona una solución robusta y precisa para el problema de la localización del robot en entornos desconocidos. En este caso a medida que el robot se va desplazando se podría capturar todo lo que visualiza y mediante puntos de vistas claves construir un mapa de su entorno. La Fig. 2.13 muestra distintas imágenes obtenidas durante la navegación

de un robot en un entorno interior (se pueden observar los distintos puntos característicos correspondiente a cada punto de vista).



**Figura 2.13 - Detección de puntos claves en la escena capturada**

**Fuente:** <http://www.cs.ubc.ca/~lowe/papers/se05.pdf>

- Registrado de imágenes

Es el proceso en el cual se construye una vista panorámica a partir de la secuencia de un conjunto de imágenes pertenecientes a la misma escena (ver Fig. 2.14). Las imágenes son comparadas entre sí hasta encontrar los vecinos más cercanos entre ellas que

permitan relacionarlas y determinar correspondencia entre cada par de imágenes, en tal caso los detectores y descriptores utilizados juegan un papel importante.

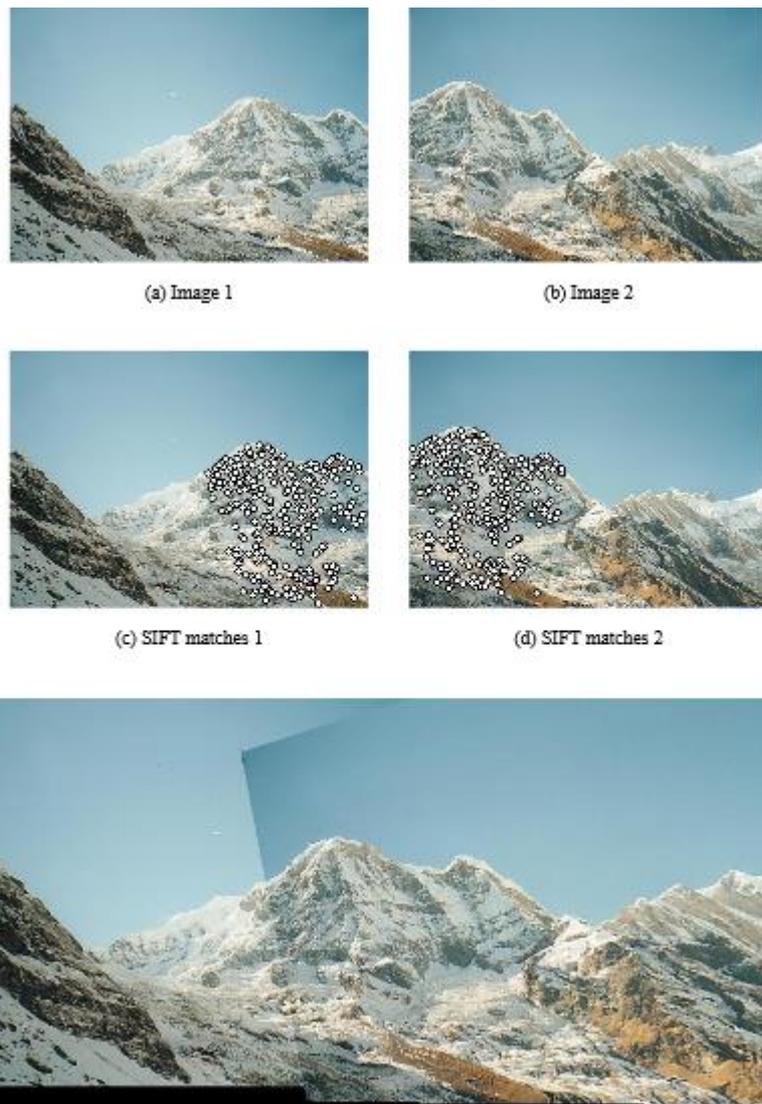


Figura 2.14 - Registrado de una escena a partir de dos imágenes

Fuente: [http://pages.cs.wisc.edu/~lizhang/courses/cs766-2008f/syllabus/10-02-feature/brown\\_lowe\\_iccv2003.pdf](http://pages.cs.wisc.edu/~lizhang/courses/cs766-2008f/syllabus/10-02-feature/brown_lowe_iccv2003.pdf)

- Software de dispositivos táctiles

Muchas aplicaciones se han realizado sobre reconocimiento de imágenes en dispositivos táctiles. Una de ellas es una aplicación turística en la que se toma foto de una imagen y se desea buscar la descripción turística del lugar. Todo esto hace uso de descriptores y procesamiento adicional para obtener tales resultados. Las distintas etapas de la aplicación móvil son ilustradas en la Fig. 2.15.

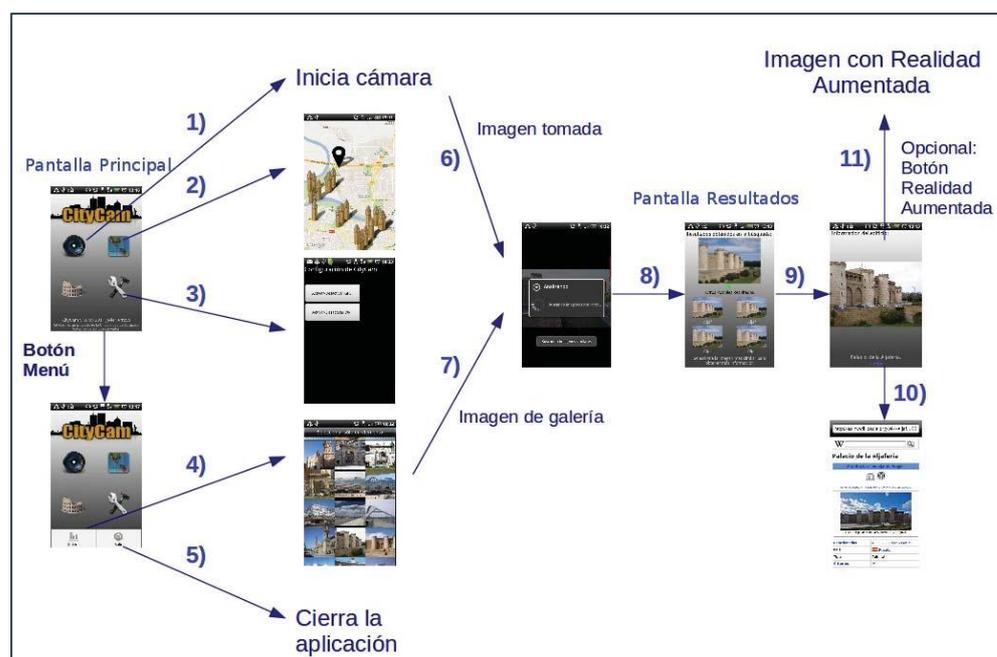


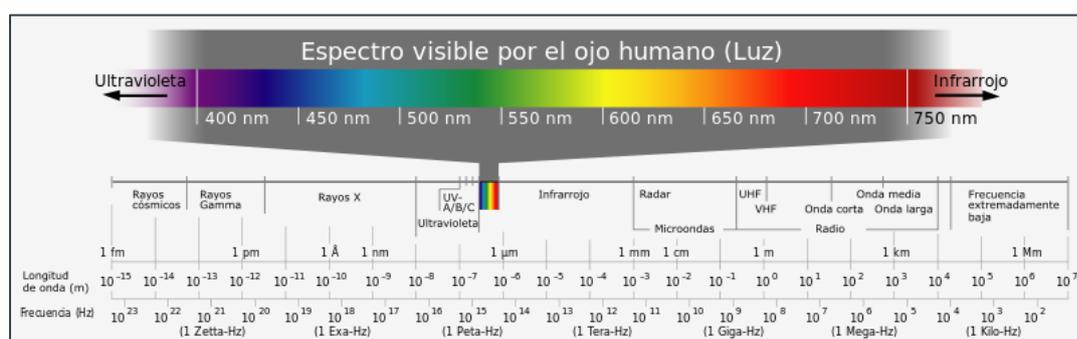
Figura 2.15 - Aplicación móvil (CityCam) para determinar la ubicación del lugar

Fuente: <http://zaguan.unizar.es/TAZ/EINA/2012/6873/TAZ-PFC-2012-038.pdf>

## 2.7. Naturaleza del campo visible e infrarrojo

### 2.7.1. Espectro visible

Se le llama espectro visible a la región del espectro electromagnético que el ojo humano es capaz de percibir. A la radiación electromagnética en este rango de longitudes de onda se le llama luz visible o simplemente luz. No hay límites exactos en el espectro visible; en general el ojo humano responde a longitudes de onda desde 400 a 700 nm aunque algunas personas pueden ser capaces de percibir longitudes de onda desde 380 a 780 nm. Este es el rango en el que el sol y las estrellas similares emiten la mayor parte de su radiación. Probablemente, no es una coincidencia que el ojo humano sea sensible a las longitudes de onda que emite el sol con más fuerza. Como se puede observar en la Fig. 2.16, la luz que vemos con nuestros ojos es realmente una parte muy pequeña del espectro electromagnético.



**Figura 2.16 - Espectro electromagnético**

Fuente: [http://es.wikipedia.org/wiki/Espectro\\_visible](http://es.wikipedia.org/wiki/Espectro_visible)

Si la radiación tiene una frecuencia en la región visible del espectro electromagnético se refleja en un objeto, por ejemplo, un tazón de

fruta, y luego golpea los ojos, esto da lugar a la percepción visual de la escena. Nuestro sistema visual del cerebro procesa la multitud de frecuencias que se reflejan en diferentes tonos y matices, y a través de este fenómeno psico-físico, la mayoría de la gente percibe un tazón de fruta; un arco iris muestra la óptica (visible) del espectro electromagnético. En la mayoría de las longitudes de onda, sin embargo, la radiación electromagnética no es visible directamente, aunque existe tecnología capaz de manipular y visualizar una amplia gama de longitudes de onda (ver ilustración Fig. 2.17).

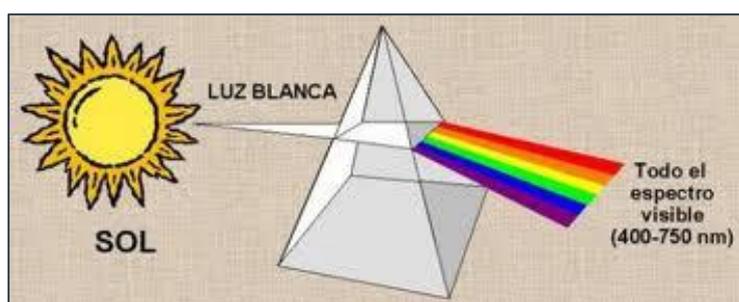


Figura 2.17 - Descomposición de un haz de luz visible

Fuente: <http://www.profesorenlinea.cl/fisica/Luzdescomposic.htm>

### 2.7.2. Infrarrojo

La radiación infrarroja, o radiación IR es un tipo de radiación electromagnética y térmica, de mayor longitud de onda que la luz visible, pero menor que la de las microondas. Consecuentemente,

tiene menor frecuencia que la luz visible y mayor que las microondas. Su rango de longitudes de onda va desde unos 0,7 hasta los 1000 micrómetros. La radiación infrarroja es emitida por cualquier cuerpo cuya temperatura sea mayor que 0 Kelvin, es decir,  $-273,15$  grados Celsius (cero absoluto). Los infrarrojos son clasificados, de acuerdo a su longitud de onda, de este modo:

- Infrarrojo cercano (de 800 nm a 2500 nm)
- Infrarrojo medio (de 2.5  $\mu\text{m}$  a 50  $\mu\text{m}$ )
- Infrarrojo lejano (de 50  $\mu\text{m}$  a 1000  $\mu\text{m}$ )

La radiación infrarroja se asocia generalmente con el calor. Ellas son producidas por cuerpos que emiten calor, aunque a veces pueden ser generadas por algunos diodos emisores de luz y algunos láseres.

Las señales son usadas para algunos sistemas especiales de comunicaciones, como en astronomía para detectar estrellas y otros cuerpos y para guías en armas, en los que se usan detectores de calor para descubrir cuerpos móviles en la oscuridad.



Figura 2.18 - Ejemplo de una imagen en campo infrarrojo

Fuente: [http://es.wikipedia.org/wiki/Radiaci%C3%B3n\\_infrarroja](http://es.wikipedia.org/wiki/Radiaci%C3%B3n_infrarroja)

### 2.7.3. Aplicaciones multiespectrales

#### PCA

Las imágenes multiespectrales guardan mucha más información que las imágenes en escalas de grises o a colores, además de brindar información puede usarse para comprimir su propia información, se puede comprimir una imagen multiespectral de  $n$  capas (imagen original) en una imagen multiespectral de un número reducido de capas (imagen comprimida) el cual será más fácil transportar y almacenar usando la transformada de KL. Cuando se necesite procesar la imagen multiespectral se puede recuperar la imagen a partir de la imagen comprimida. Una representación esquemática de este proceso es presentado en la Fig. 2.19.

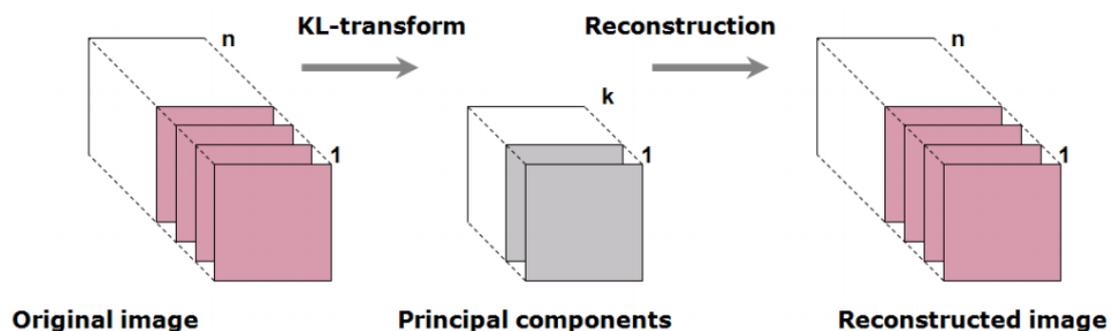
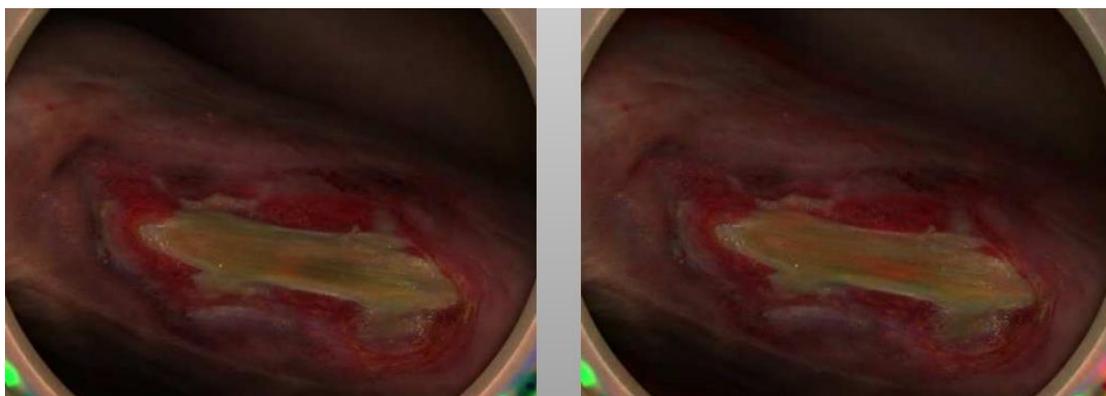


Figura 2.19 – Compresión de imágenes multiespectrales

Fuente: [http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect\\_multispec.pdf](http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect_multispec.pdf)

### Análisis de heridas en la piel

Para un mejor análisis de una herida se puede usar una imagen compuesta, esta imagen puede ser creada bajo muchos criterios, un criterio puede ser que a una imagen a color RGB de una herida a su canal R se lo fusione con una imagen del espectro infrarrojo cercano NIR; en la Fig. 2.20 se puede apreciar cómo se resalta una sección de una herida, la imagen original es la izquierda y en la parte derecha se resalta con un tono rojizo la parte interna a la zona amarilla.



**Figura 2.20 - Análisis de heridas**

Fuente: [http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect\\_multispec.pdf](http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect_multispec.pdf)

### **Comprobación de pinturas**

Se puede obtener una huella fotográfica que identifique una pintura, que pueda servir para decir si la pintura es original, esta huella no es más que una imagen multispectral que representa la reflectancia espectral de la pintura. Para obtener esta imagen se dispone de una cámara, varios filtros para diferentes rangos espectrales, la pintura y una fuente de iluminación tal como puede apreciarse en la Fig. 2.21, el número de capas depende del número de filtros.



**Figura 2.21 - Imágen multispectrales de obra "Madonna and child"**

Fuente: [http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect\\_multispec.pdf](http://lipas.uwasa.fi/~TAU/AUTO3090/Slides/lect_multispec.pdf)

Estas imágenes pueden ser tomadas con una simple cámara con sensor CCD mediante el uso de filtros, es decir que para tomar imágenes en el campo infrarrojo uso la misma cámara pero con un filtro para imágenes infrarrojas, hay que recordar que este filtro permite tomar fotos en una banda reducida y específica del espectro infrarrojo; una cámara multispectral se puede apreciar en la Fig. 2.22.



Figura 2.22 - Configuración de cámara para toma de imágenes multispectrales

Fuente: <http://www.lfb.rwth-aachen.de/en/research/basic-research/multispectral/>

### Estudios médicos

Uso de contraste, por ejemplo para examinar la sangre o los vasos sanguíneos se puede inyectar en los mismos idocyanina verde que emite una luz fluorescente a 830nm después de irradiarlo con luz a 780nm, esta luz fluorescente resalta las venas pero solo es apreciable en el campo NIR, podemos obtener la imagen en el campo visible fusionada con la del campo NIR y tener una imagen más rica en

información.

### **Aplicaciones de muestreo remoto**

**Agricultura:** Imágenes tomadas por satélite o por aeronaves son usadas para clasificar cultivos, examinar su salud, caracterizar suelos y monitorear prácticas de cultivo.

**Forestal:** reconocimiento de bosques, deforestación, inventariado de especies, estimación de biomasa, densidad de vegetación y medición de características biofísicas.

**Geología:** se puede obtener un reconocimiento de zonas del suelo midiendo la reflectancia espectral, se puede reconocer zonas minerales, yacimientos de hidrocarburos, mapeo y monitoreo de sedimentación, detección de amenazas geológicas, y mapeo planetario.

**Hidrología:** mapeo de inundaciones, zonas húmedas, cambio del nivel de agua en ríos, detección de canales de irrigación, medición del grosor de la nieve, dinámica de glaciares, entre otras.

**Océanos y costas:** se puede monitorear el nivel del océano, temperatura del agua, mapeo de rutas de navegación, identificaciones de patrones en el océano tales como corrientes matrones circulations regionales entre otros, identificación de tormentas,

ubicación de cardúmenes de peces y asentamiento de mamíferos marinos, estudios de densidad de tráfico en las rutas marinas, impacto de la actividad humana, entre otros.

### **Cultivo de precisión**

Se puede monitorear la irrigación de cultivos mediante imágenes satelitales o aún mejor por imágenes tomadas por aeronaves no tripuladas; como se aprecia en la Fig. 2.23, la imagen de la izquierda muestra con color azul las zonas de mayor forestación, la imagen intermedia muestra con color rojo las zonas de terreno secas, la imagen de la derecha resultado de las dos imágenes anteriores muestra con color amarillo las zonas donde hay alta forestación y poca irrigación de agua, podemos ver que las zonas 119 y 120 necesitan irrigación, a la falta de agua en un cultivo se le dice stress en el cultivo. La imagen izquierda es una imagen creada mediante una cámara infrarroja para poder medir la cantidad de forestación, esto se debe a que gracias a la clorofila de las plantas se refleja una cantidad apreciable de luz infrarroja con respecto al suelo, la imagen intermedia se obtiene mediante una imagen termal y la última mediante una diferencia entre las anteriores, obsérvese que estas imágenes no son netamente infrarrojas o netamente térmicas son llamadas falso color, para definir lo que significa falso color es mejor definir su contraparte

el color verdadero que es por definición una imagen idéntica a como se vería por un observador eso para el caso de imagen a color, para el caso de imagen en escala de grises se llama color verdadero si la intensidad es tal como la apreciaría un observador. Estas imágenes falso color son modificaciones que permiten presentar mejor la información.

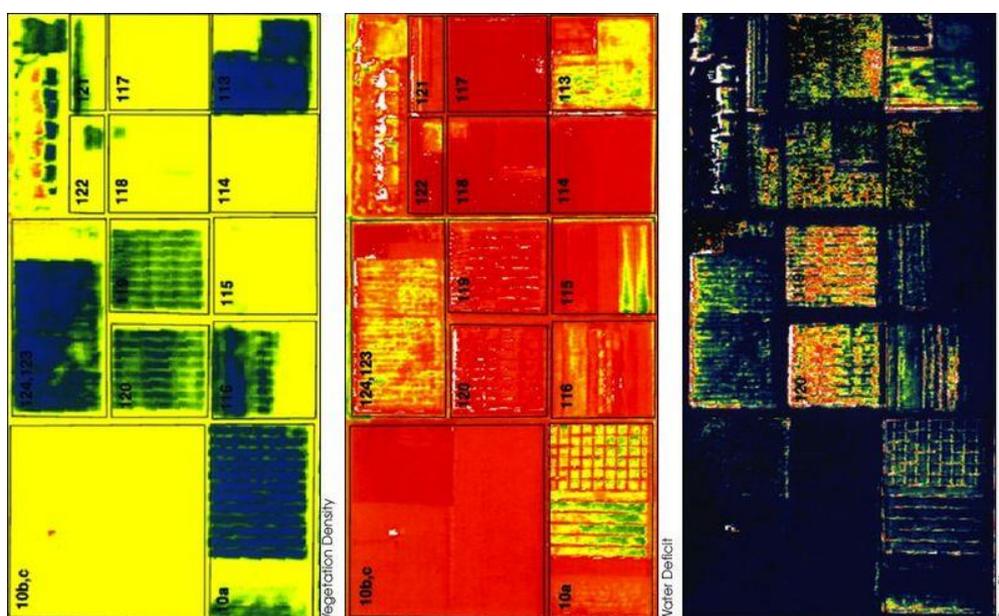
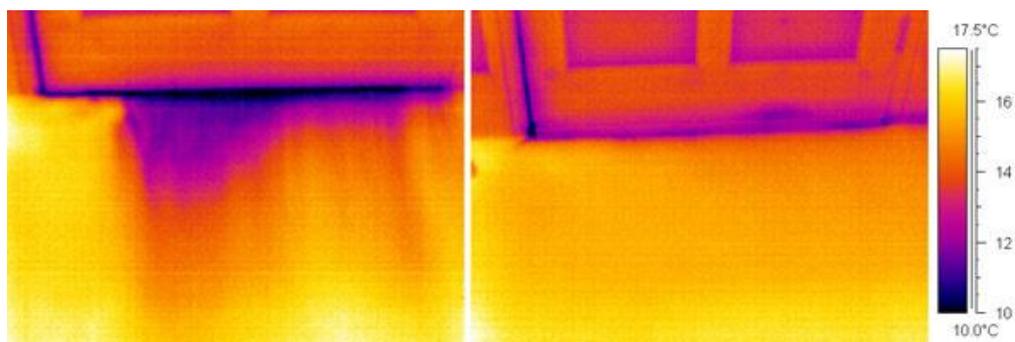


Figura 2.23 - Agricultura de precisión: imagen multispectral

Fuente: [http://en.wikipedia.org/wiki/False\\_color](http://en.wikipedia.org/wiki/False_color)

## Ingeniería

Cuando se desea verificar la confinación térmica de una habitación como por ejemplo una sala de servidores se puede usar una cámara de visión térmica para ver filtraciones existentes en rendijas o puertas (ver Fig. 2.24).



**Figura 2.24 - Filtrado de aire acondicionado por una puerta**

Fuente: <http://www.superhomes.org.uk/resources/draught-proofing/>

Para simulaciones o chequeo de partes, se puede hacer una inspección rápida sobre el funcionamiento de una maquinaria mediante una imagen térmica y el conocimiento previo sobre que partes deben estar calientes y cuales no tanto, toda esta información se puede presentar en una imagen compuesta de la imagen térmica y de una imagen visible o de una imagen térmica modificada llamada pseudo-color como se aprecia en la Fig. 2.25.



**Figura 2.25 - Imagen en pseudo-color de un tren, las partes calientes toman un color amarillo con la subida de temperatura**

Fuente: [http://en.wikipedia.org/wiki/False\\_color](http://en.wikipedia.org/wiki/False_color)

Para aplicaciones médicas, una imagen multispectral conformada por varias imágenes de niveles de resonancia magnética permite crear la imagen en pseudo-color de una rodilla donde los detalles son más fáciles de apreciar (Fig. 2.26, imagen derecha) a diferencia de una imagen simple de resonancia magnética como se aprecia en la Fig. 2.26 imagen izquierda.

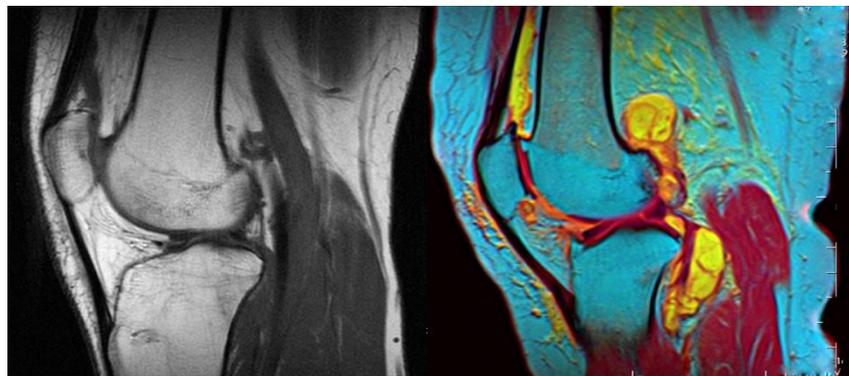


Figura 2.26 - Imagen de resonancia magnética de una rodilla

## CAPÍTULO 3

### 3. ESQUEMA DE EVALUACIÓN

#### 3.1. Funcionamiento

Para cada algoritmo y transformación se calcula una lista de números de matches correctos y porcentaje de matches correctos, cuyo tamaño depende del número de argumentos posibles para esa transformación. Entiéndase por algoritmo al par detector-descriptor detallados en la Tabla 4, así como por transformación al conjunto de problemas que afrontan los sistemas que utilizan correspondencia de puntos característicos mencionados en la sección anterior.

Es decir, para el algoritmo SURF/SURF y transformación rotación se obtendrá 36 valores de porcentaje de matches correctos y número de matches correctos, debido a que los argumentos posibles van de 0 a 360 con pasos de 10.

Para el cálculo de la lista de porcentaje y número de matches correctos de un algoritmo y transformación dado, trabajamos con la imagen en escala de grises de la imagen original, mediante el algoritmo se obtienen los puntos característicos y sus descriptores tanto para la imagen en escala de grises como para la imagen transformada de esta y se obtienen los matches mediante el método para matching descrito en la Tabla 4. Finalmente, obtenemos los matches correctos como se define en la sección 3.3.

Tabla 4 - Relación detector/descriptor con algoritmo de matching

Identificador	Detector	Descriptor	Matcher norm type
SIFT/SIFT	SIFT	SIFT	NORM_L2
SURF/SURF	SURF	SURF	NORM_L2
ORB/ORB	ORB	ORB	NORM_HAMMING
BRISK/BRISK	BRISK	BRISK	NORM_HAMMING
SURF/BRIEF	SURF	BRIEF	NORM_HAMMING
SURF/FREAK	SURF	FREAK	NORM_HAMMING

\*Todos los matchers usan fuerza bruta

### 3.2. Transformaciones

Las transformaciones son artificiales, generadas por computadora, entre estas tenemos: rotación, escala, blur y ruido aditivo, un ejemplo puede ser visto en la Fig. 3.1 y son detalladas a continuación:

**Blur:** Consiste de un filtro gaussiano aplicado iterativamente sobre la imagen. En cada iteración el tamaño del kernel del filtro ( $K \times K$ ) usado

para filtrar la imagen se actualiza como sigue:  $K=2t+1$ , donde  $t=\{1,2,\dots,9\}$ .

**Rotación en el plano:** la imagen original es rotada con respecto a su centro con un valor entre 0 y 360 grados con pasos de 10 en cada prueba.

**Cambio de escala:** se redimensiona la imagen original con un valor entre 0.2 y 2 con pasos de 0.1 en cada prueba, el valor de los píxeles es obtenido mediante interpolación lineal.

**Adición de ruido:** es un ruido aditivo que se obtiene mediante la suma de la imagen original más una personalizada. Esta imagen personalizada es creada mediante una función que genera números aleatorios uniformemente distribuidos con media igual a 0 y desviación estándar igual a  $t$ ,  $t$  varía entre 0 y 100 con pasos de 10 en cada prueba.

La Fig. 3.1 muestra una ilustración de un par de imágenes, capturadas en ambos espectros (visible e infrarrojo) con sus respectivas transformaciones descritas previamente.

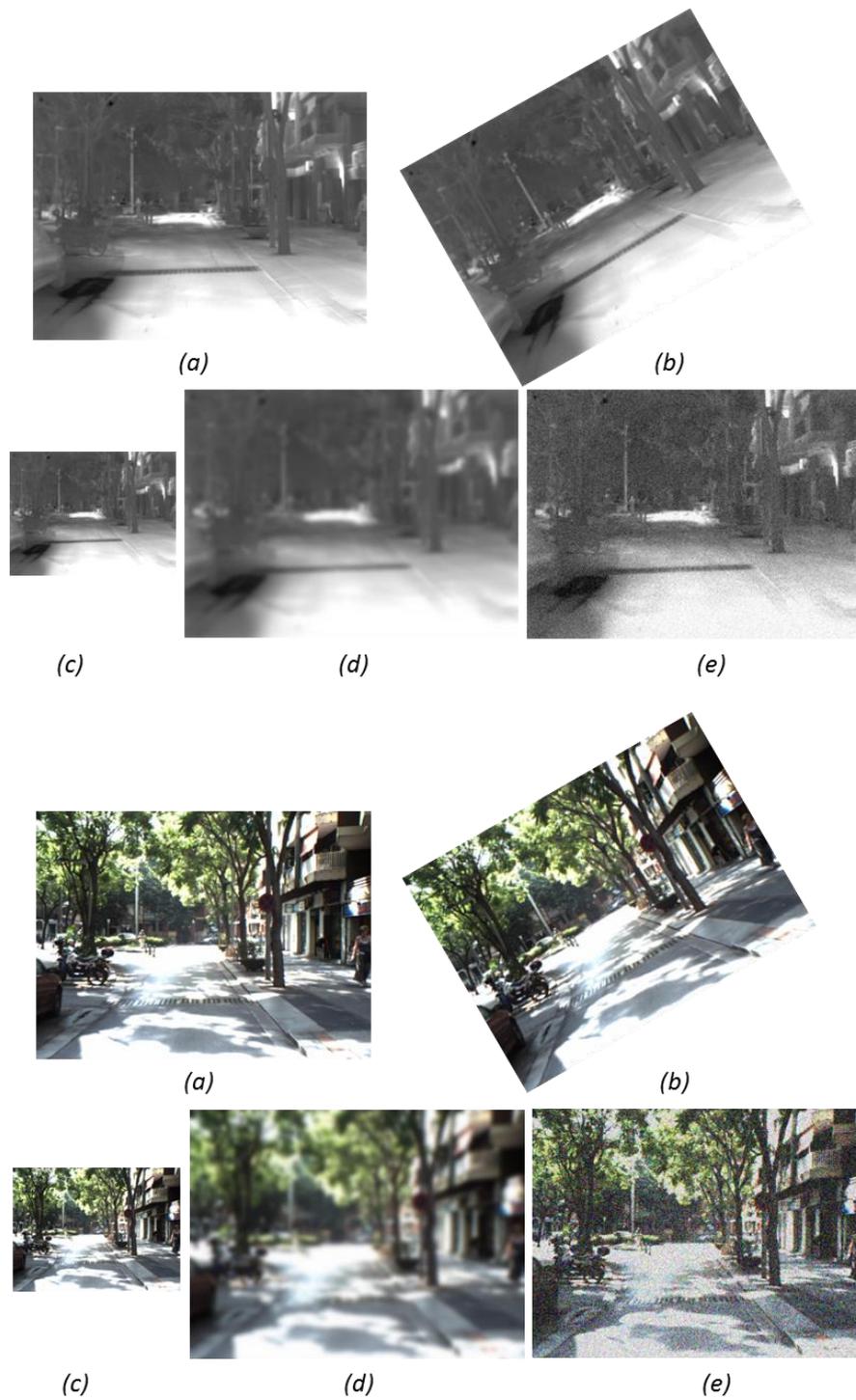


Figura 3.1 - Ilustración de un par de imágenes pertenecientes al dataset ((arriba) LWIR y (abajo) VS) con sus correspondientes transformaciones: (a) imagen original; (b) rotación; (c) escala; (d) blur; (e) ruido.

### 3.3. Cálculo de matches correctos

Para facilitar la explicación consideraremos lo siguiente: cada match tiene un punto origen y un destino que corresponde al punto detectado en la imagen original y en la imagen transformada respectivamente. Al aplicar matching, obtendremos un conjunto de matches incorrectos mucho menor que el conjunto de matches correctos, es por ello que podemos calcular una matriz de homografía bastante parecida a la matriz de homografía correcta, podemos hacer uso de esta matriz casi correcta para calcular unos puntos retro proyectados, si los puntos de interés en la imagen transformada están bien localizados al retro proyectarlos tomarían las posiciones originales en la imagen a transformar, el que estén distantes de esta posición original se debe a 2 factores: si la distancia es corta se debe a que la matriz utilizada es una aproximación a la correcta que representa la transformación y si la distancia es larga es porque no es un match correcto, corto o largo está sujeto a la percepción del usuario. Definiremos como match correcto si la distancia es menor a 2 píxeles.

Particularmente a esta evaluación y en más detalle, a partir de los matches obtenemos la matriz de homografía  $H$  usando la optimización RANSAC (para obtener una aproximación rápida) y umbral de retroproyección igual a 2. Con esta matriz  $H$  y los puntos destino del

conjunto de matches se obtienen los puntos de origen retro proyectados, se filtran los matches quedándonos con aquellos matches cuya distancia entre los puntos origen y origen retro proyectados sea menor al umbral. De estos matches filtrados se recalcula la matriz de homografía  $H'$  usando el mismo umbral pero con el método de mínimos cuadrados (contrasta con todos los puntos), esta matriz  $H'$  es más refinada que la anterior. Mediante  $H'$  se recalculan los puntos retro proyectados y se vuelve a filtrar como antes con el mismo umbral, estos matches filtrados son considerados como correctos.

## **CAPÍTULO 4**

### **4. RESULTADOS EXPERIMENTALES**

#### **4.1. Definición del dataset**

Las imágenes utilizadas son obtenidas de un sistema estero multiespectral, consiste de dos cámaras sobre una estructura que permite que estén al mismo nivel imitando el sistema de visión humano, una de las cámaras para el campo visible tipo ACE de la marca Basler con una resolución de 658x492 pixeles, la otra una cámara para el campo LWIR tipo Gobi-640-GigE de la compañía Xenixs. Las distancias focales han sido ajustadas de forma que las imágenes captadas en ambas cámaras tengan una cantidad similar de información, la Fig. 4.1 muestra algunas imágenes del dataset en ambos espectros.



Figura 4.1 - Par de imágenes en ambos espectros ((izq.) visible (VS), (der.) infrarrojo (LWIR)) contenidas en el dataset evaluado

## 4.2. Definición de métricas de evaluación

Para cada algoritmo (i) y transformación (j) se calcula el número de

matches correctos para dicho algoritmo frente al número total de correspondencias. El número de correspondencias se refiere al número total de puntos a detectar, estos son todos los puntos conocidos de la imagen original tomando como referencia al algoritmo SIFT, que para cualquier imagen transformada y algoritmo a usar sería el número máximo de matches posibles. Esto nos permite medir el performance de los algoritmos para cada una de las pruebas realizadas; el cálculo en detalle se muestra a continuación:

$$performance = \frac{\# \text{ matches correctos (Alg. (i), Transf. (j))}}{\# \text{ correspondencias (SIFT)}}$$

Notar que la medida del performance puede dar valores mayores a uno, lo cual significa que el algoritmo evaluado obtiene más características que el mismo SIFT.

Para poder extraer conclusiones adicionales proponemos el cómputo del recall, similar al estudio propuesto en [4], para cada experimento con las diferentes transformaciones. El cual es calculado como sigue:

$$recall = \frac{\# \text{ matches correctos}}{\# \text{ correspondencias}}$$

donde el #correspondencias representa el número de características detectadas-descritas en la imagen dada por el algoritmo que está

siendo probado (usado como referencia), y el número de matches correctos son los matches obtenidos después de transformar la imagen y detectar-describir puntos característicos con el algoritmo que está siendo probado. Este recall es computado para los diferentes combinaciones de algoritmos y transformaciones ( $recall_i^j$ , donde  $i=\{\text{blur, rotación, ruido, escala}\}$  y  $j=\{\text{BRISK, ORB, SIFT, BRIEF, FREAK, SURF}\}$ ) y para cada set de imágenes ( $recall_{LWIR}_i^j$ ,  $recall_{VS}_i^j$ ). Finalmente, proponemos calcular el promedio de la diferencia de vectores en LWIR y el campo visible (nos referiremos a esto como PDR: promedio de la diferencia del recall). Este valor puede ser usado para comparar el desempeño de un algoritmo en cada banda espectral (un valor negativo significa que su desempeño es mejor en el espectro visible que en el infrarrojo):

$$PDR_i^j = \frac{\sum_{k=1}^n recall_{LWIR}_{i_k}^j - recall_{VS}_{i_k}^j}{n}$$

Donde n depende de la transformación, para el caso de la rotación que contiene 36 transformaciones (ver Fig. 4.2 para más detalles). El PDR da una idea de la diferencia de desempeño en cada transformación. Debido a que este estudio se extiende al espectro LWIR identificamos el algoritmo con mejor comportamiento en este dominio.

La métrica del tiempo, es el tiempo que demora el algoritmo en

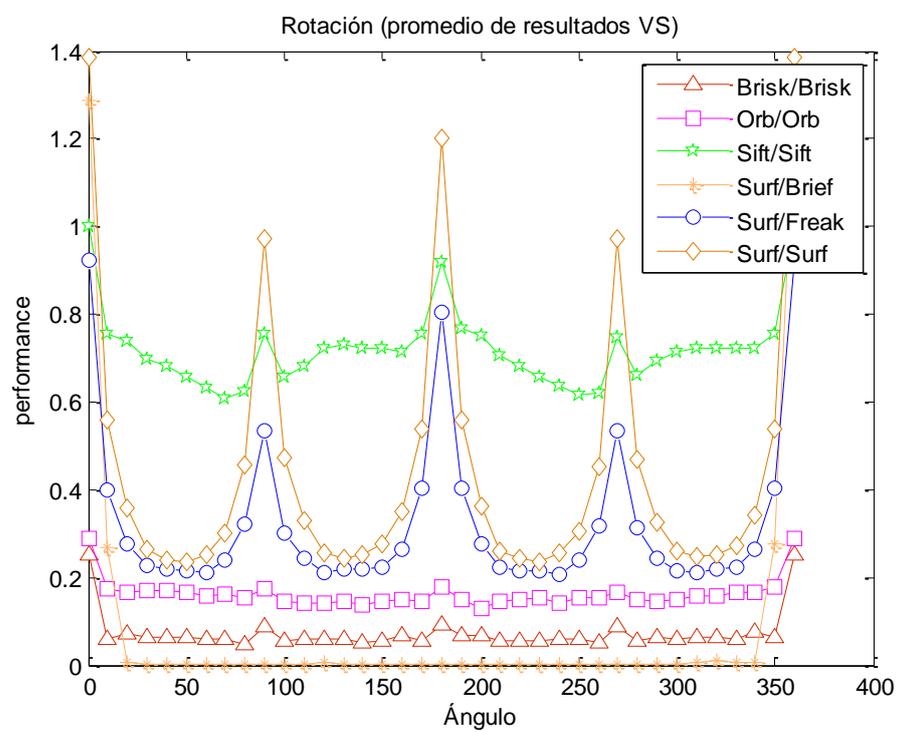
detectar y describir los puntos característicos en la imagen transformada en milisegundos. Mientras menor sea este valor, mejor.

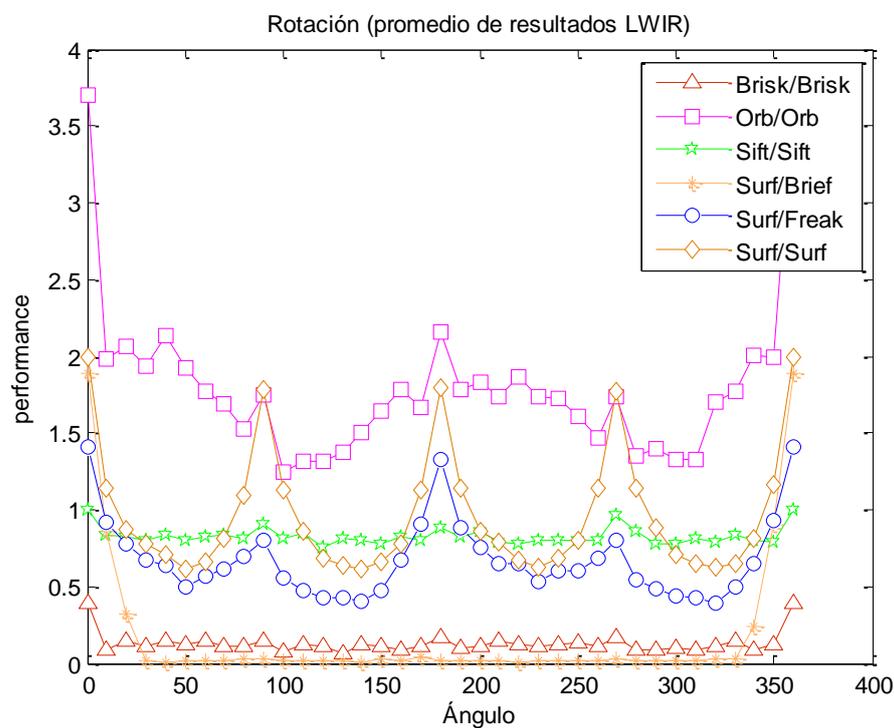
### **4.3. Contraste de resultados**

#### **4.3.1. Resultados de Performance**

La Fig. 4.2 muestra un promedio de los resultados obtenidos durante la prueba de rotación. Podemos observar en ambos casos, tanto para el caso VS como para el caso LWIR que el algoritmo más robusto a rotación es el SIFT. Su performance se mantiene muy constante a lo largo de las diferentes rotaciones (en particular en el caso LWIR); el cual solo decremento al inicio ( $\pm 10$  grados) pero luego no presenta muchos cambios. Por otro lado, el algoritmo BRIEF (usando SURF como detector) es el más sensible a rotaciones; de hecho, su performance bajo a cero justo después de aplicar una rotación de 20 grados en el campo VS y luego de una rotación de 30 grados en el campo LWIR. En el caso de SURF y FREAK, un ligero mejor performance fue apreciado en el caso LWIR donde la performance no decremento tanto como en el campo VS. También podemos apreciar que al usar el número de puntos detectados por SIFT como referencia, nos permite visualizar que el ORB tiene una considerable mayor cantidad de puntos detectados en el caso LWIR. Muy a pesar de ello, su performance no es tan bueno como el caso VS, mostrando un alto

decremento justo después de una rotación de 10 grados. Finalmente, BRISK muestra una pobre performance en ambos dominios.

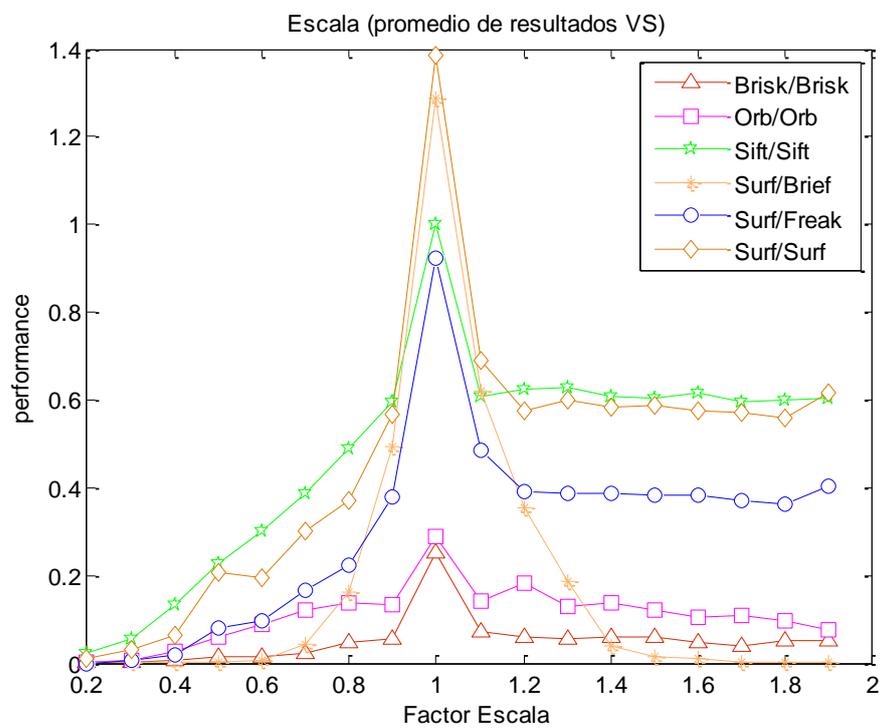


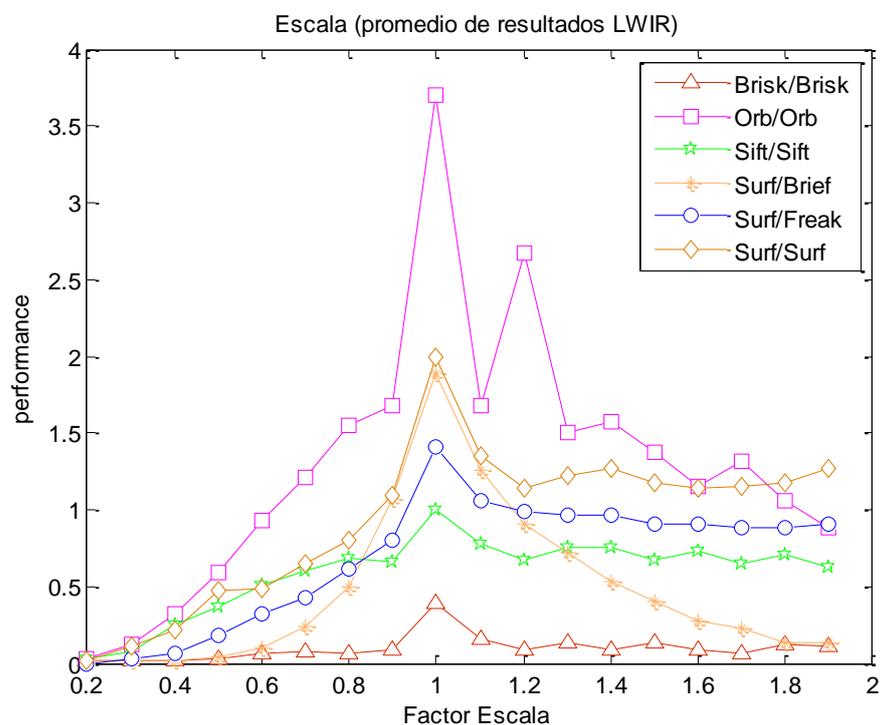


**Figura 4.2 - Performance en el caso de rotación: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR).**

La Fig. 4.3 muestra los resultados obtenidos durante la prueba de cambios de escala. El promedio de los algoritmos presenta una mejor performance en el dominio LWIR que en el dominio VS. En ambos casos BRISK muestra la peor performance muy seguido por BRIEF. Los algoritmos SIFT, FREAK y SURF son los más estables con respecto a los cambios de escala. Muy similar al anterior caso, ORB es bueno para detectar un gran número de puntos en el dominio LWIR, aunque su performance decae considerablemente, muchas de las veces es el algoritmo con más puntos detectados. Al contrario, en el dominio VS, este se encuentra entre uno de los algoritmos con

menos puntos detectados y su performance en este dominio es bastante estable.





**Figura 4.3 - Performance para cambios en escala: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR).**

La Fig. 4.4 muestra el caso de estudio de robustez de los diferentes algoritmos cuando las imágenes dadas son degradadas usando un filtro Gaussiano cuyo tamaño aumenta. En general todos los algoritmos en ambos espectros son igualmente afectados mostrando un decrecimiento en la performance con el incremento del tamaño del kernel. En el caso particular de LWIR, ORB muestra la peor performance; en otras palabras parece ser el más sensible al blur. Este hecho puede ser apreciado en la rápida caída de su performance.

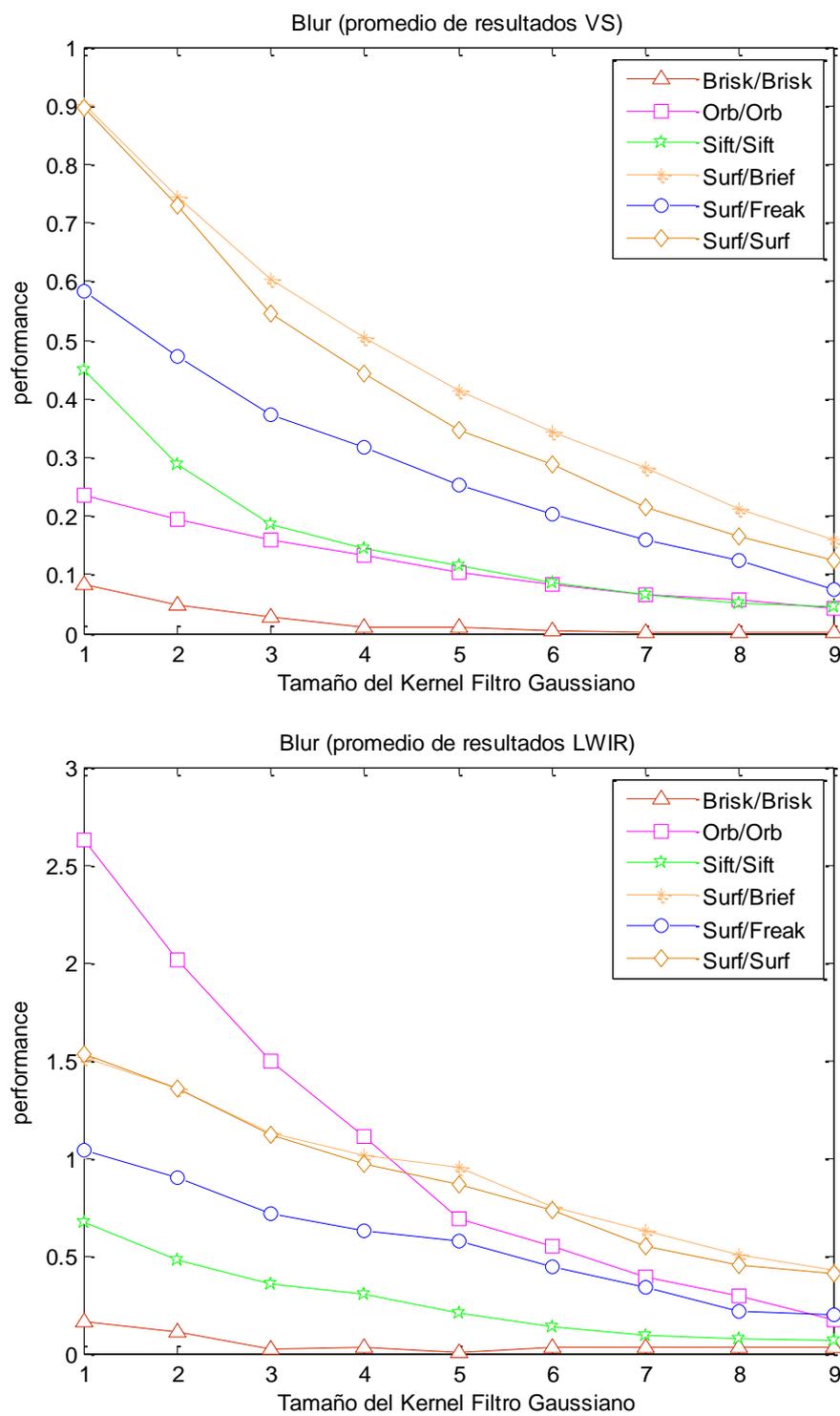


Figura 4.4 - Performance promedio ante la degradación de blur: (arriba) espectro visible (VS), (abajo) espectro infrarrojo (LWIR).

Finalmente, la Fig. 4.5 muestra las curvas obtenidas cuando ruido aditivo es aplicado. Como es esperada la performance de todos los algoritmos es degradado con ruido. Similar al caso de blur, la performance de todos los algoritmos decrece con un comportamiento similar. En el espectro visible ORB es uno de los más robustos, mientras que su performance en el campo LWIR es el peor. Este mal desempeño y su sensibilidad al ruido pueden ser explicados por la naturaleza de las imágenes (bajo contraste) en conjunto a la forma en que el algoritmo detecta los puntos característicos (basado en FAST, el cual usa un umbral de intensidad entre el pixel central y aquellos en un anillo circular). Al contrario, en el campo visible, aunque ORB es afectado como todos los algoritmos por el ruido, las imágenes tienen un mayor contraste por el ruido en el campo LWIR que en el visible.

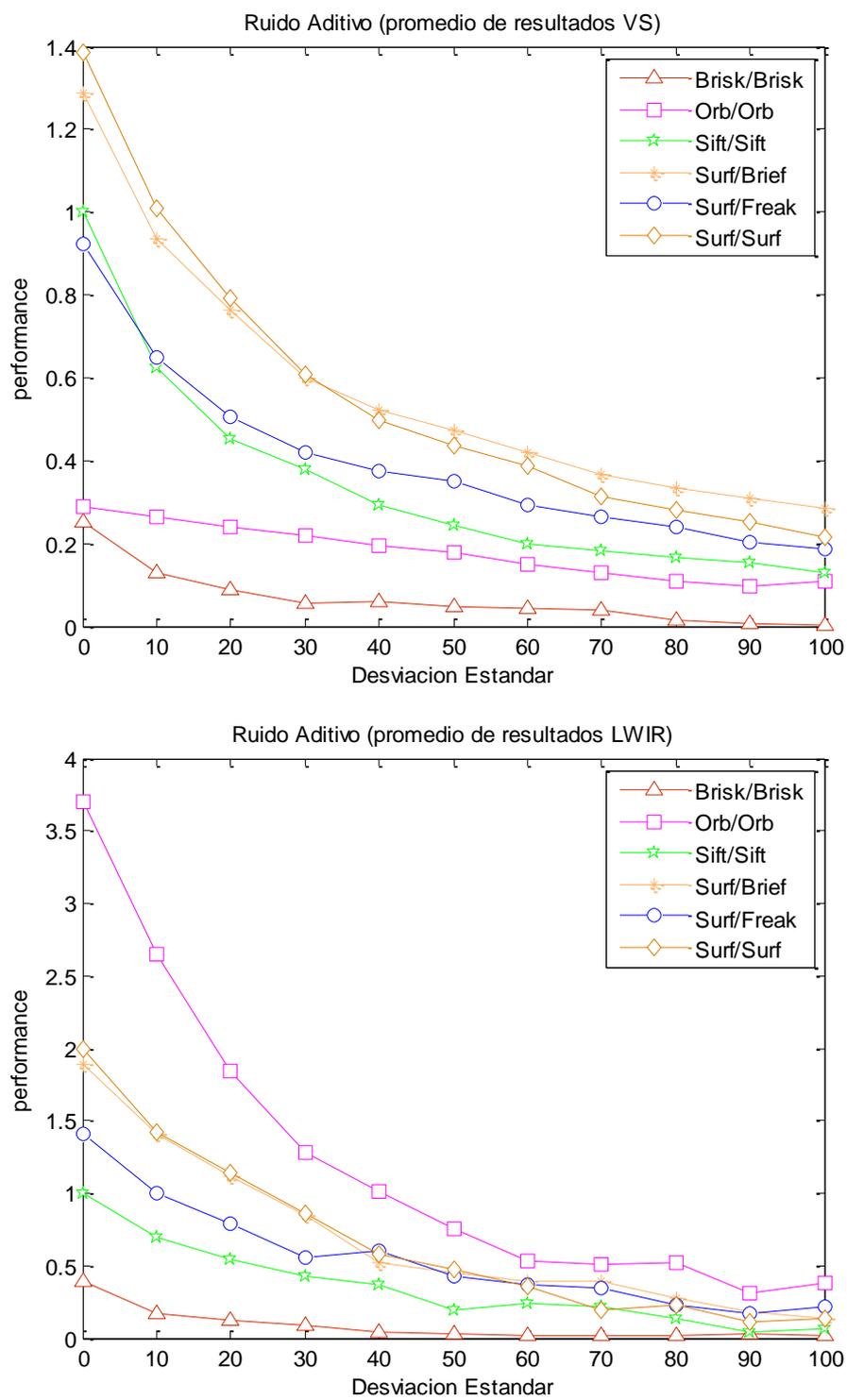


Figura 4.5 - Caso de estudio para el ruido: (arriba) espectro visible (VS), (abajo) campo infrarrojo (LWIR).

### 4.3.2. Resultados PDR

De acuerdo a los resultados mostrados en la Tabla 5 en promedio el algoritmo SURF tiene el mejor comportamiento en el LWIR para la transformación BLUR; mientras que en el caso de rotación SIFT parece ser el mejor, donde de alguna manera corrobora los resultados presentados en la Fig. 4.2, donde SIFT es el más estable de todos. En el caso de ruido todos los algoritmos tienen mal performance en el campo LWIR, siendo BRISK el más estable. Finalmente, en el caso de cambios en escala el algoritmo SIFT tiene un mejor comportamiento en LWIR que en el visible, esto puede ser apreciado comparando las curvas en Fig. 4.3, donde SIFT (seguido por FREAK) muestra un comportamiento robusto con respecto a cambios en escala en el espectro LWIR.

**Tabla 5 - Resultados PDR para cada algoritmo**

PDR	BRISK	ORB	SIFT	BRIEF	FREAK	SURF
Blur	0.0442	-0.1323	0.1064	0.1149	0.0904	0.1425
Rotación	0.0450	-0.0762	0.0726	0.0109	0.0584	0.0013
Ruido	-0.0427	-0.2921	-0.0764	-0.1266	-0.1273	-0.1106
Escala	0.0598	-0.0250	0.1564	0.0853	0.1271	0.1126

### 4.3.3. Tiempos de ejecución

Debido a la falta de textura de las imágenes en el espectro infrarrojo (LWIR), podemos observar que en general los tiempos de ejecución

en dicho espectro son bajos con respecto al campo visible. Una observación que podemos notar es la breve similitud que aparece en los tiempos de ejecución del algoritmo ORB/ORB en ambos espectros.

**Tabla 6 - Performance para el campo visible (VS)**

Algoritmo	Promedio de tiempo por frame	Promedio de tiempo por punto
BRISK/BRISK	241.266	0.533405
ORB/ORB	176.605	0.360627
SIFT/SIFT	4414.62	2.82664
SURF/BRIEF	3412.83	1.40007
SURF/FREAK	3255.03	2.62378
SURF/SURF	8490.84	3.27882

**Tabla 7 - Performance para el campo infrarrojo (LWIR)**

Algoritmo	Promedio de tiempo por frame	Promedio de tiempo por punto
BRISK/BRISK	64.421	1.01507
ORB/ORB	138.157	0.367576
SIFT/SIFT	1992.2	14.8781
SURF/BRIEF	1045.45	2.33826
SURF/FREAK	1017.74	3.72659
SURF/SURF	2569.03	4.51942

## CONCLUSIONES

Los algoritmos en el estado del arte puede ser clasificados como: i) basado en la información del gradiente. ii) basado en la información de intensidad de píxeles. Estudiamos donde sea posible encontrar una correlación entre los descriptores de una familia y la subida o caída del performance en los diferentes experimentos. La falta de textura en el campo LWIR fue uno de las consideraciones observadas en nuestro estudio. Como es una de las características de las imágenes LWIR hemos intentado ver como es afectada la performance principalmente en aquellos descriptores basados en uso de la información del gradiente. Las imágenes usadas son obtenidas directamente de la cámara, no existe procesamiento previo para mejorar o filtrar el contraste.

Este trabajo presenta una evaluación empírica de la performance de los descriptores del estado del arte cuando son usados en el dominio

LWIR y comparando los resultados con los obtenidos en el espectro visible. Sin embargo es difícil hacer una conclusión sobre cuál es el mejor algoritmo detector y descriptor de características debido a que depende de diferentes factores y de acuerdo a la Tabla 5 no hay un ganador, podemos ver que SIFT está de entre los mejores mostrando un buen desempeño en la mayoría de los experimentos. También observando los resultados presentados en la sección 4.3.1 podemos concluir que el algoritmo ORB es el que más características detecta en el campo LWIR. Esta conclusión es relacionada con la falta de textura y el bajo contraste de las imágenes del campo LWIR.

Como trabajo futuro se desea explorar el uso de técnicas de pre procesamiento para mejorar las imágenes LWIR antes de aplicar un algoritmo de detección y descripción.

## RECOMENDACIONES

Esta evaluación hace uso de algoritmos recientes o reconocidos por su trascendencia, se puede continuar esta evaluación incluyendo algoritmos más recientes tales como el kaze o el Non-Uniform SURF que no pudieron ser incluidos en esta evaluación, se debe tomar en cuenta que la implementación del algoritmo es proporcionada inicialmente por el autor y que con el pasar del tiempo esos algoritmos son optimizados por ello pueden haber evaluaciones iguales pero con resultados sensiblemente diferentes.

La métrica Performance le sugiere al lector el tiempo que le tomaría en procesar una imagen mediante un algoritmo dado y según ello escoger el algoritmo que más le convenga para su proyecto, sin embargo estos tiempos pueden parecer muy elevados, se sugiere para una evaluación futura realizar la misma evaluación pero usando la librería CUDA para acelerar el proceso usando una tarjeta gráfica y así mostrar estadísticas más alentadoras.

Se recomienda usar pasos más cortos, por ejemplo en lugar de 36 imágenes cada 10 grados en el caso de la rotación se podrían generar 72, una cada 5 grados, pasos muy grandes ocultan el comportamiento verdadero de la gráfica resultante para un algoritmo y transformación dada.

## ANEXO

### Publicaciones resultantes de este trabajo

#### Revista

Pablo Ricaurte, Carmen Chilán, Cristhian A. Aguilera-Carrasco, Boris X. Vintimilla and Angel D. Sappa, “*Feature Point Descriptors: Infrared and Visible Spectrum*”, *Sensors* (artículo aceptado para su publicación).

#### Conferencia

Pablo Ricaurte, Carmen Chilán, Cristhian A. Aguilera-Carrasco, Boris X. Vintimilla and Angel D. Sappa, “*Performance Evaluation of Feature Point Descriptors in the Infrared Domain*”, International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, January 5-8, 2014.

## BIBLIOGRAFÍA

- [1] *Measuring corner properties*. **Rosin., P. L.** 1999, Computer Vision and Image Understanding, pp. 73(2):291 – 307.
- [2] *A performance evaluation of local descriptors*. **Mikolajczyk, K. and Schmid, C.** 2005, IEEE Trans. Pattern Anal. Mach. Intell., pp. 27(10):1615–1630.
- [3] *Evaluation of local detectors and descriptors for fast feature matching*. **Miksik, O. and Mikolajczyk, K.** Tsukuba, Japan : ICPR 2012, 2012. In Proceedings of the 21st International Conference on Pattern Recognition. pp. 2681–2684.
- [4] *Evaluation of interest point detectors*. **Schmid, C., Mohr, R. and Bauckhage, C.** 2000, International Journal of Computer Vision, pp. 37(2):151–172.
- [5] Wikipedia. *Automatic number plate recognition*. [Online]
- [6] *Non-Parametric Local Transforms for Computing Visual Correspondance*. **Zabih, R. and Woodfill, J.** 1994. Proc. Third European Conf. pp. 151-158.
- [7] *Comparing several implementations of two recently published feature detectors*. **Bauer, J., Snderhauf, N. and Protzel, P.** Toulouse, France : s.n., 2007. In Proceedings of the International Conference on Intelligent and Autonomous Systems.

[8] *Object recognition from local scale invariant features*. **Lowe, D.G.** Kerkyra, Greece : s.n., September 20-27, 1999. In Proceedings of the IEEE International Conference on Computer Vision. pp. 1150–1157.

[9] *SURF: Speeded Up Robust Features*. **Bay, H., Tuytelaars, T. and Gool, L.J.** Graz, Austria : s.n., May 7-13, 2006. In Proceedings of the 9th European Conference on Computer Vision. pp. 404–417.

[10] *ORB: An efficient alternative to SIFT or SURF*. **Rublee, E., et al., et al.** Barcelona, Spain : s.n., November 6-13, 2011. In Proceedings of the IEEE International Conference on Computer Vision. pp. 2564–2571.

[11] *BRISK: Binary Robust Invariant Scalable Keypoints*. **Leutenegger S., Chli M., and Siegwart R.** Barcelona, Spain : s.n., November 6-13, 2011. In Proceedings of the IEEE International Conference on Computer Vision. pp. 2548–2555.

[12] *BRIEF: Computing a local binary descriptor very fast*. **Calonder, M., et al., et al.** 2012, IEEE Trans. Pattern Anal. Mach. Intell., pp. 1281–1298.

[13] *FREAK: Fast retina keypoint*. **Alahi, A., Ortiz, R. and Vandergheynst, P.** Providence, RI, USA : s.n., June 16-21, 2012. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 510–517.

[14] *Computer Vision Approaches to Pedestrian Detection: Visible Spectrum Survey*. **Gerónimo, D., López, A. and A.D., Sappa.** Girona, Spain : s.n.,

June 2007. In Proceedings of the 3rd. Iberian Conference on Pattern Recognition and Image Analysis. pp. 547-554.

[15] *Pedestrian Detection using Adaboost Learning of Features and Vehicle Pitch Estimation*. **Gerónimo, D., et al., et al.** Palma de Mallorca, Spain : s.n., August 2006. In Proceedings of the Int. Conf. on Visualization, Imaging, and Image Processing.

[16] *Real-time corner detection algorithm for motion estimation*. **Wang, H. and Brady, M.** s.l. : Image and Vision Computing 13 (9), 1995.

[17] *Gray level corner detection*. **Kitchen, L. and Rosenfeld, A.** s.l. : Pattern Recognition Letters, 1982, Vols. 1:95–102.

[18] *Rotational Invariant Image Operators*. **Beaudet, P.R.** s.l. : In Proceedings of the 4th International Conference on Pattern Recognition. pages 579{583, 1978}.

[19] *Using Canny's criteria to derive a recursively implemented optimal edge detector*. **Deriche, R.** s.l. : Int. J., April 1987, Vols. Vol. 1, pp. 167–187.

[20] *Towards Automatic Visual Obstacle*. **Moravec, H. P.** s.l. : Proc. 5th International Joint Conference, 1977. pp. 584.

[21] *A Combined Corner* . **Harris, C. and Stephens, M.** s.l. : Proceedings of 4th AlveyVision Conference, 1988.

- [22] *SUSAN: Smallest Univalve Segment Assimilating Nucleus*. **S.M.Smith**.  
“SUSAN -A new Approach to Low Level, s.l. : Journal of Computer, 1997,  
Vols. pp.45-78.
- [23] **Johnson., A.** *Spin-Images: A representation for 3-D surface matching*.  
s.l. : Ph.D Thesis, The Robotics Institute, 1997.