



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERIA ELECTRÓNICA

**SISTEMA DE GENERACIÓN DE MOSAICOS 2D
PARA ROBOTS MÓVILES A PARTIR DE VIDEO
MONOCULAR**

Por:

Victor Yovanni Garcia Carmona

Realizado con la asesoría de:

José de la Cruz Cappelletto Fuentes

PROYECTO DE GRADO

Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero Electrónico

Sartenejas, Marzo de 2018



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA ELECTRÓNICA

ACTA FINAL PROYECTO DE GRADO

**SISTEMA DE GENERACIÓN DE MOSAICOS 2D PARA ROBOTS
MÓVILES A PARTIR DE VIDEO MONOCULAR**

Presentado por:

Victor Yovanni Garcia Carmona

Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

José de la Cruz Cappelletto Fuentes

Novel Antonio Certad H.

Gerardo Fernandez López

Sartenejas, @día de Mayo de 2018



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA ELECTRÓNICA
**SISTEMA DE GENERACIÓN DE MOSAICOS 2D PARA ROBOTS
MÓVILES A PARTIR DE VIDEO MONOCULAR**
PROYECTO DE GRADO
PRESENTADO POR:
Victor Yovanni Garcia Carmona, Carnet: 12-10738

RESUMEN

Al realizar tareas de exploración para el análisis del suelo desde espacios aéreos o del fondo marino, es muy común emplear sistemas de adquisición basados en captura de videos para su posterior análisis. En la actualidad, el incremento de la tecnología sobre el procesamiento de datos, ha permitido que los algoritmos de visión por computadora coloquen a la cámara como principal sensor para la reconstrucción de entornos recorridos por vehículos móviles. El presente trabajo se encuentra enfocado al análisis y la implementación de distintos algoritmos para la reconstrucción de un mosaico 2D (dos dimensiones), a partir de la información proveniente de una cámara monocular ubicada en la parte inferior de un robot. El robot en cuestión puede realizar recorridos aéreos para realizar la adquisición del video, o incluso trayectorias más desafiantes como serían las aplicaciones subacuáticas. Para esto, se implementarán distintos algoritmos usando técnicas de procesamiento de imágenes y visión por computadora, para la elaboración de un sistema automatizado que permita generar un mapa en dos dimensiones de la trayectoria recorrida, con la menor distorsión posible, mejorando la detección de puntos clave, y optimizando el cálculo de las matrices de transformación para la alineación de imágenes en el mosaico, además de realizar análisis sobre el error de reproyección de dichas imágenes en el mapa del suelo generado.

Palabras clave: mosaico, video monocular, puntos clave, matriz de transformación.

Agradecimientos

Índice general

Resumen	I
Agradecimientos	I
Índice de Figuras	IV
Lista de Tablas	V
Acrónimos y Símbolos	VI
1. Introduccion	1
1.1. Antecedentes	1
1.2. Justificación y planteamiento del problema	3
1.3. Objetivos	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Estructura del trabajo	4
2. Deteccion de puntos de interes	5
2.1. Introduccion	5
2.2. Detectores de características	5
2.3. Descriptores de características	13
2.4. Emparejadores	13
2.5. Distribucion espacial	13
2.6. Resumen	13
3. Alineacion de imagenes	14
3.1. Introduccion	14
3.2. Transformaciones geometricas	15
3.3. Generacion de sub-mosaicos	16
3.3.1. Selecccion de imagen de referencia	16
3.3.2. Matriz de homografia promedio	16
3.4. Coreccion euclideana	16

3.5. Resumen	16
3.6. Algoritmos (temporal)	18
4. Union de imagenes	19
4.1. Introduccion	19
4.2. Linea de costura	19
4.2.1. Mapa mas cercano	19
4.2.2. Corte por grafo	19
4.3. Correccion de color	20
4.3.1. Ajuste de ganancia	20
4.3.2. Metodo de Reinhard	20
4.4. Fusion de imagenes	20
4.4.1. Fusion ponderada	20
4.4.2. Fusion piramidal	20
5. Resultados Experimentales	21
5.1. Comparativa de detectores	21
5.2. Correcciones de distorsion	21
5.2.1. Homografia promedio	21
5.2.2. Correccion euclideana	21
5.3. Linea de corte	22
5.4. Fusion de imagenes	22
6. Conclusiones y trabajos futuros	23
7. Conclusiones	24
A. @nombreApendice	27
A.1. @sección	27
A.1.1. @subsección	27
B. @nombreApendice	28

Índice de figuras

1.1. Logo del software Hugin	2
2.1. Puntos de interés	6
2.2. Deteccion de esquinas	7
2.3. Deteccion de esquinas	7
2.4. SIFT - Espacio de escalas	8
2.5. SURF - Espacio de escalas	10
2.6. ORB - Espacio de escalas	11
2.7. KAZE - Espacio de escalas	12
3.1. Paralaje - Efectos en el cambio de punto de vista	15

Índice de Tablas

Acrónimos y Símbolos

Dedicatoria

A @personasImportantes, por @razonesDedicatoria.

Capítulo 1

Introduccion

La navegación y exploración en áreas de difícil acceso mediante el uso de robots, es una tarea que se ha venido desarrollando en el Grupo de Investigación y Desarrollo en Mecatrónica de la USB (*GIDM*) desde hace mu tiempo. Donde una de las aplicaciones mas demandantes, es la tarea de reconstruir un mapa 2D de la superficie mapeada por los robots utilizados. En el presente capitulo se pretende introducir los trabajos previos y avances que se han tenido en el desarrollo de este tipo aplicaciones, tanto en el *GIDM* como a nivel mundial, y que dieron origen y motivación para la realización del proyecto. Además, de postular un serie de problemas que el presente trabajo busca solucionar.

1.1. Antecedentes

Cuando hablamos de construir un mosaico 2D, se hace referencia al proceso de recortar y alinear imágenes, de tal forma que puedan ser representadas todas juntas en una sola gran imagen. Partiendo de que estas imágenes sean tomadas desde diferentes ubicaciones de la cámara, a diferencia del proceso para elaborar imágenes panorámicas, en las cuales esta ubicación es una constante.

La elaboración de mosaicos para la construcción de mapas del suelo, se ha desarrollado incluso antes desde la era digital de la computadoras. Desde que el proceso de registrar fotografías ha existido, se comenzaron a usar para elaborar

mapas topográficos [1], donde imágenes adquiridas a partir de globos aerostáticos o altas colinas eran unidas manualmente. Posteriormente, producto de los avances en la creación de aviones, el interés por la aerofotografía se incrementó en gran medida. Pero debido a que no se alcanzaban grandes altitudes, y a la continua necesidad de mapear grandes áreas, era necesario que los mapas se construyeran mediante fotografías que se superpongan, de igual forma esta tarea se llevaba a cabo mediante técnicas manuales por medio de expertos.

La necesidad de registrar áreas aun mas grandes siguió avanzando, motivado por la llegada de los satélites que eran capaces de enviar a tierra la información que obtenían de las cámaras. Los avances tecnológicos en materia de computación, y el creciente aumento de datos por para esta aplicación, promovieron el desarrollo de técnicas de procesamiento de imágenes para dar solución a este tipo de problemas.

Del mismo modo, distintos centros de investigación en el área de la física, robótica y visión por computadora, han aplicado sus esfuerzos al algoritmos para la realización de estos mapas en ambientes mas desafiantes como lo son el fondo marino [2, 3, 4, 5]. Cuando se opera en este tipo de ambientes, en busca de realizar exploraciones mas eficientes y a mayor escala, se emplean vehículos operados remotamente *ROV* (del inglés: Remotely Operated Vehicles) o vehículos autónomos submarinos *AUV* (del inglés: Automated Underwater Vehicles).

En el *GIDM* se han realizado grandes avances en el desarrollo de equipos y plataformas robóticas, específicamente para su uso en aplicaciones submarinas. En este sentido, se cuenta con un vehiculo sumbarino llamado OpenROV¹

En el *GIDM* actualmente se utilizan mecanismos manuales para la elaboración de estos mapas, en específico, se hace uso de *softwares* como Hugin². Este es un programa de código abierto y gratuito bajo licencia GPL³, el cual esta dedicado a la generación de imágenes panorámicas, incluyendo funciones para el recorte, alineación, corrección de color; además de algoritmos para



FIGURA 1.1: Logo del software Hugin

¹hugin.sourceforge.net/

²hugin.sourceforge.net/

³<http://www.gnu.org/copyleft/gpl.html>

la optimización de parámetros en la cámara, y corrección de distorsión. Si bien este software esta diseñado para la creación de imágenes panorámicas, permite el uso de varios tipos de proyecciones cartográficas, entre estas la rectangular, proyectando las imágenes sobre un plano recto.

1.2. Justificación y planteamiento del problema

Para esta tarea, es común el uso de herramientas que requieren de una gran intervención por medio del usuario para la elaboración de un mapa.

Atendiendo a esta necesidad, es necesario contar con un sistema que permita realizar la reconstrucción del suelo con la menor interacción posible del ser humano. Asimismo, con el fin de poder realizar operaciones de mapeo y localización simultanea *SLAM*, haciendo uso de las herramientas y robots existentes en el laboratorio, es necesario contar con un sistema basado en visión, que genere de forma automática un mapa 2D de la superficie sobre la que navega o sobrevuela el vehículo remoto.

1.3. Objetivos

1.3.1. Objetivo General

Analizar e implementar un sistema automatizado que permita la reconstrucción de un mapa en dos dimensiones, del entorno recorrido por robot, aereo o submarino.

1.3.2. Objetivos Específicos

- Análisis comparativo de metodos vigentes en la reconstruccion de mosaicos 2D, a partir de imagenes y videos de entrada.

- Implementacion de modulo de preprocesamiento y correccion de entrada.
- Analisis comparativo de metodos de deteccion y description de puntos clave.
- Implementación de módulo de alineación de imagenes mediante la detección de puntos clave.
- Cuantificar el error de reproyeccion y distorsion en los modelos 2D generados.

1.4. Estructura del trabajo

Luego de presentar el planteamiento del problema y la descripción del proyecto, la presente investigación se encuentra dividida en 5 capítulos, organizados de la siguiente manera:

En el **Capítulo 2** se presenta una revisión del estado del arte, en el cual se exponen los distintos métodos para la detección e identificación de puntos de interés en una imagen. Además, se presentan diversos algoritmos utilizados en la actualidad para la alineación de imágenes en un mosaico. Finalmente, se muestran las técnicas que se utilizan para la fusión entre las imágenes, que logran reducir errores en el color, y la calidad del mapa final.

El **Capítulo 3** describe cada cada uno de los módulos necesarios en la construcción del mosaico, así como también una descripción detallada de los algoritmos implementados.

Los resultados experimentales son mostrados en el **Capítulo 5**, en este se contemplan los modelos generados de forma automática del suelo, tanto terrestre como submarino. Además se presenta un análisis comparativo de estos resultados sobre los que proveen herramientas no automatizadas.

Finalmente, en el **Capítulo 6** se presentan las conclusiones que se derivaron del proyecto, por otra parte, se plantean trabajos futuros que podrían dar pie a posibles mejoras en la linea de investigación del presente trabajo.

Capítulo 2

Deteccion de puntos de interes

2.1. Introduccion

En este capitulo se presenta una revisión teórica del estado actual de las aplicaciones e investigaciones que se han desarrollado en el área de procesamiento de imágenes, aplicado a la construcción de mosaicos. Además se presenta una reseña histórica de la evolución de dichos métodos. Con esto se pretende recuperar y trascender el conocimiento acumulado en esta área de estudio, además de familiarizar al lector con los conceptos básicos, necesarios para la comprensión del presente trabajo.

En primer lugar, se reseña el progreso de los algoritmos para la extracción de puntos de interés, seguidamente se presentan avances sobre los módulos de alineación de imágenes. Finalmente, en la ultima sección se describen los métodos mas utilizados en la actualidad para la etapa final del mapeo, como lo son la búsqueda de la mejor linea de corte y corrección final de color.

2.2. Detectores de características

Antes mencionar la evolución de los algoritmos de detección de puntos de interés, primero es necesario definir que son estos. Los puntos de interés, puntos

clave, o "*features*" (en español: características) como son comúnmente llamados, son regiones en una imagen que contienen patrones específicos, lo que hace que puedan ser fácilmente seguidos o ubicados en otra imagen.



FIGURA 2.1: Ejemplo de caracterización de puntos en una imagen

Atendiendo a la imagen 2.1, se puede observar que se caracterizan seis áreas de interés. Analizando estos segmentos, vemos que **A** y **B** corresponden con superficies planas, lo que hace que sea muy difícil identificar la ubicación exacta de estas superficies en la imagen original. Por otro lado, tenemos las regiones **C** y **D**, las cuales corresponden con bordes en la imagen, si bien, se puede limitar en gran medida el área de búsqueda hacia toda las regiones del mismo bordes, sigue siendo difícil encontrar la ubicación exacta. Por ultimo, analizando las regiones **E** y **F** tenemos que corresponden a esquinas de la imagen original, en este caso se puede identificar fácilmente la ubicación exacta de la región en la imagen.

A partir de esta idea, en la cual se consideran las esquinas como regiones fácilmente identificables en una imagen, en 1988 nace el primer algoritmo de detección de puntos de interés llamado Detector de esquinas de Harris [6] (nombre original en inglés: Harris Corner Detector), y como su nombre lo indica está basado en la detección de esquinas.

Retomando el concepto planteado previamente, este detector busca la diferencia de intensidad en una ventana de exploración cuando es desplazada en diferentes direcciones. Es decir, se detectará una esquina, para aquellas regiones que presenten

una alta variación de intensidad al desplazar la ventana estudiada en cualquier dirección. En la figura 2.2 se puede apreciar visualmente como funciona esta ventana de búsqueda.

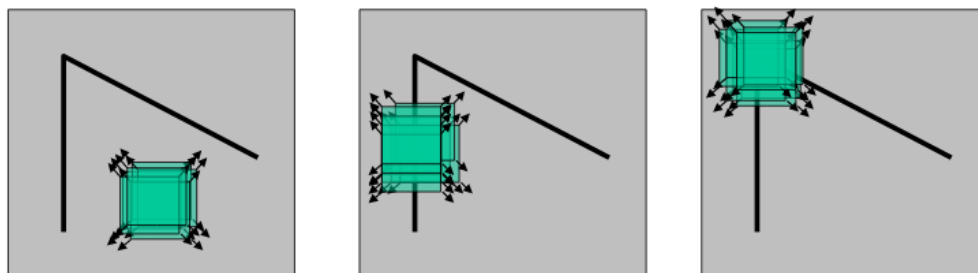


FIGURA 2.2: Desplazamiento de la ventana de búsqueda, para de detección de esquinas

Cuando se trabajan con detectores de características, se desea que estos sean invariantes ante la mayor cantidad de variables posibles, es decir, que sean capaces de detectar las mismas características en una imagen a pesar de cambios en la traslación, rotación, escala, variaciones de iluminación, variaciones del punto de vista, e incluso deformaciones debido a transformaciones geométricas. Si bien el detector presentado anteriormente es invariante ante la traslación y la rotación (ya que las esquinas se mantienen como esquinas si son rotadas o desplazadas), no funciona de la misma forma ante cambios de escala. Como se observa en la figura 2.3, una región considerada como esquina, se podría considerar plana si es ampliada.

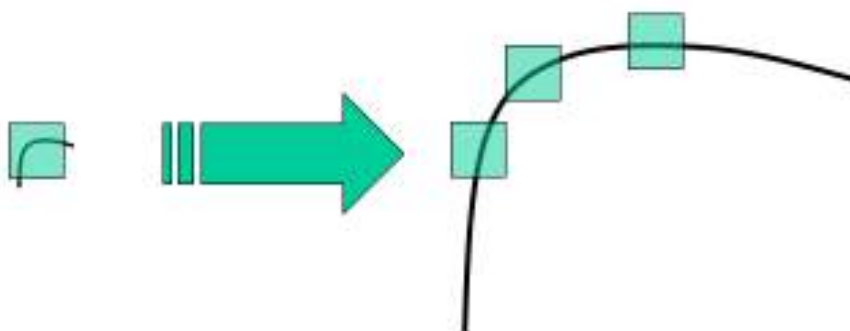


FIGURA 2.3: Desplazamiento de la ventana de búsqueda, para de detección de esquinas

Otra característica que se busca en estos algoritmos para lograr ubicar estas características ya detectada en una imagen distinta, es la que tengan la capacidad de realizar una descripción de estas regiones. Para esto, se desarrollaron los algoritmos descriptores, los cuales una vez tengan la ubicación de los puntos característicos se encargan de convertir la información de su alrededor en una serie de números, o un vector que permita diferenciar un punto clave de otro. Esta información también es necesaria que sea invariante ante las variables mencionadas previamente, para lograr una identificación eficiente del mismo punto en distintas imágenes.

Partiendo de estos problemas, en 2004 *D. Lowe* crea el detector y descriptor *SIFT* [7] (del inglés: Scale Invariant Feature Transform), el cual presenta la innovación de poseer invarianza ante cambios en la escala. El proceso para la detección y descripción de puntos de interés de este algoritmo, consta de cuatro pasos principales:

En primer lugar, realiza una detección de máximos en el espacio de la escala aplicando la diferencia gaussiana *DoG* (del inglés: Difference of Gaussians). Para esto, se aplica el filtro gaussiano con distintos tamaños de media (se tienen distintas escalas), luego restando estas imágenes para distintos pares de escalas se logra la diferencia de gaussianas. Posteriormente se buscan los máximos locales a lo largo del espacio (coordenadas X,Y) para cada correspondiente escala. Este proceso de detección se puede visualizar en la figura 2.4.

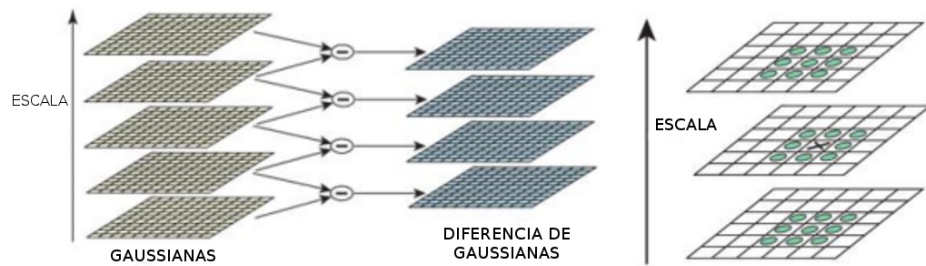


FIGURA 2.4: Detección de máximos en el espacio de la escala

En segundo lugar para la localización de puntos de interés, se descartan los puntos encontrados en el paso anterior que no superen cierto valor de umbral, es decir, que no estén lo suficientemente contrastados con su entorno. Con esta etapa el algoritmo solo toma en cuenta los puntos claves mas fuertes por cada escala. Además, con el objetivo de eliminar los bordes suficientemente contrastados que

no correspondan con esquinas, el algoritmo usa una matriz hessiana para calcular las curvaturas principales, y así quedarse solo con esquinas.

Para garantizar la invarianza con respecto a la rotación, se toman los píxeles vecinos al punto clave y se calcula la magnitud y dirección del gradiente en esa región. Con esto se hace un histograma de la magnitud del gradiente en cada dirección, donde el pico mayor del histograma indica la orientación. En el caso que exista un pico mayor al 80 % del pico principal, este se utiliza para crear otro punto de interés en la misma posición pero con la distinta rotación.

Finalmente para la descripción por cada punto clave se crea una matriz de 16x16 alrededor de este, dividida en 4 subregiones de 4x4 píxeles con un histograma de orientaciones para cada uno. Seguidamente, el descriptor del punto será el vector con los valores de los histogramas de las regiones 4x4 concatenados.

En el año 2006, un grupo de tres personas Bay, H., Tuytelaars, T. and Van Gool, L. desarrollan *SURF* [8], el cual es un detector y descriptor de características basado en SIFT, pero con modificaciones que aumentan su velocidad de detección. Si bien, sacrifica un poco de rendimiento y precisión, lo hace mas provechoso para aplicaciones embebidas que demanden mayor velocidad de computo y menor uso de recursos, como por ejemplo *SLAM* (del ingles: Simultaneous Localization and Mapping). El proceso para la extracción de características por parte de este algoritmo se compone de los siguientes pasos:

Como primer paso, en lugar de aproximar el laplaciano de Gauss *LoG* (del inglés: Laplacian of Gaussians) con la diferencia de Gaussianas (DoG) como lo hace SIFT, este algoritmo aproxima LoG con cuadrados para promediar la imagen. La ventaja de aplicar filtros con cuadrados es que con la ayuda de imágenes integrales el cálculo computacional se reduce en gran medida.

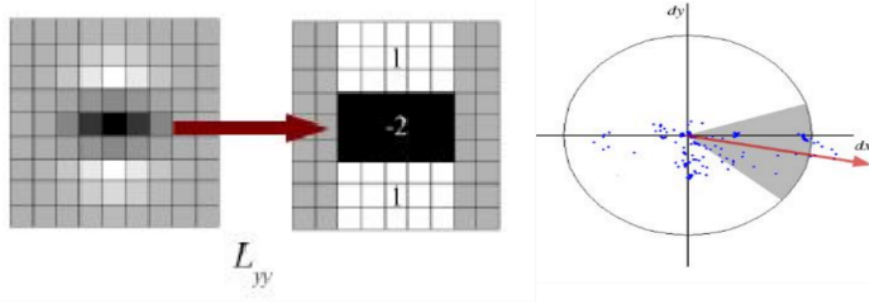


FIGURA 2.5: Detección de máximos en el espacio de la escala

En función de identificar la orientación, el algoritmo utiliza la respuesta wavelet Haar en horizontal y vertical en un vecindario de $6s$ (donde s es la escala evaluada) píxeles al rededor del punto de interés, Luego estas respuestas son representadas como puntos en el espacio, para luego calcular la orientación dominante con la suma de todos los resultados dentro de una ventana deslizante de apertura 60° . En la figura 2.5 se puede visualizar la forma del filtro que aplica el detector, y el vector de orientación en función a la distribución de puntos estudiados.

El siguiente avance importante en los algoritmos de detección aparece en el año 2011 con *ORB* [9] (del inglés: Oriented FAST and Rotated BRIEF), este utiliza una combinación del detector FAST (del inglés: Features from Accelerated Segment Test) y del descriptor BRIEF (del inglés: Binary Robust Independent Elementary Features), este nuevo algoritmo esta caracterizado por su alta velocidad de procesamiento manteniendo un buen rendimiento, gracias al uso de un descriptor binario.

Como se mencionó utiliza el algoritmo FAST el cual consiste en encontrar esquinas evaluando los píxeles en un perímetro circular, de esta forma, un punto será detectado como esquina si la cantidad de píxeles de color opuesto al evaluado, supera cierto valor de umbral (ver izquierda en la figura 2.6), posteriormente con el fin de aumentar la robustez, es aplicado el algoritmo de clasificación de esquinas de *Harris*. De igual forma se realiza con una estructura piramidal evaluando varias escalas (al igual que SIFT).

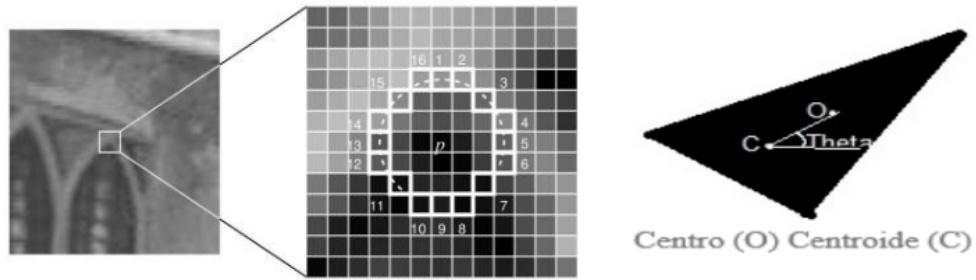


FIGURA 2.6: Detección de máximos en el espacio de la escala

Como el algoritmo FAST no toma en cuenta la orientación, en el ORB se modificó para que calculara la orientación de la siguiente forma: Se considera una región ubicada en el centro del punto estudiado, luego se calcula el centroide de la región en función a la intensidad de los puntos. De esta forma, la dirección del vector desde el punto central hasta el centroide es asignado como vector de orientación. Observando a la derecha en 2.6 se aprecia un ejemplo del lugar del centroide (C) y del centro (O) para una región en particular.

Para el descriptor utiliza BRIEF, a diferencia de los anteriores (SIFT y SURF) este es un descriptor binario y no vectorial. El descriptor BRIEF produce una palabra de n -bits usando el algoritmo *Local Binary Tests* (LBT), el problema de esta representación es que no es muy robusta ante cambios en la rotación. Para resolver esto ORB utiliza la información de la orientación previamente calculada en el paso de detección para aplicar LBT en esa orientación.

Los algoritmos de detección que se mencionaron hasta este momento tienen una característica en común, y es que cuando trabajan con el esquema piramidal lo hacen bajo el espacio de escala Gaussiano, el cual es una instancia particular de difusión lineal. De esta forma, al utilizar este filtro no se respetan los límites naturales de los objetos y se difumina del mismo nivel toda la región de la imagen cuando se avanza entre niveles de escala.

Enfocándose en esta característica, en el año de 2012 se desarrolla el detector y descriptor llamado KAZE [10] por parte de *Pablo Fernández Alcantarilla*. Este novedoso algoritmo opera completamente en un espacio de escala no lineal, para ello utilizan un esquema de división de operadores aditivos (AOS, del inglés:

Additive Operator Splitting), que les permite obtener espacios de escala no lineales de forma eficiente. De este modo se puede realizar un difuminado localmente adaptativo, posibilitando que se remueva el ruido en las imágenes, manteniendo información importante sobre los bordes de los objetos al avanzar en el espacio de escala. En la figura 2.7 se puede observar como afecta en los bordes de los objetos el aplicar un filtro de difusión lineal, y uno que no lo es, bajo el esquema propuesto por este algoritmo.

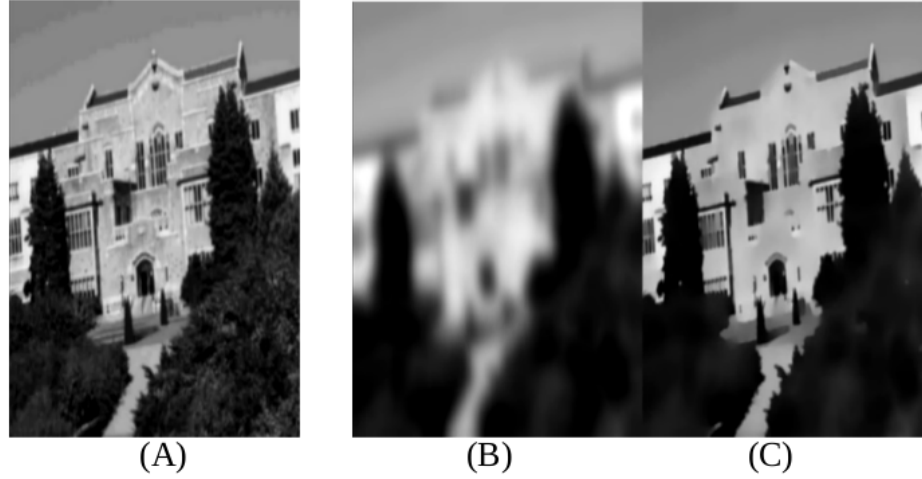


FIGURA 2.7: Detección de máximos en el espacio de la escala

Bajo este mismo esquema de difusión no lineal, el mismo autor en el año 2013 desarrolla la versión acelerada de este algoritmo que recibe el nombre de *A-KAZE* [11] (del inglés: Accelerated KAZE). Esta mejora se utiliza un esquema basado en difusión explícita rápida *FED* (del inglés: Fast Explicit Diffusion) en lugar de *AOS*, el cual es un nuevo esquema piramidal que incrementa en gran medida la velocidad de cómputo para construir el espacio de escala no lineal.

Para el cálculo de la orientación el primer algoritmo *KAZE* utiliza un descriptor para la orientación similar al que emplea SURF. Este encuentra la orientación dominante en un área circular de radio $6s$ (s corresponde con la escala), y para cada muestra del círculo se calcula la derivada de primer orden en las direcciones X e Y , y se ponderan con una gaussiana centrada en el punto de interés. Luego, las respuestas de estas derivadas son representadas como puntos en un espacio vectorial, donde la orientación dominante se haya sumando las respuestas dentro de un segmento de círculo deslizante con apertura de 60° .

Por otro lado, la versión acelerada *A-KAZE* emplea un descriptor basado en una versión modificada del algoritmo de diferencia local binaria *LDB* [12] (del ingles: Local Difference Binary), llamado M-LBD (del ingles: Modified Local Difference Binary), el cual aprovecha al máximo la información del espacio de escala no lineal. La modificación consiste en hacer un sub-muestreo de cada región que divide la zona del descriptor, en lugar de calcular el promedio de todos los píxeles de la región, es decir, se tienen muestras de cada subdivisión para distintas escalas.

2.3. Descriptores de características

Descriptores

2.4. Emparejadores

Emparejadores

2.5. Distribucion espacial

Distribucion espacial

2.6. Resumen

Resumen

Capítulo 3

Alineacion de imagenes

3.1. Introduccion

Introduccion

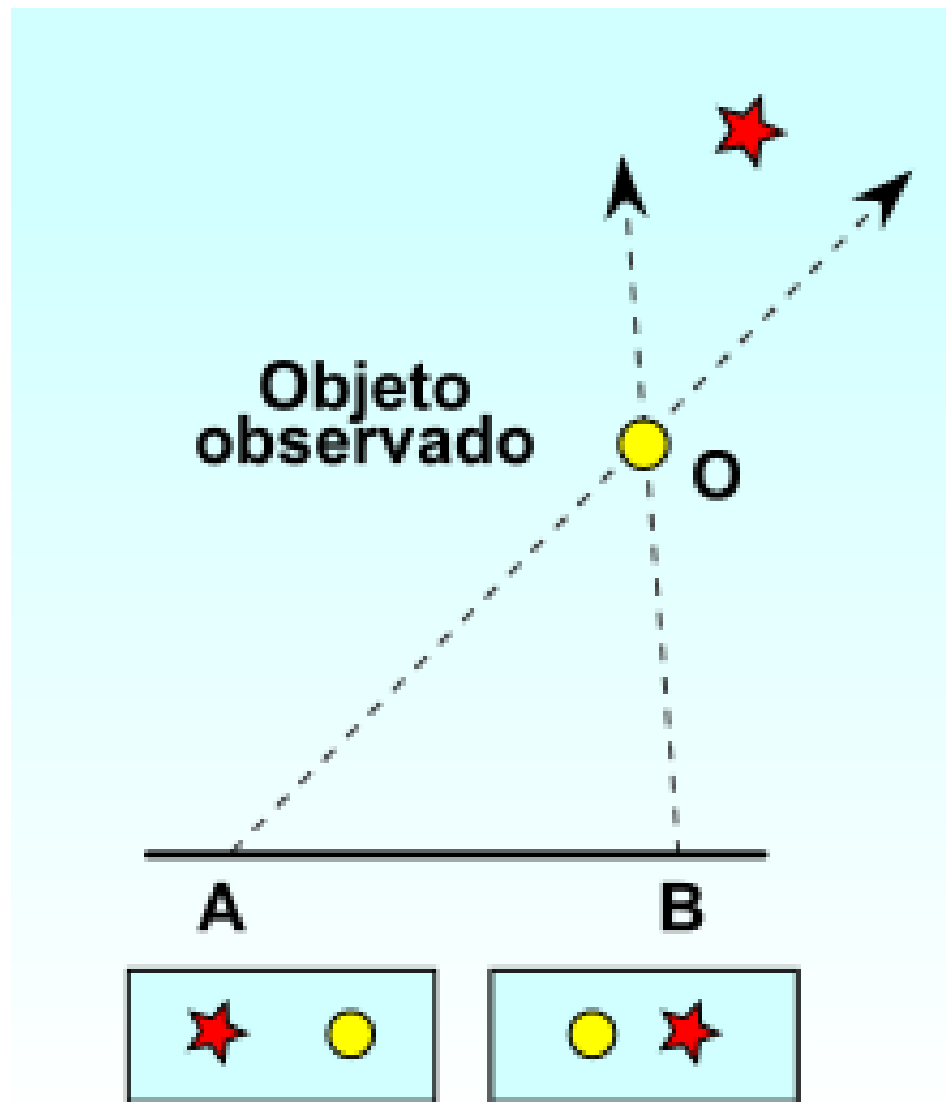


FIGURA 3.1: Efectos en el cambio del punto de vista.

Desde el punto de vista **A**, se observa la estrella a la izquierda del círculo. Mientras que desde el punto **B** la estrella es observada a la derecha del círculo.

3.2. Transformaciones geometricas

Prueba

3.3. Generacion de sub-mosaicos

Prueba

3.3.1. Seleccin de imagen de referencia

Prueba

3.3.2. Matriz de homografia promedio

Prueba

3.4. Coreccion euclideana

Prueba

3.5. Resumen

Resumen

3.6. Algoritmos (temporal)

Algoritmo 1: Calculo de matriz de homografia promedio

```

1 mientras no se alcanza el maximo de iteraciones hacer
2   seleccionar 4 puntos aleatorios del primer sub-mosaico;
3   seleccionar los 4 puntos correspondientes en el segundo sub-mosaico;
4   calcular el punto medio para cada par de puntos correspondientes;
5   calcular la transformacion desde los puntos del primer sub-mosaico hasta los
   puntos medios;
6   aplicar transformacion en el primer sub-mosaico;
7   calcular error de distorsión en el primer sub-mosaico;
8   si el error es menor que el mas bajo obtenido entonces
9     guardar el error como el mas bajo;
10    guardar la matriz de transformación como la mejor;
11  en otro caso
12    restaurar valores del primer sub-mosaico;
13  fin
14 fin

```

Algoritmo 2: Encontrar matriz de transformacion

```

15  $I_{i+1} \equiv$  Imagen nueva;
16  $I_i \equiv$  ultima imagen anadida al mosaico;
17  $V_i \equiv$  vecinos de  $I_i$ ;
18 mientras puntos emparejados  $\geq 4$  hacer
19   Emparejar puntos de  $I_{i+1}$  con  $I_i$ ;
20   si  $I_i$  tiene vecinos entonces
21     para cada vecinos de  $I_i$  hacer
22       Emparejar puntos de  $I_{i+1}$  con  $V_i$ ;
23     fin
24   fin
25   descartar malos emparejamientos; aplicar busqueda sectorizada; si puntos
   totales emparejados  $\leq 3$  entonces
26     modificar criterio para descartar;
27     si criterio para descartar llega al minimo entonces
28       terminar ; // no es posible emparejar imagen
29     fin
30   fin
31 fin

```

Capítulo 4

Union de imagenes

4.1. Introduccion

Prueba

4.2. Linea de costura

Prueba

4.2.1. Mapa mas cercano

Prueba

4.2.2. Corte por grafo

Prueba

4.3. Correccion de color

Prueba

4.3.1. Ajuste de ganancia

Prueba

4.3.2. Metodo de Reinhard

Prueba

4.4. Fusion de imagenes

4.4.1. Fusion ponderada

Prueba

4.4.2. Fusion piramidal

Prueba

Capítulo 5

Resultados Experimentales

Resumen del capitulo 5

5.1. Comparativa de detectores

Prueba

5.2. Correcciones de distorsion

Prueba

5.2.1. Homografia promedio

Prueba

5.2.2. Correccion euclideana

Prueba

5.3. Linea de corte

Prueba

5.4. Fusion de imagenes

Prueba

Capítulo 6

Conclusiones y trabajos futuros

Conclusiones

Capítulo 7

Conclusiones

Mensaje de prueba para conclusiones

Bibliografía

- [1] Paul R. Wolf. *Elements of Photogrammetry (Second Edition)*. McGraw-Hill Higher Education, 1983.
- [2] N. Gracias y J. Santos-Victor. Underwater mosaicing and trajectory reconstruction using global alignment. *MTS/IEEE Conference and Exhibition*, 2001.
- [3] H Pizarro, O. y Singh. Seabed video mosaicking with matisse: a technical overview and cruise results. *International Offshore and Polar Engineering Conference*, 2004.
- [4] Pizarro O. Singh-H. y Howland J. Eustice, R. Underwater image toolbox for optical image processing and mosaicking in matlab. *International Symposium on Underwater Technology*, 2002.
- [5] Borgetto M.-Opderbecke J. Pessel N. y Rigaud V Allais, A. G. Toward large-area mosaicing for underwater scientific applications. *IEEE Journal of Oceanic Engineering*, 2003.
- [6] Hans P. Morevec. Towards automatic visual obstacle avoidance. 1977.
- [7] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Computer Science Department, University of British Columbia*, 2004.
- [8] Tuytelaars T. y Van Gool L Bay, H. Surf: Speeded up robust features. *European Conference on Computer Vision*, 2006.
- [9] Kurt Konolige y Gary Bradski Ethan Rublee, Vincent Rabaud. Orb: an efficient alternative to sift or surf. *International Conference on Computer Vision*, 2011.
- [10] y Andrew J. Davison Pablo Fernandez Alcantarilla, Adrien Bartoli. Kaze features. *European Conference on Computer Vision*, 2012.

-
- [11] Pablo Fernández Alcantarilla. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *British Machine Vision Conference (BMVC)*, 2013.
 - [12] Kwang-Ting Cheng Xin Yang. Local difference binary for ultrafast and distinctive feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

Apéndice A

@nombreApendice

A.1. @sección

A.1.1. @subsección

“Saludo”.

Apéndice B

@nombreApendice