



**I
N
A
O
E**

**Estabilización digital de secuencias de
imágenes con precisión subpixel en sistemas
de visión catadióptricos**

Por

Francisco Daniel Sánchez Fernández

Tesis sometida como requisito parcial
para obtener el grado de

**MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE
CIENCIAS COMPUTACIONALES**

en el

**Instituto Nacional de Astrofísica, Óptica y
Electrónica**

Febrero 2014
Tonantzintla, Puebla

Supervisada por:

Dr. Leopoldo Altamirano Robles

© INAOE 2014

Derechos reservados

El autor otorga al INAOE el permiso de reproducir y
distribuir copias de esta tesis en su totalidad o en partes



Resumen

En la actualidad existe un gran número de aplicaciones donde el objetivo primordial es vigilar o rastrear algún objeto o persona en un ambiente en particular. Para obtener un ambiente con condiciones favorables en estos tipos de sistemas en la captura de imágenes, necesitamos por lo menos dos condiciones: la cobertura y un sistema ajeno a vibraciones. La cobertura implica cubrir la mayor área posible con el menor número de cámaras, debido a que mientras mayor sea el número, nos tendremos que enfrentar a una mayor cantidad de dificultades técnicas como es la calibración de cámaras, mayor tiempo de procesamiento y otros problemas generados por un elevado número de cámaras. La otra condición se refiere a eliminar las vibraciones que afectan el sistema de vigilancia, estas pueden ser provocadas por el medio en el que se encuentra nuestro sistema como pueden ser vibraciones ocasionadas por vehículos, ráfagas de aire o algún otro movimiento ajeno al sistema.

Para resolver los problemas expuestos anteriormente, en este trabajo proponemos el uso de un sistema catadióptrico de imágenes, el cual está formado por una cámara convencional, un lente y un espejo, cuya principal característica es que el campo de visión de la cámara es ampliado por el espejo del sistema. La principal desventaja que presentan estos sistemas comparado con los sistemas convencionales, es la deformación que sufre la imagen por el espejo. Sin embargo, dicha deformación puede ser revertida mediante una transformación que represente al sistema catadióptrico. Por lo general, estos sistemas poseen campos de visión de entre el rango de 180° y 360° alrededor de uno de sus ejes.

El objetivo principal de nuestro trabajo de investigación es eliminar las vibraciones que afectan a los sistemas catadióptricos, debido a que afectan el desempeño de las tareas de los sistemas de visión, como pueden ser el seguimiento o búsqueda de algún objeto o persona en particular. En nuestro trabajo utilizamos el algoritmo SURF para rastrear ciertos puntos característicos a lo largo de la secuencia de imágenes, con el objetivo de obtener un patrón del movimiento de las imágenes para realizar la compensación de las vibraciones con las que se ha afectado al sistema de visión. Nuestros resultados experimentales muestran que nuestro algoritmo para lograr la estabilización de imágenes tiene un buen desempeño en diversas condiciones de los videos, como pueden ser cambios de iluminación, objetos en movimiento, etc. El desempeño de nuestro algoritmo es medido mediante la métrica *Peak signal noise to ratio* (PSNR) entre imágenes consecutivas,

esta métrica sirve como medida de similitud entre imágenes, por lo cual una secuencia estabilizada debe de tener un índice mayor de PSNR que una secuencia que no se encuentre estabilizada. El incremento en el PSNR de las secuencias de imágenes estabilizadas fue en promedio de 12% respecto a las secuencias originales. En algunos casos se incrementó hasta 24% dependiendo de la naturaleza de la secuencia.

Abstract

Nowadays, there is number of applications where the primary objective is to monitor or track any object or person in a particular environment. For an environment with favorable conditions in these types of systems in imaging, we need at least two conditions: coverage and a system without vibrations. Coverage involves covering the largest possible area with fewer cameras, because the higher the number of cameras the greater the number of technical difficulties such as camera calibration, longer processing time and other problems generated by a large number of cameras. The other condition relates to the elimination of vibrations that affect the monitoring system, these may be caused by the environment in which our system is, such as vibrations caused by vehicles, blasts of air or some other external movement.

To solve the problems discussed above, in this work we propose using a catadioptric imaging system, which consists of a conventional camera, one lens and one mirror, whose main characteristic is that the field of view of the camera is extended by the mirror system. The main disadvantage of these systems compared to conventional systems is the deformation suffered by the mirror image. However, this deformation can be reversed by a transformation representing the catadioptric system. Usually, these systems have fields of view between the range of 180° and 360° around one of its axes.

The main objective of our research is to eliminate the vibrations that affect catadioptric systems, because they affect the performance of the tasks of vision systems, such as tracking or search for an object or person in particular. In our work we use the SURF algorithm to track certain characteristic points along the image sequence, with the aim of obtaining a pattern of motion of images in order to perform the compensation of vibrations which has affected the vision system. Our experimental results show that our algorithm for image stabilization has good performance in various conditions of the videos, such as lighting changes, moving objects and so on. The performance of our algorithm is measured by the Peak Signal Noise to Ratio (PSNR) between consecutive images, this metric is a measure of similarity between images, whereby a stabilized sequence must have a higher index of PSNR than a sequence that is not stabilized. The increase in the PSNR from the sequences of stabilized images averaged 12% over the original sequences. In some cases increased to 24% depending on the nature of the sequence.

Índice general

| | |
|---|------|
| Resumen..... | ii |
| Abstract..... | iv |
| Índice general..... | v |
| Índice de figuras..... | viii |
| Índice de tablas | xi |
| 1 Introducción | 1 |
| 1.1 Motivación | 1 |
| 1.2 Objetivo General | 3 |
| 1.3 Objetivos específicos | 3 |
| 1.4 Solución Propuesta..... | 3 |
| 1.5 Contribuciones | 4 |
| 1.6 Organización del documento | 4 |
| 2 Fundamentos Teóricos | 5 |
| 2.1 Sistema catadióptrico | 5 |
| 2.2 Modelo del sistema catadióptrico | 7 |
| 2.2.1 Sistemas no centrales | 7 |
| 2.2.2 Sistemas centrales | 7 |
| 2.3 Calibración..... | 8 |
| 2.4 Descriptores de puntos característicos | 10 |
| 2.4.1 <i>Scale Invariant Feature Transform (SIFT)</i> | 10 |
| 2.4.2 <i>Speeded Up Robust Features (SURF)</i> | 13 |

| | | |
|-------|--|----|
| 2.5 | Correspondencia entre puntos característicos | 17 |
| 2.6 | Transformaciones de imágenes | 17 |
| 2.6.1 | Transformaciones geométricas | 17 |
| 2.7 | Medidas de estabilización de imágenes | 19 |
| 2.8 | Regresión polinomial | 21 |
| 2.9 | Resumen..... | 22 |
| 3 | Trabajo relacionado | 24 |
| 3.1 | Métodos para la reconstrucción de imágenes catadiópticas. | 24 |
| 3.1.1 | Pano-mapping Table (PT)..... | 25 |
| 3.2 | Estabilización de imágenes | 29 |
| 3.3 | Métodos de estabilización directos | 30 |
| 3.4 | Métodos de estabilización basados en características. | 34 |
| 3.5 | Resumen..... | 36 |
| 4 | Método Propuesto | 37 |
| 4.1 | Descripción general del método..... | 37 |
| 4.2 | Rastreo de puntos | 38 |
| 4.3 | Manejo de vectores de movimientos ajenos al movimiento del sistema.. | 40 |
| 4.4 | Precisión subpixel | 42 |
| 4.5 | Compensación Rotacional..... | 44 |
| 4.6 | Transformación de la imagen..... | 45 |
| 4.7 | Compensación Traslacional | 46 |
| 4.8 | Reducción de la imagen | 47 |
| 4.9 | Resumen..... | 48 |
| 5 | Experimentos y resultados | 49 |
| 5.1 | Reconstrucción de la imagen | 49 |
| 5.2 | Selección de Parámetros | 50 |
| 5.3 | Estabilización de la imagen..... | 52 |
| 5.4 | Desempeño contra algoritmos para sistemas convencionales..... | 56 |
| 5.5 | Discusión de resultados..... | 58 |

| | | |
|-----|------------------------------------|----|
| 5.6 | Resumen..... | 59 |
| 6 | Conclusiones y Trabajo futuro..... | 61 |
| 6.1 | Contribución | 61 |
| 6.2 | Conclusiones | 62 |
| 6.3 | Trabajo futuro | 63 |
| 7 | Bibliografía | 65 |

Índice de figuras

| | |
|---|----|
| Figura 1.1 Sistemas de imágenes con sus respectivos campos de visión a) convencional, b) rotacional, c) <i>fish-eye</i> y d) catadióptrico | 1 |
| Figura 1.2 a) Imagen omnidireccional, b) Imagen panorámica reconstruida..... | 2 |
| Figura 1.3 Método de estabilización digital, la a) imagen de referencia es utilizada para alinear la b) imagen actual y obtener una c) imagen compensada. | 2 |
| Figura 2.1 Imagen tomada mediante un lente <i>fish-eye</i> , mostrando la distorsión que genera el lente. | 6 |
| Figura 2.2 Transformación de una imagen omnidireccional a una imagen panorámica y muestra de la pérdida de resolución. | 6 |
| Figura 2.3. Sistema catadióptrico no central y sus diversos puntos focales..... | 7 |
| Figura 2.4. Sistema catadióptrico central y su punto focal. | 7 |
| Figura 2.5. Configuración del sistema catadióptrico utilizado en este trabajo de investigación. | 8 |
| Figura 2.6. Sistema coordinado del sistema catadióptrico, b) plano del sensor y c) plano de la imagen. | 9 |
| Figura 2.7. Reproyección de los patrones de calibración a sus respectivas coordenadas en el mundo real. | 10 |
| Figura 2.8. Diferencias de gaussianas usadas para la construcción del descriptor SIFT. | 11 |
| Figura 2.9. Selección de puntos extremo en el algoritmo SIFT. | 11 |
| Figura 2.10. Histograma de orientaciones de un punto de interés en el algoritmo SIFT | 13 |
| Figura 2.11. Filtro gaussiano de segundo orden de la derivada parcial en: a) dirección y, b) dirección xy, aproximación del filtro gaussiano usando filtros de caja en c) dirección y, d) dirección xy..... | 15 |
| Figura 2.12. Filtros de Haar utilizados en el descriptor SURF, el color negro indica un valor de -1 y el color blanco un valor de 1. a) Filtro en x, b) filtro en y..... | 16 |
| Figura 2.13. Formación del descriptor SURF mediante 4 subregiones y sus respectivas respuestas a los filtros Haar..... | 16 |

| | |
|---|----|
| Figura 3.1. Transformación de a) una imagen omnidireccional a b) una imagen panorámica..... | 24 |
| Figura 3.2. Configuración del sistema catadióptrico para la transformación de imágenes, se observa las diferentes medidas necesarias para la creación de la tabla panorámica..... | 25 |
| Figura 3.3. Algoritmo de estimación de movimiento basado en bloques a) imagen de referencia, b) imagen actual y su respectivo vector generado por la búsqueda del bloque..... | 31 |
| Figura 3.4. Caso ideal de la función delta centrada en desplazamiento del movimiento. | 32 |
| Figura 3.5. Función delta del movimiento de una imagen, mostrando el desplazamiento de la imagen y el ruido que contiene. Las coordenadas del pico más alto son el desplazamiento que sufre la imagen. | 33 |
| Figura 3.6. Bloque de búsqueda para los algoritmos directos de estabilización, en la imagen actual y su parámetro p..... | 33 |
| Figura 3.7. Zonas utilizadas en la imagen utilizadas en[39] para realizar la estabilización de imágenes..... | 35 |
| Figura 4.1. Esquema del algoritmo propuesto para la estabilización de 2 imágenes consecutivas en sistemas catadióptricos. | 38 |
| Figura 4.2. Tablas de puntos rastreados entre dos imágenes consecutivas y sus respectivas coordenadas..... | 39 |
| Figura 4.3. Concatenación y eliminación de puntos característicos en el historial de puntos..... | 39 |
| Figura 4.4. Rastreo de puntos SURF entre dos imágenes consecutivas..... | 40 |
| Figura 4.5. Ejemplo de creación de historial de movimiento de una secuencia de imágenes. | 41 |
| Figura 4.6. Truncamiento de histograma de movimiento para la eliminación de valores ajenos al modelo de las vibraciones. | 42 |
| Figura 4.7. Segmento de una imagen para ejemplificar a) resolución original de la imagen, b) resolución subpixel $\frac{1}{2}$ | 43 |
| Figura 4.8. Segmento de una imagen para ejemplificar a) resolución subpixel $\frac{1}{4}$ y b) resolución subpixel $\frac{1}{8}$ | 44 |
| Figura 4.9. Representación de la compensación rotacional entre dos imágenes consecutivas..... | 45 |
| Figura 4.10. Radios de las zonas de interés en imágenes omnidireccionales. | 45 |
| Figura 4.11. a) Imagen panorámica sin resolución uniforme, b) imagen panorámica con resolución uniforme mediante interpolación..... | 46 |
| Figura 4.12. a) Polinomio para la compensación de las columnas en la imagen panorámica, b) Polinomio para la compensación de los renglones en la imagen panorámica..... | 47 |

| | |
|--|----|
| Figura 4.13. a) Imagen compensada mostrando las zonas donde fue compensada (zonas en negro) b) imagen sin las zonas de compensación. | 47 |
| Figura 5.1. Número mínimo de valores rastreados a los largo de la secuencia de imágenes. | 50 |
| Figura 5.2 Desviación estándar del Video 1 respecto al número mínimo de valores rastreados. | 51 |
| Figura 5.3. Porcentaje de valores eliminados del histograma de movimiento. | 51 |
| Figura 5.4 Desviación estándar del Video 1 respecto al porcentaje de valores eliminados del histograma de movimiento. | 51 |
| Figura 5.5. Grado del polinomio para modelar el movimiento de la imagen. | 52 |
| Figura 5.6 Desviación estándar del Video 1 respecto al grado del polinomio para modelar el movimiento de la imagen. | 52 |
| Figura 5.7 Compensación rotacional en sistemas convencionales, se ilustra cómo se pierde información de las esquinas de la imagen panorámica. | 58 |

Índice de tablas

| | |
|---|----|
| Tabla 2.1 Diferentes transformaciones geométricas de imágenes. | 19 |
| Tabla 3.1. Tabla panorámica para la transformación de imágenes omnidireccionales. | 27 |
| Tabla 3.2. Complejidad de los algoritmos de reconstrucción de las imágenes omnidireccionales. | 29 |
| Tabla 5.1. Resultados del algoritmo de transformación <i>Pano Mapping Table</i> en sus dos versiones. | 49 |
| Tabla 5.2. Resultados del algoritmo de estabilización sin precisión subpixel y sus diferentes etapas. | 54 |
| Tabla 5.3. Resultados del algoritmo de estabilización con precisión subpixel y sus diferentes etapas. | 55 |
| Tabla 5.4. Tiempos de procesamiento de los algoritmos con y sin precisión subpixel. | 56 |
| Tabla 5.5 Resultados de algoritmos convencionales implementados en las secuencias de prueba. | 57 |

1 Introducción

1.1 Motivación

Los sistemas de imágenes convencionales se encuentran bastante limitados por su campo de visión (*FOV*, por sus siglas en inglés), este parámetro es básicamente el área que observa el sensor, la cual está definida por un rectángulo que está establecido por los parámetros de fabricación del sensor. El enfoque que se le da a este problema es el de incorporar un espejo al sistema convencional de imágenes para incrementar su campo de visión.

Algunas soluciones que se han implementado para solucionar la limitación del campo de visión han sido por ejemplo, rotar el sistema de imágenes en su centro de proyección o usar un lente *fish-eye*. En la Figura 1.1 se muestran los campos de visión de algunas soluciones. La mayoría de las soluciones propuestas incorporan sistemas mecánicos, los cuales han sido adoptados para aplicaciones en tiempo real, sin embargo su velocidad se ve comprometida por la respuesta de los actuadores, dado que es muy lenta. Otro factor al que se ven expuestos, es que también son susceptibles a descalibración de las piezas mecánicas.

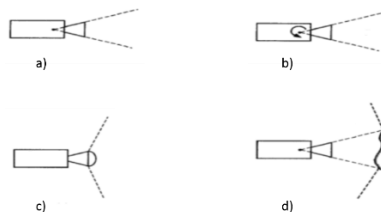


Figura 1.1 Sistemas de imágenes con sus respectivos campos de visión a) convencional, b) rotacional, c) *fish-eye* y d) catadióptrico

En la actualidad han surgido cámaras en las cuales el campo de visión es ampliado por diferentes dispositivos tales como espejos o lentes, estas cámaras se les denomina catadióptricas, las cuales tienen la propiedad de entregarnos imágenes panorámicas del entorno (360°). Este tipo de imágenes han adquirido gran popularidad para supervisión, monitoreo, teleconferencias y otras aplicaciones donde se requieran imágenes de 360° del entorno. Una de las ventajas de las imágenes panorámicas, es la reducción del número de cámaras que se necesitan para realizar vistas panorámicas del ambiente [1]. Otra de las ventajas de este tipo de sistemas es la eliminación de piezas móviles que afectan con problemas de calibración a los sistemas de imágenes.

Uno de los grandes retos a superar en las cámaras catadióptricas, es la reconstrucción de la imagen reflejada en el espejo, ya que dicha imagen se encuentra distorsionada por la naturaleza del espejo (circular, parabólico, hiperbólico). También es importante esta reconstrucción para poder realizar procesamientos a nuestras imágenes, ya que la mayoría de estos están definidos para espacios rectangulares. En la Figura 1.2 se muestra una imagen obtenida del sistema catadióptrico y su reconstrucción.

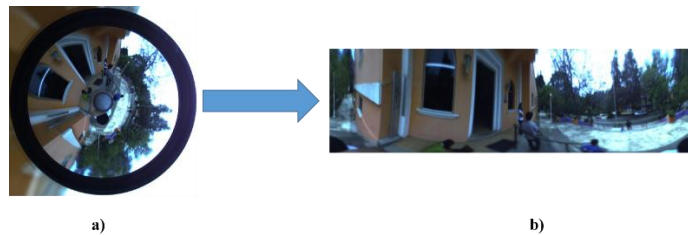


Figura 1.2 a) Imagen omnidireccional, b) Imagen panorámica reconstruida

Otro gran reto de los sistemas de imágenes, es el movimiento al que se ven expuestos, ya que esto provoca que la secuencia de imágenes sea inestable por lo cual la detección o seguimiento de objetos se ve seriamente afectado. Desde hace varios años han existido sistemas de estabilización mecánicos, estos sistemas han dado buenos resultados pero con la miniaturización de los sistemas de imágenes, este tipo de estabilizadores han pasado a ser demasiado costosos y no tan sencilla su migración a dispositivos tan pequeños [2].

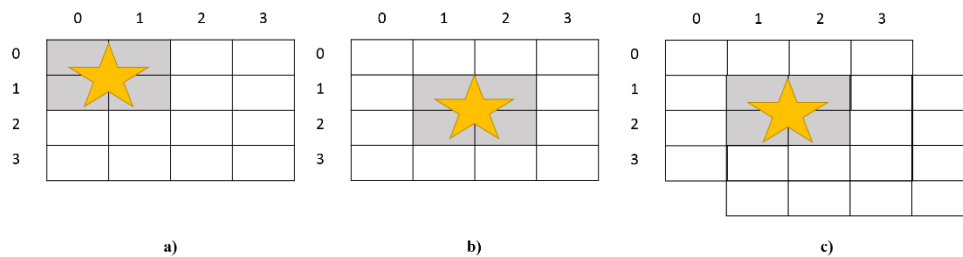


Figura 1.3 Método de estabilización digital, la a) imagen de referencia es utilizada para alinear la b) imagen actual y obtener una c) imagen compensada.

Una alternativa a este problema es la implementación de algoritmos de estabilización electrónica, los cuales se aplican directamente a la imagen capturada. Estos algoritmos básicamente lo que hacen es tomar una imagen de referencia, compararla con la imagen que se tomó y tratar de compensar el movimiento que se produjo, ya que si no se compensa la imagen tomada saldrá borrosa debido al

movimiento [3]. En la Figura 1.3 se ejemplifica el proceso que realizan los algoritmos de estabilización electrónica.

1.2 Objetivo General

El objetivo general de este trabajo de investigación es desarrollar un algoritmo con precisión subpixel capaz de realizar la estabilización digital de secuencias de imágenes obtenidas de un sistema catadióptrico, que sea robusto a objetos en movimiento en las escenas y a cambios de iluminación en ambientes no controlados.

1.3 Objetivos específicos

Los objetivos específicos para este trabajo son:

- Proponer el modelo analítico del sistema catadióptrico de imágenes.
- Proponer e implementar un algoritmo para la transformación de imágenes en el plano hiperbólico al plano cartesiano.
- Diseñar y desarrollar un algoritmo para la estimación de movimiento en imágenes subsecuentes.
- Diseñar y desarrollar un algoritmo para clasificar los vectores de movimiento que son ajenos al movimiento de la cámara.
- Proponer el modelo para la compensación de las imágenes a partir de los vectores de movimiento.

1.4 Solución Propuesta

Nosotros presentamos un algoritmo de estabilización de imágenes en sistemas de imágenes catadióptricos utilizando el descriptor SURF para realizar un historial de puntos a lo largo de la secuencia de imágenes y con esto eliminar la mayoría de los puntos que no pertenecen a las vibraciones del sistema. El siguiente paso para eliminar los valores restantes es el truncamiento del histograma. Los pasos mencionados anteriormente se realizan en 2 etapas, una antes de la transformación de la imagen omnidireccional y otra en la etapa traslacional, con el objetivo de eliminar el error acumulativo.

El modelo de las vibraciones del sistema se obtiene mediante el ajuste de un polinomio a los valores restantes del historial. El polinomio nos da una representación de los movimientos a través de la imagen, con lo cual podemos realizar la compensación de la imagen.

Los resultados de este trabajo de investigación son basados en la medida de estabilización de imágenes *peak signal to noise ratio* o PSNR por sus siglas en inglés, mediante esta métrica podemos evaluar la similitud entre imágenes. Una secuencia de imágenes estabilizada posee un mayor PSNR respecto a la secuencia original debido a que se elimina el movimiento entre las imágenes y estas presentan mayor similitud entre ellas, partiendo de esta hecho, podemos mencionar que nuestro algoritmo de estabilización de imágenes logro un incremento en promedio del 12% de PSNR de las secuencias estabilizadas respecto a la secuencia original.

1.5 Contribuciones

Las principales contribuciones de nuestro proyecto de investigación son las siguientes:

- Un algoritmo de estabilización de imágenes exclusivamente diseñado para sistemas de visión catadióptricos.
- Un método para encontrar los movimientos que son parte de las vibraciones que afectan al sistema de visión catadióptrico.
- Un algoritmo con precisión subpixel para compensar los movimientos rotacionales sobre el eje Z del sistema de visión catadióptrico.
- Un algoritmo con precisión subpixel para compensar los movimientos traslacionales sobre el eje Z del sistema de visión catadióptrico.

1.6 Organización del documento

Este documento está organizado de la siguiente manera. El capítulo 2 presenta algunos conceptos básico para la comprensión del modelo del sistema catadióptrico y algunos conceptos de cómo medir el movimiento en las imágenes y el nivel de estabilización. El capítulo 3 presenta un pequeño resumen de los diferentes trabajos que existen en la reconstrucción de imágenes omnidireccionales y métodos para la estabilización de imágenes. El capítulo 4 trata acerca de nuestro método propuesto y las diferentes etapas de las que está compuesto. En el capítulo 5 describimos los experimentos que realizamos para probar el desempeño de nuestro algoritmo, así como las formas de evaluación y los resultados obtenidos. Por último, en el capítulo 6, discutimos las conclusiones que son derivadas de este trabajo de investigación. También hablamos de las principales contribuciones y de las posibilidades que existen para un trabajo futuro.

2 Fundamentos Teóricos

2.1 Sistema catadióptrico

Las cámaras catadióptricas son ampliamente usadas en sistemas de vigilancia, navegación de robots y para reconstrucción 3D, donde el campo de visión debe de ser mayor que el proporcionado por los sistemas convencionales. Las cámaras convencionales con un campo de visión limitado, continuamente pueden perder de vista al objeto cuando el punto de referencia cambia significativamente debido a algún movimiento del sistema, del objeto o de ambos. Los sensores catadióptricos omnidireccionales, formados generalmente por una cámara, un espejo y lentes, pueden seguir objetos en un rango más amplio que en las cámaras convencionales [4].

Algunos intentos del uso de sensores omnidireccionales incluyen colocar diversas cámaras en diferentes posiciones [5], usar diferentes arreglos de cámaras con dispositivos mecánicos que proporcionan un movimiento rotacional sobre uno de sus ejes e incluso utilizar espejos planos para aumentar el campo de visión [6], [7]. Los sistemas mencionados anteriormente presentan diversos problemas en los que se encuentran, principalmente, el registro de las imágenes (proceso de transformación de diversos conjuntos de imágenes a un sistema de coordenadas unificado) y las vibraciones propias del sistema.

Las cámaras con lentes *fish-eye* también han sido utilizadas para incrementar el campo de visión [8], sin embargo estas cámaras han demostrado que la distorsión de los objetos más cercanos a la cámara es irreversible y que no existe un único punto de vista en la cámara, este fenómeno lo podemos ver representado en la Figura 2.1.

Los sensores catadióptricos consisten en una cámara que se encuentra fija, usualmente montada verticalmente, además de un espejo que es rotacionalmente simétrico y que se encuentra suspendido sobre la cámara. Los espejos más comunes en los sistemas catadióptricos son con forma parabólica, hiperbólica y elipsoidal.

Las ventajas de los sensores catadióptricos están basadas en que el escaneo de su campo de visión omnidireccional (360°) es casi instantáneo, no como en el sistema rotacional que requieren más tiempo debido a los diversos actuadores mecánicos que tienen que interactuar para cubrir todo el campo de visión omnidireccional. Un menor tiempo de captura significa que tendrá una menor exposición a problemas relacionados por los movimientos y las vibraciones de la cámara, o por objetos en movimiento.

Por lo anteriormente mencionado, es evidente que este tipo de sistemas es idóneo para ambientes dinámicos porque una de las características de los sistemas catadióptricos es la de mantener por más tiempo los objetos dentro del campo visual aun cuando el objeto realice movimientos rápidos. Entonces, los sistemas de visión catadióptricos son ideales para situaciones de navegación de vehículos [9], estimación de movimiento [10] y sistemas de vigilancia [11].



Figura 2.1 Imagen tomada mediante un lente *fish-eye*, mostrando la distorsión que genera el lente.

La principal desventaja de los sistemas catadióptricos es que la resolución de las imágenes no es constante debido a la forma de los espejos, este fenómeno lo podemos observar en la Figura 2.2. Debido a esta distorsión en las imágenes omnidireccionales, no se puede aplicar el clásico procesamiento de imágenes que se aplica a un sistema convencional, por esta razón las imágenes que se obtienen del sistema de visión catadióptrico deben de someterse a una transformación para poder aplicar diferentes tipos de procesamiento.

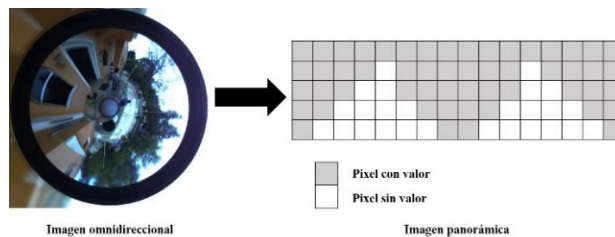


Figura 2.2 Transformación de una imagen omnidireccional a una imagen panorámica y muestra de la pérdida de resolución.

2.2 Modelo del sistema catadióptrico

Los sistemas catadióptricos, como mencionamos anteriormente están formados por una cámara y un espejo, sin embargo existen diversas configuraciones de estos sistemas que a continuación explicamos.

2.2.1 Sistemas no centrales

Los sistemas catadióptricos son formados principalmente por un arreglo combinado de espejos y cámaras de perspectiva (convencionales). Un sistema catadióptrico es llamado sistema no central (es decir, no tienen un solo punto focal) cuando los rayos de luz que inciden sobre la cámara y que son reflejados por el espejo, no tienen un punto en común donde intersecten. Este tipo de sistemas se ejemplifican con la Figura 2.3.

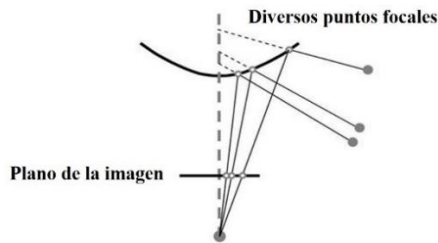


Figura 2.3. Sistema catadióptico no central y sus diversos puntos focales.

2.2.2 Sistemas centrales

Por el lado contrario, los sistemas centrales cumplen con la propiedad de tener un solo punto focal en el sistema. Cada rayo de luz que es reflejado por la superficie del espejo, que intersecta un solo punto, llamado punto focal. En la Figura 2.4 se ejemplifica este tipo de sistemas.

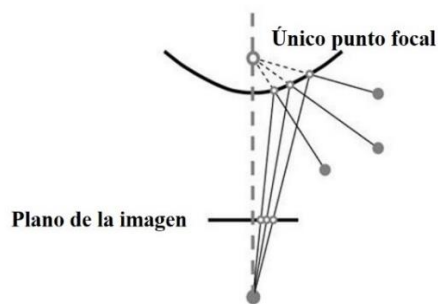


Figura 2.4. Sistema catadióptico central y su punto focal.

En este trabajo de investigación utilizamos un sistema catadióptrico central, el cual está configurado de la siguiente manera:

- Una cámara y un espejo hiperbólico.
- El centro óptico de la cámara tiene que coincidir con el foco de la hipérbola, con esto se asegura que los rayos de luz reflejados hacia la cámara, intersectan en un mismo punto, el foco del espejo hiperbólico.

En la Figura 2.5 se representa el tipo de sistema que usamos en nuestro trabajo de investigación, en la cual podemos observar todas las características que mencionamos anteriormente.

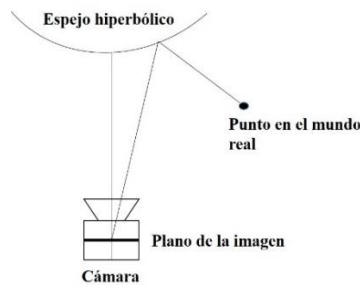


Figura 2.5. Configuración del sistema catadióptrico utilizado en este trabajo de investigación.

2.3 Calibración

El proceso de calibración consiste en obtener los parámetros que modelan las diferentes transformaciones, las cuales son aplicadas a los objetos en el mundo real para obtener como resultado final una imagen digital. En los sistemas catadióptricos, dicho proceso trata de encontrar el mejor conjunto de parámetros para modelar las transformaciones a través del espejo del sistema. En este proyecto utilizamos el modelo de Scaramuzza [12] para obtener y modelar los parámetros del sistema catadióptrico, A continuación ahondamos en la técnica usada en dicho método para obtener los parámetros.

En la Figura 2.6 podemos observar cómo está constituido el sistema catadióptrico, tenemos el espejo y dos planos paralelos a él. Los planos representan la proyección en el sensor de la cámara y la proyección en el plano de la imagen. El plano del sensor y el plano de la imagen están relacionados mediante una transformación afín, esto para modelar la distorsión que sufre la imagen por el lente de la cámara.

La proyección en el plano del sensor la definimos como:

$$\mathbf{u}'' = \begin{bmatrix} u'' \\ v'' \end{bmatrix}$$

Y la proyección en el plano de la imagen se representa como:

$$\mathbf{u}' = \begin{bmatrix} u' \\ v' \end{bmatrix}$$

Entonces la relación entre \mathbf{u}'' y \mathbf{u}' queda expresada de la siguiente manera:

$$\mathbf{u}'' = A\mathbf{u}' + t$$

Donde A representa una rotación y un escalamiento, t representa una traslación. La representación de un punto en las coordenadas del mundo real mediante su posición en el plano de la imagen, Scaramuzza [12] lo expresa de la siguiente manera:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix}$$

Donde,

$$f(u, v) = a_0 + a_1\rho'' + a_2\rho''^2 + \dots + a_n\rho''^n$$

$$\rho'' = \sqrt{u''^2 + v''^2}$$

Para resolver el problema de la calibración debe de encontrarse los parámetros $[A, t, a_0, a_1, a_2, \dots, a_n]$, esto se logra realizando una minimización no lineal mediante el algoritmo de Levenberg y Marquadt [13].

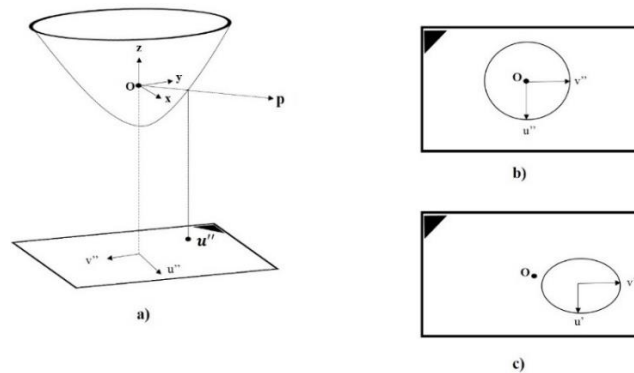


Figura 2.6. Sistema coordenado del sistema catadióptrico, b) plano del sensor y c) plano de la imagen.

El proceso de calibración utiliza unos patrones para conocer la geometría de un objeto en el espacio y poder tener como referencia un objeto. En general la calibración consiste en conocer las medidas de varios puntos en el mundo real y compararlas con las proporciones de los mismos puntos capturados en la imagen, para así calcular la mejor transformación que modele las proporciones de la imagen a las del mundo real. El número de puntos en el mundo real depende del patrón de calibración que se está utilizando, en nuestro caso utilizamos un patrón cuadrangular de 30 puntos (6x5). El autor recomienda que por lo menos se utilicen 10 imágenes distintas del patrón en diferentes posiciones para obtener un mejor resultado de la calibración.

En la Figura 2.7 podemos observar cómo se realiza la proyección de los puntos estimados de los patrones mediante el proceso de calibración a las coordenadas del espacio.

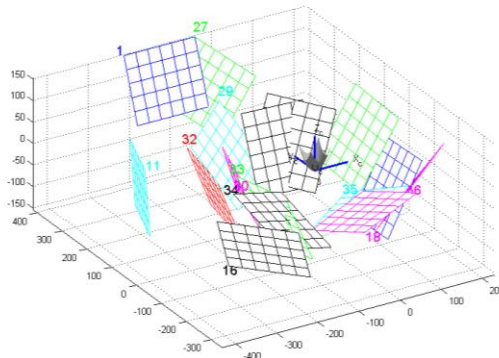


Figura 2.7. Reproyección de los patrones de calibración a sus respectivas coordenadas en el mundo real.

Cabe mencionar que el modelo propuesto por Scaramuzza [12], al momento de realizar la reproyección de los puntos del patrón en el espacio de la imagen, sólo posee un error de 0.25 píxeles respecto a la posición original del punto en la imagen.

2.4 Descriptores de puntos característicos

2.4.1 *Scale Invariant Feature Transform (SIFT)*

El descriptor *Scale Invariant Feature Transform (SIFT)* fue desarrollado por Lowe [14] como un algoritmo capaz de detectar puntos característicos en una imagen.

Estos puntos son invariantes frente a diferentes transformaciones como traslación, escala, rotación, cambios de iluminación y transformaciones afines.

El algoritmo SIFT se compone principalmente de cuatro etapas que se describen siguiendo la implementación de Lowe [14]:

1. Detección de extremos en el espacio escala:

La primera etapa del algoritmo realiza una búsqueda sobre las diferentes escalas y dimensiones de la imagen, identificando aquellos puntos que sean invariantes a los cambios de orientación y escala. Esto se lleva a cabo mediante la función diferencia de gaussiana, la cual está definida como:

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

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

Donde G representa el filtro gaussiano y k la escala del filtro, la diferencia de gaussiana se ejemplifica en la Figura 2.8.

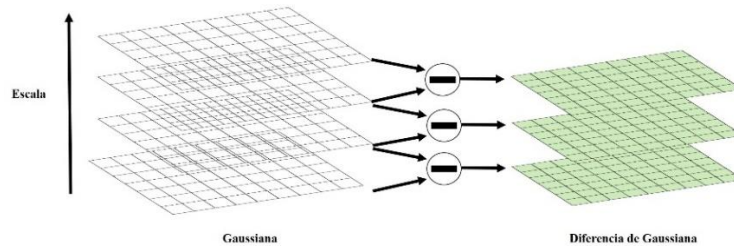


Figura 2.8. Diferencias de gaussianas usadas para la construcción del descriptor SIFT.

Una vez que se han aplicados los filtros y tenemos las diferencias de gaussianas, procedemos a encontrar los extremos, esto se logra comparando cada punto con sus 26 vecinos, como se muestra en la Figura 2.9. El punto que se encuentra bajo análisis corresponde a un extremo, si es el mayor o el menor de sus 26 vecinos.

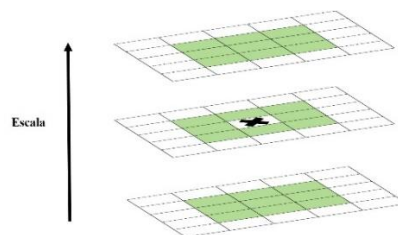


Figura 2.9. Selección de puntos extremo en el algoritmo SIFT.

2. Localización de los puntos de interés:

Para seleccionar los puntos de interés, de forma precisa, se aplica una medida de estabilidad sobre todos ellos para descartar aquellos que no sean adecuados. Utilizando la matriz Hessiana de $D(x, y, \sigma)$ eliminamos los puntos que tienen bajo contraste y aquellos que pertenecen a un borde.

3. Asignación de la Orientación:

En este paso se calcula la magnitud y orientación del gradiente alrededor de los puntos de interés, se asignan una o más orientaciones a cada punto de interés extraído de la imagen, basándose en las direcciones locales presentes en la imagen. El primer paso es calcular la magnitud y orientación del gradiente de cada punto clave que paso la etapa anterior, la magnitud y orientación se calcula con las siguientes ecuaciones:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$
$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right)$$

Un histograma de orientación se forma a partir de las orientaciones de los gradientes de los puntos que se encuentran dentro de la región que rodea al punto de interés. El histograma de orientación posee 36 divisiones que abarcan los 360° .

Los picos en el histograma de orientaciones corresponden a las direcciones que predominan en los gradientes locales, con esto se procede a detectar el mayor pico del histograma, y entonces, si existen otros máximos superiores al 80 % del pico principal, éstos se utilizarán para crear otros puntos de interés con nuevas orientaciones.

4. Descriptor del punto de interés:

La última etapa hace referencia a la representación de los puntos de interés como una medida de los gradientes locales de la imagen en las proximidades de dichos puntos y respecto de una determinada escala. Una vez que ya se tienen todos los puntos de interés de la imagen, se tiene que segmentar la vecindad del punto de interés en 4×4 regiones con tamaño de 4×4 píxeles. Una vez que ya se ha dividido la vecindad del punto clave, se genera un histograma de orientación de gradiente para cada región. Estos histogramas poseen ocho divisiones cada uno como se observa en la Figura 2.10.

Cada punto de interés corresponde a un vector de características compuesto por 128 elementos, el cual está compuesto por las 4 x 4 regiones con un histograma de 8 divisiones cada una, que le proporciona una invarianza parcial a deformaciones y a como cambios de iluminación.

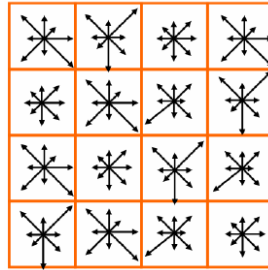


Figura 2.10. Histograma de orientaciones de un punto de interés en el algoritmo SIFT

2.4.2 *Speeded Up Robust Features (SURF)*

El descriptor SURF, cuyas siglas hacen referencia al algoritmo *Speeded Up Robust Features*, fue desarrollado por Herbert Bay et al. [15] como un detector de puntos de interés. El descriptor SURF guarda cierta similitud con la metodología del descriptor SIFT [14] pero presenta notables diferencias, que quedan expuestas en la explicación de su funcionamiento. Los autores hacen hincapié en que este descriptor presenta principalmente 2 mejoras resumidas en los siguientes conceptos:

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

Estas mejoras se consiguen mediante la reducción de la dimensionalidad y complejidad en el cálculo de los vectores de características de los puntos de interés que se obtienen de las imágenes, mientras continúan siendo bastante descriptivos e igualmente repetitivos.

A continuación se presentan las diferencias más sobresalientes respecto del descriptor SIFT:

- La normalización o longitud de los vectores de características de los puntos de interés es considerablemente menor, concretamente se trata de vectores con una dimensionalidad de 64, lo que supone una reducción de la mitad de la longitud del descriptor SIFT.

- El descriptor SURF utiliza siempre la misma imagen, la original. El algoritmo SIFT utiliza diferentes tamaños de la imagen submuestreada, sin embargo SURF utiliza imágenes integrales.
- Utiliza el determinante de la matriz Hessiana para calcular tanto la posición como la escala de los puntos de interés.

A continuación se describirán a detalle las etapas para la creación de los descriptores SURF:

1. Detección de puntos:

El descriptor SURF hace uso de la matriz Hessiana, más concretamente, del valor del determinante de la matriz, para la localización y la escala de los puntos. Primero se realiza el cálculo de la matriz Hessiana para cada valor σ , donde σ es el valor de la escala del filtro gaussiano. La matriz Hessiana $H(p, \sigma)$ de un punto $p = (x, y)$ en una imagen $I(x, y)$ se encuentra definida como:

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

Donde $L_{xx}(p, \sigma)$ es la convolución de segundo orden de la Gaussiana con la imagen original en el punto p . Los otros términos de la matriz Hessiana $L_{xy}(p, \sigma)$ y $L_{yy}(p, \sigma)$, se pueden definir de la misma forma como:

$$L_{xx}(p, \sigma) = \frac{\partial^2}{\partial x^2} G(\sigma) * I(p)$$

Las aproximaciones de las derivadas parciales se denotan como D_{xx} , D_{xy} y D_{yy} , el determinante de la matriz Hessiana se puede calcular de la siguiente manera:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$$

En la Figura 2.11 se puede observar la representación de la derivada parcial de segundo orden de un filtro gaussiano discretizado y la aproximación de la derivada implementada en el caso del descriptor SURF.

La imagen de salida obtenida tras la convolución de la imagen original con un filtro de dimensiones 9 x 9, que corresponde a la derivada parcial de segundo orden de una gaussiana con $\sigma = 1.2$, es considerada como la escala inicial. Las capas

sucesivas se obtienen mediante la aplicación gradual de filtros de mayores dimensiones.

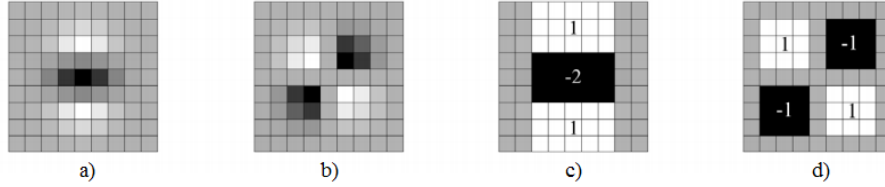


Figura 2.11. Filtro gaussiano de segundo orden de la derivada parcial en: a) dirección y, b) dirección xy, aproximación del filtro gaussiano usando filtros de caja en c) dirección y, d) dirección xy.

Finalmente para calcular la localización de todos los puntos de interés en todas las escalas, se realiza mediante la eliminación de los puntos que no cumplan la condición de máximo en una vecindad de $3 \times 3 \times 3$, este principio consiste en descartar todos los valores que no sean el máximo en una región, dejando como resultado solo el valor máximo.

2. Asignación de la orientación:

La siguiente etapa en la creación del descriptor corresponde a la asignación de la orientación de cada uno de los puntos de interés obtenidos en la etapa anterior. En esta etapa es donde se otorga al descriptor de cada punto, la invarianza ante la rotación mediante la orientación del punto de interés. El primer paso para otorgar la mencionada orientación consiste en el cálculo de la respuesta de los filtros Haar en las direcciones x e y mediante los filtros Haar mostrados en la Figura 2.12.

Una vez obtenidas las respuestas Haar, estas son ponderadas utilizando una gaussiana con $\sigma=2.5$, centrada en el punto de interés. Las respuestas son representadas como vectores en un espacio con la respuesta horizontal a lo largo del eje x y la respuesta vertical a lo largo del eje y . La orientación dominante es estimada calculando la suma de todas las respuestas con una ventana corrediza cubriendo un ángulo de $\pi/3$. Las respuestas horizontales y verticales dentro de la ventana son sumadas. El vector más grande dentro de los 6 sectores en los que se divide el área, es el que asigna su orientación al punto de interés.

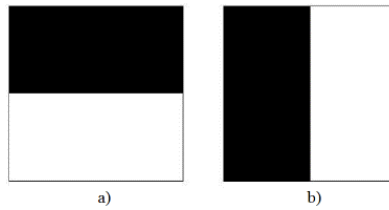


Figura 2.12. Filtros de Haar utilizados en el descriptor SURF, el color negro indica un valor de -1 y el color blanco un valor de 1. a) Filtro en x, b) filtro en y.

3. Creación del descriptor:

El primer paso es construir una ventana cuadrada alrededor del punto de interés, orientada en relación a la orientación obtenida en el paso anterior. Esta región es a su vez dividida en 4 x 4 subregiones dentro de cada una de las cuales se calculan las respuestas de los filtros Haar, consideramos d_x y d_y las respuestas de los filtros Haar relativas a la orientación del punto de interés, en las direcciones horizontal y vertical respectivamente. Cada una de las subregiones quedan representadas por un vector definido como:

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

Finalmente el vector resultante engloba las 4x4 subregiones y sus 4 respuestas de los filtros Haar, lo cual nos da una longitud total de 64 elementos. El cálculo de las respuestas de las subregiones se puede ejemplificar en la Figura 2.13.

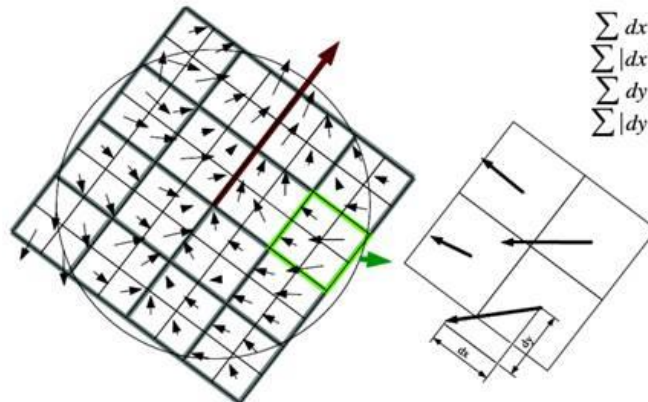


Figura 2.13. Formación del descriptor SURF mediante 4 subregiones y sus respectivas respuestas a los filtros Haar.

2.5 Correspondencia entre puntos característicos

La correspondencia de puntos característicos entre imágenes tiene como finalidad, el cálculo de un valor que represente el grado de similitud entre las dos imágenes y que con dicho valor de similitud se puedan establecer diferentes conclusiones. El cálculo de este valor, representado como distancia se realiza mediante la aplicación de una métrica o fórmula de la distancia entre ambas imágenes.

Para calcular el grado de similitud entre los puntos característicos existen varias alternativas. Sin embargo, en este proyecto nos hemos centrado en una de ellas, la distancia euclidiana.

La correspondencia entre puntos característicos se lleva a cabo mediante el cálculo de la distancia euclidiana entre los vectores de características pertenecientes a diferentes puntos de interés. Este cálculo genera a su vez otro valor, conocido como umbral, que será utilizado para determinar cuál de los puntos de la imagen comparada corresponde con su homólogo, en la imagen de referencia. La distancia euclidiana la podemos definir como:

$$d_i = \sqrt{(a_i - b_i)^2}$$
$$K = \sum_{i=1}^n d_i$$

Donde d_i es la distancia euclidiana entre el elemento a_i de un descriptor de la imagen A y el mismo elemento b_i de un descriptor en la imagen B. La variable K es la suma de las distancias euclidianas que hay entre los n elementos de ambos descriptores, donde n puede tomar el valor de 64 o 128 dependiendo del descriptor que se esté utilizando.

2.6 Transformaciones de imágenes

2.6.1 Transformaciones geométricas

El objetivo principal de las transformaciones geométricas es la de modificar las relaciones espaciales que existen entre los píxeles de una imagen, sin embargo, el valor de cada píxel continua siendo el mismo. La imagen puede ser más grande, pequeña, rotada, trasladada o transformada de otras maneras [16].

Supongamos que tenemos (u, v) y (x, y) , los cuales son sistemas coordenados llamados espacio de entrada y espacio de salida, respectivamente [17]. Una transformación de coordenadas geométricas la podemos definir como aquella que lleva el espacio de coordenadas de entrada a un espacio de coordenadas de salida, como se ilustra en la siguiente ecuación:

$$(x, y) = T\{(u, v)\}$$

Donde T es la transformación que hace el cambio de espacio coordenado de entrada al espacio de salida, también T tiene su transformación inversa que transforma las coordenadas del espacio de salida a las coordenadas del espacio de entrada, esta transformación está definida por la siguiente ecuación:

$$(u, v) = T^{-1}\{(x, y)\}$$

En esta sección sólo abordaremos el caso de las transformaciones geométricas, que son las transformaciones afines. Estas transformaciones son las que utilizamos en el desarrollo de este trabajo de investigación. Las transformaciones afines están definidas de tal forma, que son funciones lineales, es decir, están expresadas por una multiplicación de matrices y una parte aditiva. Esta transformación puede ser descrita como en la siguiente ecuación:

$$[x \ y] = [u \ v] \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + [b_1 \ b_2]$$

La transformación afín puede ser simplificada añadiendo una tercera coordenada, esta adición es para que el cálculo de la transformación sea más fácil en el sentido matemático y computacional. Esta simplificación queda descrita en la siguiente ecuación:

$$[x \ y \ 1] = [u \ v \ 1] \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ b_1 & b_2 & 1 \end{bmatrix}$$

Variando los coeficientes de la matriz de transformación se pueden obtener diferentes efectos como escalamiento, rotación, traslación, deformación y reflexión. Las ecuaciones de los diferentes tipos se muestran en la Tabla 2.1. En este trabajo las principales transformaciones que se usaron fueron las transformaciones de rotación, traslación y escalamiento.

| Tipo | Matriz de transformación | Ecuación de las coordenadas |
|------------------------|--|--|
| Identidad | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = u$ $y = v$ |
| Escalamiento | $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = s_x u$ $y = s_y v$ |
| Rotación | $\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = u \cos \theta - v \sin \theta$ $y = u \sin \theta + v \cos \theta$ |
| Deformación horizontal | $\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = u + \alpha v$ $y = v$ |
| Deformación vertical | $\begin{bmatrix} 1 & \beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = u$ $y = \beta u + v$ |
| Reflexión vertical | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = u$ $y = -v$ |
| Reflexión horizontal | $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = -u$ $y = v$ |
| Traslación | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \delta_x & \delta_y & 1 \end{bmatrix}$ | $x = u + \delta_x$ $y = v + \delta_y$ |

Tabla 2.1 Diferentes transformaciones geométricas de imágenes.

2.7 Medidas de estabilización de imágenes

Las métricas de calidad de video son métodos objetivos con los cuales podemos medir la calidad de un video que fue sometido a un procesamiento. Los métodos para la evaluación de la compensación de movimiento en los algoritmos de

estabilización están basados en el hecho de que cuando el movimiento ha sido perfectamente compensado en las secuencias de imágenes, la diferencia entre la imagen estabilizada y la imagen de referencia es minimizada, también en teoría debería de ser cero para los objetos que permanecen constantes entre la secuencia, como el fondo de la imagen.

El PSNR por sus siglas en inglés (*Peak Signal to Noise Ratio*), es la medida más utilizada para medir la estabilización de imágenes[18]. Esta medida nos puede indicar que tanto se traslapan dos imágenes, un PSNR máximo es alcanzado cuando se logra una estabilización confiable. La ecuación del PSNR se define como:

$$PSNR(I_r, I_s) = 10 \log \frac{255}{MSE(I_r, I_s)}$$

Donde I_r es la imagen de referencia, I_s es la imagen estabilizada y MSE es el error cuadrático medio entre dos imágenes, el cual está definido por la siguiente ecuación:

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N |I_r(x, y) - I_s(x, y)|^2$$

Donde M y N son el ancho y la altura de las imágenes de las secuencias.

El resultado de la ecuación anterior esta dado en decibels *db*, como se mencionó anteriormente, mientras mayor sea el PSNR mayor será el grado de estabilización logrado. Esto lo podemos observar a partir de la ecuación del PSNR, ya que si el MSE es cercano a cero, es decir las imágenes son muy parecidas, el resultado de la división será muy grande. El efecto inverso se verá si el MSE es más grande, es decir las imágenes no son muy parecidas, por lo tanto el resultado de la división será pequeño.

Una vez que sabemos cómo medir el nivel de estabilización entre dos imágenes, debemos saber el nivel de estabilización de una secuencia completa de imágenes. Este nivel es conocido como *Interframe Transformation Fidelity* (ITF) [19], el cual está definido como:

$$ITF = \frac{1}{\#imágenes - 1} \sum_{k=2}^{\#imágenes} PSNR(I_{k-1}, I_k)$$

Como observamos en la ecuación anterior, lo que obtenemos es un promedio de PSNR entre cada una de las imágenes tomando como referencia la imagen anterior. El ITF se aplica a las dos secuencias bajo experimentación, la secuencia sin

estabilizar y la secuencia estabilizada; en principio la secuencia estabilizada debe de poseer un ITF mayor al de la secuencia sin estabilizar.

2.8 Regresión polinomial

En este proyecto de investigación para modelar las distintas vibraciones que sufren las imágenes, debemos de ajustar los datos a una curva que es descrita por un polinomio de grado m , todo esto con la finalidad de obtener el modelo que describa los movimientos que sufren las imágenes y a partir de dicho modelo, realizar la compensación de la imagen.

Supongamos que tenemos un conjunto de datos distintos:

$$x_1, x_2, x_3, \dots, x_n$$

Y también contamos con las observaciones que se realizaron con los puntos anteriores:

$$y_1, y_2, y_3, \dots, y_n$$

Ahora consideremos que la ecuación de la regresión polinomial entre la variable independiente x y la variable dependiente y , puede ser representada de la siguiente forma:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \dots + \beta_p x^p$$

Escrito de forma matricial, el modelo de la regresión polinomial quedaría de la siguiente forma:

$$Y = \beta^T X$$

Donde Y es un vector de tamaño $1 \times n$ y X es una matriz de tamaño $n \times (p+1)$, donde p es el grado del polinomio al que queremos ajustar los datos. También hay que mencionar que la primera columna de X , todos sus elementos son 1. β es un vector de tamaño $(p+1) \times 1$.

El objetivo principal de ajustar un polinomio a los datos del movimiento de las imágenes es para suavizar las variaciones que tienen los datos debido a errores aleatorios o no controlados [20], como pueden ser objetos que se mueven dentro de la imagen o cambios en la iluminación de la imagen.

Ahora, como mencionamos antes, tenemos a X y Y de la siguiente forma:

$$Y = \begin{bmatrix} Y_{11} \\ Y_{21} \\ Y_{31} \\ \vdots \\ \vdots \\ Y_{n1} \end{bmatrix} \quad X = \begin{bmatrix} X_{11} & \cdot & \cdot & \cdot & X_{1(p+1)} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ X_{n1} & \cdot & \cdot & \cdot & X_{n(p+1)} \end{bmatrix}$$

Para estimar los coeficientes β del polinomio que deseamos ajustar, utilizamos el método de mínimos cuadrados:

$$E = \sum \epsilon^2 = \sum (y - \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdot \cdot \cdot + \beta_p x^p)^2$$

Donde lo que se busca minimizar es la suma de los cuadrados del error ϵ , es decir, encontrar los coeficientes β que hagan que el polinomio que se desea ajustar tenga el menor error entre Y y el polinomio que se propuso. Con dicho método se busca eliminar los valores que sean ajenos al conjunto de datos pertenecientes al sistema. En este trabajo de investigación, utilizamos la regresión polinomial para eliminar los valores que no pertenecen a las vibraciones que afectan a las imágenes del sistema catadióptrico de imágenes.

2.9 Resumen

En este capítulo se revisaron los principales tópicos para la comprensión del desarrollo de este trabajo de investigación; comenzamos definiendo como se forma, cuáles son las principales ventajas, desventajas y clasificación de los sistemas de visión catadióptricos. Después describimos el proceso de calibración de sistemas catadióptricos y su utilidad para la obtención de diversos parámetros para el desarrollo de este trabajo. En el siguiente paso, se habló de los dos principales descriptores de puntos característicos y la forma para medir la correspondencia entre ellos, como parte fundamental de este trabajo, dado que es el medio por el cual se puede estimar el movimiento entre 2 imágenes consecutivas. También hicimos una introducción a las transformaciones geométricas, las cuales son usadas para la compensación de imágenes. Para la evaluación de nuestro algoritmo de estabilización realizamos una introducción de las principales métricas para estimar la estabilización de una secuencia de imágenes. Finalmente, se habló de la regresión polinomial como herramienta para encontrar el modelo de movimiento entre imágenes consecutivas del sistema de visión catadióptrico.

En el siguiente capítulo nos adentramos en los algoritmos para la reconstrucción de las imágenes omnidireccionales a panorámicas y en los diferentes trabajos en lo que se refiere a estabilización de imágenes, mostrando las diferentes etapas que deben de tener y haciendo una clasificación entre métodos directos y basados en características.

3 Trabajo relacionado

3.1 Métodos para la reconstrucción de imágenes catadióptricas.

En la actualidad, con el gran uso de las cámaras para diversas aplicaciones, se ha vuelto una necesidad que los sistemas maximicen su área de cobertura con el menor número de cámaras; todo esto para que se pueda reducir el costo y la complejidad de instalar y calibrar un gran número de cámaras. Sin embargo, las cámaras poseen un campo de visión limitado, por las características antes mencionadas, esto conlleva a un problema de instalación de sistemas que utilicen un gran número de cámaras.

Los sistemas catadióptricos son ideales para solucionar el problema que se mencionó anteriormente. Este tipo de sistemas usan un tipo especial de espejo para reflejar los rayos de luz de un entorno de 360° para formar una imagen omnidireccional. Las imágenes omnidireccionales tienen diferentes propiedades respecto a las imágenes tomadas con una cámara convencional, en específico la deformación de la imagen por la forma del espejo. Esta deformación hace que no puedan ser utilizadas las imágenes directamente de la cámara, sino que deben de pasar por un proceso de transformación. Esta transformación se refiere a pasar de una imagen omnidireccional a una imagen panorámica, esto se ejemplifica en la Figura 3.1.

Existen tres principales métodos para transformar las imágenes omnidireccionales a imágenes panorámicas, estos son: *Pano-mapping Table (PT)*, *Discrete Geometry Techniques (DGT)* y *Log-polar Mapping (LM)*[21]. A continuación explicamos con más detalle el método PT, el cual tiene diversas ventajas sobre los otros dos, que se mencionan más adelante.

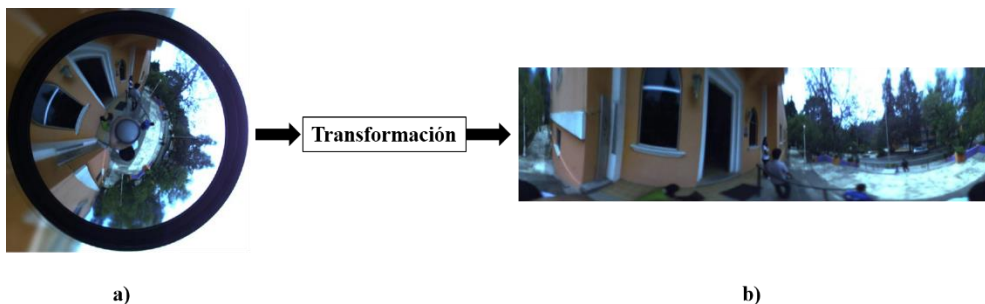


Figura 3.1. Transformación de a) una imagen omnidireccional a b) una imagen panorámica.

3.1.1 Pano-mapping Table (PT)

En este método existen 3 principales etapas [22], las cuales son: selección de puntos, creación de la tabla panorámica y creación de la imagen panorámica.

Selección de puntos

En esta etapa se eligen diferentes puntos de la imagen y sus respectivas proyecciones en el sistema coordenado del mundo real, dichos puntos deben de cumplir la característica que deben de ser fácilmente medibles e identificables tanto en el sistema coordenado de la imagen (p_k) como en el del mundo real (P_k). Jeng y Tsai [23] sugieren que se elijan al menos 5 puntos los que se tomen en cuenta para realizar la transformación de las imágenes omnidireccionales. El primer paso es configurar el sistema catadióptrico como se ilustra en la Figura 3.2.

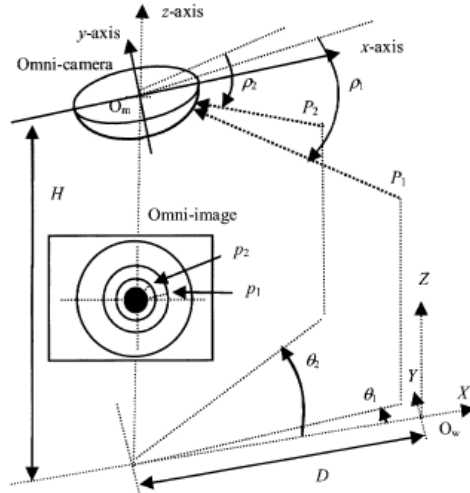


Figura 3.2. Configuración del sistema catadióptico para la transformación de imágenes, se observa las diferentes medidas necesarias para la creación de la tabla panorámica.

El sistema catadióptico de la Figura 3.2, debe ser configurado cumpliendo las siguientes condiciones:

- El origen del sistema coordenado del mundo real O_w puede ser cualquier punto en el mundo real que facilite las mediciones de los puntos.
- El sistema coordenado del mundo real está formado por: (X, Y, Z) .
- El origen del sistema coordenado catadióptico (el centro del espejo) O_m debe de estar colocado respecto a O_w en la posición $(-D, 0, H)$, estos

parámetros deben ser conocidos por el usuario que se encuentra realizando la transformación.

- El sistema coordenado catadióptrico está formado por: (x, y, z) .

También en la Figura 3.2 se muestra los ángulos de acimut (θ) y elevación (ρ). El ángulo de acimut se mide respecto al eje X del sistema coordenado del mundo real y el ángulo de elevación se mide respecto al plano formado por x-y del sistema coordenado catadióptrico.

Una vez que ya se configuro el sistema, se procede a medir los puntos deseados respecto al sistema coordenado del mundo real (X_k, Y_k, Z_k) y respecto al centro de la imagen (u_k, v_k) . El centro de la imagen puede ser fácilmente calculado mediante un método de calibración, como el explicado en el capítulo anterior.

Creación de la tabla panorámica

El primer paso para la creación de esta tabla es dividir el rango del ángulo de acimut, el cual comprende de 0 a 2π y se divide en N fragmentos. El rango del ángulo de elevación comprende entre ρ_i y ρ_f , el cual varía dependiendo del tipo de espejo que se utilice en el sistema, por lo general entre 0 y 90 grados. También este rango es dividido en M fragmentos. Por lo que tendremos una tabla T de tamaño $M \times N$.

Por la propiedad no lineal de la superficie del espejo, cada pixel de la imagen se modela como se muestra en la siguiente ecuación:

$$\begin{aligned}u &= f_r(\rho) \cos \theta \\v &= f_r(\rho) \sin \theta\end{aligned}$$

Donde $f_r(\rho)$ es conocida como una función de extensión radial, la cual modela la deformación del espejo radialmente y está definida de la siguiente manera:

$$f_r(\rho) = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4$$

Los coeficientes a_0 hasta a_4 serán calculados con la información extraída de los puntos seleccionados y mediante los siguientes pasos:

Paso 1:

Transformar los puntos del mundo real con coordenadas (X_k, Y_k, Z_k) a puntos con coordenadas respecto a O_m . Esto se logra restándole a todos los puntos los siguientes valores: $(-D, 0, H)$.

Paso 2:

Para cada par de puntos (p_k, P_K) , se calcula el ángulo de elevación ρ_k para P_K y la distancia radial r_k para p_k mediante las siguientes ecuaciones:

$$\rho_k = \tan^{-1} \left[\frac{Z_k}{D_k} \right]$$

$$r_k^2 = u_k^2 + v_k^2$$

Donde D_k está definida como la distancia en el plano x-y desde el punto de interés hasta el origen del sistema coordenado catadióptrico.

Paso 3

Sustituir todos los datos ρ_k y r_k para cada uno de los puntos seleccionados, con lo cual tendremos n ecuaciones simultaneas:

$$f_r(\rho_1) = a_0 + a_1\rho_1 + a_2\rho_1^2 + a_3\rho_1^3 + a_4\rho_1^4$$

$$f_r(\rho_2) = a_0 + a_1\rho_2 + a_2\rho_2^2 + a_3\rho_2^3 + a_4\rho_2^4$$

⋮
⋮
⋮

$$f_r(\rho_n) = a_0 + a_1\rho_n + a_2\rho_n^2 + a_3\rho_n^3 + a_4\rho_n^4$$

Para obtener los coeficientes, habrá que resolver el sistema de ecuaciones mediante un método numérico. Si el número de puntos seleccionados es mayor a 5, deberá escogerse un método para resolver sistemas de ecuaciones sobre determinados. Una vez que se obtienen los coeficientes, se llena la tabla con las respectivas coordenadas de la imagen obtenidas mediante las ecuaciones que modelan los pixeles de la imagen. La tabla panorámica queda formada como se muestra en la Tabla 3.1.

| | θ_1 | θ_2 | ⋅ ⋅ ⋅ | θ_N |
|----------|--------------------|--------------------|-------|--------------------|
| ρ_1 | (u_{11}, v_{11}) | (u_{12}, v_{12}) | ⋅ ⋅ ⋅ | (u_{1N}, v_{1N}) |
| ρ_2 | (u_{21}, v_{21}) | (u_{22}, v_{22}) | ⋅ ⋅ ⋅ | (u_{2N}, v_{2N}) |
| ⋮ | ⋮ | ⋮ | ⋅ ⋅ ⋅ | ⋮ |
| ρ_M | (u_{M1}, v_{M1}) | (u_{M2}, v_{M2}) | ⋅ ⋅ ⋅ | (u_{MN}, v_{MN}) |

Tabla 3.1. Tabla panorámica para la transformación de imágenes omnidireccionales.

Generación de la imagen panorámica

Dada la imagen omnidireccional y la tabla creada en la etapa anterior, podemos comenzar la creación de una imagen panorámica. Cada celda de la tabla panorámica es la coordenada de un pixel en la imagen omnidireccional, de esta posición se extrae el valor de color del pixel. El tamaño de la imagen panorámica será del mismo tamaño que de la tabla, en este caso $M \times N$ por lo que el color extraído tendrá como posición en la imagen panorámica los índices de la columna y el renglón de su posición en la tabla. Cabe recordar que cada entrada en la tabla no será siempre un número entero, entonces para obtener el valor del color del pixel adecuado se deberá usar una técnica de interpolación para obtenerlo.

Como se mencionó anteriormente, se eligió el método PT debido a que en los métodos para reconstruir imágenes omnidireccionales, es el que presenta diversas ventajas sobre los demás. Wong et al. [21] presentan una comparación de los diversos métodos para la reconstrucción en el cual las principales características que se comparan son las siguientes:

Resolución de la imagen generada

Los algoritmos LM y DGT son los que presentan menor tamaño de la reconstrucción de las imágenes dado que el tamaño de la imagen panorámica está definido por el tamaño del perímetro de la imagen omnidireccional, en cambio en el algoritmo PT la imagen panorámica puede ser de la resolución deseada por el usuario. Esto es debido a que el tamaño de la tabla especificada por el usuario será el tamaño de la imagen panorámica del sistema, por lo que manipulando el tamaño de la tabla se podrá obtener la resolución subpixel deseada que se propuso en este trabajo.

Calidad de imagen

El algoritmo PT es el que presenta menor distorsión de los objetos reales, es decir conservan su forma original. Este fenómeno se puede apreciar mejor en las líneas rectas, donde los algoritmos LM y DGT presentan una mayor curvatura de las líneas. En cambio el algoritmo PT aunque presenta distorsión de las líneas, es menor respecto a los otros dos algoritmos.

Complejidad del algoritmo

En la Tabla 3.2 se muestra la complejidad en ciertas operaciones de los diferentes algoritmos para la reconstrucción de las imágenes omnidireccionales. En la Tabla 3.2, X está definida como la longitud de la imagen panorámica y Y como la altura de la imagen panorámica. Como podemos observar en la tabla, tomando en cuenta

todas las operaciones podemos ver que el algoritmo PT presenta una menor complejidad que los demás algoritmos.

| | DGT | Log-polar mapping | Pano-mapping Table |
|----------------|-----------|-------------------|--------------------|
| Suma | $O(XY^2)$ | $O(X^2Y^2)$ | $O(Y^2)$ |
| Resta | | | |
| Multiplicación | $O(Y)$ | $O(X^2)$ | $O(Y^2)$ |
| Division | | | |

Tabla 3.2. Complejidad de los algoritmos de reconstrucción de las imágenes omnidireccionales.

3.2 Estabilización de imágenes

La estabilización de imágenes consiste básicamente en generar un video donde se han removido los movimientos indeseables como por ejemplo, la vibración de la mano que sostiene la cámara o el movimiento del vehículo donde va transportada la cámara [24]. El objetivo principal de la estabilización es retirar el movimiento de la secuencia de imágenes que no pertenece a ella.

Los algoritmos de estabilización pueden dividirse principalmente en tres principales fases [25]:

- Cálculo del movimiento. En esta etapa se realiza el cálculo de todos los movimientos en una imagen respecto a una imagen anterior en la secuencia temporal de imágenes.
- Filtrado del movimiento. En esta parte se filtran los movimientos que no son generados por el movimiento de la cámara, como pueden ser objetos entrando a la escena, cambios de iluminación, etc. Los movimientos que se calcularon en la etapa anterior se dividen en dos categorías [26]. Uno es el movimiento global, este tipo corresponde a los movimientos que son parte del sistema en el que se encuentra la cámara, como son los que provocan la vibración de la mano o algún otro tipo de vibración. El otro tipo de movimiento es llamado local, es aquel que es parte de la imagen, como pueden ser personas en movimiento, autos o alguna escena cambiante. Como podemos observar, el tipo de movimiento que se desea eliminar es el global.
- Compensación. En esta etapa, mediante los vectores de movimiento global se trata de encontrar la mejor transformación que modele dichos vectores para que con la función inversa de la transformación se proceda a realizar la compensación de la imagen.

Para el cálculo del movimiento existen básicamente dos tipos de algoritmos: directos y basados en características. Los algoritmos directos se basan en dividir la imagen en bloques y buscar cada bloque en la imagen subsecuente y generar un vector de movimiento para cada bloque, la principal desventaja de este tipo de métodos es que los objetos tienden a sufrir pequeñas transformaciones que dificultan su búsqueda en las imágenes subsecuentes [27]. Los algoritmos basados en características tienden a buscar en cada imagen cierto tipo de características especiales como bordes o esquinas por lo que las características en cada imagen son casi invariantes, una vez encontradas dichas características se buscan en la imagen siguiente. La desventaja de los algoritmos basados en características es que computacionalmente son más complejos pero son capaces de encontrar los objetos aunque sufran deformaciones [28].

La complejidad de los algoritmos de estabilización consiste básicamente en discernir cuales movimientos son inherentes al movimiento de la cámara y cuales son parte de una escena cambiante, como por ejemplo personas, autos, animales en movimiento. Otro gran problema al que se presentan estos algoritmos son a los cambios de iluminación, ya que estos cambios se pueden percibir como movimientos y por lo tanto se podría generar una compensación innecesaria [29].

3.3 Métodos de estabilización directos

Los métodos de estabilización directos son los más utilizados, con una implementación sencilla y con resultados de alta calidad. En este método se asumen que los movimientos sólo son de traslación y no se producen cambios de iluminación.

En los algoritmos de estabilización directos cada una de las imágenes de la secuencia de video se divide en bloques rectangulares. Este método detecta el movimiento entre dos imágenes basándose respecto a los bloques que las constituyen. Los bloques de la imagen actual serán comparados con los bloques de la imagen de referencia, creando un vector que une ambas regiones, esto se representa en la Figura 3.3.

Cuando se determina la máxima similitud entre bloques entre todos los candidatos dentro de la ventana de búsqueda, se realiza la correspondencia entre bloques. Si el bloque elegido no se encuentra en la misma posición en ambas imágenes, significa que se ha movido. La distancia que hay entre ambos bloques es el vector de movimiento estimado, este se le asigna a todos los píxeles de un mismo bloque.

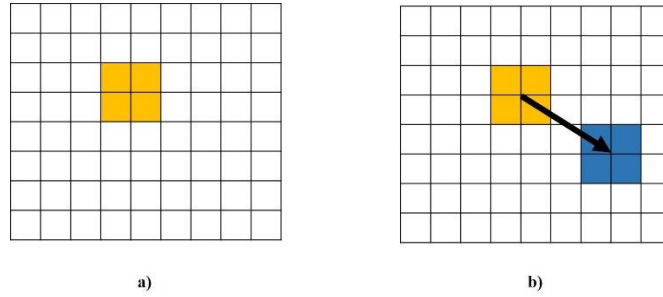


Figura 3.3. Algoritmo de estimación de movimiento basado en bloques a) imagen de referencia, b) imagen actual y su respectivo vector generado por la búsqueda del bloque.

En estos algoritmos, la principal diferencia que tienen entre ellos es la forma en como detectan el movimiento entre imágenes consecutivas. A continuación se detallara los algoritmos más significativos de este tipo.

En los trabajos de Kwon et al. [30] y Yan et al. [31] utilizan el algoritmo de correlación de fase para calcular el movimiento entre imágenes, dicho algoritmo está basado en la transformada de Fourier. La correlación de fase nos brinda un método únicamente para calcular el movimiento traslacional entre dos imágenes.

Supongamos que tenemos dos imágenes consecutivas, f_k y f_{k+1} , que se encuentran relacionadas mediante desplazamientos d_1 y d_2 , esto se representa en la siguiente ecuación:

$$f_k(x, y) = f_{k+1}(x + d_1, y + d_2)$$

Por la propiedad de corrimiento de la transformada de Fourier sabemos que un corrimiento en el dominio espacial de dos imágenes tiene como consecuencia una diferencia lineal en fase en el dominio de la frecuencia de las transformadas de Fourier. La transformada de la ecuación queda de la siguiente forma:

$$F_k(x, y) = F_{k+1}(x, y)e^{-i(xd_1+yd_2)}$$

La correlación de fase está definida como el espectro de potencia cruzada normalizada (*normalized cross power spectrum*) entre f_k y f_{k+1} , la cual está definida por la siguiente ecuación:

$$G(x, y) = \frac{F_{k+1}F_k^*}{|F_{k+1}F_k^*|} = e^{-i(xd_1+yd_2)}$$

Si f_k y f_{k+1} son funciones continuas, entonces la transformada inversa de Fourier de $G(x, y)$ será una función delta centrada en el desplazamiento (d_1, d_2) de f_{k+1} respecto a f_k . En la Figura 3.4 se ejemplifica la transformada inversa de $G(x, y)$.

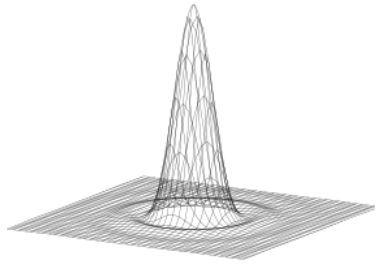


Figura 3.4. Caso ideal de la función delta centrada en desplazamiento del movimiento.

En la Figura 3.4, se muestra el desplazamiento mediante la función delta, dicha función se encuentra perfectamente centrada en el desplazamiento de las imágenes. Sin embargo en casos reales, la función delta del desplazamiento mostrara un comportamiento como en la Figura 3.5, donde presentara ruido y la función delta no estará completamente centrada, por lo cual habrá que hacer algún tipo de filtrado para encontrar el desplazamiento correcto.

Por otra parte, en los algoritmos que están basados en el movimiento por bloques, la imagen se divide en bloques rectangulares que no están traslapados entre ellos y son de tamaño fijo. Se asume en este problema que sólo se producen traslaciones, siendo el modelo más usado en otras aplicaciones como en compresión de vídeo, principalmente por su facilidad de implementación sobre hardware. No obstante, presenta problemas ante movimientos más generales como podría ser una transformación afín que incluya escalado y/o rotación.

La conjetura sobre la que se sustentan estos métodos es que un bloque de tamaño $M \times N$ (donde M y N pueden ser del mismo tamaño) en la imagen k centrado en el punto (x_0, y_0) coincide con un bloque del mismo tamaño en la imagen $k+1$ con un desplazamiento de por medio. Esto expresado mediante una ecuación, se expresa de la siguiente forma:

$$k(x_0, y_0) = k + 1(x_{0+a}, y_{0+b})$$

Dicho de otra manera, cada uno de los bloques que componen cada una de las imágenes de la secuencia de vídeo posee un equivalente, esto es, un bloque del

mismo tamaño que representa la misma porción de la escena en otra imagen consecutiva de la secuencia de vídeo.

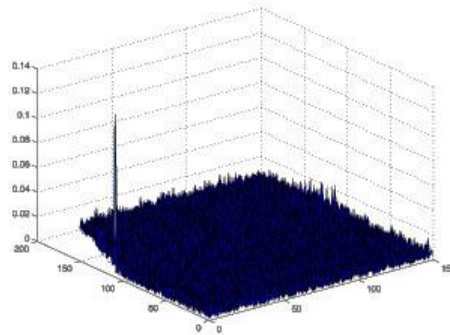


Figura 3.5. Función delta del movimiento de una imagen, mostrando el desplazamiento de la imagen y el ruido que contiene. Las coordenadas del pico más alto son el desplazamiento que sufre la imagen.

El procedimiento que se sigue en este tipo de métodos de estimación de movimiento es dividir la imagen actual de la secuencia de vídeo en una matriz de bloques y luego buscar este mismo bloque en las siguientes imágenes consecutivas del vídeo. Los tamaños de bloque que suelen considerarse típicamente se encuentran en torno a 16x16 píxeles. La zona de búsqueda queda determinada por un parámetro p denominado parámetro de búsqueda. El significado de este parámetro queda ilustrado en la Figura 3.6.

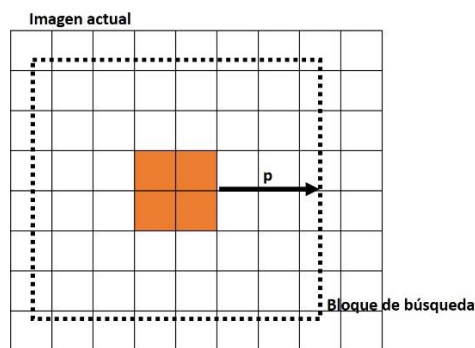


Figura 3.6. Bloque de búsqueda para los algoritmos directos de estabilización, en la imagen actual y su parámetro p .

También existen otros métodos que están basados en el algoritmo de flujo óptico, el cual tiene algunas variantes como pueden ser los métodos de Horn – Schunck [32] o el método Lucas – Kanade [33]. Estos algoritmos hacen uso de varias suposiciones para calcular el movimiento en imágenes consecutivas, como son, los objetos en la imagen son rígidos y la iluminación es constante [34]. Las características anteriores

son difíciles de cumplir y asociado a esto, el algoritmo de flujo óptico es computacionalmente complejo, por lo cual este tipo de algoritmos para la estabilización no son utilizados con frecuencia.

3.4 Métodos de estabilización basados en características.

Los métodos basados en características tienen como principal aportación el uso de descriptores de puntos característicos tales como SIFT y SURF para cuantificar el movimiento entre imágenes. Los algoritmos que emplean dichos descriptores son los que presentan mejores incrementos en el índice de PSNR de las secuencias de imágenes; esto debido principalmente a que los descriptores son robustos a diversas transformaciones como rotación, traslación y escalamiento, también son robustos a los cambios de iluminación. A continuación detallaremos algunos de los trabajos más importantes que utilizan los descriptores de puntos característicos.

En el trabajo de Battiato et al. [35], se aplica el descriptor SIFT entre imágenes consecutivas para obtener un vector de movimiento local para cada punto característico. Para realizar la correspondencia entre puntos, utilizan la distancia euclidiana con un umbral de 0.6, en lugar del 0.8 utilizado en el algoritmo original de SIFT. Utilizando ese umbral, los autores garantizan que eliminan las correspondencias erróneas entre los pares de puntos. El modelo de las vibraciones lo representa mediante una rotación, un escalamiento y una traslación en los dos ejes de la imagen.

La estabilización de imágenes desarrollada en el trabajo de Yao et al. [36] utiliza el descriptor SIFT para encontrar los puntos característicos que son comunes entre imágenes, después utiliza el algoritmo RANSAC[37] para eliminar los valores que no son parte del conjunto que representa el movimiento entre imágenes. En este trabajo, poseen una etapa más para el refinamiento de la estimación de movimiento que proporciona el algoritmo RANSAC; hacen uso del filtro de partícula para eliminar el error acumulativo que acarrearán los valores que no pertenecen al sistema.

El trabajo realizado por Zhou [38] toma como punto de partida las ventajas que presenta el algoritmo SURF respecto a SIFT para extraer los puntos característicos entre imágenes, como es la velocidad de procesamiento. En la parte que corresponde al modelo, hacen uso de una transformación afín que utiliza 4 parámetros para describir la transformación que sufre la imagen por las vibraciones. El algoritmo RANSAC es usado al igual que en otros trabajos, para eliminar los valores que no pertenecen al sistema. Además utilizan la cantidad de movimiento compensado entre las 2 imágenes anteriores para hacer un promedio con el movimiento de la imagen actual y así obtener un resultado más confiable.

El método propuesto por Li y Liu [39] hacen uso del descriptor SIFT para rastrear los puntos entre imágenes y para el modelo de la compensación solo hacen uso de rotación, traslación y escalamiento porque consideran que las transformaciones afín y proyectiva son difíciles de observar en las imágenes, debido al corto tiempo entre imágenes. En la eliminación de valores ajenos del sistema también utilizan el algoritmo RANSAC como en otras implementaciones. Otra parte novedosa en este algoritmo es la utilización de algunas secciones de la imagen como se muestra en la Figura 3.7, esto se hace con el objetivo de reducir el tiempo de procesamiento, debido a que SIFT es la parte más costosa en este rubro.

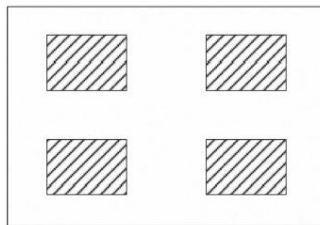


Figura 3.7. Zonas utilizadas en la imagen utilizadas en[39] para realizar la estabilización de imágenes.

Como podemos ver en los trabajos anteriormente presentados, existen básicamente 3 etapas para realizar la estabilización de imágenes mediante un descriptor de puntos característicos, las etapas son las siguientes:

1. Extracción de puntos característicos entre imágenes consecutivas mediante algún descriptor como SIFT, SURF, BRIEF, etc.
2. Eliminación de los puntos que no pertenezcan a las vibraciones que afectan a las imágenes del sistema, en este paso los algoritmos más usados son RANSAC y mínimos cuadrados.
3. Compensación de la imagen mediante la elección previa de un modelo de la transformación que sufren las imágenes debido a las vibraciones.

En los métodos que se mencionaron previamente, la evaluación de los algoritmos se hace mediante el PSNR de las imágenes, sin embargo es difícil realizar una comparativa entre ellos debido a la naturaleza de las imágenes, dado que en algunas imágenes se presentara un incremento mayor de PSNR que en otras. Todos los algoritmos presentan un incremento entre el PSNR de la secuencia estabilizada respecto a la secuencia original pero el porcentaje puede variar dependiendo de la configuración de la imagen. Pongamos como ejemplo una secuencia de una persona con un fondo de color negro, donde el fondo es la parte que más abarca en la imagen, si aplicamos cualquier algoritmo de estabilización obtendríamos como resultado un índice muy pequeño de incremento del PSNR. Por otro lado, con una

imagen regular podríamos obtener un índice mayor. La única constante en los algoritmos que presentamos, es el incremento del índice de PSNR en cada una de las secuencias a las que aplicaron sus respectivos algoritmos.

3.5 Resumen

En el desarrollo de este capítulo se habló primero de los algoritmos para la reconstrucción de imágenes omnidireccionales, los cuales se agrupan principalmente en 3 grupos: *Pano-mapping Table (PT)*, *Discrete Geometry Techniques (DGT)* y *Log-polar Mapping (LM)*[21]. Después de haber analizado varias características de estos algoritmos, el algoritmo PT fue el elegido para la transformación de imágenes omnidireccionales en este trabajo de investigación. En la siguiente parte se profundizó en la estabilización de imágenes y las etapas que deben de contener dichos algoritmos, siendo principalmente: estimación de movimiento, filtrado del movimiento y compensación de la imagen. También se aborda la clasificación de estos algoritmos, directos y basados en características, siendo estos últimos sobre los que se basa nuestro trabajo de investigación debido a su robustez a diferentes transformaciones y condiciones.

En el próximo capítulo analizamos a detalle todas las etapas implementadas en el desarrollo de nuestro algoritmo de estabilización, tanto las etapas tradicionales de las que hablamos anteriormente como de las etapas que agregamos para lograr la estabilización en sistemas catadióptricos.

4 Método Propuesto

4.1 Descripción general del método

En anteriores capítulos hemos hablado de las etapas que contienen los algoritmos de estabilización, las cuales se pueden resumir en las siguientes:

1. Rastreo de puntos característicos entre imágenes consecutivas mediante algún descriptor de puntos.
2. Eliminación de puntos característicos que no pertenecen a las vibraciones que afectan al sistema de visión catadióptrico.
3. Compensación de las vibraciones mediante el modelo del sistema previamente seleccionado para describir las vibraciones.

El método desarrollado en este trabajo de investigación tiene como etapas las que se describen a continuación:

1. Rastreo de puntos en imágenes omnidireccionales consecutivas mediante el descriptor SURF.
2. Eliminación de puntos ajenos a las vibraciones mediante el historial de puntos que se mantienen activos a lo largo de las imágenes, es decir, solo utilizamos los puntos que se mantienen en el mayor número de imágenes.
3. Aproximación del modelo de movimiento mediante el ajuste de un polinomio a los datos del historial de puntos característicos.
4. Compensación de la imagen omnidireccional mediante el modelo rotacional.
5. Transformación de la imagen omnidireccional a una imagen panorámica.
6. Transformación de los puntos rastreados en las imágenes omnidireccionales al sistema coordinado de las imágenes panorámicas.
7. Aproximación del modelo de movimiento traslacional mediante el ajuste de un polinomio a los datos restantes del historial de puntos característicos.
8. Compensación de la imagen panorámica mediante un modelo de traslación en los dos ejes coordinados.
9. Reducción de la imagen para eliminar las zonas en donde se aplicó la compensación.

Como podemos observar, nuestro algoritmo también cumple con las tres etapas que debe de tener cualquier algoritmo de estabilización, para el rastreo de puntos utilizamos el descriptor SURF, en la etapa de selección de puntos utilizamos el

historial de puntos y una regresión polinomial, por ultimo utilizamos un modelo de compensación tanto en la imagen omnidireccional como en la imagen panorámica. La principal contribución de nuestro trabajo de investigación es un algoritmo de estabilización de imágenes exclusivamente diseñado y basado en las características de este tipo de sistemas utilizando una precisión subpixel, ya que como se mencionó previamente, no existen trabajos relacionados en el campo de estabilización de imágenes hacia este tipo de sistemas.

En la Figura 4.1 se muestran los pasos y las diferentes transformaciones que se realizan en nuestro algoritmo para la compensación entre 2 imágenes consecutivas.

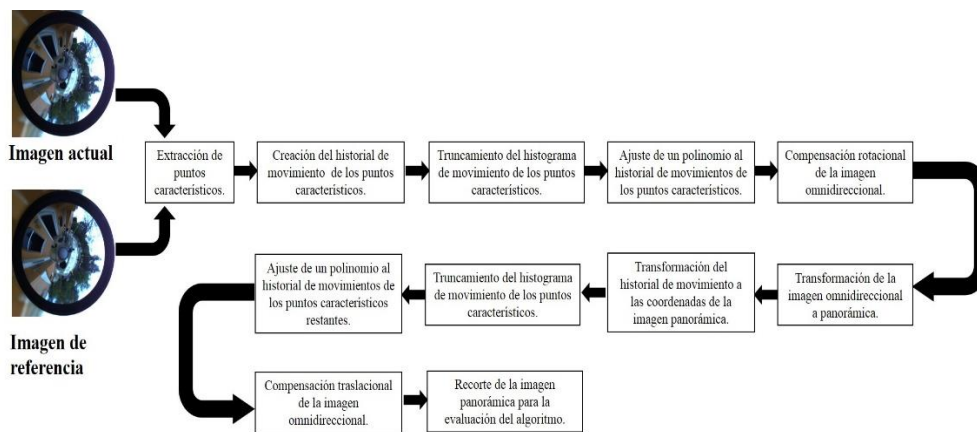


Figura 4.1. Esquema del algoritmo propuesto para la estabilización de 2 imágenes consecutivas en sistemas catadióptricos.

En las siguientes secciones profundizamos en las partes más importantes que se implementan en nuestro algoritmo de estabilización.

4.2 Rastreo de puntos

En este paso de nuestro método se trata de localizar puntos que sean constantes entre las imágenes para poder usarlos como referencia en el cálculo del movimiento de las imágenes. La estrategia que se utiliza en esta etapa es la siguiente:

1. Partimos de una imagen inicial llamada I_0 y la imagen subsecuente en la secuencia llamada I_1 , a cada una de las imágenes se les aplica el descriptor SURF para encontrar sus puntos característicos. Una vez que se tienen los puntos característicos de ambas imágenes se procede a discriminar aquellos que no cumplen cierto umbral establecido respecto a la distancia euclidiana

entre cada punto. Los puntos característicos que son iguales en ambas imágenes se almacenan en una tabla, P, y la posición de dichos puntos respecto a cada imagen se almacena en una tabla, D, esto se ejemplifica en la Figura 4.2.

| D | |
|------------|------------|
| I_0 | I_1 |
| $P_1(x,y)$ | $P_1(x,y)$ |
| . | . |
| . | . |
| $P_n(x,y)$ | $P_n(x,y)$ |

| P | |
|-------|-------|
| I_0 | I_1 |
| P_1 | P_1 |
| . | . |
| . | . |
| P_n | P_n |

Figura 4.2. Tablas de puntos rastreados entre dos imágenes consecutivas y sus respectivas coordenadas.

2. El siguiente paso es encontrar los puntos característicos O de la siguiente imagen en la secuencia de imágenes, en este caso sería la imagen I_2 . También se almacena la ubicación de O en I_2 , la cual es llamada M.
3. Los puntos O encontrados en el paso anterior se comparan con la última columna de la tabla P, se vuelve a utilizar la distancia euclidiana para discriminar los puntos característicos que son similares.
4. Una vez encontrados los puntos iguales entre O y la última columna de P, se procede a eliminar los renglones de los puntos que no son similares entre la última columna de P y O, también se eliminan los renglones de D y M usando como referencia los índices de los renglones que se eliminaron de P y O. Después de eliminar los renglones, se procede a concatenar P con O y también D con M. Este procedimiento se ilustra mejor en la Figura 4.3.

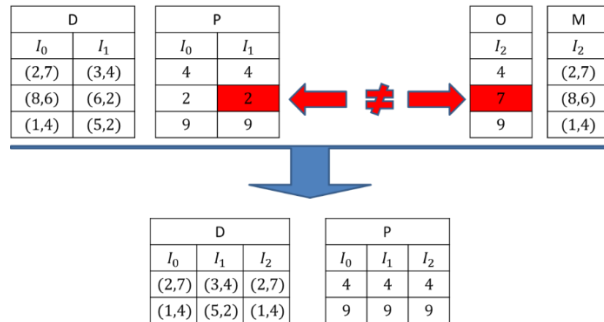


Figura 4.3. Concatenación y eliminación de puntos característicos en el historial de puntos.

5. Los pasos 2, 3 y 4 se repiten con las siguientes imágenes de la secuencia hasta que se alcanza un número mínimo de puntos característicos en la

tabla P. Si este mínimo de puntos es alcanzado se pasa la tabla P al siguiente paso que es el de modelado del movimiento para la compensación.

6. Dado que las tablas D y P son transferidas a la siguiente etapa, volvemos a generar dos nuevas tablas para el resto de las imágenes tomando como referencia o imagen inicial, la última imagen de la cual fueron extraídos los puntos característicos. En otras palabras, se vuelve a repetir los pasos 1, 2, 3, 4 y 5 para el resto de la secuencia de imágenes.

El procedimiento descrito anteriormente nos asegura que los puntos con una breve duración temporal son eliminados, con esto se eliminan aquellos puntos que pueden pertenecer a pequeños cambios de iluminación, objetos entrantes o salientes en la imagen. En la Figura 4.4 se observa los puntos SURF que quedan después de realizar el proceso que se describió anteriormente.

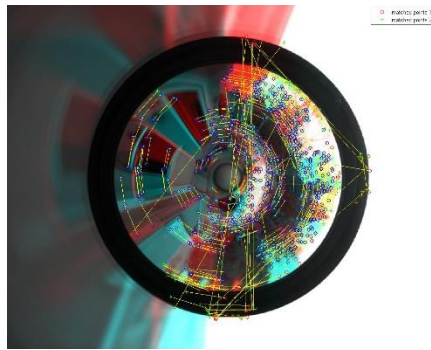


Figura 4.4. Rastreo de puntos SURF entre dos imágenes consecutivas.

4.3 Manejo de vectores de movimientos ajenos al movimiento del sistema

El siguiente paso en nuestro método implementado consiste en eliminar aquellos movimientos que no pertenecen a las vibraciones, que perturban nuestro sistema o que son parte del medio ambiente de la escena, como pueden ser personas u objetos en movimiento. El primer paso para eliminar estos puntos fue mediante el rastreo de puntos, debido a las siguientes razones:

- Si existieran personas u objetos en movimiento en la secuencia de imágenes, el rastreo de puntos los eliminaría si salieran de la imagen dado que ese punto ya no lo encontraría en las siguientes imágenes.

- Si los objetos en movimiento dentro de la imagen no desaparecen de la secuencia, estadísticamente representaría un porcentaje pequeño lo cual no afectaría el cálculo para la compensación del movimiento. Aunque estos valores no representan una mayoría, más adelante se presenta un método para eliminar estos valores.
- Los puntos que predominan en las tablas P y O son en su mayoría puntos que corresponden al fondo de la imagen, eso debido a que por lo general el fondo de la imagen se mantiene constante a menos que sea ocluido por algún objeto. Esta característica hace que las tablas puedan ser fácilmente filtradas entre los vectores que pertenecen al movimiento de la cámara y los que no pertenecen.

A partir de las características anteriormente descritas, procedemos a separar los valores que no corresponden al movimiento de la cámara. A partir de ahora la tabla que más nos interesa es la tabla P, esto debido a que contiene las posiciones de los puntos característicos a lo largo de la secuencia de imágenes y con esto podemos hacer una aproximación del comportamiento de las vibraciones, que podemos nombrar como historial de movimiento. Mediante la tabla D, se forma una nueva tabla llamada E que estará formada por la diferencia entre las columnas pertenecientes a D, esto se ejemplifica en la Figura 4.5.

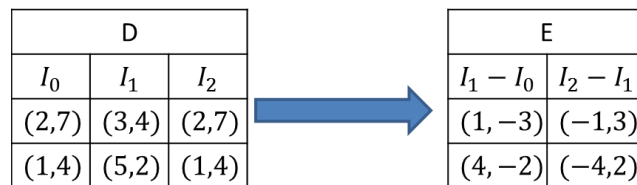


Figura 4.5. Ejemplo de creación de historial de movimiento de una secuencia de imágenes.

Con este procedimiento, lo que obtenemos es la diferencia de posición de los puntos característicos respecto a una imagen anterior. Si tomamos cada renglón de E, podemos obtener un historial de como se ha movido ese punto en los ejes X e Y en la secuencia de imágenes. Para eliminar aquellos valores atípicos, usaremos dos pasos, los cuales serán:

1. Creación de histograma
2. Eliminación de límite inferior y superior del histograma.

El primer paso consiste en crear el histograma de cada columna de la tabla E, con este histograma se obtiene la distribución de movimiento de los puntos característicos de una imagen actual respecto a una imagen de referencia. Una vez

que tenemos el histograma, lo que prosigue es eliminar un porcentaje de los límites inferior y superior del histograma, suponiendo una distribución gaussiana de los puntos, como se muestra en la Figura 4.6. El porcentaje que se elimina es de aproximadamente el 10%, la elección de este valor se discutirá más a fondo en la sección de resultados.

El histograma con los valores eliminados nos servirá para calcular la media del movimiento entre imágenes. La principal contribución de haber eliminado dicho porcentaje de valores es la de reducir el valor de la desviación estándar del histograma, con lo cual se podrá obtener una mejor aproximación del movimiento entre imágenes.

Una vez que obtenemos el histograma con los valores pertenecientes al movimiento que generan las vibraciones, tratamos de ajustar los datos a un polinomio de grado n . El polinomio que mejor ajusta los datos del histograma de los movimientos es el polinomio de tercer grado, no se utiliza otro polinomio de mayor grado principalmente por que a medida que aumentamos el grado del polinomio el sistema de ecuaciones tiende a ser más inestable, y es más probable que más adelante existan oscilaciones. La elección del grado del polinomio se discute con más profundidad en el capítulo 5.

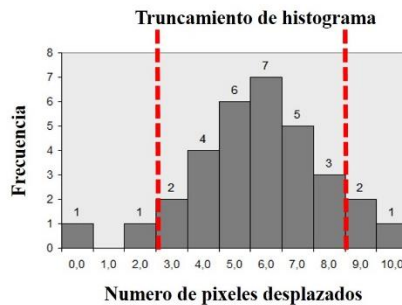


Figura 4.6. Truncamiento de histograma de movimiento para la eliminación de valores ajenos al modelo de las vibraciones.

4.4 Precisión subpixel

El siguiente paso es crear una imagen con precisión subpixel para poder hacer la compensación de la imagen de una manera más exacta y por lo tanto obtener un mayor índice de PSNR entre imágenes consecutivas. El proceso para obtener la imagen con precisión subpixel consiste básicamente en realizar una interpolación entre píxeles hasta alcanzar la resolución deseada. Las resoluciones subpixel son siempre potencias negativas de 2: $\frac{1}{2}$, $\frac{1}{4}$, etc.

El proceso para obtener una imagen de resolución subpixel es el siguiente. Primero, supongamos que tenemos dos pixeles cualquiera como en la Figura 4.7. A partir de estos pixeles se realiza la división por la mitad del segmento que los separa, como se ejemplifica en la misma figura, mediante interpolación se encuentra el valor en color de ese pixel. Al finalizar este paso se ha obtenido una resolución subpixel de $\frac{1}{2}$.

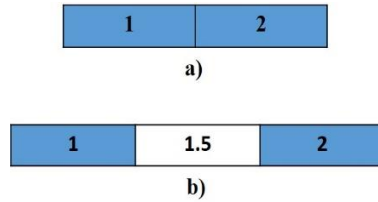


Figura 4.7. Segmento de una imagen para ejemplificar a) resolución original de la imagen, b) resolución subpixel $\frac{1}{2}$

Para lograr el siguiente nivel de resolución subpixel, se vuelve a realizar el mismo procedimiento de dividir a la mitad el segmento que divide a los pixeles y subpixeles y también mediante interpolación se encuentra su valor de color. El siguiente nivel de resolución sería el de $\frac{1}{4}$ y quedaría representado como se ilustra en la Figura 4.8. Este proceso de incrementar la resolución de la imagen se aplica a toda la imagen tomando las dos dimensiones, tanto columnas y renglones.

En cuanto a niveles subpixel, podemos hacer el mismo proceso hasta llegar al nivel deseado. Cabe mencionar que en el campo de procesamiento de imágenes los niveles más usados son los de $\frac{1}{4}$ y $\frac{1}{8}$, este último representado como en la imagen 32.

También debemos tener en cuenta que cada nivel subpixel que aumentemos, estaremos incrementando al doble el tamaño de la imagen original por ejemplo si tenemos una imagen y escogemos un nivel subpixel de $\frac{1}{4}$, la imagen final será 4 veces más grande que la original.

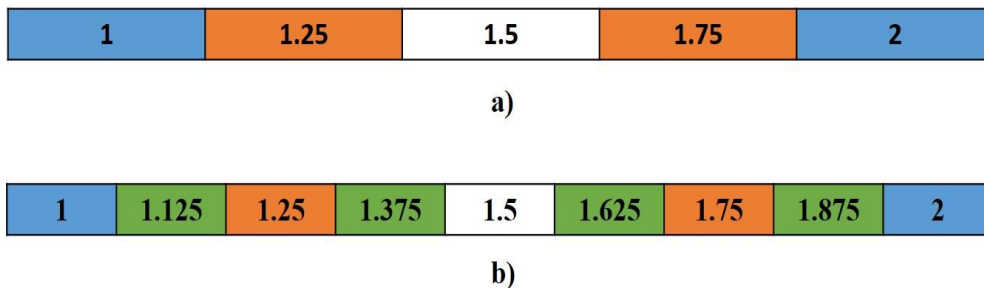


Figura 4.8. Segmento de una imagen para ejemplificar a) resolución subpixel $\frac{1}{4}$ y b) resolución subpixel $\frac{1}{8}$.

La finalidad de crear una imagen con resolución subpixel tiene que ver con la compensación del movimiento, ya que los puntos que obtenemos mediante el descriptor SURF poseen una precisión subpixel entonces el cálculo para compensar el movimiento de las imágenes también debe de presentar una precisión subpixel. Por lo tanto, si usamos una precisión subpixel para compensar el movimiento, obtendremos un error menor en el cálculo del movimiento y por consiguiente el índice de PSNR entre imágenes consecutivas incrementara. Digamos por ejemplo que usamos la imagen original para crear el modelo de compensación, este modelo tendrá como error máximo un rango entre ± 0.5 *pixeles*, sin embargo si usamos un nivel de resolución de $\frac{1}{8}$, el error máximo de la compensación sería de ± 0.0625 *pixeles*, con lo cual al momento de realizar la compensación del movimiento será más precisa y el índice de PSNR aumentara debido a la mayor igualdad entre pixeles.

4.5 Compensación Rotacional

El paso descrito en la sección anterior se aplica a la secuencia de imágenes omnidireccionales, pero previamente las coordenadas de los puntos SURF se transforman a coordenadas polares, para que podamos obtener el movimiento representado mediante un ángulo y una magnitud. La transformación de las coordenadas se hace respecto al centro de la imagen, que se obtuvo previamente mediante la calibración del sistema catadióptrico. Las razones anteriores son el motivo por el cual primero realizamos la compensación rotacional, ya que utilizando la forma de la imagen omnidireccional (circular), podemos compensar la imagen mediante una simple rotación respecto a su centro representando previamente los pixeles en otro sistema coordenado.

Siguiendo el procedimiento, obtendríamos dos polinomios que describen el movimiento angular de la imagen y otro que describe el movimiento mediante una magnitud. En nuestro caso, solo utilizaremos el polinomio que describe el movimiento angular respecto al centro de la imagen, para realizar la compensación utilizamos una transformación de rotación, como se describió en el capítulo 2. La compensación rotacional entre 2 imágenes se ilustra en la Figura 4.9.

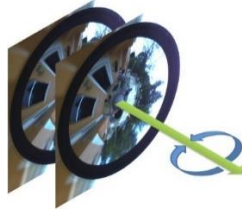


Figura 4.9. Representación de la compensación rotacional entre dos imágenes consecutivas.

Una vez que se ha compensado la imagen rotacionalmente, se procede a actualizar los puntos a las nuevas coordenadas de la imagen panorámica compensada, con el objetivo de que estos puntos sirvan más adelante para la etapa de la compensación traslacional.

4.6 Transformación de la imagen

Como hemos mencionado anteriormente, las imágenes omnidireccionales están representadas como se ve en la Figura 4.10. La porción visible del sistema catadióptrico está definida por el área definida entre los círculos formados por los radios r_1 y r_2 , dicha región es la única de interés para la reconstrucción de la imagen. Los radios r_1 y r_2 pueden ser definidos mediante el proceso de calibración, estos radios definen el área de la imagen donde el espejo refleja todavía rayos de luz hacia el sensor de la cámara.



Figura 4.10. Radios de las zonas de interés en imágenes omnidireccionales.

Si reconstruimos la porción de la imagen omnidireccional comprendida por r_1 y r_2 , obtendríamos una imagen representada como en la Figura 4.11. Dicha imagen tendría huecos representados en negro, los cuales son generados debido a que la imagen omnidireccional no presenta una resolución uniforme. Debido a este problema, para llenar los huecos se hace uso de la interpolación bicúbica, con este método logramos obtener una imagen con resolución uniforme como se muestra en la Figura 4.11.

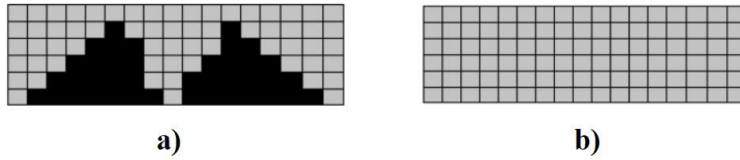


Figura 4.11. a) Imagen panorámica sin resolución uniforme, b) imagen panorámica con resolución uniforme mediante interpolación.

El tamaño de la imagen panorámica se encuentra limitado tanto por la altura como por el ancho. La altura de la imagen está determinada por la diferencia entre los radios r_1 y r_2 , esto queda expresado en la siguiente ecuación:

$$\text{Altura de la imagen panorámica} = r_1 - r_2$$

Para el caso del ancho de la imagen panorámica, está determinado por el perímetro del círculo con radio r_1 debido a que en este nivel es donde se presenta una resolución uniforme. Lo anterior queda expresado en la siguiente ecuación;

$$\text{Ancho de la imagen panorámica} = 2\pi r_1$$

Los puntos rastreados en el primer paso de nuestro algoritmo también son transformados a las coordenadas de la imagen panorámica, esto con el objetivo de que puedan ser utilizados más adelante en el cálculo de la compensación traslacional.

4.7 Compensación Traslacional

Una vez realizada la transformación de la imagen omnidireccional a una imagen panorámica, procedemos a realizar la compensación de la imagen panorámica en sus dos ejes coordenados. Hasta este paso de nuestro algoritmo hemos rastreado, filtrado y transformado los puntos característicos para realizar la compensación de la imagen en la etapa rotacional, ahora toca el turno de realizar la estabilización en una etapa de traslación.

La compensación traslacional comienza con las tablas de los puntos característicos rastreados en las etapas anteriores, cada tabla representa el movimiento de un punto característico respecto a una imagen anterior y existe una tabla para cada eje coordenado de la imagen. A cada tabla ya le han sido eliminados los valores que no corresponden a las vibraciones que afectan al sistema catadióptrico.

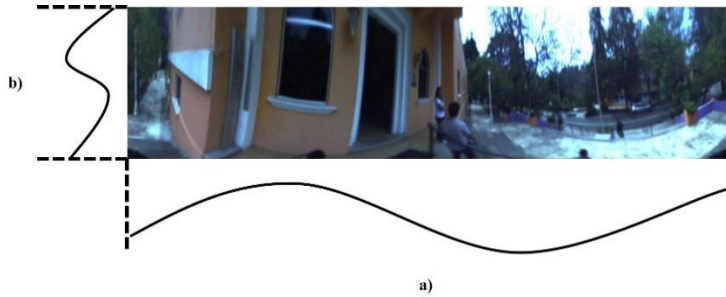


Figura 4.12. a) Polinomio para la compensación de las columnas en la imagen panorámica, b) Polinomio para la compensación de los renglones en la imagen panorámica.

El proceso que utilizamos para la compensación rotacional lo volvemos a utilizar, esta vez ajustamos un polinomio a los datos de movimiento para cada eje coordenado. Obtendremos dos polinomios que describen el movimiento de la imagen a través del eje X e Y de la imagen; dichos polinomios son utilizados para mover todos los renglones y columnas de las imágenes como se ilustra en la Figura 4.12. El proceso de volver a ajustar los polinomios a las tablas de los puntos rastreados se vuelve a realizar para eliminar aquellos valores que pudieron haber pasado la etapa de filtrado en la compensación rotacional.

4.8 Reducción de la imagen

En esta parte de nuestro algoritmo eliminamos aquellas zonas en las cuales la imagen sufrió un desplazamiento como parte de la compensación traslacional. Esta reducción de la imagen se ilustra en la Figura 4.13.

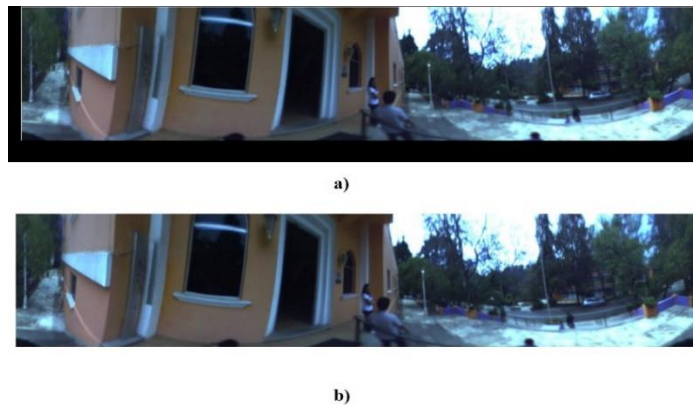


Figura 4.13. a) Imagen compensada mostrando las zonas donde fue compensada (zonas en negro) b) imagen sin las zonas de compensación.

Como podemos observar en la Figura 4.13, eliminamos aquellas zonas (en negro) donde la imagen tuvo que desplazarse para compensar el movimiento generado por las vibraciones en el eje Z del sistema catadióptrico. Esta reducción de la imagen sólo tiene como fin un mejoramiento estético de la imagen, en ningún momento mejora el rendimiento de nuestro algoritmo. Esta parte es importante para poder realizar una adecuada evaluación del algoritmo.

4.9 Resumen

En este capítulo se desglosa las partes más importantes del desarrollo e implementación de nuestro algoritmo de estabilización para sistemas catadióptricos. Primero se dio un esbozo general del método con sus respectivos pasos y transformaciones que sufren las imágenes a lo largo de nuestro algoritmo. En la siguiente sección se analizó la forma en que se rastrean los puntos característicos SURF a lo largo de la secuencia de imágenes, para después formar el historial de puntos. El historial de puntos es una parte fundamental de nuestro algoritmo, dado que es una fuente para modelar el movimiento al que se somete el sistema y también tiene la función de filtrar aquellos valores que no tienen una duración temporal alta y que pertenecen a objetos que no forman parte del movimiento global de la imagen. El siguiente paso en nuestro algoritmo fue darle una precisión subpixel a la imagen para reducir el error del algoritmo en las etapas de compensación de la imagen. Las siguientes etapas de nuestro algoritmo consistieron en compensar rotacionalmente la imagen omnidireccional, después, las imágenes fueron transformadas a imágenes panorámicas para continuar con la etapa de compensación traslacional. Por último, las imágenes fueron sometidas a la eliminación de las zonas donde fueron compensadas para que se pudiera realizar una correcta evaluación del PSNR de las imágenes.

En el siguiente capítulo se analizan los resultados de nuestro algoritmo en sus diversas etapas, así como la elección de los diversos parámetros de nuestro algoritmo para obtener el mejor incremento de PSNR de las imágenes. Finalmente se hace una comparación de nuestro método con otros algoritmos de estabilización en sistemas convencionales.

5 Experimentos y resultados

5.1 Reconstrucción de la imagen

En esta sección analizaremos los resultados obtenidos en la realización de este trabajo de investigación. El primer análisis está basado en la etapa de la transformación de la imagen, en esta parte se investigó en la literatura y se escogió el método más adecuado para este tipo de problema; es decir, el que presenta menos distorsión en la imagen final.

El algoritmo que se estableció para usar en este proyecto fue el algoritmo *Pano Mapping Table*, sin embargo existen dos versiones. La versión de Jeng et al. [23] y la versión de Xiong et al. [40]. Ambas versiones difieren en el tamaño de la tabla y el tiempo de ejecución del algoritmo, en estos rubros el algoritmo que tiene mejor rendimiento es el reportado en [40].

A continuación se muestran diversos resultados de la implementación de los dos algoritmos en la Tabla 5.1.

| Tamaño de la imagen omnidireccional | Tamaño de la imagen panorámica | Tiempo de transformación método [23] (s) | Tiempo (s) de transformación método [40] (s) | Relación de mejora en tiempo |
|-------------------------------------|--------------------------------|--|--|------------------------------|
| 1024 x 768 | 640x144 | 3.689259 | 1.412237 | 2.6123 |
| 1024 x 768 | 1280x288 | 14.717020 | 5.621864 | 2.6178 |
| 1024 x 768 | 2560x576 | 59.057698 | 22.557367 | 2.6181 |
| 1024 x 768 | 5120x1152 | 234.046250 | 90.355914 | 2.5903 |
| 1024 x 768 | 10240x2304 | 918.584759 | 366.599097 | 2.5057 |
| 1024 x 768 | 20480x4608 | 3769.567375 | 1401.507126 | 2.6897 |
| 1024 x 768 | 40960x9216 | 17944.291056 | 5385.974649 | 3.3317 |

Tabla 5.1. Resultados del algoritmo de transformación *Pano Mapping Table* en sus dos versiones.

Como podemos observar en la Tabla 5.1, el algoritmo Xiong et al. [40] presenta una mejora de aproximadamente 2.7 veces respecto al tiempo de ejecución, también como se explicó en el estado del arte estos algoritmos difieren en el tamaño de la tabla para la creación de la imagen, dicha relación es de 1/8 a favor del algoritmo de Xiong et al. El tamaño de la imagen panorámica varía en la tabla solamente para probar el tiempo de procesamiento entre los distintos métodos, sin embargo tanto el ancho y la altura de la imagen panorámica tienen sus restricciones como se mencionó en el capítulo anterior. Por lo tanto, si excedemos los valores permitidos para el ancho y altura de la imagen, podemos ocasionar la distorsión de la imagen panorámica.

Por las ventajas anteriormente mencionadas el algoritmo de Xiong et al. [40] es el utilizado en este proyecto para la reconstrucción de las imágenes omnidireccionales.

5.2 Selección de Parámetros

En la ejecución de nuestro algoritmo de estabilización, tenemos que seleccionar tres parámetros para poder obtener el mayor índice de PSNR de la secuencia estabilizada. Los parámetros son los siguientes:

1. Número mínimo de los puntos característicos rastreados en la secuencia de imágenes.
2. Porcentaje de valores eliminados de las trayectorias de los puntos característicos a través de la secuencia de imágenes
3. Grado del polinomio para el modelo de compensación de la secuencia de imágenes.

La selección de parámetros se basa en escoger aquellos con los cuales se obtiene un índice de PSNR mayor. La medición de los parámetros se realizó calculando el PSNR de todas las secuencias y realizando un promedio de estos, bajo diferentes combinaciones de parámetros.

En el caso para el número mínimo de puntos característicos que se rastrean en la secuencia de imágenes, el mejor parámetro fue de 100 puntos, como se puede observar en la Figura 5.1 y en Figura 5.2 mostramos el comportamiento de la desviación estándar del Video 1 variando dicho parámetro. Este parámetro es el número mínimo de características que debe de tener la tabla de puntos rastreados, si el número de elementos es menor, se procede a realizar un nuevo historial de puntos a partir de la última imagen.

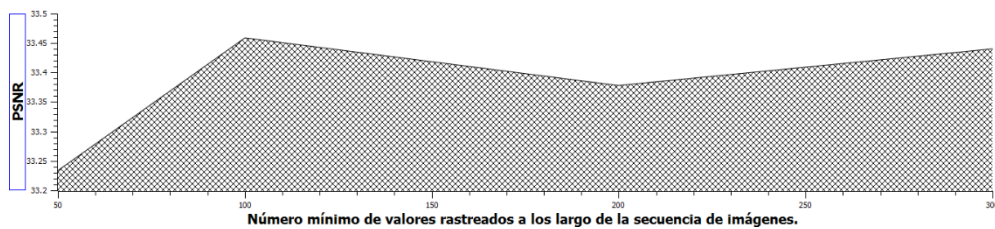


Figura 5.1. Número mínimo de valores rastreados a los largo de la secuencia de imágenes.

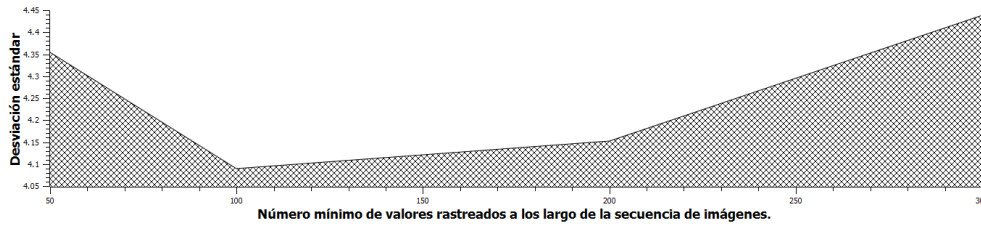


Figura 5.2 Desviación estándar del Video 1 respecto al número mínimo de valores rastreados.

El siguiente parámetro es porcentaje de valores que se eliminan del historial de movimientos, este consiste en eliminar cierto porcentaje de los valores más altos y bajos del histograma. El valor del parámetro con el que se obtuvo un mayor PSNR en las secuencias fue eliminando un 10% de los valores. En la Figura 5.3 se ilustran los diferentes valores con los que se implementó este parámetro y en la Figura 5.4



Figura 5.3. Porcentaje de valores eliminados del histograma de movimiento.

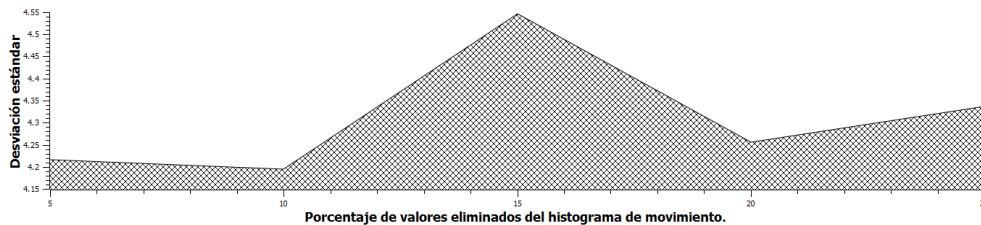


Figura 5.4 Desviación estándar del Video 1 respecto al porcentaje de valores eliminados del histograma de movimiento.

El último parámetro del algoritmo es el grado del polinomio que sirve para realizar el modelo de las vibraciones del sistema, el grado del polinomio que mejores resultados presente fue el de tercer grado. La elección de un polinomio de mayor grado no es posible debido a que el polinomio presentara mayores oscilaciones. Los resultados de este parámetro se ilustra en la Figura 5.5 y en la Figura 5.6 se muestra la desviación estándar del Video 1 variando dicho parámetro.

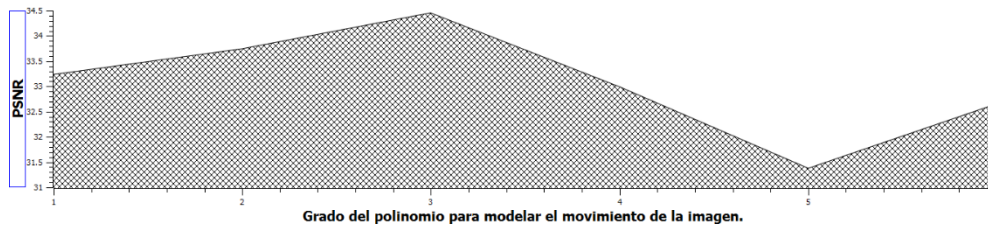


Figura 5.5. Grado del polinomio para modelar el movimiento de la imagen.

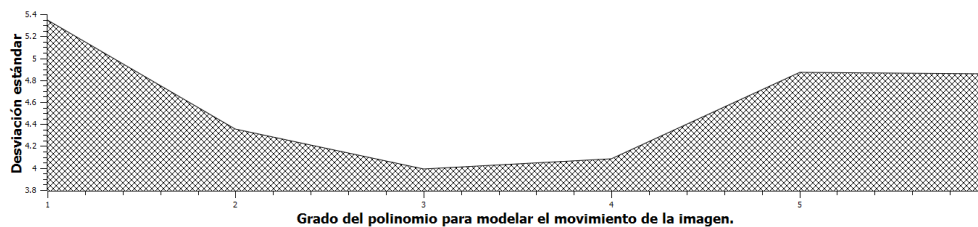


Figura 5.6 Desviación estándar del Video 1 respecto al grado del polinomio para modelar el movimiento de la imagen.

5.3 Estabilización de la imagen

En esta sección analizaremos los diversos resultados en la etapa de estabilización de las imágenes, el algoritmo se utilizó en diferentes tipos de videos adquiridos con nuestro sistema de visión catadióptrico, con diferentes características que se enlistan a continuación:

- Videos sujetos a vibraciones del sistema catadióptrico, sin objetos en movimiento (Videos 1-5).
- Videos sujetos a vibraciones del sistema catadióptrico con objetos en movimiento en la escena (Videos 6-10).
- Videos sujetos a vibraciones del sistema catadióptrico montado en objeto móvil con objetos en movimiento en la escena (Videos 11-15).

Las secuencias de imágenes se tomaron en diferentes escenarios donde ninguna variable fue controlada como: iluminación, vibraciones, cantidad o velocidad de objetos en movimiento, todo con el motivo de contar con secuencias más naturales para la evaluación de nuestro algoritmo. Anteriormente mencionamos que la evaluación de nuestro algoritmo se hizo en diferentes secuencias con características específicas y que cada una de estas secuencias está formada por 300 imágenes, por lo cual al final de nuestra etapa de evaluación, nuestro algoritmo habrá sido

evaluado en alrededor de 4500 imágenes. Por estas razones y la utilización de secuencias no artificiales, consideramos que tenemos un buen y diverso conjunto para la evaluación para nuestro algoritmo de estabilización.

En la Tabla 5.2 se muestran las diferentes etapas del algoritmo de estabilización, mostrando la métrica para medir la estabilización que se eligió anteriormente. La tabla muestra los resultados obtenidos en las diferentes etapas del algoritmo, las cuales se explican a continuación se explica con más detalles los resultados para cada etapa.

Etapa rotacional

En esta etapa la mayoría de las secuencias de imágenes presentan una mejoría en cuanto al promedio del PSNR entre imágenes, sin embargo los videos 11-15 presentan un pequeño incremento en el PSNR debido a que en estas secuencias no se presenta este tipo de movimiento. Dichas secuencias fueron tomadas con el sistema catadióptrico montado en un automóvil transitando por diferentes tipos de calles, razón por la cual la mayoría de las vibraciones son en la etapa traslacional. Los resultados de esta etapa están representados en la columna 2 de la Tabla 5.2, esta columna representa el PSNR de la secuencia después de haber sido transformada y compensada rotacionalmente.

Etapa Traslacional

Una vez que los videos han sido estabilizados en la etapa rotacional, estos pasan a la etapa de estabilización traslacional para eliminar dichos movimientos. Todos los videos en esta etapa presentaron una mejora en cuanto al PSNR debido a que es el movimiento más común en este tipo de sistemas. En la columna 3 se presentan los resultados de esta etapa, representa el PSNR de la secuencia cuando se le han aplicado las dos etapas de estabilización, tanto rotacional como traslacional.

Etapa de recorte

Dado que para compensar la imagen, esta se debe de mover en sentido contrario al de las vibraciones, por este motivo en la secuencia que fue estabilizada existirán zonas donde no existen pixeles debido a que fueron desplazados para compensar el movimiento. La eliminación de estas zonas tiene que ver directamente con el PSNR de la secuencia, debido a que si restamos un pixel negro (zona de compensación)

de otro pixel con un valor distinto, esto hará que el PSNR entre imágenes disminuya. La columna 5 contiene los valores de PSNR de la secuencia original que ha sido transformada a una secuencia panorámica y recortada para poder realizar una correcta comparación de PSNR contra la secuencia que ha sido estabilizada. La columna 6 presenta el valor final de la secuencia que hasta el momento ha sido transformada, estabilizada y recortada.

| | PSNR video estabilizado etapa rotacional | PSNR video estabilizado etapa traslacional | PSNR video estabilizado y recortado. | PSNR video sin estabilizar y recortado. | Relación de mejora respecto a la secuencia original (%). |
|-----------------|---|---|---|--|---|
| Video 1 | 34.236 | 36.154 | 37.113 | 29.9774 | 23.8032 |
| Video 2 | 32.5379 | 34.2177 | 34.6619 | 28.7827 | 20.4261 |
| Video 3 | 33.5356 | 35.1179 | 35.4977 | 28.9733 | 22.5186 |
| Video 4 | 33.3688 | 34.8053 | 35.1555 | 29.1086 | 20.7735 |
| Video 5 | 34.0321 | 34.2453 | 34.4564 | 29.1354 | 18.263 |
| Video 6 | 31.9567 | 32.9204 | 33.1394 | 29.3027 | 13.0933 |
| Video 7 | 31.6435 | 32.3484 | 32.4971 | 29.6259 | 9.6915 |
| Video 8 | 32.9675 | 33.8601 | 34.1622 | 30.7487 | 11.1012 |
| Video 9 | 33.7323 | 34.1942 | 34.8279 | 30.9561 | 12.5073 |
| Video 10 | 32.9432 | 33.4012 | 33.5623 | 30.1735 | 11.231 |
| Video 11 | 31.9354 | 32.3564 | 32.7951 | 31.4873 | 4.1534 |
| Video 12 | 31.0234 | 31.4637 | 31.6345 | 30.5837 | 3.4358 |
| Video 13 | 31.6579 | 32.5797 | 32.4775 | 31.1738 | 4.182 |
| Video 14 | 31.0415 | 31.7332 | 31.7882 | 30.8724 | 2.9664 |
| Video 15 | 30.8598 | 31.9007 | 32.1016 | 30.6209 | 4.8355 |

Tabla 5.2. Resultados del algoritmo de estabilización sin precisión subpixel y sus diferentes etapas.

La última columna representa el porcentaje de incremento de PSNR de la secuencia estabilizada respecto a la secuencia panorámica original, mediante estos resultados podemos observar que con nuestro algoritmo obtuvimos un promedio de incremento de PSNR del 12.2%, sin embargo podemos observar que en algunos casos el incremento es menor o mayor, esto debido a los diferentes escenarios de las secuencias de imágenes.

| | PSNR video estabilizado etapa rotacional | PSNR video estabilizado etapa traslacional | PSNR video estabilizado y recortado. | PSNR video sin estabilizar y recortado. | Relación de mejora respecto a la secuencia original (%). |
|-----------------|---|---|---|--|---|
| Video 1 | 33.4358 | 37.0685 | 37.3378 | 30.0318 | 24.3275 |
| Video 2 | 32.6372 | 34.5956 | 35.4387 | 28.812 | 22.9996 |
| Video 3 | 32.9082 | 35.7117 | 36.2607 | 29.5266 | 22.807 |
| Video 4 | 33.2541 | 35.4878 | 35.8368 | 29.6318 | 20.6026 |
| Video 5 | 33.3459 | 34.9322 | 35.2766 | 29.497 | 19.5934 |
| Video 6 | 31.8153 | 33.604 | 34.1648 | 30.076 | 12.9298 |
| Video 7 | 31.8647 | 32.839 | 32.8135 | 29.7519 | 10.2905 |
| Video 8 | 32.8475 | 33.8766 | 34.6422 | 31.0576 | 11.5417 |
| Video 9 | 33.5546 | 34.622 | 35.3555 | 31.4221 | 12.518 |
| Video 10 | 32.2153 | 34.177 | 34.7062 | 31.1566 | 10.4298 |
| Video 11 | 31.5036 | 32.7288 | 33.7313 | 32.3755 | 4.1876 |
| Video 12 | 31.8719 | 31.9444 | 32.1235 | 30.9454 | 3.8068 |
| Video 13 | 31.235 | 33.3692 | 33.3503 | 31.9527 | 4.3741 |
| Video 14 | 30.4306 | 32.0523 | 32.477 | 31.2235 | 4.0146 |
| Video 15 | 31.8371 | 32.3426 | 32.9916 | 31.0935 | 6.1043 |

Tabla 5.3. Resultados del algoritmo de estabilización con precisión subpixel y sus diferentes etapas.

En la Tabla 5.3 mostramos los resultados obtenidos mediante la estabilización de imágenes con una precisión subpixel de $\frac{1}{4}$, en la cual podemos observar un incremento en el PSNR de las secuencias de imágenes. Sin embargo, el incremento de PSNR no es significativamente mayor respecto al algoritmo sin precisión subpixel. El promedio de la relación de mejora del algoritmo con precisión subpixel ronda aproximadamente entre el 12.70 % respecto a la secuencia original. Los resultados del algoritmo con precisión subpixel presentan la misma distribución de la Tabla 5.2.

En la Tabla 5.4 se muestran los tiempos de procesamiento de los algoritmos de estabilización con y sin precisión subpixel de $\frac{1}{4}$, donde podemos observar que el tiempo de procesamiento se incrementa más del doble con precisión subpixel de $\frac{1}{4}$. Este incremento se debe principalmente por el nivel de precisión, dado que, hemos cuadruplicado la cantidad de información de la imagen.

| | Tiempo de procesamiento sin precisión subpixel. (s) | Tiempo de procesamiento con precisión subpixel. (s) |
|----------|---|---|
| Video 1 | 523.2894 | 1234.4098 |
| Video 2 | 538.3132 | 1225.3429 |
| Video 3 | 517.2745 | 1235.6453 |
| Video 4 | 524.0573 | 1243.1874 |
| Video 5 | 537.5943 | 1236.8734 |
| Video 6 | 532.4954 | 1231.4753 |
| Video 7 | 519.8756 | 1227.234 |
| Video 8 | 563.9485 | 1234.5678 |
| Video 9 | 532.4562 | 1240.8945 |
| Video 10 | 538.9146 | 1237.0948 |
| Video 11 | 553.4583 | 1232.3847 |
| Video 12 | 537.6183 | 1221.3474 |
| Video 13 | 528.7193 | 1246.4934 |
| Video 14 | 543.5483 | 1235.3848 |
| Video 15 | 532.8435 | 1226.7436 |

Tabla 5.4. Tiempos de procesamiento de los algoritmos con y sin precisión subpixel.

5.4 Desempeño contra algoritmos para sistemas convencionales.

Anteriormente mencionamos que no existe trabajo relacionado en el estado del arte donde el algoritmo de estabilización sea desarrollado en un sistema catadióptrico. Sin embargo, nos planteamos la tarea de comparar nuestro método con otros métodos convencionales para tener una forma de evaluación de nuestro trabajo respecto a otros trabajos.

Primero, elegimos un trabajo donde también utilizan el descriptor SURF para realizar la estimación de movimiento, en este caso es el algoritmo propuesto por Zhou[38]. El segundo paso fue elegir una plataforma comercial que implemente un algoritmo de estabilización para secuencias de imágenes, el software elegido es el proporcionado por el editor de videos de la compañía Youtube[41]. El proceso para realizar la comparación fue primero transformar las secuencias omnidireccionales

con precisión subpixel a secuencias panorámicas, dado que los dos enfoques que vamos a utilizar son implementados sobre sistemas convencionales. Una vez que tenemos las secuencias panorámicas procedemos a la aplicación de ambos algoritmos a las secuencias y la respectiva comparación respecto a los resultados obtenidos de nuestro algoritmo propuesto con precisión subpixel. Los resultados de las pruebas se muestran en la Tabla 5.5.

| | PSNR video estabilizado mediante método Zhou[38] | PSNR video estabilizado mediante editor Youtube[41] | PSNR video estabilizado mediante método propuesto. |
|-----------------|---|--|---|
| Video 1 | 33.8917 | 35.9483 | 37.3378 |
| Video 2 | 32.8295 | 33.2197 | 35.4387 |
| Video 3 | 33.0209 | 33.6877 | 36.2607 |
| Video 4 | 32.3277 | 32.7104 | 35.8368 |
| Video 5 | 33.1498 | 34.3915 | 35.2766 |
| Video 6 | 32.3627 | 32.7964 | 34.1648 |
| Video 7 | 31.9404 | 32.0268 | 32.8135 |
| Video 8 | 32.1734 | 33.5619 | 34.6422 |
| Video 9 | 33.7209 | 34.1179 | 35.3555 |
| Video 10 | 32.3486 | 33.443 | 34.7062 |
| Video 11 | 32.5608 | 32.7783 | 33.7313 |
| Video 12 | 31.2492 | 31.7331 | 32.1235 |
| Video 13 | 32.0145 | 32.2165 | 33.3503 |
| Video 14 | 31.3006 | 31.4297 | 32.477 |
| Video 15 | 31.1486 | 31.3151 | 32.9916 |

Tabla 5.5 Resultados de algoritmos convencionales implementados en las secuencias de prueba.

Mediante la Tabla 5.3 y la Tabla 5.5, podemos inferir el porcentaje de mejora de los algoritmos respecto a la secuencia original, el algoritmo de Zhou[38] obtuvo en promedio una mejora en el porcentaje de PSNR respecto a la secuencia original alrededor del 6.13% y el software de Youtube alrededor de un incremento del 8.18%. El mayor incremento de PSNR en nuestro método propuesto se debe básicamente a que hacemos diversos procesamientos en las imágenes

omnidireccionales, como por ejemplo el rastreo de puntos característicos o la compensación rotacional.

El rastreo de puntos característicos realizado en la imagen omnidireccional, nos ayuda a evitar el error generado por la distorsión de la transformación a una imagen panorámica y por lo tanto podemos obtener un modelo de movimiento que no contenga error acumulativo por la etapa de compensación y de transformación de la imagen.

La compensación rotacional en nuestro algoritmo es realizada respecto al centro de la imagen omnidireccional, después de realizar la compensación, no perdemos porciones de la zona de interés (definida por los radios r_1 y r_2) de la imagen omnidireccional dado que la zona de interés es más pequeña que la imagen omnidireccional. Por otro lado, en los algoritmos convencionales, la estabilización de las imágenes se hace respecto al centro de la imagen panorámica siendo en este caso donde se pierden zonas de interés como se ilustra en la Figura 5.7.

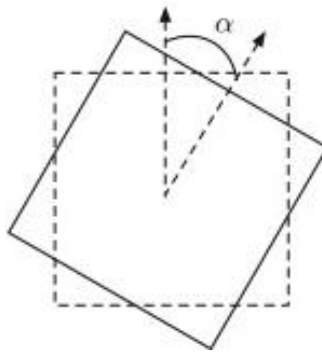


Figura 5.7 Compensación rotacional en sistemas convencionales, se ilustra cómo se pierde información de las esquinas de la imagen panorámica.

Los resultados en esta sección nos indican que nuestro algoritmo es superior en desempeño, al obtener un índice mayor de PSNR respecto a las secuencias originales de imágenes. Sin embargo, este desempeño superior se cumple bajo ciertas características, como es la de utilizar los algoritmos convencionales en las imágenes panorámicas y no directamente en las imágenes omnidireccionales como es el caso de nuestro método propuesto.

5.5 Discusión de resultados

El desempeño de nuestro algoritmo como se mencionó anteriormente esta medido mediante el índice de PSNR de la secuencia de imágenes. El incremento del índice de PSNR nos indica que nuestro algoritmo de estabilización está realizando

adecuadamente su propósito, sin embargo el incremento es variable en cada secuencia de imágenes debido a su naturaleza, es por ello que en algunas secuencias que no presentan demasiados cambios espaciales entre sus imágenes consecutivas se obtiene un mayor índice de PSNR.

Los mayores incrementos de PSNR en nuestros videos de prueba fueron en aquellas secuencias de imágenes (1-5) donde no existe movimiento por parte del sistema catadióptrico, solamente existen vibraciones que afectan a dicho sistema. Por otro lado, los menores incrementos se dieron en las secuencias (11-15) donde se presenta movimiento de todo el sistema catadióptrico, esto se debe básicamente a la poca relación espacial que existe entre los pixeles de dos imágenes consecutivas. Sin embargo, en todos los trabajos revisados en el estado del arte, el principal objetivo de los algoritmos de estabilización de imágenes es obtener un incremento en el índice de PSNR.

Los trabajos revisados en el estado del arte, las implementaciones de los algoritmos de estabilización son realizados en sistemas convencionales de imágenes, no se encontró referencia alguna en estabilización de sistemas catadióptricos. La principal razón por la cual no podríamos comparar nuestro algoritmo de estabilización en sistemas convencionales radica en que la primera parte de nuestro algoritmo, la etapa de rastreo de puntos y la compensación rotacional, son realizadas en las imágenes omnidireccionales tomando en cuenta las propiedades de dichas imágenes. En nuestro algoritmo hacemos uso tanto del centro de la imagen omnidireccional y esto nos facilita la estimación del movimiento en un sistema de coordenadas polar.

5.6 Resumen

En este capítulo hablamos de los diferentes resultados de nuestro algoritmo de estabilización, primero abordamos los resultados en tiempo de ejecución y espacio de almacenamiento del algoritmo de transformación *Pano Mapping Table* en sus dos versiones[23][40]. Después se presentaron diversos resultados de nuestro algoritmo para la elección de nuestros tres principales parámetros como son: el número mínimo de valores que se rastrean a lo largo de las secuencias de imágenes, porcentaje de valores que se eliminan del historial de movimiento y el grado del polinomio que se ajusta a los puntos rastreados para encontrar el modelo de movimiento de la imagen. En la siguiente parte, se habló de los resultados de nuestro algoritmo de estabilización y sus resultados a lo largo de sus diferentes etapas. Los resultados de nuestro algoritmo de estabilización sin precisión subpixel presentaron en promedio un incremento del 12.2% en su PSNR y en su versión con precisión subpixel alrededor de 12.7%.

En el siguiente capítulo se discuten las contribuciones y conclusiones de este trabajo de investigación, así como las principales tareas que se pueden realizar en un futuro para mejorar el desempeño de dicho algoritmo.

6 Conclusiones y Trabajo futuro

6.1 Contribución

Los sistemas convencionales de imágenes, como se ha mencionado, poseen un campo de visión limitado. Por otro lado, los sistemas catadióptricos de imágenes poseen un amplio campo de visión, que generalmente abarca desde 180° a 360° . Cualquier sistema de imágenes se ve afectado por vibraciones pertenecientes al medio donde se encuentra, para eliminar este tipo de vibraciones existen métodos basados en técnicas de procesamiento de imágenes. La motivación principal de nuestro trabajo de investigación es la ausencia de un algoritmo de estabilización diseñado específicamente para sistemas de visión catadióptricos.

El método propuesto en este trabajo de investigación consiste básicamente en rastrear puntos característicos en las imágenes omnidireccionales para después realizar un historial de movimientos que nos ayude a describir el movimiento de las imágenes y poder generar un modelo de movimiento. El modelo de movimiento sirve para compensar la imagen en una etapa rotacional en la imagen omnidireccional y en otra etapa traslacional en la imagen panorámica. El método propuesto se implementó agregando precisión subpixel a las imágenes, donde demostró un mejor desempeño respecto a la implementación sin precisión subpixel.

La principal aportación de este trabajo radica en el desarrollo de un algoritmo de estabilización de imágenes para sistemas catadióptricos de imágenes. Nuestro algoritmo realiza la estabilización de imágenes alrededor del eje Z del sistema catadióptico. El historial de movimiento que creamos para descartar aquellos puntos que no tienen una gran duración temporal, es una parte fundamental de nuestro algoritmo dado que es la encargada de seleccionar aquellos puntos que son más confiables para la estabilización de la imagen.

Otra aportación es el algoritmo para realizar la clasificación de los vectores de movimiento que pertenecen a las vibraciones del sistema, esta parte es fundamental para lograr una correcta estabilización de imágenes. Nuestro algoritmo es rápido y robusto, dado que solamente necesitamos dos etapas para separar aquellos puntos que pertenecen a las vibraciones del sistema.

La precisión subpixel utilizada en este trabajo de investigación, demostró que funciona para obtener un mejor desempeño en la estabilización de imágenes. Utilizando dicha precisión logramos un incremento en el PSNR de las secuencias de imágenes, este incremento se debe principalmente a la mejor aproximación de los vectores de movimiento que se obtiene mediante el aumento de la precisión.

6.2 Conclusiones

En este proyecto de investigación se desarrolló un algoritmo de estabilización en sistemas de visión catadióptricos, en el cual podemos observar mediante los resultados obtenidos las características de los tipos de escenas donde el algoritmo presenta mejoras en la relación de PSNR, este tipo de características se presentan a continuación:

- Secuencias con vibraciones generadas por el medio donde se encuentra el sistema catadióptico (vibraciones de la mano, del ambiente, etc.).
- Secuencias con objetos en movimiento tales como personas, autos, etc.
- Secuencias con cambios de iluminación.

Como podemos observar, se cumplieron los objetivos planteados para el proyecto de investigación, los cuales consistían básicamente en reconocer aquellos movimientos que pertenecen o no a la secuencia de imágenes. Identificar las vibraciones del medio es el caso más sencillo de nuestro algoritmo, por lo cual en los resultados es donde se presentaron mejores incrementos en el índice de PSNR de las secuencias. Estos incrementos en el PSNR se deben básicamente a que el fondo de la imagen permanece constante y solo se ve afectado por las vibraciones, por lo cual es sencillo realizar el rastreo de los puntos a lo largo de toda la secuencia de imágenes.

El segundo caso es reconocer entre los movimientos generados por las vibraciones y los movimientos de los objetos que son parte de la escena. En este tipo de escenas también se obtuvo un buen desempeño de nuestro algoritmo, principalmente por la utilización del historial de movimiento, dado que si algún objeto entra o sale de la imagen será eliminado rápidamente del historial de movimientos.

Por último también se probó el algoritmo en cambios de iluminación, ya que esto puede provocar que los objetos desaparezcan o que los objetos se distorsionen y parezca que se movieron, sin embargo debido a la utilización del algoritmo SURF para la extracción de puntos característicos estos cambios no afectan el desempeño de nuestro algoritmo. En estos tres casos se obtuvieron resultados satisfactorios, dado que el incremento en el PSNR fue notable.

En las pruebas realizadas hay un caso en particular donde incremento del PSNR no es semejante al de los casos que se mencionaron anteriormente, esta prueba fue realizada montando el sistema catadióptico en un vehículo que recorrió diversas calles. En este tipo de pruebas nuestro algoritmo presentó el más bajo incremento en el nivel de PSNR, esto se debe principalmente a que en nuestro algoritmo no se tomó en cuenta algún modelo de movimiento del sistema catadióptico, en este caso

el modelo de movimiento del auto. Otra razón para que el incremento del PSNR no fuera mayor, es debido a la naturaleza de las imágenes, ya que el PSNR de la imagen original es bajo por la cantidad de movimiento que existe en ella.

Otro punto importante a considerar es que no existe una mejora adicional en cuanto al uso de la precisión subpixel, esto debido a que el uso de la precisión subpixel conlleva el incremento de la información de la imagen y por lo tanto el tiempo de procesamiento de la imagen. Simplemente al incrementar un nivel la precisión subpixel, se está incrementando el doble de la información y el PSNR de la imagen estabilizada sufre un incremento muy pequeño que no justifica el tiempo de procesamiento incrementado.

Por último, realizamos una comparación de nuestro método propuesto contra dos algoritmos desarrollados para sistemas convencionales. En las pruebas realizadas con estos algoritmos, nuestro método propuesto mostro un desempeño superior al incrementar el índice de PSNR de las imágenes. Sin embargo, no podemos asegurar esta superioridad completamente dado que nuestro algoritmo está desarrollado para sistemas catadióptricos utilizando algunas propiedades de estos sistemas y los algoritmos convencionales no toman ventajas de dichas propiedades.

6.3 Trabajo futuro

Algunas acciones futuras que se podrían implementar sobre este proyecto de investigación, podrían ser las siguientes.

- Realizar una implementación en tiempo real que al menos pueda realizar una estabilización de imágenes con una tasa de 30 imágenes por segundo para poder utilizar en aplicaciones tales como vigilancia, seguimiento de personas, etc. Por ejemplo, se podría paralelizar la etapa de transformación de la imagen.
- Implementar una transformación de la imagen omnidireccional a panorámica que se ajuste a las características de los espejos utilizados, debido a que los algoritmos que se utilizaron para la transformación de imágenes en este proyecto son genéricos y sirve para cualquier tipo de espejo (circular, parabólico, hiperbólico). Lo anterior hace que la distorsión de la transformación no pueda ser minimizada para un problema en específico.
- Modelar el desplazamiento generado por las vibraciones en los ejes X e Y del sistema catadióptrico, dado que estos movimientos generan distintos

tipos de transformaciones en la imagen y se necesita una combinación de estas transformaciones para modelarlas.

- Implementar la estabilización de la imagen en los ejes restantes (X e Y) del sistema catadióptrico para lograr mitigar los efectos de escalamiento en las imágenes omnidireccionales por las vibraciones.

7 Bibliografía

- [1] L. Chen, M. Zhang, B. Wang, Z. Xiong, and G. Cheng, “Real-Time FPGA-Based Panoramic Unrolling of High-Resolution Catadioptric Omnidirectional Images,” in *2009 International Conference on Measuring Technology and Mechatronics Automation*, 2009, pp. 502–505.
- [2] S. Ertürk and T. J. Dennis, “Image sequence stabilisation based on DFT filtering,” *IEE Proc. - Vision, Image, Signal Process.*, vol. 147, no. 2, p. 95, 2000.
- [3] K. Wang, “Experiment research on electronic image stabilization of omnidirectional camera system,” in *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management*, 2011, vol. 150001, pp. 1682–1685.
- [4] L. Spacek, “A catadioptric sensor with multiple viewpoints,” *Rob. Auton. Syst.*, vol. 51, no. 1, pp. 3–15, Apr. 2005.
- [5] R. Swarninathan and S. Nayar, “Non-metric calibration of wide-angle lenses and polycameras,” in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, pp. 413–419.
- [6] S. Kang and R. Szeliski, “3-D scene data recovery using omnidirectional multibaseline stereo,” *Int. J. Comput. Vis.*, vol. 25, no. 2, pp. 167–183, 1997.
- [7] H. Ishiguro, M. Yamamoto, and S. Tsuji, “Omni-directional stereo,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 257–262, 1992.
- [8] S. Shah and J. K. Aggarwal, “Mobile robot navigation and scene modeling using stereo fish-eye lens system,” *Mach. Vis. Appl.*, vol. 10, no. 4, pp. 159–173, Dec. 1997.
- [9] L. Delahoche, C. Pegard, B. Marhic, and P. Vasseur, “A navigation system based on an omnidirectional vision sensor,” in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, 1997, vol. 2, pp. 718–724.

- [10] Y. Yagi, W. Nishii, K. Yamazawa, and M. Yachida, "Rolling motion estimation for mobile robot by using omnidirectional image sensor HyperOmniVision," in *Proceedings of 13th International Conference on Pattern Recognition*, 1996, pp. 946–950 vol.1.
- [11] M. Wang, C. Huang, and H. Lin, "An Intelligent Surveillance System Based on an Omnidirectional Vision Sensor," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, 2006, pp. 1–6.
- [12] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion," in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, 2006, no. IcvS, pp. 45–45.
- [13] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
- [14] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [16] T. Moeslund, "Geometric Transformations," in *Introduction to Video and Image Processing SE - 10*, Springer London, 2012, pp. 141–153.
- [17] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*, 2nd ed., vol. 2. Mc Graw Hill, 2010.
- [18] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 1998, vol. 5, pp. 2789–2792.
- [19] L. Marcenaro, G. Vernazza, and C. S. Regazzoni, "Image stabilization algorithms for video-surveillance applications," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 2001, vol. 1, pp. 349–352.

- [20] D. Saville and G. Wood, "Polynomial Regression," in *Statistical Methods: The Geometric Approach SE - 16*, Springer New York, 1991, pp. 431–460.
- [21] W. K. Wong, W. ShenPua, C. K. Loo, and W. S. Lim, "A study of different unwarping methods for omnidirectional imaging," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 433–438.
- [22] S. W. Jeng and W. H. Tsai, "A Unified Approach to Unwarping of Omni-Images into Panoramic and Perspective-View Images Using Pano-Mapping Tables," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 2401–2406.
- [23] S. Jeng and W. Tsai, "Using pano-mapping tables for unwarping of omni-images into panoramic and perspective-view images," *IET Image Process.*, vol. 1, no. 2, p. 149, 2007.
- [24] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–63, Jul. 2006.
- [25] S. Auberger and C. Miro, "Digital video stabilization architecture for low cost devices," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, 2005, pp. 474–479.
- [26] S. Lee, S. Jeon, and E. Kang, "Fast digital image stabilizer based on Gray-coded bit-plane matching," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 598–603, 1999.
- [27] H. Chen, "Integration of Digital Stabilizer With Video Codec for Digital Video Cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 7, pp. 801–813, Jul. 2007.
- [28] S. Battiato, A. R. Bruna, and G. Puglisi, "A Robust Block-Based Image/Video Registration Approach for Mobile Imaging Devices," *IEEE Trans. Multimed.*, vol. 12, no. 7, pp. 622–635, Nov. 2010.
- [29] G. Puglisi and S. Battiato, "A Robust Image Alignment Algorithm for Video Stabilization Purposes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1390–1400, Oct. 2011.

- [30] O. Kwon, J. Shin, and J. Paik, "Video stabilization using Kalman filter and phase correlation matching," *Image Anal. Recognit.*, pp. 141–148, 2005.
- [31] H. Yan and J. Liu, "Robust Phase Correlation based feature matching for image co-registration and DEM generation," *Remote Sens. Spat. Inf. Sci.*, pp. 1751–1756, 2008.
- [32] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, Aug. 1981.
- [33] B. Lucas D. and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Int. Jt. Conf. Artif. Intell.*, pp. 674–679, 1981.
- [34] J. Zhang, F. Zhang, and M. Ito, "Motion estimation based on optical flow and an artificial neural network (ANN)," *Artif. Life Robot.*, vol. 14, no. 4, pp. 502–505, Jan. 2010.
- [35] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "SIFT Features Tracking for Video Stabilization," in *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, 2007, no. Iciap, pp. 825–830.
- [36] Y. Shen, P. Guturu, T. Damarla, B. Buckles, and K. Namuduri, "Video stabilization using principal component analysis and scale invariant feature transform in particle filter framework," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1714–1721, Aug. 2009.
- [37] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [38] M. Zhou and V. K. Asari, "A Fast Video Stabilization System Based on Speeded-up," *7th Int. Symp. ISVC 2011, Las Vegas, NV, USA, Sept. 26-28, 2011. Proceedings, Part II*, pp. 428–435, 2011.
- [39] C. Li and Y. Liu, "Global motion estimation based on SIFT feature match for digital image stabilization," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, 2011, pp. 2264–2267.

- [40] Z. Xiong, I. Cheng, A. Basu, W. Wang, W. Xu, and M. Zhang, “Efficient omni-image unwarping using geometric symmetry,” *Mach. Vis. Appl.*, vol. 23, no. 4, pp. 725–737, Dec. 2010.
- [41] Youtube, “Editor Youtube.” 2013.