PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO DE UN ALGORITMO DE ESTABILIZACIÓN DE VIDEO ORIENTADO A LA DETECCIÓN DE PERSONAS

Tesis para optar el Título de **INGENIERO ELECTRÓNICO,** que presenta el bachiller:

Alberto Hiroshi Inafuku Yoshida

ASESOR: Renán Alfredo Rojas Gómez

Lima, mayo de 2015

Resumen

El presente trabajo de tesis tiene como objetivo principal el desarrollo de un algoritmo de estabilización de video robusto y eficiente frente a cambios de escala, rotación e iluminación. La estabilización de video es una etapa de pre procesamiento utilizada para eliminar o reducir el ruido que se adhiere debido a movimientos involuntarios en los videos. Su importancia radica en que procesamientos posteriores requieren de imágenes alineadas y libres de distorsión espacial.

El documento está divido en cuatro capítulos descritos a continuación:

En el capítulo 1 se presenta la problemática de la estabilización de videos así como las aplicaciones en diversos campos. Se explicará cómo los videos pueden ser afectados por factores ajenos a la cámara. Entre las aplicaciones se mencionarán procesamientos posteriores que requieren estabilización como paso previo y aplicaciones finales en temas de seguridad, industria y entretenimiento.

En el capítulo 2 se mencionan las alternativas de solución del problema. Se presentan tanto las alternativas como sus características, ventajas y desventajas. Entre ellas se describen algunas alternativas mecánicas, que involucran equipos y sistemas sofisticados, y alternativas digitales como registro de imágenes, Structure from Motion y Geometría Epipolar.

El capítulo 3 describe la metodología a emplear. Se utilizan gráficas, diagramas e imágenes para mostrar de manera sencilla cómo se piensa atacar el problema. Se describen los parámetros utilizados en cada etapa.

El capítulo 4 muestra las simulaciones y los resultados del algoritmo implementado. Mediante imágenes se muestra los resultados de las etapas descritas en el capítulo anterior.

Índice

Introducción

La estabilización de video es una etapa de pre procesamiento utilizada para eliminar o reducir el movimiento involuntario adquirido durante su adquisición. Mediante Procesamiento Digital de Imágenes (PDI) se puede modificar el video para reducir los movimientos bruscos de manera rápida y eficiente [2, 3]. La detección de personas a través de PDI sería de gran utilidad para optimizar procesos en diversas áreas [7-11]. En seguridad, industria o entretenimiento, podemos encontrar procesos que involucran reconocimiento. Por ejemplo, reconocer rostros o movimientos corporales para controlar e interactuar con una interfaz o realizar un seguimiento, identificación o conteo de personas.

Para la implementación del algoritmo de estabilización se utilizarán conceptos de Geometría Epipolar y Registro de imágenes, para conseguir representar a las personas como objetos con movimiento independiente al de la cámara, diferenciándolos de objetos o escenarios estáticos. La Geometría Epipolar se basa en la geometría proyectiva obtenida a partir de dos vistas [4, 19]. Un objeto en el espacio observado desde una cámara en movimiento puede ser representado en fotografías sucesivas mediante dos planos que guardan relaciones de geometría proyectiva en una matriz fundamental. El Registro de Imágenes es utilizado para obtener información de escenas y hacer seguimiento y/o transformaciones entre cuadros [3, 13].

El objetivo de la presente investigación es diseñar un algoritmo de estabilización de video orientado a la detección de personas, capaz de corregir la posición de los frames, brindando resultados físicamente coherentes frente a cambios rotacionales y de escala, y frente a cambios de iluminación, además de distinguir objetos estáticos de dinámicos. Para ello se realizará previamente el estudio del estado del arte en estabilización de video y una comparación de las metodologías existentes.

Capítulo 1

Estabilización de Video

El avance tecnológico ha permitido que hoy en día sea accesible para una gran mayoría poder grabar un video desde un teléfono móvil, una tablet, o desde vehículos, como automóviles, aviones o drones [7-10]. Lamentablemente al adquirir estos videos, debido a condiciones externas a las cámaras, como movimientos involuntarios, se tiene un ruido particular conocido como jitter, un corrimiento en los cuadros que altera el movimiento de la cámara. El jitter, en señales, es una variación en la posición ideal de una señal en un tiempo determinado [6]. Como se muestra en figura 1.1. La señal original en línea punteada sufre un retardo aleatorio que modifica su valor ideal, dando como resultado una señal con jitter.

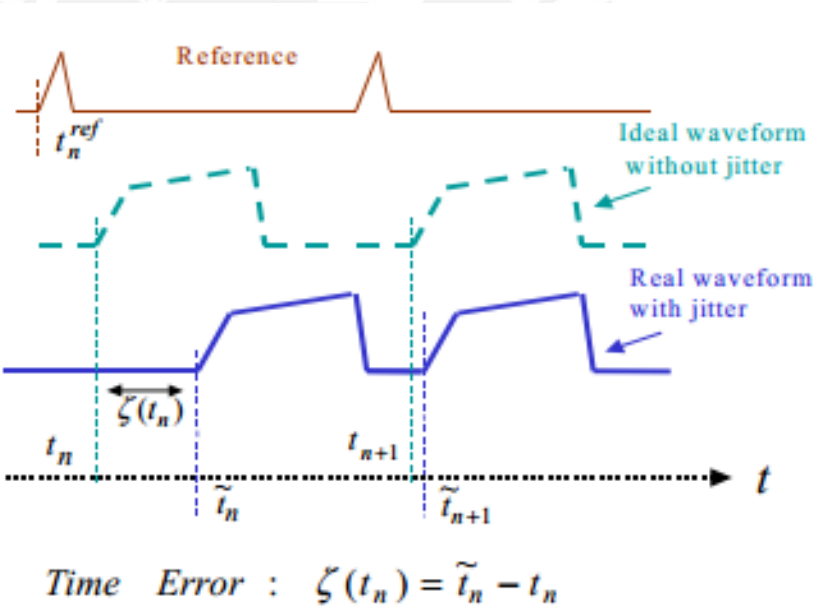


Figura 1.1. Jitter en señales

Fuente: Zamek [6]

El movimiento tembloroso de las manos en videos tomados desde dispositivos móviles [7], el viento y vibraciones de los motores en videos tomados desde vehículos aéreos no tripulados [9], o baches en las pistas en videos tomados desde autos [10], son algunos ejemplos de cómo este efecto se puede presentar.

El efecto jitter perjudica la calidad de las imágenes, lo que reduce la eficiencia de etapas de post procesamiento como segmentación o reconocimiento. Cuando se hace un seguimiento o tracking a un punto en particular de la imagen, se puede apreciar cómo este punto presenta un movimiento distinto al que debería seguir. La figura 1.2 describe la apariencia de un video con ruido y el video resultado luego de que este es estabilizado.

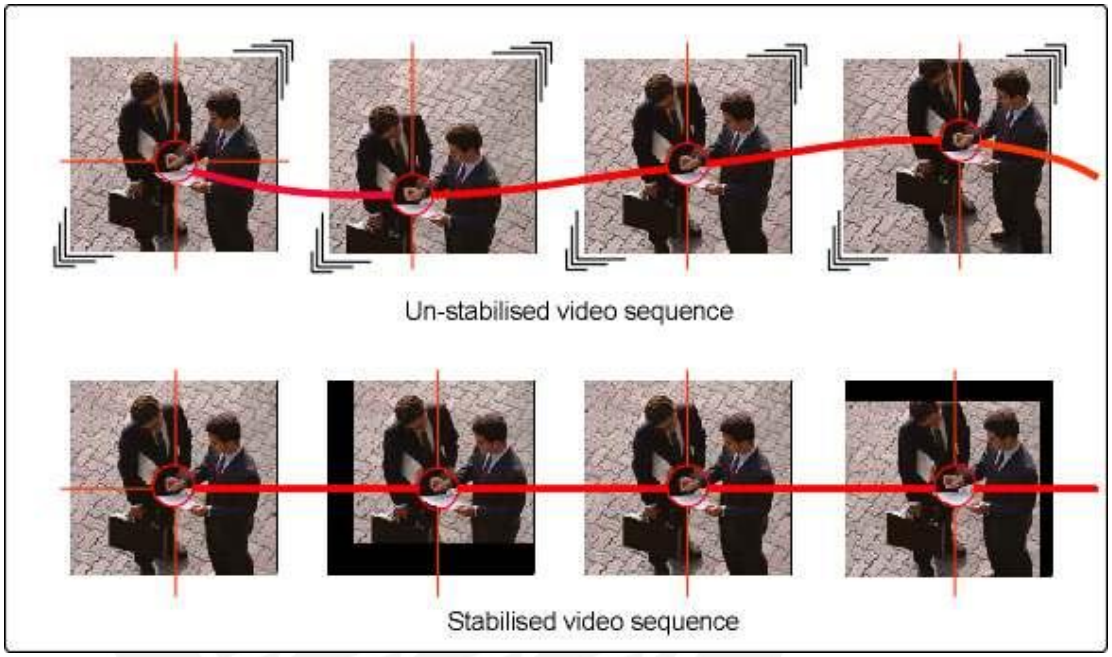


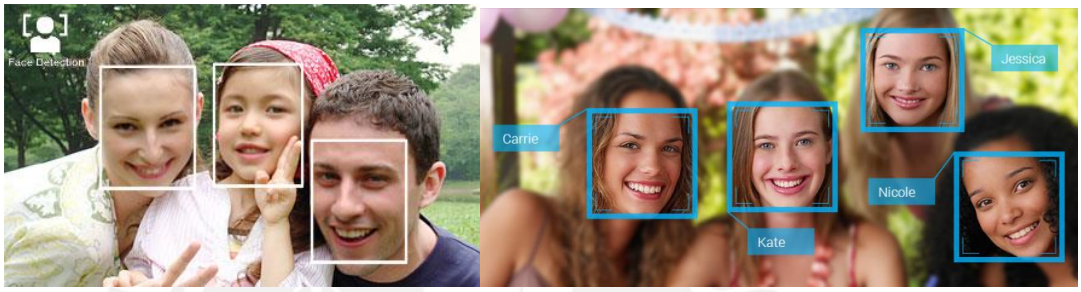
Figura 1.2. Secuencia de video con jitter y Secuencia de video estabilizada

Fuente: http://www.ovation.co.uk/video-stabilization.html

La estabilización de video es un proceso previo necesario para procesamientos como segmentación, reconocimiento, identificación o seguimiento de objetos. La tesis busca la representación de personas como objetos dinámicos dentro de los videos. Su importancia radica en que estos procesos necesitan que la información sea coherente y libre de distorsiones espaciales para dar lugar a resultados de alta precisión [8, 9, 10].

- Reconocimiento e Identificación

Las imágenes presentan regiones individuales las cuales representan objetos o patrones. El reconocimiento de objetos busca encontrar patrones característicos pertenecientes a una clase predeterminada de objeto para encontrarlo en una imagen. La identificación, además de reconocer la clase del objeto, le proporciona una identidad, en el caso de personas puede ser un nombre, edad, género, etc. La figura 1.3 muestra una fotografía con reconocimiento y otra con identificación de rostros



A B

Figura 1.3. A. Reconocimiento de rostros en cámara digital, B. Identificación de personas con asignación de nombres propios.

Fuentes:
https://www.techinasia.com/5-cool-face-recognition/
http://www.sony-asia.com/article/271940/section/product/product/dsc-wx1

- Segmentación

El proceso de segmentación busca separar las imágenes en objetos coherentes mediante la agrupación de píxeles con características similares o en base a cambios bruscos de intensidad. La segmentación es un proceso cuyo nivel de detalle depende de la aplicación. Para el caso de segmentación de personas, se puede determinar cuántas personas hay en una imagen [1]. Las figura 1.4 muestra el resultado de aplicar segmentación a una imagen.

A B

Figura 1.4. A. Imagen de dos personas, B. Imagen segmentada de A

Fuente: C.Peñaloza, Introduction to Computer Vision and Machine Learning. Lecture 5. Segmentation and Motion

- Seguimiento de objetos

El seguimiento de objetos busca representar movimiento de estos en dos y tres dimensiones. Es utilizado para predicción de movimiento, identificación de eventos, estructuración de movimiento y monitoreo de colas. Este se consigue encontrando la correspondencia de las características o features de los objetos entre los cuadros, representados de manera simplificada o *sparse*. La figura 1.5 muestra el resultado de aplicar seguimiento de objetos durante una secuencia de imágenes.

Figura 1.5. Tracking de personas

Fuente: http://crcv.ucf.edu/courses/CAP5415/Fall2013/Lecture-10-KLT.pdf

1.1 Aplicaciones

La detección de personas en aplicaciones de seguridad, control de tráfico, navegación, etc., debe ser rápida y confiable. A continuación se mencionan algunas aplicaciones.

1.1.1 Seguridad

En temas de seguridad, la detección y reconocimiento de personas es vital para tomar acciones rápidas. En seguridad ciudadana, identificar personas que cometan delitos o infracciones resulta relevante y es un proceso que podría ser mejorado e incluso automatizado [11]. Otra aplicación se da en seguridad vial. Los vehículos del futuro son capaces de conducirse solos, pero la seguridad tanto de pasajeros como transeúntes debe estar garantizada. Los automóviles deben ser capaces de reconocer y tomar acciones inmediatas si alguna persona o algún objeto a traviesa su camino [10]. Por lo tanto estos deben poseer cámaras que monitoreen su trayectoria. Estos sistemas requieren de estabilizadores que reduzcan las oscilaciones en las cámaras debido al terreno y a la frecuencia de oscilación del propio vehículo.

1.1.2 Navegación de robots

La estabilización de imágenes puede utilizarse en aplicaciones de navegación de robots. En temas de robótica para que un robot pueda ser considerado inteligente, este tiene que tomar decisiones dependiendo de las condiciones del medio. Para el entrenamiento de los robots, estos necesitan obtener información mediante cámaras y sensores para analizar su entorno, pero esta debe estar libre de distorsiones. Un robot inspector que viaja por los túneles de las minas para revisar su estado tiene que enfrentarse a un terreno hostil lleno de obstáculos, por lo que su andar presenta movimientos bruscos y aleatorios [8]. Un video estabilizado es importante porque le da al inspector información detallada sobre el estado de la mina.

Las características atmosféricas pueden ser causantes de variaciones (temperatura, humedad, presión, etc.) espacio temporales en las imágenes capturadas por aeronaves [9]. Además, el corregir estos movimientos puede resultar perjudicial si se confunde el movimiento de objetos con el movimiento de la escena.

1.1.3 Entretenimiento

En lo que se refiere a entretenimiento y comunicaciones, los dispositivos móviles como Smartphone, Tablet, iPads, PDAs, etc., cuentan con video cámaras las cuales permiten a los usuarios grabar videos en cualquier momento [7]. El avance tecnológico y la disminución de precios de estos aparatos han permitido que las ventas de estos se incremente. En [29] se informa que el 78% de jóvenes latino-americanos tiene un Smartphone. En el Perú para del año 2012 al 2013 la importación de estos equipos pasó del 20% al 35% lo que significa que 3 de 10 teléfonos móviles son Smartphone. Además este porcentaje se incrementará entre 40 y 45% al final de este año. [30].

Hoy en día, las cámaras son baratas, ligeras y fáciles de usar, por lo que están presentes en celulares, autos, computadoras y prendas tecnológicas o ("wearables"). Por ejemplo existen cámaras como las GoPro [28], las cuales permiten grabar videos en primera persona (first-person cameras). Estas cámaras no se manipulan con las manos como las videocámaras convencionales. Estas pueden aferrarse a cascos o arneses, lo que permite grabar mientras se practican deportes como surf, skiing, montañismo, paracaidismo, deportes de aventura, etc. como se muestra en la figura 1.6.

Figura 1.6. Cámara GoPro instalada en casco de protección.

Fuente: http://skitheworld.com/2011/10/pov-promoting-your-skiing/

1.2 Problemática

El sistema de estabilización debe procurar reducir el tiempo de ejecución sin descuidar la calidad en la detección. Sin embargo, durante la adquisición de los videos se presentan factores que perjudican la etapa de detección. Algunos factores se mencionan a continuación.

1.2.1 Distorsiones espaciales

El corrimiento de imágenes debido al efecto jitter puede ser representado por transformaciones afines, entre las que se encuentran los cambios de escala, rotación y traslación, las cuales son las transformaciones lineales más simples. La figura 1.7 muestra ejemplos de estos tipos de distorsión.

A B C

Figura 1.7. A. Imagen Original, B. Distorsiones Espaciales: Rotación, Traslación y Escala, C. Imagen con jitter.

1.2.2 Cambios de iluminación

Los cambios de iluminación perjudican el performance de los sistemas de estabilización dado que afectan la etapa de detección de puntos característicos [7]. Este problema puede presentarse de manera gradual o cambiar radicalmente. La figura 1.8 muestra cómo se presenta el cambio gradual en una secuencia de imágenes.

Figura 1.8. Imagen con distintos valores de iluminación.

Fuente: C. Peñaloza, Introduction to Computer Vision and Machine Learning. Lecture 1. Introduction to Computer Vision

1.2.3 Oclusión

La oclusión de objetos es un problema que surge cuando un objeto desaparece de la imagen, esto debido a que sale de la toma, se oculta o es tapado por otro objeto. La pérdida parcial o total de los objetos en las imágenes perjudica de manera significativa dado que se pierde información durante el seguimiento de puntos característicos. La figura 1.9 muestra como los objetos de un video pueden ocultarse o ser ocultados.

Figura 1.9. Oclusión de Objetos

Fuente: C. Peñaloza, Introduction to Computer Vision and Machine Learning. Lecture 1. Introduction to Computer Vision

1.2.4 Paralaje

Paralaje es una transformación de perspectiva de una imagen producida por el movimiento del observador o por el movimiento de lo observado, que produce un efecto visual de cambio de profundidad y distancia [12]. Mientras más cerca se encuentre el objeto, se verá más grande y si el observador se mueve, su movimiento será más rápido.

Por otro lado si el objeto se encuentra lejos del observador, su tendrá un tamaño más pequeño y su movimiento en caso el observador se mueva, será lento. El efecto de paralaje resulta ser problemático cuando se tiene la necesidad de llevar los objetos detectados en un plano de dos dimensiones a uno de tres. En el Anexo se incluye una animación que describe este efecto.

Debido a que las personas son objetos no-rígidos, los cuales cambian constantemente de estructura, en lo que se refiere a la detección, se debe tener cuidado para lograr distinguirlas. Una transformación que modifica a toda una imagen perdería la información del movimiento particular de los objetos.

1.3 Síntesis del capítulo

A lo largo del capítulo 1 se presentaron las diversas aplicaciones en donde se puede aplicar la estabilización de video, así como los factores que perjudican su performance.

- El efecto jitter es un ruido que se adhiere a los videos debido a movimientos indeseados durante su adquisición.

- La estabilización de video es un pre procesamiento necesario para procesos como segmentación, reconocimiento, identificación y seguimiento de objetos.

- Las aplicaciones que requieren videos estabilizados pueden encontrarse en seguridad, industria, entretenimiento, etc.

- Durante la grabación de los videos no solo el efecto jitter puede afectar su calidad, existen también problemas con cambios de iluminación, oclusión y paralaje.

En el capítulo 2, se presentarán las metodologías existentes para la estabilización de videos. Se mencionarán las investigaciones realizadas para estabilizar videos y se explicarán los pasos a seguir para representar objetos en 3D con Structure From Motion y Geometría Epipolar.

Capítulo 2

Metodologías de Estabilización de Video

La estabilización de video puede aplicarse por medios mecánicos y por medios digitales, a continuación se presentarán algunos ejemplos de estas opciones. Se mencionarán las investigaciones realizadas para estabilizar videos y se explicarán los pasos a seguir para representar objetos en 2D con Registro de Imágenes y en 3D con Structure From Motion y Geometría Epipolar.

2.1 Soluciones Mecánicas

Las alternativas mecánicas hacen referencia a equipos, sistemas o mecanismos que solucionan el problema del jitter mediante la prevención de movimientos involuntarios o la corrección de la posición de la cámara.

2.1.1 Sistemas de corrección de movimiento

En la industria del cine, una plataforma móvil manejada por dos personas conocida como Dolly Cam, permite la grabación de videos en trayectos predeterminados separando el movimiento de la cámara del movimiento del operador. En los últimos años, se han fabricado cámaras de video sofisticadas que permiten evitar movimientos involuntarios. Algunos ejemplos son la SteadiCam o las cámaras GoPro. Algunas de estas alternativas de solución se muestran en las figuras 2.1.

A B

Figura 2.1. A. SteadiCam, B. Cámaras GoPro

Fuentes: http://www.neogaf.com/forum/showthread.php?t=535197
http://fr.gopro.com/

2.1.2 Limitaciones

Estas opciones eliminan o reducen el jitter previniendo o corrigiendo in situ el movimiento indeseado, pero tienen la limitación de ser opciones costosas, difíciles de implementar o manipular [19, 22]. Los dispositivos mecánicos mencionados no se pueden adaptar a todos los sistemas de adquisición de datos por lo que resultan ser soluciones específicas para determinados casos. Además estas son soluciones on-line o en vivo, sólo se pueden corregir videos cuando estos se están grabando, por lo tanto los videos adquiridos previamente no pueden ser corregidos.

2.2 Soluciones Digitales

Las alternativas digitales hacen referencia a la corrección de la posición de las imágenes mediante procesamiento digital a través de un computador. Las metodologías existentes dentro del procesamiento de videos dependen de las características de los videos. Los algoritmos existentes pueden solucionar uno de dos problemas, ya sea robustez frente a transformaciones o rapidez en procesamiento [2, 3]. Las alternativas de soluciones que se presentarán a continuación son: Registro de Imágenes, Structure From Motion y Geometría Epipolar.

2.2.1 Registro de Imágenes

El registro de imágenes consiste en encontrar un modelo de transformación que lleve una imagen de partida (input) a una de llegada (output). Para ello se tienen dos alternativas: Registro Directo y Registro Basado en Características. En el registro directo se busca llevar toda la información disponible en la imagen a la imagen de salida. Estas metodologías son precisas pero son computacionalmente costosas. Por otro lado, las metodologías basadas en características particulares de las imágenes, conocidas como features, son más eficientes además de presentar mejores resultados que las metodologías directas, por lo que son más utilizadas [2].

A continuación se mencionan los pasos a seguir para el registro de imágenes además de nombrar algunos algoritmos.

2.2.1.1 *Detección de Features*

Una imagen presenta características relevantes conocidas como *features*, las cuales pueden representar a esta de una manera simple, o *sparse*, con la cual se pueden realizar aplicaciones como reducir el peso de las imágenes, mejorar la transmisión de estas o proporcionar invarianza frente a transformaciones y cambios de intensidad [2]. Los algoritmos de detección pueden clasificarse según el tipo de feature que se detecta. Usualmente los son puntos, líneas, o regiones. Los puntos son los más utilizados, dado que con líneas y regiones finalmente se encuentran puntos (cruces y centroides respectivamente) [5].

En 2005, se realizó un recuento de los detectores de features más destacados, explicando las aplicaciones, características, ventajas y desventajas de cada uno [16].

- Detector Harris

Entre los algoritmos más robustos frente a cambios de escala y rotación se tienen al detector Harris, basado en el detector implementado por Harris y Stephens [14].

El detector de Harris evalúa la existencia de un punto por medio de la gradiente de la imagen. Para ello se conforma la Matriz de estructura local M a partir de las derivadas parciales de la imagen $I_{(u,v)}$ en las direcciones vertical y horizontal.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Donde

$$A_{(u,v)} = I^2_{x(u,v)}$$

$$B_{(u,v)} = I^2_{y(u,v)}$$

$$C_{(u,v)} = I_{xy(u,v)}$$

Luego, cada una de las componentes de la matriz *M* se convoluciona con un filtro gaussiano obteniendo una nueva matriz $\bar{M}$. Esta matriz es utilizada para conformar la función de respuesta de punto $Q_{(u,v)}$.

$$Q_{(u,v)} = (\bar{A}\bar{B} - \bar{C}) - \alpha\,(\bar{A} - \bar{B})^2$$

Donde $0.4 < \alpha < 0.6$

Finalmente se evalúa si se trata de un punto mediante la comparación con un umbral y con los valores del vecindario.

$Q_{(u,v)} > Th$

Donde $10000 < Th < 1000000$

$Q_{(u,v)} es\ máximo\ local$

Si se cumplen las dos condiciones entonces el punto evaluado sí es considerado.

- Diferencia de Gaussianas (DoG) [17]

Esta metodología consiste en generar pirámides de imágenes que aumentan en escala. Luego se realiza una resta para encontrar los puntos máximos para un vecindario establecido. La principal ventaja del DoG es que es una simplificación del Laplaciano de Gaussiano (LoG), por lo que reduce los costos computacionales [1].

Para obtener una representación de una imagen $I_{(x,y)}$ en espacio escala $L_{(x,y,\sigma)}$, se realiza una convolución (*) en x y en y con un filtro gaussiano $G_{(x,y,\sigma)}$.

$$L_{(x,y,\sigma)} = G_{(x,y,\sigma)} * I_{(x,y)}$$

Donde

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

La diferencia de Gaussianos $D_{(x,y,\sigma)}$ se obtiene mediante la diferencia de dos imágenes espacio-escala separadas un factor k:

$$D_{(x,y,\sigma)} = L_{(x,y,k\sigma)} - L_{(x,y,\sigma)}$$

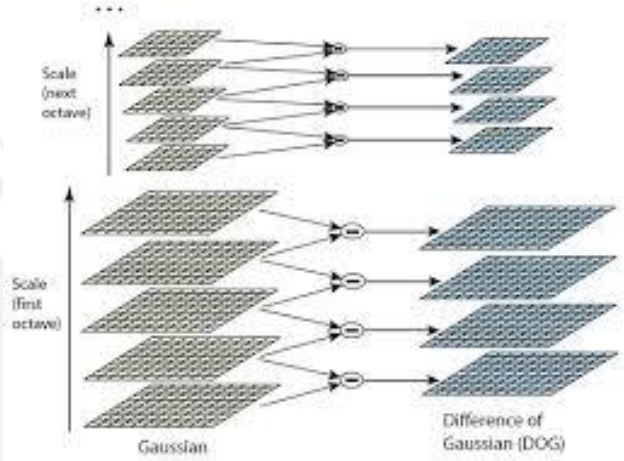


Figura 2.2. SIFT: Detección de puntos por Diferencia de Gaussianos.

Fuente: Lowe [17]

Finalmente se para la evaluación de los puntos se compara la intensidad del punto candidato con el valor de sus 8 vecinos, con los 9 vecinos de una escala posterior y 9 de una escala anterior.

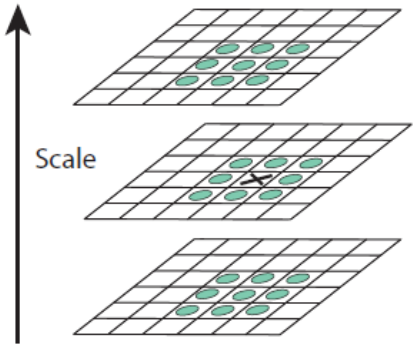


Figura 2.3. SIFT: Evaluación de puntos máximos en vecindario local

Fuente: Lowe [17]

2.2.1.2 *Descriptores*

- SIFT (Scale-Invariant Features Transform)

El descriptor más conocido es SIFT ó Scale-Invariant Features Transform [17]. Entre las propiedades de SIFT se destacan su invarianza frente a cambios de escala, rotación, iluminación, además de su robustez frente a ruido aditivo. El principal problema de SIFT es el tiempo de procesamiento. No es eficiente en objetos planos (objetos ricos en texturas) y frente a distorsión muy severa.

La primera etapa de SIFT consiste en la detección de puntos de interés mediante DOG. La segunda etapa consiste en obtener la magnitud $m(x,y)$ y orientación $\theta(x,y)$ de la gradiente para la imagen filtrada $L(x,y)$.

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

La invarianza frente a rotación se consigue mediante la rotación de las coordenadas del descriptor y la orientación de la gradiente relativa a la orientación del punto. Luego se aplica un filtro gaussiano representado por un círculo en la figura. Finalmente es conforma un vector descriptor que contiene a todos los valores del histograma. El mejor resultado se obtiene para un arreglo de 4 x 4 de histogramas de 8 bins. Por lo tanto el vector descriptor de cada punto estará conformado por 128 elementos.

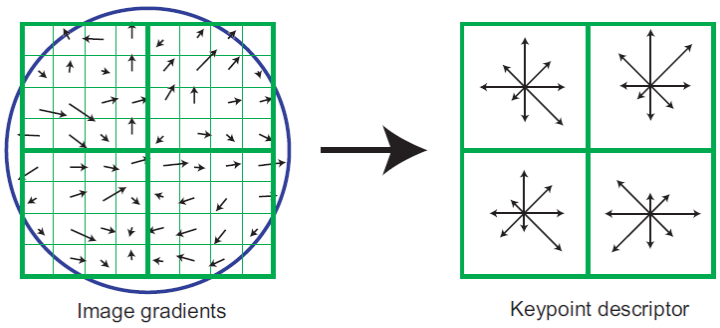


Figura 2.4. A. SIFT: Detección de puntos mediante Diferencia de Gaussianos, B. SIFT: Descriptor basado en gradientes.

Fuente: Lowe [17]

- KLT (Lukas Kanade Tracker)

El algoritmo KLT [18], busca encontrar el movimiento de los puntos de interés a través de una secuencia de frames. Para ello el primer paso es la detección de los puntos. Luego se busca un modelo de transformación para encontrar el movimiento que se hay entre frames, mediante una minimización de la diferencia entre la imagen de llegada y la transformación de la imagen actual.

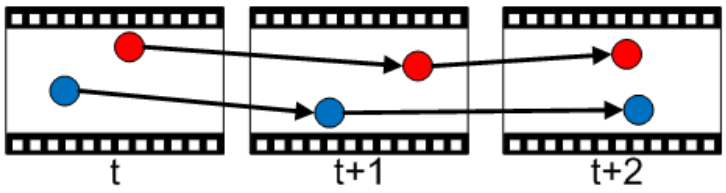


Figura 2.5. Puntos detectados a lo largo de una secuencia.

2.2.1.3 *Modelamiento y Transformación*

Es necesario relacionar las imágenes mediante un modelo de transformación. Los modelos de transformación ya sean lineales o no lineales se describen en [3, 4]. Lo más usual es considerar un modelo lineal de transformación de afinidad, dado que contiene transformaciones usuales como de rotación y de escala y es mucho más fácil de trabajar que si se asumiese un modelo no lineal. Luego de tener el modelo más cercano de la transformación entre cuadros, se procede a encontrar la matriz de transformación.

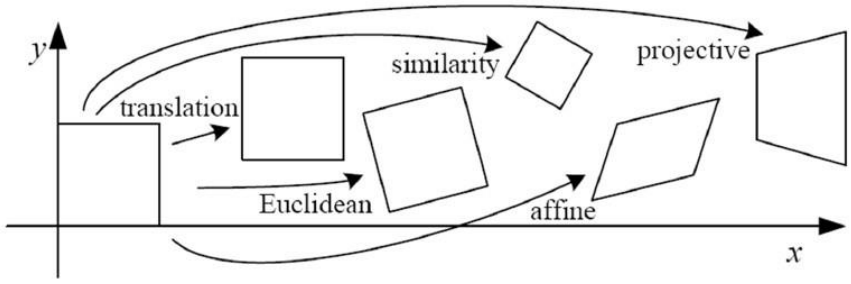


Figura 2.6. Modelos de Transformaciones Lineales

Fuente: Szeliski [3]

La figura 2.4 muestra el resultado luego de una transformación afín.

Figura 2.7. A. Imagen Original "Bote", B. Imagen A luego de transformación de afinidad.

Fuente: http://homepages.inf.ed.ac.uk/rbf/HIPR2/libboa1.htm

2.2.1.4 *Matching*

Luego de encontrar el descriptor de las imágenes se debe encontrar las correspondencias entre los puntos de las imágenes de llegada y las de salida. Para ello se utilizan algoritmos que buscan comprobar que estos realmente puedan encontrarse en la siguiente imagen. Es por ello que se utilizan algoritmos como RANSAC (Random Sample Concensus) [21] para distinguir los resultados acertados, conocidos como inliers, de los erróneos, o outliers. RANSAC es un algoritmo que busca identificar inliers y outliers mediante la búsqueda del modelo que mejor describa el grupo de datos. El primer paso de la metodología es elegir al azar un grupo de datos. Luego se busca el modelo que mejor describa. Se elige otro grupo aleatorio y se calcula el modelo. Este se compara con el modelo hallado anteriormente y se busca el que modelo que mejor describa teniendo un error mínimo. Este proceso se repite un número máximo N veces determinado por el usuario o al encontrar la representación más óptima.

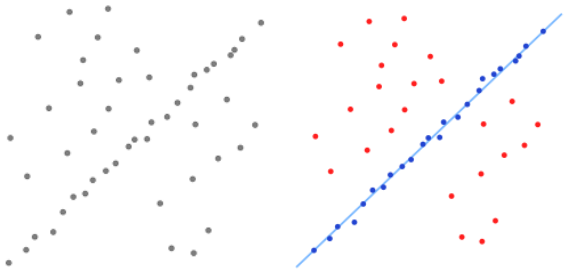


Figura 2.8. RANSAC: búsqueda del modelo que mejor represente al sistema.

Fuente: http://robotica.unileon.es/mediawiki/index.php/PCL

2.2.1.5 *Remuestreo*

Finalmente, conocidas la representación de la imagen y la transformación, se procede a llevar la imagen al espacio que le corresponde. El remuestreo utiliza los puntos de la imagen hallados inicialmente y mediante interpolaciones se puede obtener la imagen completa. Entre los métodos de remuestreo más conocidos se tienen a Vecino más cercano e Interpolación Bilineal.

- Vecino Más Cercano

El remuestreo por vecino más cercano consiste en encontrar la correspondencia para el punto de la imagen resultado en la imagen fuente, o referencia. Para ello las coordenadas del punto en la imagen de llegada (x_o, y_o) son llevadas a la imagen de partida (x_i, y_i) mediante la inversa de la función de Transformación que relaciona entrada y salida. En la imagen de partida las coordenadas (x_i, y_i) pueden no se enteras, por lo tanto al punto (x_o, y_o), se le asigna la intensidad del vecino más cercano del punto en donde cayó en la imagen de partida, es decir del valor entero más cercano de x_i e y_i [5].

- Interpolación Bilineal

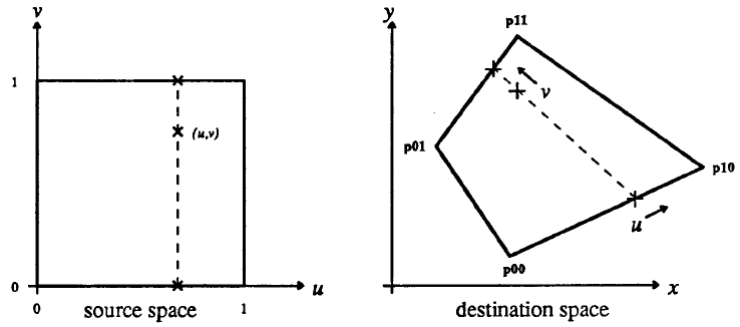


Figura 2.9. Interpolación Bilineal

Fuente: Heckbert [23]

Con la interpolación bilineal se puede determinar la posición de un punto respecto a una celda en la imagen destino. Esto mediante la ecuación:

$$(x, y) = (1-u)(1-v)p_{00} + u(1-v)p_{10} + (1-u)vp_{01} + uvp_{11}$$

Utilizando notación matricial

$$(x\ y) = (uv\ u\ v\ 1)\begin{bmatrix} a & e \\ b & f \\ c & g \\ d & h \end{bmatrix}$$

$$p_e = W\ A$$

Donde

$$A = \begin{bmatrix} a & e \\ b & f \\ c & g \\ d & h \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} V1r & V1c \\ V2r & V2c \\ V3r & V3c \\ V4r & V4c \end{bmatrix}$$

$$A_c = Z\ V_c$$

2.2.2 Structure From Motion

Structure From Motion busca construir un modelo tridimensional a partir una escena capturada en muchas imágenes mientras minimiza un error de re proyección, el cual consiste en una minimización no lineal robusta de los errores de medición [2].

A B

Figura 2.10. A. Adquisición de varias vistas del objeto. B. Reconstrucción del objeto en tres dimensiones.

Fuente: C. Peñaloza, Introduction to Computer Vision and Machine Learning, Lecture 5. Segmentation and Motion

La entrada del algoritmo es un grupo de imágenes con puntos de correspondencia. La salida es la ubicación tridimensional de los puntos, además de los parámetros de movimiento de la cámara.

SFM busca minimizar la suma de re proyecciones mediante Bundle Adjustment. El cual utiliza la información de las vistas y relaciones geométricas para refinar la posición tridimensional de los puntos. SFM es utilizado para Modelado 3D, Navegación de robots, creación de mapas, o efectos Visuales [2].

Las principales limitaciones de SFM son la complejidad computacional y la restricción de que los objetos tienen que ser estáticos.

2.2.3 Frame Warping

La etapa de warping consiste en deformar una imagen de entrada a una de salida, mediante la minimización de energías de similaridad y distorsión local. Content Preserve Unwarping consiste en una técnica de warping diseñada especialmente para estructuras 3D [22]. Este algoritmo recibe como entradas los puntos de control en las imágenes de partida y de llegada, los cuales guardan una relación tridimensional, la cual podría obtenerse con SfM. Establece una grilla con vértices V en la imagen de partida, los cuales se mueven para formar una nueva grilla en la imagen de llegada. Para ello se deben minimizar las energías de data y similaridad.

Además se necesita establecer un modelo de movimiento de cámara pre definido.

Sus resultados son modificados para dar resultados visualmente aceptables. Pero estos carecen de información espacial físicamente coherente.

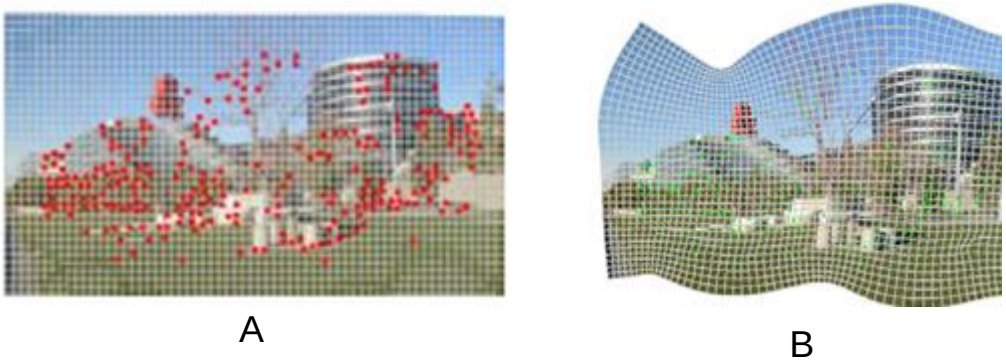


Figura 2.11. A. Grilla en imagen de partida, B. Grilla modificada en imagen de llegada.

Fuente: Liu [22]

$$E = E_d + \alpha E_s \qquad E_d = \sum_k \left\| w_k^T V_k - P_k \right\|^2$$

$$E_s(V_1) = w_s \| V_1 - (V_2 + u(V_3 - V_2) + vR_{90}(V_3 - V_2)) \|^2$$

$$donde\ R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

2.2.3.1 *Término Data*

El término de data minimiza la distancia entre la salida del punto proyectado P_k y la ubicación interpolada entre la celda de salida correspondiente a la celda de entrada la cual contiene a P_k. Con interpolación bilineal podemos hallar una relación entre un punto P_k y los vectores V_k en la celda que lo contiene [23]:

$$P_k = w_k^T V_k \qquad donde\ w_k = \begin{bmatrix} (1-u)(1-v) \\ (1-u)\,v \\ (1-v)\,u \\ u\,v \end{bmatrix}$$

Para minimizar todos las distancias:

$$E_d = \sum_k \left\| w_k^T V_k - P_k \right\|^2$$

2.2.3.2 *Término de Similitud*

El término de similitud mide la desviación de cada celda de la grilla de salida desde una transformación de similaridad a su correspondiente grilla de entrada. Una celda es separada en dos triángulos y se encuentra la distancia de un vértice con respecto al lugar en donde debería estar según una trasformación de similaridad.

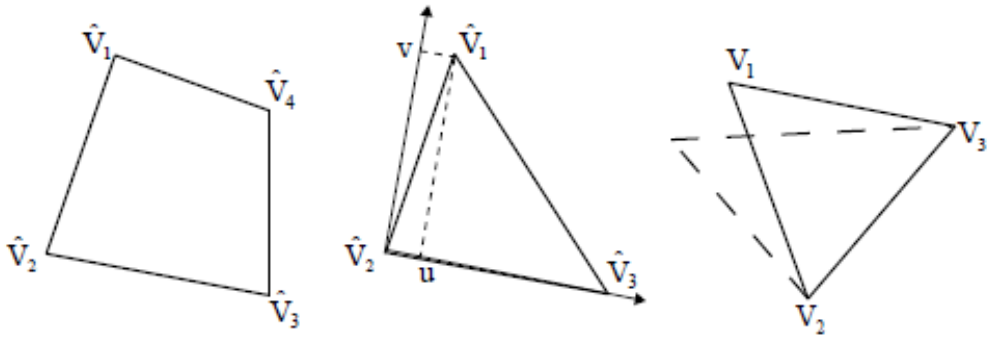


Figura 2.12. Relaciones de similaridad entre vértices

Fuente: Liu [22]

Se puede obtener la posición del vértice V_1 a partir de V_2 y V_3. Dado que el punto hallado no coincide exactamente con V_1, se minimiza la distancia entre ellos.

$$V_1 = V_2 + u(V_3 - V_2) - vR_{90}(V_3 - V_2), \quad donde\ R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

En el término de similitud se tienen dos parámetros más: $\alpha\ y\ ws$

2.2.4 Geometría Epipolar

La geometría epipolar hace referencia a las relaciones de proyecciones geométricas entre dos vistas. Estas relaciones se encuentran en una matriz de 3x3 conocida como matriz fundamental. Un punto P en el espacio puede ser visto por dos cámaras con centros O_i y O_r. Las imágenes capturadas por las cámaras forman un plano cada una π_i y π_r respectivamente. El punto P es visto en π_i como x y x' en π_r.

La línea epipolar (l) es la intersección de un plano epipolar con el plano de una imagen. Todas las líneas epipolares es intersectan en el epipolo. El plano epipolar intersecta a las vistas en líneas epipolares y establece la relación:

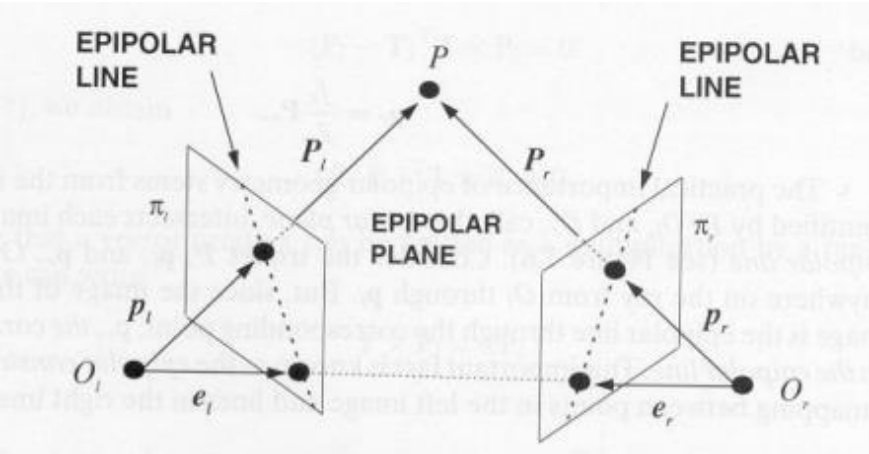


Figura 2.13. Relaciones Proyectivas en Geometría Epipolar

Fuente: Hartley [4]

$$x^{T\prime} F\, x = 0 \qquad l' = F\, x$$

2.2.4.1 *Matriz Fundamental*

Mediante Geometría Epipolar se pueden representar objetos tridimensionales sin necesidad de utilizar la información tridimensional de los objetos, evitando los costos computacionales que se generan al trabajar con esta [4, 19]. Esto se logra utilizando proyecciones de los puntos obtenidos en los frames pasados y futuros imágenes.

- *Algoritmo de 8 puntos*

Vectores Aumentados para la representación de un mismo punto en dos vistas (x y x')

$$\mathbf{x} = (x, y, 1)^T$$
$$\mathbf{x}' = (x', y', 1)^T$$

Se puede encontrar una función que relaciona las coordenadas del punto y los nueve valores de la matriz fundamental:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'^{f_{23}} + xf_{31} + yf_{32} + f_{33} = 0$$

$$(x'x.x'y,\ x',y'x,y'y,y',x,y,1)f = 0$$

$$A\,f = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} f = 0$$

- *RANSAC*

  Para N iteraciones

  Escoge 8 puntos aleatoriamente

  Calcula la Matriz Fundamental a partir del algoritmo de 8 puntos

  Compara si el modelo encontrado es mejor que el anterior, si lo es

  F = F actual

  Repite

2.3 Síntesis del Capítulo

- Considerar estabilización en 3D implica asumir una trayectoria para la cámara y modelar a los objetos en 3D a partir de movimiento en un video.
- Las metodologías basadas en features son mucho más veloces que las directas y pueden llegar a tener mejores resultados, pero dependen de la detección de los puntos de control.
- Los métodos tradicionales de descriptores SIFT y SURF no pierden vigencia, dada su exactitud y son tomados como referencia para comparaciones con nuevos métodos.
- Los métodos binarios son mucho más veloz pero decrecen en robustez frente a los cambios mencionados.
- Se utiliza RANSAC para la evaluación de la detección de los puntos. Esto permite que los resultados sean espacialmente coherentes dado que se eliminan los puntos que no representan el modelo o también conocidos como outilers.

Capítulo 3

Diseño del algoritmo de Estabilización de Video

En este capítulo se explica, mediante diagramas, esquemas e imágenes, el diseño del algoritmo de estabilización de video. En el Anexo 1 se encuentran los pseudo-códigos correspondientes. Para comenzar se plantean los objetivos para el presente trabajo, posteriormente se describen los videos de entrada del sistema, mencionando las características y los requerimientos, además de mencionar las características de implementación. A continuación se muestra el diagrama de flujo general del sistema. Luego se entra a detalle en cada una de las etapas del proceso. Finalmente se presenta una síntesis del capítulo.

3.1 Objetivos

En base a los problemas vistos hasta ahora, se proponen alcanzar los siguientes objetivos:

Objetivo General

Diseñar un algoritmo de estabilización de video orientado a la detección de personas, eficiente y robusto ante cambios de rotación, escala e iluminación.

Objetivos Específicos

• Realizar un estudio del estado del arte de la estabilización de video.

• Diseñar de un método robusto y eficiente frente a movimientos involuntarios de cámara y cambios de iluminación.

• Implementar el algoritmo en un lenguaje de alto nivel.

3.2 Consideraciones de diseño

El método a implementar debe ser invariante ante cambios rotacionales y de escala, de modo que se mejore la calidad visual de los videos. Los problemas de oclusión (pérdida parcial o total de los objetos) no serán considerados para este trabajo. El costo computacional no será de interés.

3.2.1 Videos de entrada

Los videos utilizados han sido descargados de la página web de Goldstein y Fattal [19], y simulados en MATLAB. Para dar resultados cuantitativos se opta por la creación de videos sintéticos en donde se simule los movimientos de cámara.

Características de los videos:

Formato:	.avi, .mp4 ó .wmv
Duración:	8 – 30 segundos
Resolución:	640 x 320 a 1280 x 720 pixeles
Peso:	6 - 30 MB

Restricciones de los videos:

- No debe haber oclusión (pérdida parcial o total de objetos en el video).
- No deben tener cambios de iluminación y movimiento de cámara muy severos (no se tendrán suficientes puntos de control).
- Videos con objeto móvil que abarca toda la toma (mayoría de puntos detectados son móviles).

3.3 Diseño del algoritmo de Estabilización de Video

Por lo expuesto anteriormente, para diseñar un algoritmo de estabilización que obtenga el modelo tridimensional de las personas detectadas, se utiliza Geometría Epipolar y registro de imágenes. La figura 3.1 muestra el diagrama de flujo seguido.

3.3.1 Diagrama de Flujo General

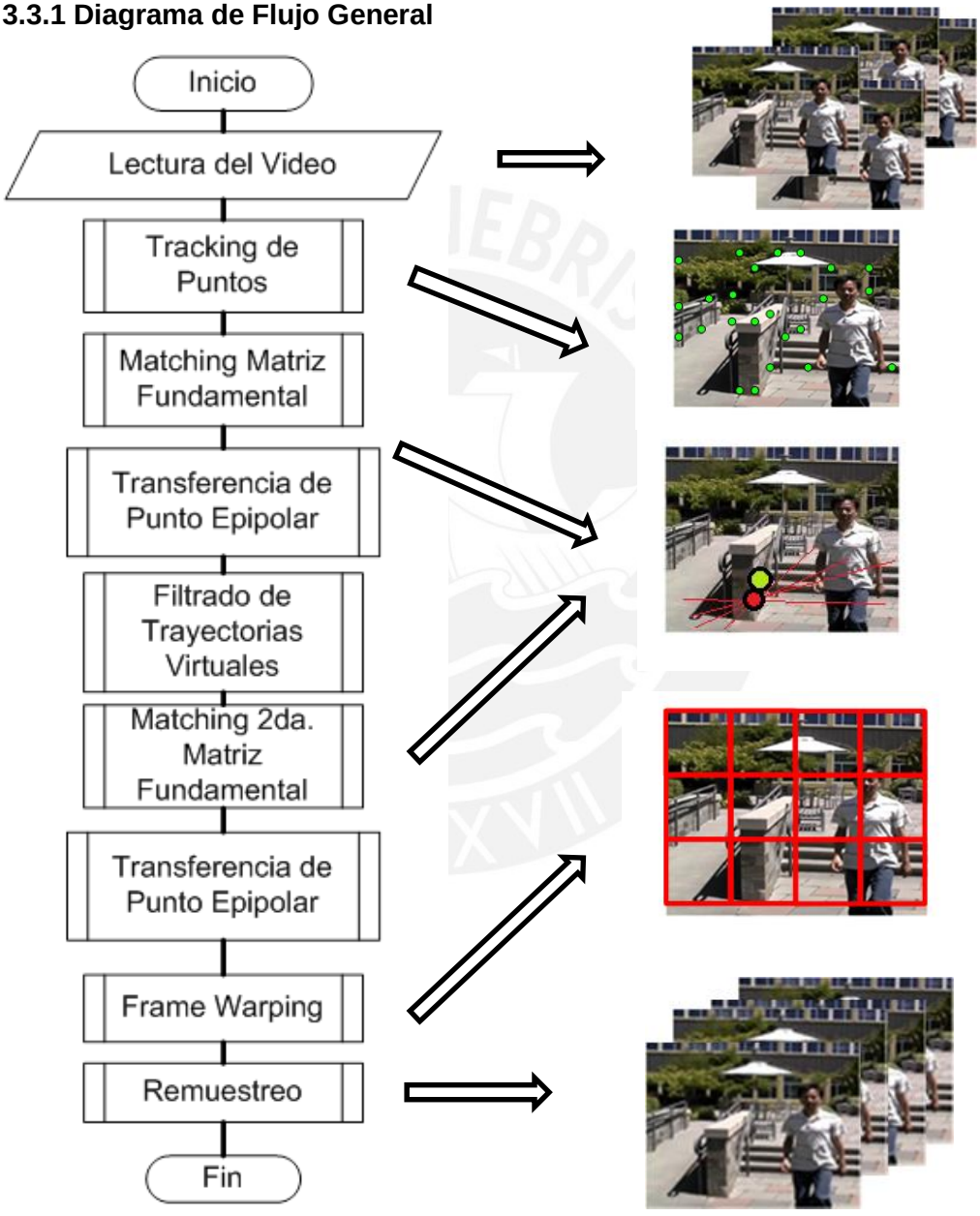


Figura 3.1. Diagrama de Flujo

Nuestro sistema recibe como entradas videos afectados por jitter. A la salida se obtienen videos con movimientos más sutiles, en los cuales el jitter ha sido eliminado o reducido. La etapa de detección recibe una imagen y retorna una representación simplificada o sparse. Para obtener las matrices fundamentales se utilizan el algoritmo de 8 puntos y RANSAC. Con las matrices halladas se utiliza la Función de Transferencia Epipolar en dos ocasiones para encontrar los puntos que corresponden a las imágenes corregidas. En la última etapa se realiza un remuestreo de la imagen utilizando Frame Warping.

A continuación se describen las tres partes principales del algoritmo. Luego de la descripción de cada una de las etapas, se presenta un esquema o pseudocódigos que describen cada etapa.

3.3.2 Detección y Seguimiento de puntos

3.3.2.1 *Esquema de Seguimiento de puntos*

La figura 3.2 muestra los pasos de la detección y seguimiento de puntos.

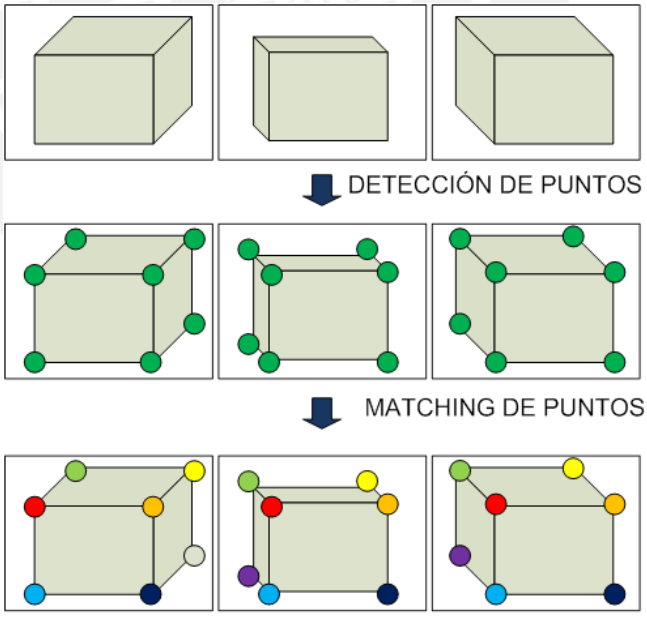


Figura 3.2. Seguimiento de Puntos: Detección y Matching de Puntos

3.3.2.2 *Detección y Tracking*

Para esta etapa se utilizará el programa Voodoo Tracker [26], el cual nos permite encontrar trayectorias de puntos en videos. Se elige detección y tracking por Harris y KLT debido a que las distorsiones presentes de frame a frame no son tan severas.

Entradas: Secuencia de Imágenes

Salidas: Archivos .pnt

Los archivos .pnt contienen la información relevante para el algoritmo de detección. Tiene una bandera que indica si el punto fue detectado en el frame anterior, las coordenadas de este en el frame actual y pasado. Se obtiene un archivo. pnt por cada frame del video. La información se muestra en la tabla 3.1.

1	x	Coordenada Actual de Columna
2	y	Coordenada Actual de Fila
3	Manual	Punto detectado manualmente (1: Sí, 0: No)
4	type 3D	Tipo de Punto (0: No es 3D, 1: Esférico, 2: Cartesiano)
5	Px	Coordenada 3D: X
6	Py	Coordenada 3D: Y
7	Pz	Coordenada 3D: Z
8	Ident	ID del punto
9	Hasprev	¿Fue detectado en el Frame Anterior? (1: Sí, 0: No)
10	Pcx	Coordenada de Columna en Frame Anterior
11	Pcy	Coordenada de Fila en Frame Anterior
12	Support	Máscara Inlier/Outlier

Tabla 3.1. Información de un archivo .pnt

3.3.3 Trayectorias Útiles

Durante un video, los puntos detectados para un frame determinado pueden mantenerse o desaparecer en el siguiente. En los archivos .pnt se tiene un número P de puntos definidos por lo que se tiene un índice P de puntos iniciales. Si un punto aparece y se mantiene, la información efectivamente le corresponderá. Si este desaparece ocurre un salto en el índice del archivo .pnt. Si el punto p en el índice i es detectado en el frame f, pero en el frame f + 1 no es detectado, el punto p' que en el frame f fue detectado en el índice i+1 ocupará su lugar, cambiando del índice i+1 a i, siempre y cuando no hayan desaparecido más puntos antes del índice i.

El algoritmo buscará trayectorias de puntos que se mantengan por 20 frames como mínimo. Se establece la condición:

- El ID del punto debe mantenerse. (ítem 8 del archivo .pnt).

Para encontrar las trayectorias útiles para lo que resta del algoritmo se definen dos máscaras. Las máscaras indicarán la presencia o ausencia de trayectorias, así como la longitud de las trayectorias detectadas. La matriz de salida contiene información sobre las trayectorias detectadas.

Máscara de Trayectoria:

Dimensiones:	P x F	P: Nro. De Puntos, F: Nro. De Frames
Valores:	0.	Punto no encontrado
	0.5	Punto encontrado, Trayectoria $\leq$ 10 Frames
	1.	Punto encontrado, Trayectoria $>$ 10 Frames

Máscara de Coherencia Temporal

Dimensiones:	P x F	donde P: Nro. De Puntos F: Nro. De Frames
Valores:	0	$t < ta$ ó $te < t$
	0.02 t	$ta \leq t < ta + T$
	1	$ta + T \leq t < te - T$
	0.02 t	$(t - te)\ te - T \leq t < te$

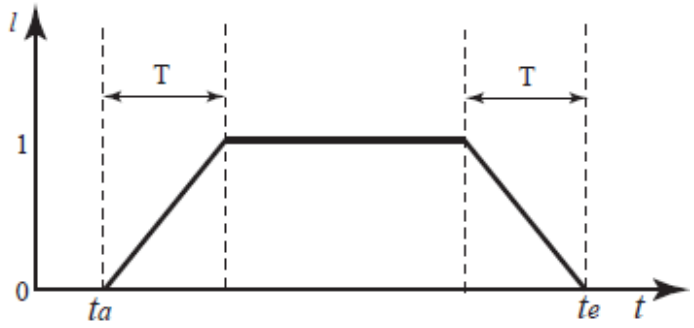


Figura 3.3. Función de Coherencia Temporal

Fuente: Liu [22]

La figura 3.3 muestra la función de coherencia temporal. Donde *ta*, *te* y T son Inicio de trayectoria, Final de trayectoria y Umbral.

Trayectoria (Matriz):

Dimensiones: T x 3 donde T: Nro. De Trayectorias detectadas

Parámetros: Inicio de Trayectoria
Fin de Trayectoria
Índice de Punto detectado

3.3.4 Función de Transferencia Epipolar

La Función de Transferencia Epipolar [19], consiste en utilizar las matrices fundamentales que relacionan un punto en un frame t con los puntos respectivos en frames pasados y futuros para encontrar un punto virtual en el frame actual.

3.3.4.1 *Esquema*

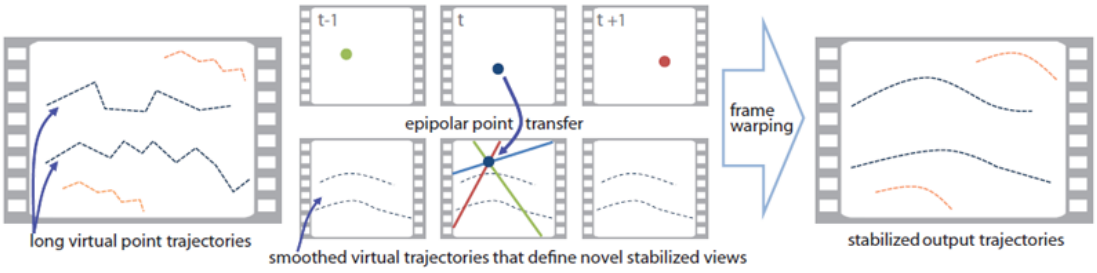


Figura 3.4. Función de Transferencia Epipolar

Fuente: Goldstein y Fattal [19]

Para ello sigue el siguiente esquema.

1. Encontrar matrices fundamentales y líneas epipolares
2. Filtrado de líneas epipolares
3. Intersecciones de líneas epipolares

3.3.4.2 *Primeras matrices fundamentales*

Durante esta etapa se encontrarán las matrices fundamentales que relacionan un frame con diez frames anteriores, luego se trazaran líneas epipolares para determinar las intersecciones y por último la posición correcta de los puntos [19]. Se encuentran las matrices epipolares correspondientes a diez frames anteriores al actual (frame t), como se muestra en la figura.

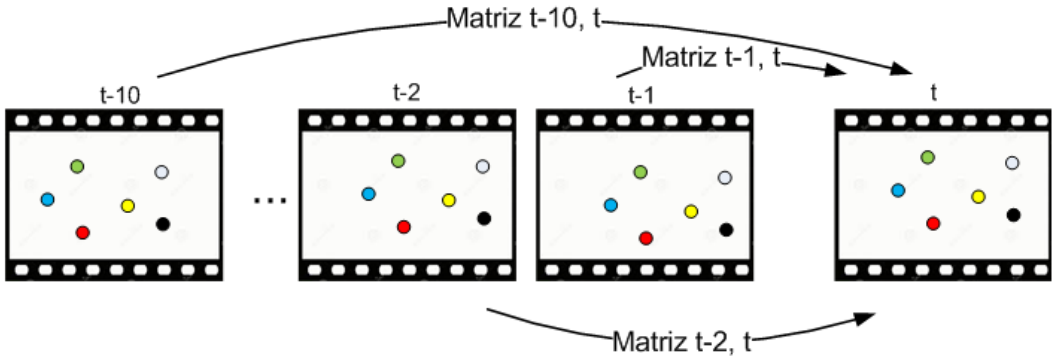


Figura 3.5. Matrices Fundamentales para frames pasados

El primer paso es utilizar los puntos que efectivamente siguen una trayectoria de al menos 10 frames. Para ello se utiliza la máscara como índice para filtrar los puntos que no cumplen la condición.

El segundo paso es formar las parejas de puntos correspondientes.

$$(P_{t-1}, P_t), (P_{t-2}, P_t), \dots, (P_{t-10}, P_t)$$

Utilizando la función *estimateFundamentalMatrix.m* la cual realiza el algoritmo de 8 puntos normalizado para encontrar una matriz fundamental y RANSAC para encontrar la Matriz óptima, por lo que sus parámetros de entrada a partir de los puntos, son el método elegido, la distancia mínima y el número de intentos
La salida será la matriz FM de 4 dimensiones:

FM: 3 x 3 x 10 x N N: Nro. De Frames – 10 frames iniciales
Las primeras 2 dimensiones corresponden a la matriz de 3 x 3. El tercer elemento indica a qué frame anterior le pertenece la matriz y el último el frame actual.

3.3.4.3 *Líneas Epipolares*

Una vez hallada la matriz epipolar correspondiente se procede a hallar las líneas epipolares correspondientes al frame pasado sobre el frame actual. Para ello se utiliza la función epipolarLines.m, la cual recibe como entradas las matrices epipolares y los puntos en el frame pasado.

$$x\,F = l'$$

Al finalizar este paso se cuenta con diez grupos de líneas epipolares pertenecientes a todos los puntos.

La salida obtenida será la matriz L de 4 dimensiones

L: Pi x 3 x 10 x N Donde N: Nro. De frames, Pi: Puntos de interés
El primer elemento indica qué puntos pertenece la línea epipolar, el segundo los tres elementas de una línea epipolar A, B y C

$$Ax + By + C = 0$$

El cuarto indica a qué frame pasado pertenece y el último el frame actual.

3.3.4.4 *Filtrado de líneas epipolares*

Para encontrar las intersecciones no se utilizará todas las líneas halladas. Por lo tanto se realiza un filtrado de las líneas a partir del ángulo que estas forman entre sí y de su norma. Se establece un ángulo mínimo de 0.15 y una norma mínima de 0.01. La norma se obtiene a partir de la raíz cuadrada de la suma de los cuadrados de los tres elementos las líneas. El ángulo mínimo se determina hallando el ángulo que forma una línea con las 9 restantes.

3.3.4.5 Intersecciones

La tercera etapa consiste en encontrar las intersecciones que las líneas forman en la matriz actual como se muestra en la figura. Utilizando minimización por mínimos cuadrados (Least Squares) se encuentran las intersecciones entre las líneas epipolares.

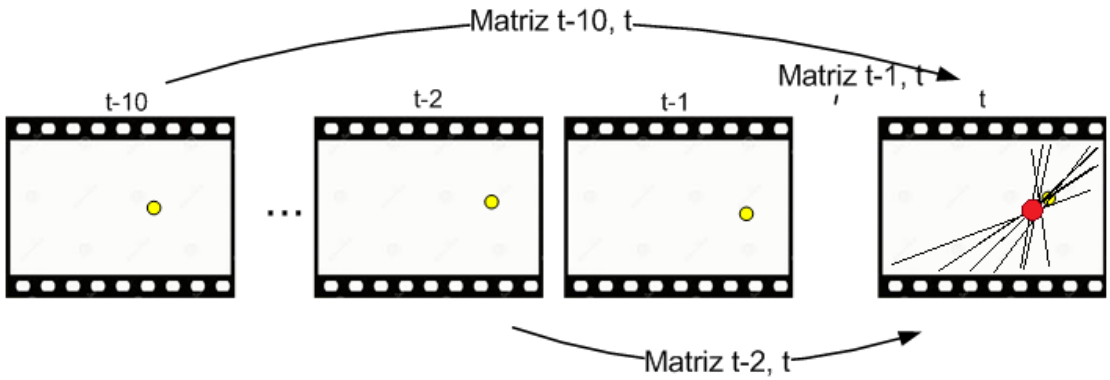


Figura 3.6. Intersecciones de Líneas Epipolares

A partir de:

$$Ax + By + C = 0$$

Se obtiene:

$$[x\ y] = ([A\ B]^T[A\ B])^{-1}[A\ B]^T(-C)$$

3.3.5 Filtrado

La etapa de filtrado de trayectoria consiste en aplica un filtro gaussiano que suavice las trayectorias. Esto se realiza mediante la convolución con un filtro Gaussiano de longitud 25 frames y varianza 5 Los parámetros del filtro se adecuan según la longitud de las trayectorias.

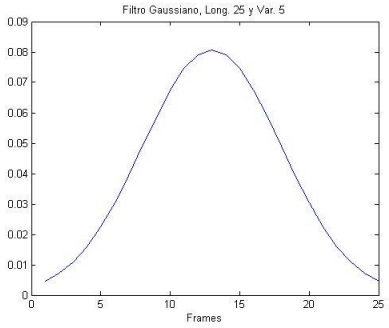


Figura 3.7. Filtro Gaussiano N 25 y V 5

Implementación en Matlab 2013

3.3.6 2da. Función de Transferencia Epipolar y filtrado

Para esta etapa se tienen dos diferencias. La primera es que los puntos a relacionar serán los puntos virtuales filtrados de la etapa anterior y los puntos originales del frame actual. La segunda diferencia está en la elección de los frames para encontrar matrices epipolares. En esta etapa se eligen los 5 frames anteriores y los 5 posteriores como se muestra en la figura. Los nuevos puntos virtuales son filtrados nuevamente con el algoritmo. La varianza para esta etapa es menor.

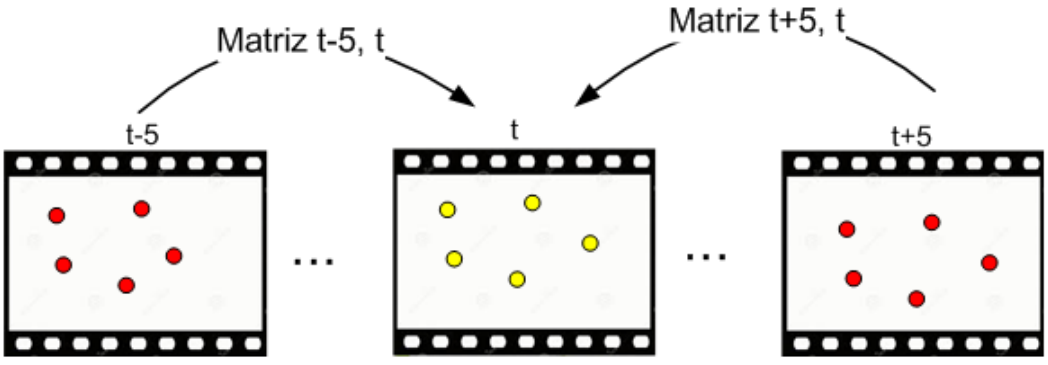


Figura 3.8. Matrices Fundamentales de frames pasados y futuros

3.3.7 Frame warping

La etapa de frame warping consiste en establecer una grilla que se modifique para llevar la información del frame actual al estabilizado Para ellos se utilizara los puntos originales y los puntos estabilizados de cada frame. Se elige esta metodología debido a que evita los costos computacionales de trabajar con data tridimensional. Para ello se define una trayectoria de cámara. Este aspecto del método hace que el resultado sea visualmente correcto, pero no en realidad no es coherente físicamente. Al utilizar geometría epipolar para encontrar la trayectoria de los nuevos puntos, sí se mantiene una coherencia física en las imágenes resultado.

El algoritmo buscará la posición óptima de los vértices en la imagen destino al minimizar las energías de data y similaridad. Para el algoritmo es necesario tener como entradas los Puntos originales y Puntos virtuales, la matriz de pesos, el número de frame, el tamaño máximo de fila y el tamaño de celda. Cabe recalcar que para simplificar los cálculos se recorta la imagen al tamaño R x R.

El primer paseo es filtrar los puntos que serán de utilidad, para ello se utiliza la matriz w. lo segundo es establecer una grilla a partir de la imagen original se establecer celdas tendrá la grilla. Para llevar la grilla original a la modificada se utilizaran los vértice V del frame actual en conjunto con los puntos originales Pk' y los puntos estabilizados Pk para obtener los vértices Vk en el frame estabilizado.

Se puede apreciar que para 16 celdas (Grilla de 4 x 4), se obtienen 25 vértices

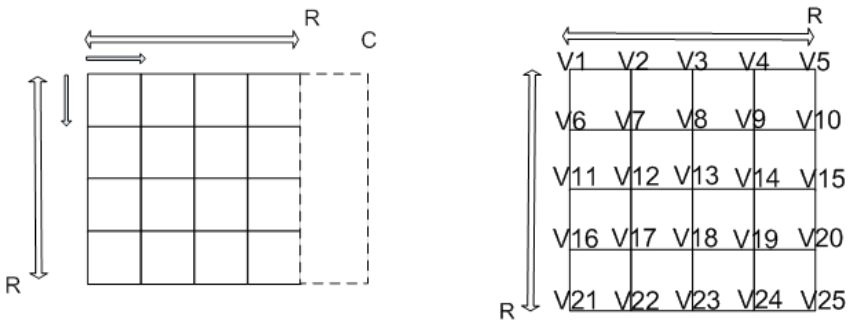
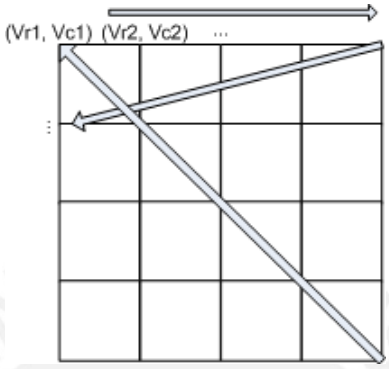


Figura 3.9. A. Grilla en imagen distorsionada. B. Vértices encontrados y numerados

A partir de los vértices hallados, se conforma un vector de vértices V

1. Primero se almacenan las filas (rows). Se empieza en el Vértice (Vr1, Vc1) y se avanza hacia la siguiente columna.



2. Cuando se llega al final de la fila, se avanza a la siguiente fila y se avanza por columnas nuevamente.

3. Cuando se llega al final, se regresa al inicio y ahora se almacenan las columnas.

$$V = \begin{bmatrix} Vr_1 \\ Vr_2 \\ . \\ . \\ . \\ Vr_{n+1} \\ Vc_1 \\ Vc_2 \\ . \\ . \\ . \\ Vc_{n+1} \end{bmatrix}$$

Figura 3.10. Formación del Vector de Vértices

Elaboración Propia

$$V: 2\ (n+1)^2\ x\ 1 \qquad donde \quad n^2 = número\ de\ celdas$$

Lo mismo ocurre con los puntos a utilizar:

$$P = \begin{bmatrix} Pr_1 \\ Pr_2 \\ . \\ . \\ . \\ Pr_P \\ Pc_1 \\ Pc_2 \\ . \\ . \\ . \\ Pc_P \end{bmatrix}$$

$P: 2\ Np\ x\ 1$ Np: Número de puntos útiles

Para el algoritmo, el vector de pesos w_k de valores $\alpha, \beta, \gamma, \delta$ será almacenado en una matriz dispersa M.

Matriz dispersa M

Dimensiones: (2 x Nro. De Puntos) x (2 x $(c+1)^2$)

$$M = \begin{bmatrix} \alpha 1 & \beta 1 & \gamma 1 & \delta 1 & \dots & 0 & \dots & \alpha 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \alpha 2 & \beta 2 & \dots & 0 & \dots & \alpha 2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \beta 3 & \gamma 3 & \dots & 0 \\ 0 & \gamma 4 & \beta 4 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \gamma n & \alpha n & 0 & \dots & 0 & \dots & 0 & \delta n & 0 & \dots & 0 \end{bmatrix}$$

La matriz M hallada será multiplicada por los pesos de la matriz de coherencia temporal hallada anteriormente.

En el término de similitud se tienen dos parámetros más: $\alpha \; y \; ws$
Para esta implementación serán tomados como valores constantes 20 y 1 respectivamente.

Se define la matriz dispersa N para almacenar los pesos respectivos de la ecuación.
Matriz dispersa N

Dimensiones: (16 x Nro. De Celdas) x (2x $(c+1)^2$)

$$N = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & -1 & \dots & 1 & \dots & -1 & 0 & 0 & \dots & 1 \\ 0 & -1 & 1 & 0 & \dots & 0 & \dots & 0 & -1 & 0 & \dots & -1 \\ -1 & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & -1 & 1 & \dots & 0 & \dots & 0 & -1 & 0 & \dots & 0 \end{bmatrix}$$

3.3.8 Remuestreo

La etapa de remuestreo consiste en llevar la información de una imagen de partida a una de llegada. Para ello se utiliza la interpolación bilineal. Dado que en la etapa anterior se obtuvieron los vértices de la grilla en el frame estabilizado, se encontrarán los puntos correspondientes a la imagen de partida en la imagen de llegada. Es importante conocer la correspondencia entre los puntos encontrados y sus grillas para que la interpolación pueda dar resultados coherentes.

3.4 Síntesis del capítulo 3

A lo largo del capítulo 3 se presentaron las etapas a seguir durante la implementación, así como los parámetros y componentes que se utilizan en cada una de ellas.

- La etapa de detección se realizará con Voodoo Tracker, con detector de Harris y KLT. Se utilizan trayectorias mayores o iguales a 20 frames, según lo establecido [19].
- Durante la etapa de Geometría Epipolar se calculan las matrices epipolares, líneas epipolares y puntos de intersección para obtener los nuevos puntos virtuales.
- La etapa de Frame Warping busca generar una grilla que se acomode a la nueva imagen estabilizada. Esto se consigue mediante la minimización de un vector de vértices.

Capítulo 4

Simulaciones y Análisis de Resultados

En este capítulo se mostrarán los resultados de la simulación y prueba del algoritmo, así como el análisis de los resultados obtenidos. Como se mencionó en el capítulo anterior, las pruebas se realizarán con los videos de la base de datos de [19].

4.1. Simulaciones

4.1.1 Detección y Tracking de Puntos

La detección y tracking de puntos se realiza con el programa Voodoo Tracker [26]
Detección: Detector de esquinas de Harris.
Parámetros: Nro. Máx. Puntos: 4000, Sigma Gaussiano: 0.7, Mín. Relativo: 1.0e-5, Escala: 0.4

Tracking: KLT
Parámetros: Nro. De Puntos Máx. 4000, Revisión de Constancia: Deshabilitado, se verifica si la transformación es de similaridad, afinidad, o traslación.

La figura 4.1 muestra la detección en el programa Voodoo Tracker.

Figura 4.1 Detección de puntos con Voodoo Tracker

Máscaras de trayectorias y Máscara de Coherencia Temporal:

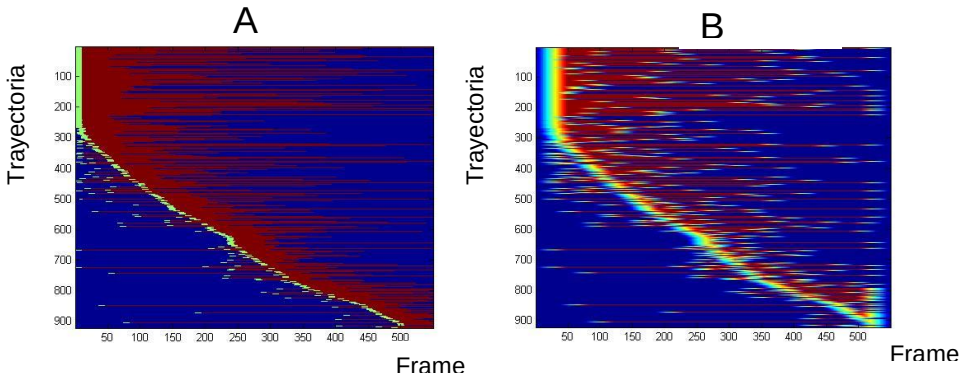


Figura 4.2 A. Máscara de trayectoria. B. Máscara de Coherencia Temporal

4.1.2 Función de Transferencia Epipolar

Se obtienen los puntos virtuales mediante las funciones:

- Algoritmo de 8 puntos
- RANSAC

Parámetros para RANSAC:
Nro. De Iteraciones 15e3
Distancia Euclidiana Mínima 0.001

La figura 4.3 muestra la intersección encontrada entre líneas epipolares. La figura 4.4 muestra el resultado de filtrar las líneas epipolares que no cumplen con las condiciones de ángulo y norma.

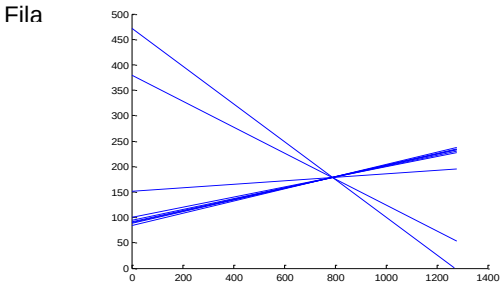
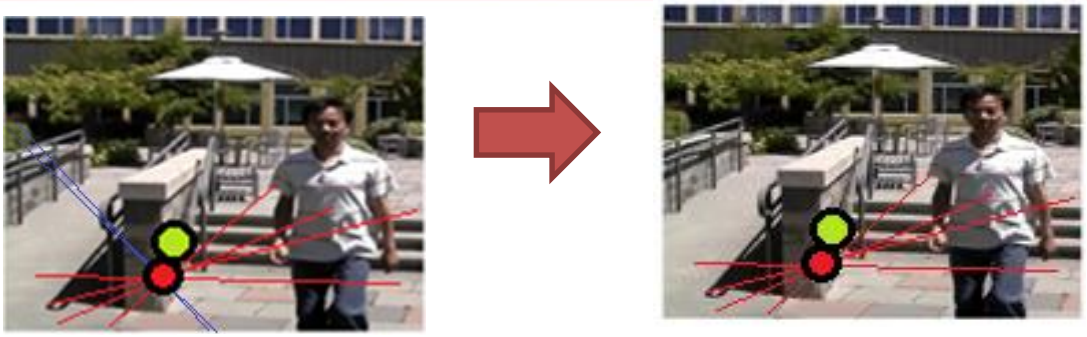


Figura 4.3 Intersección de Líneas Epipolares

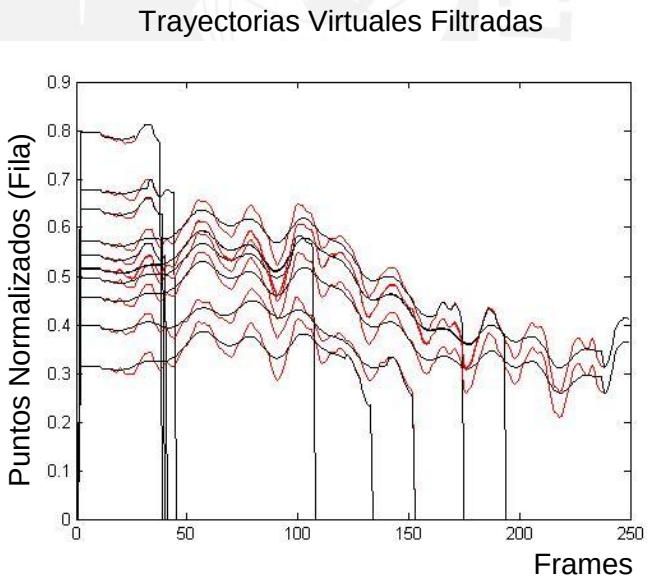


A B

Figura 4.4 Filtrado de líneas epipolares. A). Todas las líneas encontradas. En color azul líneas que no cumplen las condiciones. B). Líneas filtradas.

4.1.3 Filtrado

La etapa de filtrado consiste en la convolución de las trayectorias de los puntos tanto para filas como columnas con un filtro gaussiano, debido a que permite rechazar altas frecuencias y no genera artefactos.



Trayectoria (Fila) de 1 punto
ROJO: Trayectoria Original
NEGRO: Trayectoria Filtrada
Filtro Aplicado: Gaussiano, Longitud 25, Varianza 5

Figura 4.5 Filtrado de trayectorias

4.1.4 Frame Warping

La malla de inicio está conformada por los vértices de los cuadrantes. En la figura 4.4 se muestran en rojo. La malla de salida se muestra en azul.

Figura 4.6. Grillas en Frame Warping. Rojo: Original Azul: Estabilizada

4.2 Resultados

A continuación se presentan tablas informativas acerca de los videos fuente y resultado. Los videos descritos en la Tabla 4.1 se encuentran en el Anexo. La Tabla 4.2 muestra las características de los videos estabilizados. En la Tabla 4.3 se aprecian los tiempos obtenidos para cada etapa del algoritmo.

Tabla 4.1. Videos en Anexo.

	Descripción	Estado	Fuente
Video 1A	Persona caminando	Distorsionada	Banco de Videos [19]
Video 1B		Resultado	
Video 2A	Autopista, autos	Distorsionada	Banco de Videos [19]
Video 2B		Resultado	
Video 3A	Plaza, personas caminando	Distorsionada	Banco de Videos [19]
Video 3B		Resultado	
Video 4A	Sintético, cubos en movimiento	Original	Video creado
Video 4B		Distorsionada	
Video 4C		Estabilizada	
Video 4D		Comparación	
Video 5A	Iluminación, persona caminando	Distorsionada	Video grabado
Video 5B		Resultado	

Tabla 4.2. Descripción de Videos resultado

	Persona caminando	Sintético	Plaza (Cono)	Iluminación
Núm. de frames	250	115	450	500
Duración del video estabilizado (seg.)	9 s	6 s	19 s	25 s
Tamaño de Imagen (pixels)	1280x720	556x460	640x360	1920x1080 (Calidad HD)
Trayectorias Válidas	3462 (> 40 frames)	453 (>20 frames)	10685 (> 20 frames)	3206 (> 20 frames)
Puntos en Frame Warping (prom)	822	103.7	2736.1	252.2
Núm. de frames finales	188	96	400	490

Tabla 4.3. Resultado de tiempos de implementación

Tiempos	Persona caminando	Sintético	Plaza (Cono)	Iluminación
Detección de puntos y Tracking	8 m 17 s	1 m 29 s	22 m 29 s	41 m 39 s
Función de transferencia epipolar	2 m 23 s	23 s	10 m 54 s	6 m 34 s
Frame Warping	28 m 35 s	4 m 48 s	18 m 8 s	2 h 56 m 24 s
Construcción del video	6 s	4 s	25 s	36 s

4.3 Evaluación de resultados

Para realizar un análisis cuantitativo del algoritmo, se busca una métrica que compare los videos distorsionados y estabilizados con videos originales sin distorsión. Este análisis resulta complicado cuando se utilizan videos ya grabados como los de la base de datos. Como solución se propone utilizar videos sintéticos los cuales simulen el ruido adquirido durante la adquisición de los videos.

Para ello se crean videos sintéticos, con el programa *create_cubes.m*, utilizando rendering y opengl, al cual se le añade señales sinusoidales y ruido aditivo de distribución normal con varianza variable y media 0, para simular un video con jitter. A este nuevo video se le aplicará el algoritmo para obtener un video estabilizado el cual pueda ser comparado con el original. Los videos sintéticos nos permiten tener una versión sin jitter y una con jitter, para poder comparar el resultado con el video resultado.

4.3.1 Procedimiento para la evaluación

Paso 1. Creación de video original y distorsionado

Como se puede apreciar en el video sintético, los cubos laterales azules emulan el fondo de un video real mientras que el cubo rojo que se mueve distintamente emula a un objeto con movimiento independiente al de la cámara.

Paso 2. Obtener el video estabilizado con el algoritmo implementado.

Paso 3. Puntos Característicos

Para cada frame de los videos se utiliza el algoritmo de SIFT hallamos descriptores para los tres videos (original, distorsionado y estabilizado)

Paso 4. Correspondencias entre puntos

Para encontrar las correspondencias entre videos original-distorsionado y original-estabilizado, se hace uso de Nearest Neighbor Distance Ratio o NNDR, para encontrar la alternativa más adecuada dado un punto en el frame original. Se compara la relación entre la distancia del primer y segundo vecino más cercano. Este ratio debe ser menor a un umbral dado.

$$\frac{\|D_A - D_B\|}{\|D_A - D_C\|} < t$$

Donde

$D_A: Descriptor\ en\ evaluación$

$D_B: Descriptor\ Vecino\ más\ cercano$

$D_C: Segundo\ Vecino\ más\ cercano$

$t: Umbral$

Luego utilizando RANSAC se eliminan outliers. De esta manera encontramos las correspondencias. Para que la comparación se realiza con los mismos puntos, se realiza una búsqueda de los puntos originales detectados en los dos pares de videos. La figura muestra el resultado hasta este paso.

Figura 4.7 Matching de puntos original-distorsionado

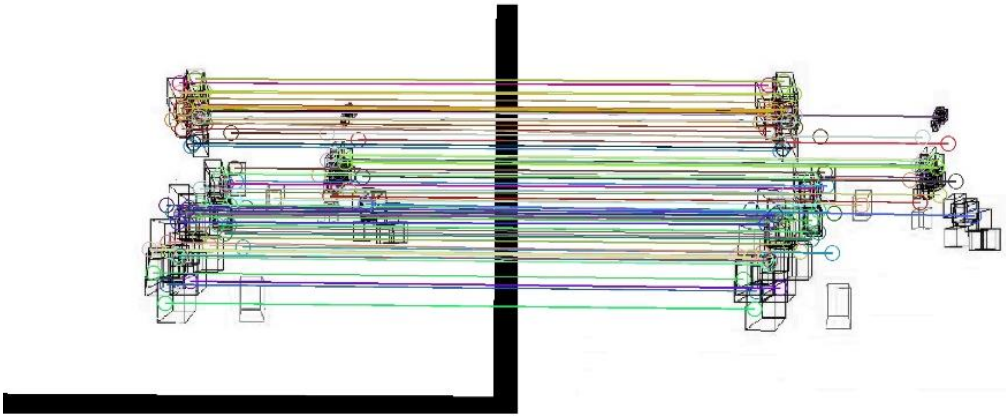


Figura 4.8. Matching de puntos original-estabilizado

Paso 5. Evaluación

Los puntos del video distorsionado y estabilizado que correspondan al mismo punto original, son utilizados para hallar una distancia de error. Para cada par de videos, obtenemos las distancias de error máxima y promedio de cada frame. Las gráficas de las figuras 4.9 y 4.10 muestran las distancias medidas.

$$d = \sqrt{(x_o - x)^2 + (y_o - y)^2}$$

$$Error\ x\ Frame = \frac{1}{N} * \sum_{p=1}^{N} d(p)$$

Las gráficas azules representan el error por frame en el video distorsionado y la roja el error por frame en el video estabilizado. Se evaluaron 90 frames del video sintético. Se trabajaron 76.3 puntos en promedio para los videos.

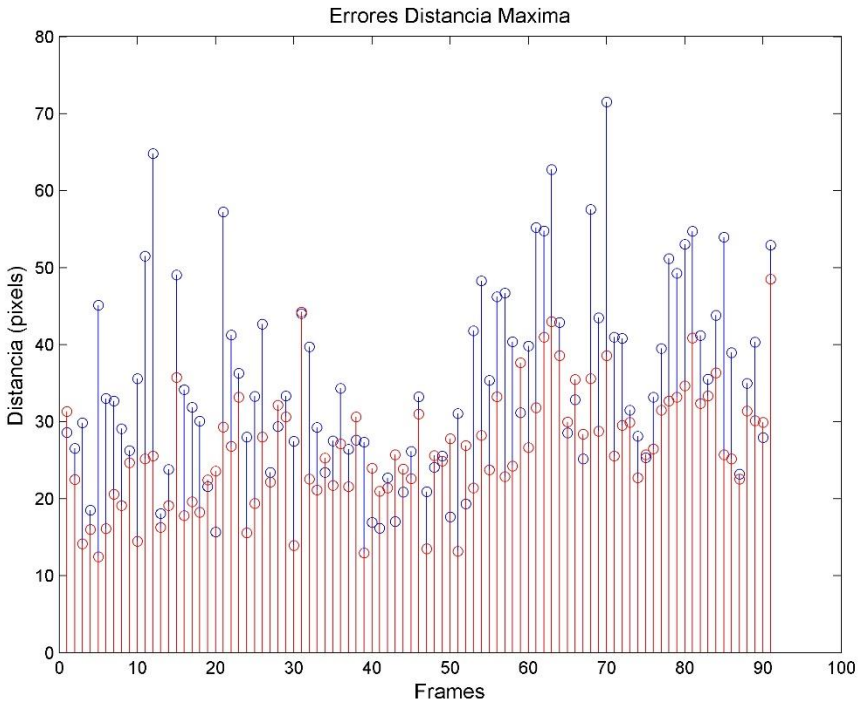


Figura 4.9. Error distancia máxima por frame. En azul: original-distorsionado en rojo: original-estabilizado

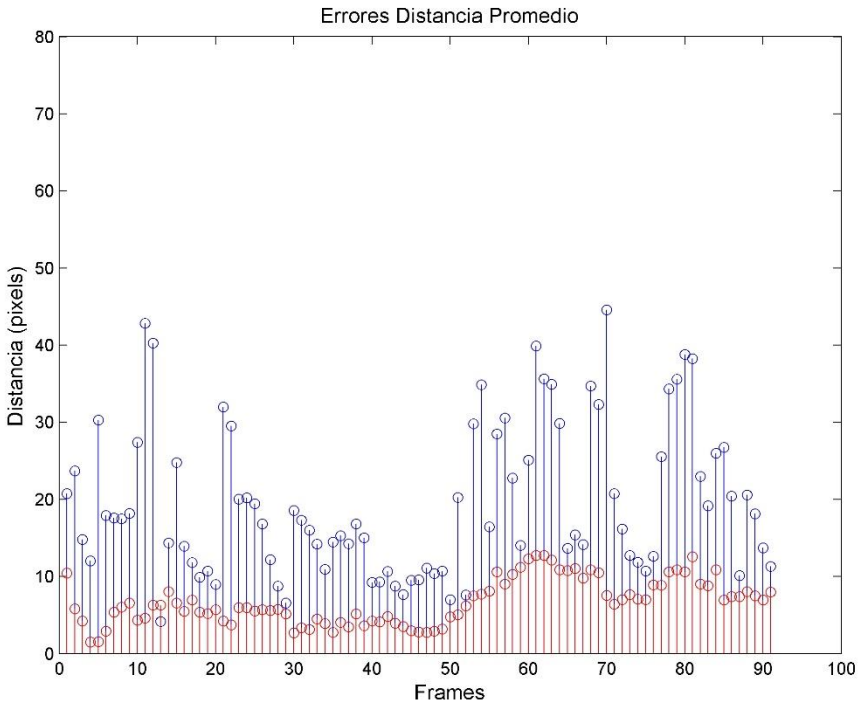


Figura 4.10. Error distancia promedio por frame. En azul: original-distorsionado en rojo: original-estabilizado

En las figuras 4.11 y 4.12, se muestra la variación de este error comparando las distancias error encontradas. Esto mediante la fórmula.

$$Diferencia\ Error\ en\ \% = 100\ x\ \frac{Dist.Distorsionado - Distancia\ Estabiliz.}{Dist.Distorsionado}$$

Finalmente para todo el video se obtuvo en promedio una diferencia máxima de **12.33%** y una diferencia promedio de **28.76%**.

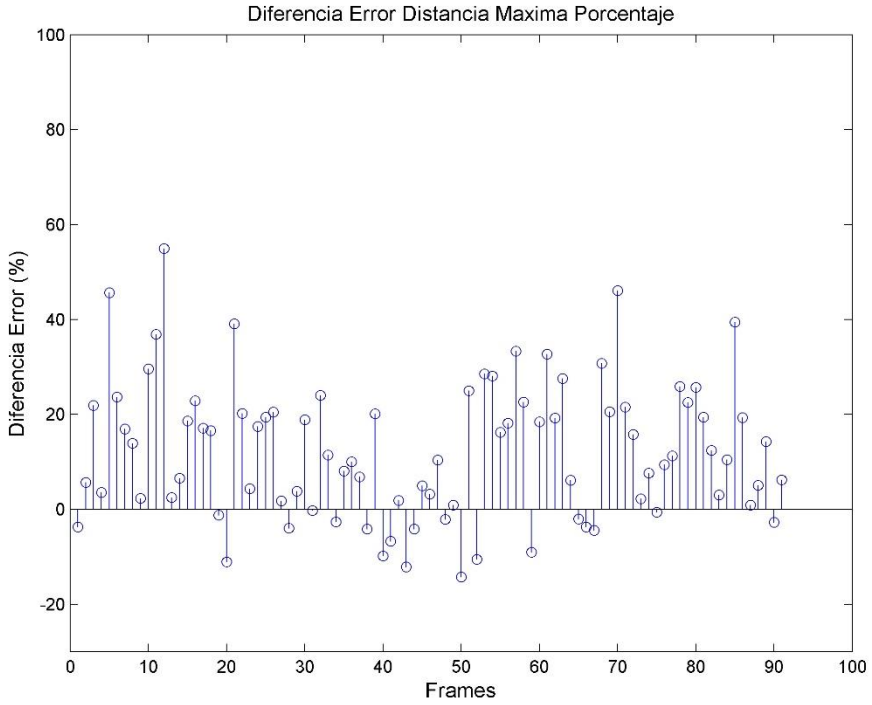


Figura 4.11. Diferencia distancia error máxima por frame (en porcentaje).

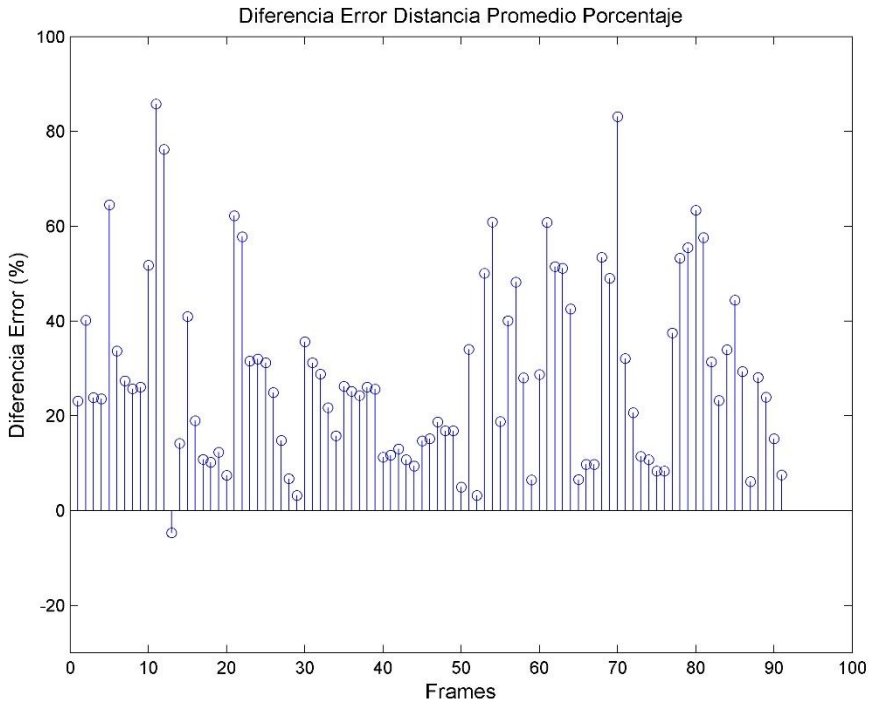


Figura 4.12. Diferencia Error Distancias. A). Distancias Máximas. B). Distancias Promedio

4.4 Análisis de resultados

- Con el video sintético creado se puede apreciar la magnitud de error existente entre los videos originales, los distorsionados y los estabilizados. Como se aprecia en las gráficas de las figuras 4.11 y 4.12 existen frames en donde la diferencia de error es negativa. Para el caso de las distancias máximas, esto se debe a que el video estabilizado presentó un punto en particular con mayor distorsión. Las diferencias negativas disminuyen para el caso del error promedio, debido a que se obtiene una muestra más representativa, la cual indica cómo se han corregido los puntos.

- Cuando el número de trayectorias encontradas no es suficiente o su duración es menor a 10 frames, los videos no pueden ser estabilizados por el algoritmo. Para el video que presenta cambios de iluminación, se tuvo que utilizar una cámara Full HD para obtener mayor detalle en los videos y así conseguir un mayor número de trayectorias.

- El nivel de ruido presente en el video sintético es determinante para el correcto funcionamiento del algoritmo. Para valores muy severos o cambios muy repentinos, el algoritmo pierde precisión. En los videos 1-3 el efecto jitter no es severo, por lo que resulta más sencillo obtener trayectorias virtuales coherentes.

- El detector utilizado no es robusto frente a cambios de iluminación, sin embargo tiene mayor repetitividad. Se tiene que considerar que en videos reales, los cambios bruscos de intensidad afectan la etapa de detección de puntos.

- Los videos de la base de datos de [19] son parte de los videos de prueba utilizados en la comunidad de image processing. Los videos sintéticos, en cambio, son hechos desde cero con la función *create_volume.*

- En las tablas 4.2 y 4.3 se puede observar una relación directa entre el tiempo de procesamiento y la calidad de los videos. El video grabado en calidad HD demoró 3 horas 45 minutos para su estabilización.

4.5 Síntesis del capítulo 4

- Se mostraron las simulaciones del algoritmo y se muestra el análisis de los videos resultados. Se utilizaron Harris y SIFT para la detección de puntos, obteniéndose un mayor número de puntos con la detección de Harris.

- Se propone utilizar videos sintéticos para corroborar que los resultados sean coherentes. Se utilizó distancia entre puntos como medida del error entre las imágenes originales y estabilizadas. El primer paso es encontrar puntos de control para las tres imágenes. Luego se encuentran los puntos correspondientes a las tres. Se encuentran las relaciones entre los videos: original-distorsionado y original-estabilizado. Finalmente se aplica distancia euclidiana para hallar una medida de error.

- Se obtienen resultados coherentes para todos los videos utilizados. Lo que demuestra que el algoritmo funciona.

Conclusiones

1. La Geometría Epipolar utiliza relaciones geométricas de proyección para encontrar un punto en el espacio visto desde dos puntos de vista. Dado que las relaciones son almacenadas en una matriz de 3x3 llamada matriz fundamental, no es necesario almacenar la información del punto tridimensional, por lo que se ahorra costo computacional. Además la geometría epipolar permite, a diferencia de otras metodologías de establización de video, representar de manera físicamente coherente la trayectoria seguida por la cámara y el movimiento independiente de los objetos.

2. Los valores de ruido severos, movimientos muy bruscos de cámara o cambios severos de iluminación. afectan el funcionamiento del algoritmo. Se pierde efectividad debido a que el seguimiento de puntos no logra ubicar un número significativo de puntos confiables (inliers) o las trayectorias son demasiado cortas para aplicar la transferencia de punto epipolar. El número de puntos detectados en la primera etapa condiciona el resto de la implementación.

3. Para la evaluación cuantitativa resultó necesaria la creación de un video sintético para evaluar el resultado estabilizado con respecto al video sin jitter. Como se presentó en la sección 4.3, los resultados mostraron que el algoritmo de estabilización reduce el error de distancia de las coordenadas de un mismo punto con respecto al video original en un 28.76% en promedio. Los parámetros de ruido, el número de cubos, el movimiento de cámara y la velocidad son factores que influyen de manera significativa en los resultados.

4. Mediante el uso de las tablas 2 y 3 se concluye que el tiempo de procesamiento del algoritmo depende de la calidad de los videos.

5. Se demuestra que sí es posible estabilizar un video dadas las transformaciones de rotación y escala, y transformaciones de intensidad.

Recomendaciones

1. Luego de la detección de puntos de control, se recomienda evaluar la longitud de trayectorias obtenidas, para verificar que estas tienen una longitud aceptable. En caso no encontrarse la cantidad suficiente de puntos, se recomienda probar con un detector más robusto o utilizar la función de transferencia epipolar para hallar nuevos puntos de interés como se menciona en [19]. Este cambio significaría un mayor costo computacional.

2. Para representar a los objetos en movimiento de manera más coherente, en [19] se propone utilizar Time-View Reprojection. Esta implementación evalúa la posición de los puntos de control en el tiempo y forma en base a ello se encuentra la posición en donde debería encontrarse el punto mediante una re proyección dinámica. Consiste en derivar las posiciones para hallar la velocidad y aceleración y en base a estos parámetros minimizar el error de proyección.

3. Para optimizar y mejorar la velocidad de procesamiento, se puede llevar el código a un sistema de cómputo heterogéneo.

4. En los videos 1-3 se aprecia durante algunos frames franjas negras a los extremos de imagen. Este detalle es controlado recortando la imagen de salida. La falta de textura en la etapa de Frame Warping y las franjas negras que aparecen en los videos estabilizados debido a las limitaciones del warping y remuestreo, pueden solucionarse mediante Motion Inpainting y Plane-Based Warps descritos en [24-25].

Bibliografía

[1] R. González y R. Woods
2007, “Digital Image Processing” 3era Edición. Prentice Hall.

[2] R. Szeliski
2010, “Computer vision: algorithms and applications” Springer.

[3] R. Szeliski
2006, “Image alignment and stitching: a tutorial”
Foundations and Trends in Computer Vision Vol. 2, No. 1.

[4] R. Hartley y A. Zisserman
2000, “Multiple view geometry in computer vision” Cambridge University Press.

[5] A. Goshtasby
2005, “2-D and 3-D image registration for medical, remote sensing, and industrial applications” Wiley & Sons.

[6] I. Zamek y S. Zamek
2005, “Definitions of jitter measurement terms and relationships”. Test Conference, 2005. Proceedings. ITC 2005. IEEE International.

[7] G. Puglisi
2011, “A robust video alignment for video stabilization purposes” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 21, No. 10, October 2011.

[8] W. Zhu
2011, “A real-time scheme of video stabilization for mine tunnel inspectional robot” IEEE Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics December 15 -18, 2007, Sanya, China.

[9] O. Oreifej
2013, “Simultaneous video stabilization and moving object detection in turbulence” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 2, February 2013.

[10] Y. Liang
2004, “Video stabilization for a camcorder mounted on a moving vehicle,” IEEE Vehicular Technology, IEEE Transactions. Vol. 53.

[11] R. Cucchiara
2004, “Using computer vision techniques for dangerous situation detection in domotic applications”.
Intelligent Distributed Surveilliance Systems, IEEE Transactions, 23 Feb. 2004.

[12] B. Rogers y M. Graham
1979, "Motion parallax as an independent cue for depth perception “ Perception 8 (2) 125 -134.

[13] B. Zitová
2003, “Image registration methods: a survey” Image and Vision Computing.

[14] C. Harris y M. Stephen
1988, “A Combined Corner and Edge Detector”, Proceedings on 4th Alvey Vision Conference, Manchester.

[15] W. Burger
2007, “Digital image processing an algorithm introduction using java” Springer.

[16] K. Mikolajczyk y C. Schmid
2005, “A Performance Evaluation of Local Descriptors”, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[17] D. Lowe
2004, “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision.

[18] J. Shi y C. Tomasi
1994, “Good features to track”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94). 593– 600.

[19] A. Goldstein y R. Fattal.
2012, “Video stabilization using epipolar geometry”. ACM Trans. Graph., 32(5).

[20] R. Hartley
1997, “In defense of 8 point algorithm” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 6, June 1997.

[21] M. Fisher y R. Bolles,
1981, Random sample consensus: A paradigm for modeling fitting with applications to image analysis and automated cryptography, Comm. ACM 24 (6) (1981) 381–395.

[22] F. Liu
2009, “Content preserving warps for 3D video stabilization” ACM SIGGRAPH 2009.

[23] P. Heckbert
1989, “Fundamentals of texture mapping and image warping” Tech. Rep. UCB/CSD-89-516, EECS Department, University of California, Berkeley, Jun.

[24] Z. Zhou
2013, “Plane based content preserving warps” IEEE Computer Vision and Pattern Recognition (CVPR) 2013.

[25] Y. Matsushita
2006, “Full-frame video stabilization with motion inpainting” Pattern Analysis and Machine Intelligence IEEE Transactions.

[26] Voodoo Tracker
Programa especializado en detección y seguimiento de puntos. http://www.viscoda.com/index.php/en/products/non-commercial/voodoo-camera-tracker. Consultado el: 5 de Septiembre del 2014.

[27] Mathworks
http://www.mathworks.com/

[28] GoPro
GoPro es una marca de cámaras especializadas para grabar sin utilizar las manos. http://es.gopro.com/

[29] El Comercio
Noticia relacionada al aumento de smartphones.
http://elcomercio.pe/tecnologia/actualidad/78-jovenes-latinoamericanos-tiene-smartphone-noticia-1768629. Consultado el: 10 de Octubre del 2014.

[30] El Comercio
Noticia relacionada al aumento de la importación de smartphnes en el país.
http://elcomercio.pe/economia/peru/importacion-smartphones-peru-crecio-15-ano-noticia-1714442. Consultado el: 10 de Octubre del 2014.