

ALGORITMO PARA LA GENERACIÓN DE IMÁGENES EN 360 GRADOS MEDIANTE  
LA TÉCNICA DE *IMAGE STITCHING* EN CÁMARAS ROTATORIAS.

FELIPE ANDRÉS VILLARREAL PEÑARANDA

GISELA ANDREA GÓMEZ URREA

Trabajo de Grado Como Requisito para Obtener el Título de:

Ingeniero Electrónico

Director

JUAN DAVID BASTIDAS RODRÍGUEZ

Ingeniero Electrónico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS

ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE

TELECOMUNICACIONES

INGENIERÍA ELECTRÓNICA

BUCARAMANGA

2017

## **Agradecimientos**

### **Felipe Andrés Villarreal Peñaranda**

En este trabajo no solamente formamos parte los autores sino también todas las personas que nos brindaron su apoyo a lo largo de este proceso y quiero expresar toda mi gratitud a cada uno de ellos, ya que este es el fruto de un esfuerzo conjunto y cada uno apporto su granito de arena para llegar hasta este punto.

Inicialmente le agradezco a nuestra institución la universidad Industrial de Santander por permitirnos tener una formación profesional en una de las universidades más prestigiosas del país.

A los docentes que tuve la fortuna de conocer y de los cuales pude aprender, porque gracias a su sabiduría pude avanzar en mis metas de llegar a ser un profesional.

A Said David Pertuz Arroyo por su guía permanente, su apoyo, por sus enseñanzas a lo largo del desarrollo de este proyecto, las cuales nos ayudaron a crecer como estudiantes y como personas, y sobre todo por su gran compromiso y sentido del deber, que son dignos de admiración.

A Juan David Bastidas por su buena disposición y su colaboración a lo largo de todo este proceso.

A mi compañera en esta etapa final de mis estudios Gisela Andrea Gómez Urrea, por su empeño, dedicación y buena energía, sin importar lo difícil que fueran las pruebas a superar.

A mis padres, mis tías y familiares, por ser una voz de aliento en mi vida, por contagiarme su alegría y sus buenas energías a pesar de la distancia, su fe incondicional, sus buenos deseos, por estar presentes en todo momento y por enseñarme que si me lo propongo todo en esta vida es posible.

**Gisela Andrea Gómez Urrea**

Quiero agradecer a Dios y a todas aquellas personas que con su apoyo y aliento han contribuido para alcanzar esta meta.

A mis padres por amarme y educarme como una persona solidaria y útil para la sociedad. A mi padre, que aunque la vida no le permitió acompañarme hasta este punto, sé que es mi ángel guardián, quien con su corazón amoroso y generoso cuidado de mí, creyó en mis capacidades desde el principio y me brindó la oportunidad de iniciar este proceso. A mi madre, motor de mi vida, por su incesante amor y paciencia, quien me ha enseñado a creer en mis capacidades y que con esfuerzo y dedicación todo es posible.

A mis hermanos, cuñados y sobrinos, quienes han sido la luz en este camino, gracias por su cariño y compañía.

A la familia Rangel Urrea, quienes son un gran ejemplo y apoyo invaluable.

A mi compañero de tesis Felipe Andrés Villarreal Peñaranda, por su esfuerzo, entrega y dedicación para llevar a cabo esta gran tarea.

Por último, quiero expresar mi gratitud a los profesores Said Pertuz Arroyo y Juan David Bastidas, por su guía y dedicación en el desarrollo de este proyecto.

Y a todos aquellos que de alguna manera contribuyeron en la consecución de este logro,  
gracias.

## Resumen

Las imágenes panorámicas tienen diversas aplicaciones en la actualidad como lo son el desarrollo de entornos virtuales para videojuegos, la creación de mapas satelitales y aplicaciones biomédicas, entre otras.

En el presente proyecto se desarrolla un sistema de composición de imágenes tomadas por una cámara rotatoria. La entrada de dicho sistema es un conjunto de imágenes tomadas a una misma escena, desde diferentes perspectivas. Posteriormente se debe extraer información de cada una de estas imágenes. Cada imagen capturada tiene su propio sistema coordenado, pero solo se pueden unir cuando dichos sistemas son iguales. Para llevar una imagen de su propio sistema coordenado al de otra imagen, es necesario aplicar una transformación proyectiva. Finalmente se unen las imágenes y después se hace necesario suavizar la transición de una imagen a otra. La salida es una imagen panorámica de 360 grados formada por mosaicos resultantes de la composición de pares de imágenes.

En este trabajo se realiza un estudio, selección y evaluación de los métodos que realicen cada una de las funciones anteriormente nombradas. Se presenta un diagrama funcional de los métodos elegido, y se implementan algoritmos que realicen dichas tareas de forma automática. Todo esto sobre el entorno de programación MATLAB.

## Tabla de Contenido

Introducción .....	17
1. Descripción del Proyecto .....	19
1.1. Identificación del Problema .....	19
1.2. Justificación de la Solución .....	19
1.3. Objetivos .....	21
2. Fundamentos y Antecedentes.....	22
2.1. Geometría de la formación de imágenes.....	22
2.2. Calibración de Cámaras .....	27
2.3. Generación de imágenes Panorámicas.....	31
3. Metodología .....	33
3.1. Adquisición de Imágenes.....	33
3.2. Calibración de la cámara.....	36
3.3. Detección de puntos de interés .....	37
3.3. Estimación de homografías.....	47
3.4. Generación de imágenes panorámicas .....	58
4. Resultados .....	62
5.1. Comparación de algoritmos de detección de puntos de interés .....	62
5.2. Desempeño en la estimación de homografías .....	67
5.3. Panorámicas de 2 imágenes .....	71
5.4. Panorámicas con N imágenes .....	75

	10
6. Conclusiones .....	79
7. Recomendaciones.....	81
Referencias bibliográficas.....	82
Apéndices.....	88

## Lista de Tablas

Tabla 1. Número promedio de puntos principales encontrados para cada una de las imágenes analizadas con los métodos a comparar. ....	64
Tabla 2. Número promedio para cada uno de los métodos analizados .....	65

## Lista de Figuras

Figura 1. Modelo Pinhole de la proyección perspectiva.....	22
Figura 2. Modelado de la cámara a partir del método Pinhol .....	24
Figura 3. Transformación entre el sistema de referencia en el espacio y el de la cámara.....	26
Figura 4. Cámara PTZ modelo SNC-EP520.....	34
Figura 5. La forma en que se conecta la cámara IP al computador, mediante un puerto de red. ....	36
Figura 6. Ilustración del procedimiento para encontrar la diferencia gaussiana a diferentes escalas .....	38
Figura 7. Descripción gráfica de pasos necesarios para encontrar la geometría epipolar .....	44
Figura 8. Diagrama de flujo del método Match.....	46
Figura 9. Diagrama de flujo del método DLT normalizado .....	52
Figura 10. Diagrama de bloques de método RANSAC .....	57
Figura 11. Feathering or center weighting image <i>Blending</i> .....	59
Figura 12. <i>Pyramid Blending</i> .....	59
Figura 13. Diagrama de flujo del método <i>Blending piramidal</i> .....	61
Figura 14. Diagrama de barras que contiene el desempeño para el cálculo de la matriz de homografía .....	68
Figura 15. (a) Imagen de entrada en coordenadas cilíndricas. (b) Imagen de muestra transformada con la homografía encontrada con el método RANSAC. (c) Imagen de entrada transformada con la matriz de homografía encontrada con el método DLT.....	69
Figura 16. Puntos coincidentes entre un par de imágenes pertenecientes a la misma escena. ....	70
Figura 17. Par de imágenes de la Figura 16, transformadas con la homografía calculada con el método RANSAC ..	70
Figura 18. Extracción de características de un par de imágenes tomadas por un teléfono celular.....	72
Figura 19. Correspondencias entre las imágenes de la Figura 18.....	72
Figura 20. Imagen panorámica obtenida con dos imágenes tomadas a una misma escena, tomadas por otro tipo de cámara .....	73
Figura 21. Imagen panorámica obtenida con imágenes tomadas a una misma escena con la cámara PTZ.....	74
Figura 22. (a) Primera imagen de entrada. (b) Segunda imagen de entrada. (c) Tercera imagen de entrada.....	75

Figura 23. Mosaico obtenido con la primera y segunda imagen.....	76
Figura 24. Mosaico obtenido a partir de la segunda y tercera imagen.....	76
Figura 25. Imagen final a partir de 3 imágenes de entrada. ....	77
Figura 26. Imagen panorámica de 360 grados a partir de 18 imágenes de entrada.....	78
Figura 27. Cilindro de radio unitario .....	90
Figura 28. Diagrama de flujo del algoritmo de deformación de imágenes en coordenadas cilíndricas .....	92

## Lista de Apéndices

Apéndice A. Diagrama del proceso que genera de imágenes panorámicas .....	88
Apéndice B. Parámetros característicos de la cámara PTZ SNC-EP520 .....	89
Apéndice C. Deformación de imágenes a coordenadas cilíndricas .....	90

## RESUMEN

**TITULO:** ALGORITMO PARA LA GENERACIÓN DE IMÁGENES EN 360 GRADOS MEDIANTE LA TÉCNICA DE IMAGE STITCHING EN CÁMARAS ROTATORIAS \*

**AUTORES:** Gisela Andrea Gómez Urrea & Felipe Andrés Villarreal Peñaranda \*\*

**PALABRAS CLAVES:** Reconstrucción, proyección, panorámicas, Stitching.

**DESCRIPCION:** Es posible adquirir imágenes a distancias más próximas a los objetos de interés, que tendría más detalle, pero se perder la posibilidad de capturar la información deseada desde un punto de vista más amplio en una sola imagen. Aquí es donde surge la idea de combinar varias imágenes con diferentes perspectivas obtenidas con la misma cámara. Un sistema automático de reconstrucción de imágenes digitales permitiría utilizar una cámara giratoria para obtener imágenes de alta resolución y una amplia gama de visión. Aprovechando la cámara giratoria de configuración remota, que permite controlar parámetros como rotación, inclinación y enfoque entre otros; Tomar diferentes planos de una escena a lo largo de su camino y unirlos para formar una imagen única.

Uno de los temas de estudio en el campo del procesamiento de imágenes es la extracción de información de una escena a partir de una imagen capturada. Esta operación puede ser tediosa ya que una sola imagen no contiene toda la escena que se pretende analizar y es necesario analizar varias capturas por separado, teniendo en cuenta que cada captura corresponde a un sistema de coordenadas diferente. En este proyecto se implementa un sistema que, a partir de imágenes tomadas desde diferentes ángulos, con variaciones en la orientación de la misma escena, permite crear una sola imagen (360 ° panorámica) resultado de la combinación de N imágenes de entrada. Facilitar la extracción de información de una escena, gracias al hecho de que solo se analiza una sola imagen, en lugar de n capturas de la misma escena.

Al final de este proceso se obtiene la imagen en 360 grados de alta resolución. Esta nueva imagen puede utilizarse para estudios posteriores, basados en la información que se puede extraer de la escena capturada, minimizando el procedimiento de extracción de información de la misma.

\* Tesis de Grado

\*\*Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería eléctrica electrónica y de telecomunicaciones.  
Director: Ana Beatriz Ramírez, Ingeniero Electrónico.

## ABSTRACT

**TITLE:** ALGORITHM FOR THE GENERATION OF IMAGES IN 360 DEGREES THROUGH THE IMAGE STITCHING TECHNIQUE ON ROTARY CAMERAS\*

**AUTHORS:** Gisela Andrea Gómez Urrea & Felipe Andrés Villarreal Peñaranda\*\*

**KEYWORDS:** Reconstruction, projection, panoramic, Stitching.

**DESCRIPCION:** It is possible to acquire images at distances closer to the objects of interest, which would have greater detail, at the cost of losing the possibility of capturing the desired information from a wider point of view in a single image. This is where the idea of combining several images with different perspective obtained with the same camera arises. An automatic digital image reconstruction system would allow a rotating camera to be used to obtain high resolution images and a wide range of vision. Taking advantage of a rotating camera of remote configuration, which allows controlling parameters such as rotation, tilt, and focus among others; To take different shots of a scene along its path and join them to form a unique image.

One of the topics of study within the field of image processing is the extraction of information from a scene from a captured image. This operation can be tedious since a single image does not contain the whole scene that is intended to be analyzed and it is necessary to analyze several captures separately, taking into account that each capture corresponds to a different coordinate system. In this project is implemented a system that, from images taken from different angles, with variations in the orientation of the same scene, allows to create a single image (360 ° panoramic) result of the combination of N input images. Facilitating the extraction of information from a scene, thanks to the fact that only a single image is analyzed, instead of n captures of the same scene.

At the end of this process you get the image in 360 degrees high resolution. This new image can be used for later studies, based on the information that can be extracted from the scene captured, minimizing the procedure of extracting information from it.

\*Bachelor thesis

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería eléctrica electrónica y de telecomunicaciones.  
Director: Ana Beatriz Ramírez, Ingeniero electrónico.

## Introducción

El proceso de formación de una imagen se puede modelar en 3 etapas: capturar información de una escena 3D, transformar la escena capturada al sistema 2D de la cámara (Szeliski R. , 2010), y digitalizar la captura para producir una imagen digital en 2D. A partir de estas imágenes se pueden extraer características de la escena, tales como las dimensiones de los objetos o la intensidad de la luz. En una imagen se pueden apreciar todo tipo de formas, las cuales, son el resultado de aplicar una transformación 3D para llegar a un sistema de coordenadas 2D (Szeliski R. , 2004). Si se tienen dos imágenes de una misma escena, se puede aplicar una transformación geométrica para llevarlas al mismo sistema de coordenadas 3D (misma perspectiva), concatenarlas, y posteriormente generar una imagen panorámica o mosaico de la escena. Para este propósito, la técnica de *Stitching* es una de las más utilizadas en el campo de visión por computador, para el desarrollo de entornos virtuales, el procesamiento de imágenes de tipo satelital, la generación de mapas digitales, mosaicos de alta resolución, y el análisis de estructuras en 3D [ (Huang , Lin, & Chen, 2015), (Szeliski R. , 2004), (Tang & Jia, 2008), (Hua, Li, & Li, 2010)]. También es utilizada en otros campos como la medicina, para generar imágenes biomédicas de con una vista más amplia (Wang & Wang, 2010).

Con este proyecto, se pretende indagar en el campo de la fotogrametría, estudiar los conceptos básicos utilizados en el procedimiento de creación de una imagen panorámica, y aplicarlos para brindar una base sólida a futuros trabajos de investigación en dicho campo. El objetivo de este

proyecto es implementar y evaluar el desempeño de un algoritmo para la generación de imágenes panorámicas. Un ejemplo directo es la reconstrucción 3D, en donde se desea extraer las características geométricas de la imagen panorámica, tales como direcciones de líneas, posiciones de puntos y vértices, y distancia entre planos; y reproducirlas en nuestro propio sistema de coordenadas en 3D con el fin de reconstruir la escena capturada (H.Shum, 1998).

## **1. Descripción del Proyecto**

### **1.1. Identificación del problema**

Es posible adquirir imágenes a distancias más próximas a los objetos de interés, con lo que se tendría mayor detalle, a costa de perder la posibilidad de capturar la información deseada desde un punto de vista más amplio en una única imagen. Este problema ha sido abordado, al menos de forma parcial, en distintos escenarios [ (Hua, Li, & Li, 2010), (Wang & Wang, 2010), (Tang & Jia, 2008), (Szeliski R. , 2004)]. Sin embargo, las técnicas que obtienen resultados de mayor calidad suelen requerir información adicional, o requieren de fotografías tomadas de una forma determinada.

Aquí es donde surge la idea de combinar varias imágenes con diferente perspectiva obtenidas con la misma cámara. Un sistema automático de reconstrucción de imágenes digitales permitiría utilizar una cámara rotatoria para obtener imágenes de alta resolución y amplio rango de visión. Aprovechando las ventajas de una cámara giratoria de configuración remota, que permite controlar parámetros como rotación, inclinación, y enfoque de la cámara entre otros; para obtener distintas tomas de una escena a lo largo de su trayectoria y unir las para formar una imagen única.

### **1.2. Justificación de la solución**

Uno de los temas de estudio dentro del campo del procesamiento de imágenes es la extracción de información de una escena a partir de una imagen capturada. Esta operación puede ser algo

tediosa ya que una sola imagen no contiene toda la escena que se pretende analizar y es necesario analizar varias capturas por separado, teniendo en cuenta que cada captura corresponde a un sistema coordinado diferente. En este proyecto se implementa un sistema que, a partir de imágenes tomadas desde distintos ángulos, con variaciones en la orientación de una misma escena, permita crear una única imagen (panorámica de 360°) resultado de la combinación de N imágenes de entrada. Facilitando la extracción de información de una escena, gracias a que solo se analiza una sola imagen, en lugar de N capturas de la misma escena.

Específicamente, la tarea de unir las imágenes se realiza mediante la técnica de *Stitching*, que es un proceso con el cual se combinan dichas imágenes fotográficas de una misma escena para crear una imagen panorámica. Este problema se puede solucionar en cuatro etapas que son la calibración y captura de imágenes, la identificación de los puntos principales o coincidentes, la combinación de las imágenes y, por último, un suavizado de la imagen final. En el primer paso, la calibración, es necesaria para realizar correcciones de perspectiva y aberración en las imágenes. En segunda instancia, es necesario realizar el proceso de detección de coincidencias (o correspondencia) entre imágenes. Para encontrar una transformación adecuada que lleve las imágenes a un mismo sistema coordinado se implementa el método RANSAC (Fischler & Bolles, 1981), el cual es útil para encontrar modelos matemáticos dentro de un grupo de datos en presencia de *outlayers* (A.Zisserman, 2004). Finalmente se aplica un suavizado mediante la técnica del *Blending* (Levin, 2003) a la imagen panorámica, para ajustar su tonalidad.

Al finalizar este proceso se obtiene la imagen en 360 grados de alta resolución. Esta nueva imagen puede ser utilizada para estudios posteriores, basados en la información que se puede extraer de la escena capturada, minimizando el procedimiento de extracción de información de la misma.

### 1.3. Objetivos

#### 1.3.1. Objetivo general

Implementar un algoritmo para generar imágenes panorámicas de 360° de una escena usando cámaras rotatorias mediante el software MATLAB.

#### 1.3.2. Objetivos específicos

- Desarrollar una herramienta en MATLAB que permita el control de una cámara rotatoria
- Seleccionar un algoritmo de registro de imágenes para encontrar los puntos coincidentes entre las imágenes capturadas con una cámara rotatoria.
- Implementar un algoritmo que permita encontrar la matriz de transformación entre dos imágenes para llevar una imagen desde su propio sistema de coordenadas al sistema de coordenadas de la imagen con la que se está comparando.
- Implementar un algoritmo que permita unir las imágenes capturadas en un mismo sistema de coordenadas mediante las técnicas del *Stitching* y *Blending*.

## 2. Fundamentos y Antecedentes

### 2.1. Geometría de la formación de imágenes

#### 2.1.1. Modelo Pin-hole

El modelo básico del cámara también llamado, “Pinhole” (A.Zisserman, 2004) se basa en la forma en que se generan las imágenes en una cámara y representa la transformación de las coordenadas de los puntos de la escena en las coordenadas de la imagen. Todos los rayos de luz se proyectan sobre un plano imagen, tras atravesar un único punto en el espacio, este punto es llamado punto focal de la cámara.

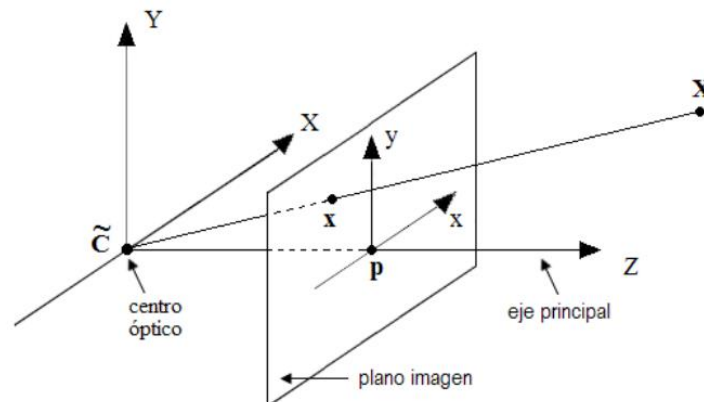


Figura 1. Modelo Pinhole de la proyección perspectiva. El punto  $C$  es el centro óptico y  $P$  es el punto principal. El centro óptico es el centro de la cámara y es donde se fija el origen de coordenadas de la cámara (A.Zisserman, 2004)

De acuerdo a la Figura 1, el modelo Pinhole define los siguientes parámetros:

*Punto principal:* El punto intersección (punto  $P$ ) entre el plano de la imagen y el eje óptico (recta perpendicular al plano de la imagen que pasa por el centro de cámara  $\tilde{C}$ ).

*Distancia focal:* Es la distancia existente entre el punto  $\tilde{C}$  y el punto principal.

El modelo Pinhole aplica una matriz de proyección para transformar las coordenadas de los puntos del objeto en coordenadas de la imagen, de la siguiente manera:

$$\mathbf{m} = \mathbf{P} \mathbf{M} \quad (1)$$

donde

$$\mathbf{M} = [x_w, y_w, z_w, 1]^t \quad (2)$$

es el vector que contiene las coordenadas del punto en el sistema de referencia exterior a la cámara,  $\mathbf{P}$  es una matriz de 3x4 denominada matriz de proyección, y  $\mathbf{m}$  es el vector de las coordenadas del punto proyectado en la imagen,

$$\mathbf{m} = [u, v, 1]^t \quad (3)$$

### 2.1.2. Parámetros intrínsecos y extrínsecos

El proceso de proyección de una imagen mediante el modelo Pinhole presentado en la sección anterior se puede descomponer algebraicamente mediante la concatenación de dos operaciones matriciales, haciendo  $\mathbf{P} = \mathbf{K} \mathbf{D}$ :

$$\mathbf{m} = \mathbf{K} \mathbf{D} \mathbf{M}$$

Específicamente, la proyección perspectiva se puede modelar mediante una matriz de parámetros intrínsecos  $\mathbf{K}$  y una matriz de parámetros extrínsecos  $\mathbf{D}$ , tal como se describe a continuación.

### *Parámetros intrínsecos*

Los parámetros intrínsecos son aquellos que definen la geometría interna y la óptica de la cámara. Éstos determinan cómo la cámara proyecta los puntos del mundo al plano de la imagen en 2D, siendo constantes en tanto no varíen las características y posiciones relativas entre la óptica y el sensor imagen. Tales parámetros son el punto principal y distancia focal (ver Figura 1).

El modelo Pinhole descrito anteriormente (Figura 1), fija un sistema de coordenadas proyectivas en el centro óptico, en donde el eje Z del sistema coincide con el eje principal de la cámara, es decir que se convierte en el sistema de referencia de la cámara. Además, el plano imagen se fija en el plano  $Z = f$ . (Pé, 2007). Mediante relación de triángulos, como se observa en la Figura 2, podemos determinar la correspondencia entre un punto cualquiera del espacio y su proyección en el plano imagen:

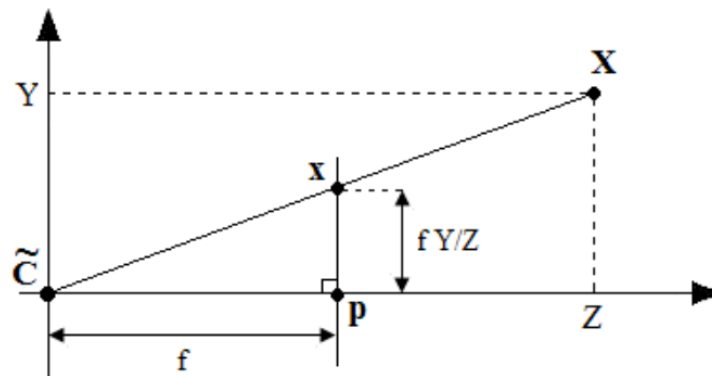


Figura 2. Modelado de la cámara a partir del método Pinhole (Pé, 2007)

Tomando un punto del espacio respecto al sistema de referencia de la cámara

$$X = (X_c, Y_c, Z_c)^T \rightarrow (x, y)^T = \left( \frac{fX_c}{Z}, \frac{fY_c}{Z} \right)^T \quad (4)$$

Usando coordenadas homogéneas, se puede expresar la ecuación (4) en forma matricial

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX_c \\ fY_c \\ Z_c \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \Rightarrow \mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

La ecuación (5) abreviada, es equivalente a:

$$\mathbf{P} = \text{diag}(f, f, 1)[I|0] = \mathbf{x} = \mathbf{P}\mathbf{X}_c \quad (6)$$

donde  $\text{diag}(f, f, 1)$  es una matriz diagonal, y sus parámetros corresponden a la distancia focal de la cámara y  $[I|0]$  equivale a una matriz identidad 3x3 concatenado con un vector columna nulo 3x1.  $\mathbf{X}_c$ , es un vector columna que representa las coordenadas homogéneas de un punto en el espacio,  $\mathbf{x}$  es el vector columna, representa las coordenadas homogéneas de un punto en la imagen, finalmente  $\mathbf{P}$  es la matriz homogénea de proyección de la cámara, cuyo tamaño es de 3x4 (Pé, 2007).

### ***Parámetros extrínsecos***

También es importante detenerse en los parámetros extrínsecos, estos relacionan los sistemas de referencia del mundo real y la cámara, describiendo la posición y orientación de la cámara en el sistema de coordenadas del mundo real. Dichos parámetros son el vector de translación y la matriz de rotación (ver Figura 1).

A continuación, se representa el cambio del sistema de referencia del objeto al de la cámara

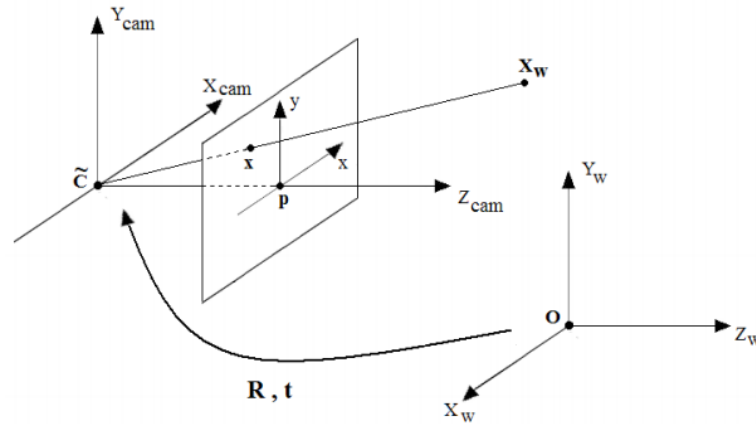


Figura 3. Transformación entre el sistema de referencia en el espacio y el de la cámara (Pé, 2007)

En la Figura 3,  $X_w$  representa las coordenadas no homogéneas de un punto respecto al sistema de referencia del espacio.  $X_c$  representa las coordenadas del mismo punto respecto al sistema de referencia de la cámara.  $\tilde{C}$  es el centro del sistema de referencia, y  $\mathbf{R}$  es la matriz de rotación 3x3 que representa la orientación del sistema de referencia de la cámara (Pé, 2007).

La matriz de parámetros intrínsecos representa la orientación de la cámara respecto a un marco de referencia externo. Específicamente, se define a partir de una rotación y una traslación como:

$$\mathbf{D} = [\mathbf{Rt}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Siendo  $\mathbf{R}$  la matriz de rotación y  $\mathbf{t}$  el vector de traslación. (González, 2003)

## 2.2. Calibración de cámaras

La calibración es el proceso mediante el cual se obtienen los parámetros que definen las condiciones de formación de la imagen, incluyendo geometría interna y óptica de la cámara; además de obtener la orientación y posición de la misma respecto a un objeto de referencia, es decir la estimación de los parámetros intrínsecos y extrínsecos descritos en las secciones anteriores, además de los parámetros de distorsión de la imagen.

Es muy importante realizar una buena calibración del sistema ya que, de ello depende fundamentalmente, el resultado obtenido y representa un paso importante en el proceso de reconstrucción de imágenes. Para dicha calibración, uno de los algoritmos más reconocidos en el estado del arte es el método de Zhang (Zhang, 1998).

El proceso de calibración de cámaras inicia con capturas realizadas a un patrón de ajedrez, después de obtener los puntos donde se ubican las esquinas dentro del patrón, se procede a buscar los parámetros intrínsecos y extrínsecos de la cámara. Para ello es necesario buscar el vector de distorsión, en el cual se encuentran contenidos los coeficientes de distorsión radial y tangencial, mediante la ecuación (8).

$$D = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \quad (8)$$

La distorsión radial se presenta cuando las líneas rectas en una imagen se tornan curvas, estos coeficientes se encuentran utilizando las siguientes expresiones:

$$x_r = x * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \quad (9)$$

$$y_r = y * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \quad (10)$$

$x_r$  y  $y_r$  con las coordenadas, en el mundo real, de las esquinas encontradas con el método Harris (Zhang, 1998), pero sin distorsión y  $r$  se define como:

$$r = x^2 + y^2 \quad (11)$$

Donde  $x$  y  $y$  son las coordenadas, en el mundo real, de los puntos encontrados en la imagen sin distorsión. Dichas coordenadas se obtienen al conocer las medidas de los recuadros en el patrón de ajedrez utilizado. Para encontrar los coeficientes de la distorsión tangencial utilizamos el siguiente sistema de ecuaciones:

$$x_t = x + (2p_1 * xy + p_2 * (r^2 + 2x^2)) \quad (12)$$

$$y_t = y + (p_1 * (r^2 + 2y^2) + 2p_2 * xy) \quad (13)$$

$x_t$  y  $y_t$  son las coordenadas del mundo real sin distorsión tangencial (Zhang, 1998). Al aplicar el vector de distorsión a los puntos analizados se llega a la siguiente expresión:

$$x_d = x_r + x_t \quad (14)$$

$$y_d = y_r + y_t \quad (15)$$

los parámetros  $x_d$  y  $y_d$  son las coordenadas sin distorsión de los puntos analizados en el mundo real. Ahora se transforman estas coordenadas a coordenadas imagen, para ello se utiliza la matriz de la cámara  $\mathbf{K}$  :

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

En la matriz anterior encontramos los parámetros intrínsecos de la cámara los cuales son las distancias focales  $f_x$  y  $f_y$ , en los ejes  $x$  y  $y$  respectivamente, y las coordenadas del centro óptico

de la cámara  $c_x$  y  $c_y$  en los ejes  $x$  y  $y$ . (Mordvintsev & Abid, 2013) Una vez obtenida la matriz  $\mathbf{K}$  se pasan los puntos sin distorsión a coordenadas imagen:

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \mathbf{K} * \begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} \quad (17)$$

$x_p$  y  $y_p$  son las coordenadas imagen de las esquinas detectadas. Al resolver todo el sistema de ecuaciones planteado en esta sección es posible determinar los coeficientes de distorsión y la matriz de la cámara.

### 2.2.1. Transformaciones de imágenes y homografías

Una vez se conoce la distorsión de la cámara, el proceso de calibración permite obtener cada uno de los componentes para la proyección perspectiva:

$$X_C = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]X_W \quad (18)$$

Con lo anterior la expresión general para la matriz de proyección de la cámara es

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \Rightarrow \mathbf{P} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \end{bmatrix} \quad (19)$$

La matriz de proyección se puede expresar en función de la matriz de calibración de la cámara y una matriz de rotación y traslación. Es decir, la matriz de proyección captura el efecto combinado de los parámetros intrínsecos y extrínsecos.

### 2.2.2. Geometría proyectiva

La geometría 2D proyectiva estudia las propiedades del plano proyectivo  $\mathbb{P}^2$  que son invariantes bajo un grupo de transformaciones proyectivas u homografías (A.Zisserman, 2004).

Una proyectividad es una correspondencia lineal entre vectores expresados en coordenadas homogéneas de 3 dimensiones:

$$x = \mathbf{H}x = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (20)$$

Este tipo de transformaciones son invertibles y su inversa es a su vez es una proyectividad. Si multiplicamos la matriz  $\mathbf{H}$  por un factor no nulo, la transformación proyectiva no varía debido, en coordenadas homogéneas, puntos que difieren en un factor de escala no nulo son equivalentes. Si se normaliza la matriz  $\mathbf{H}$  de forma que  $h_{33} = 1$ , esta matriz tendrá 8 grados de libertad en el caso más general

Para efectos de este trabajo, interesa la transformación proyectiva dada por la siguiente matriz:

$$x' = \mathbf{H}_p x = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{pmatrix} x \quad (21)$$

En la ecuación anterior  $\mathbf{v} = (v_1, v_2)^T$ ,  $\mathbf{A}$  es una matriz 2x2 no singular dada por  $\mathbf{A} = s\mathbf{R}\mathbf{K} + \mathbf{t}\mathbf{v}^T$ ,  $s$  corresponde a un factor de escalamiento,  $\mathbf{R}$  es una matriz de rotación 2D,  $\mathbf{K}$  representa una matriz triangular superior normalizada y  $\mathbf{t}$  es un vector de traslación 2D. La matriz tiene ocho grados de libertad, por lo que se necesitan al menos 4 puntos en correspondencia. Las

transformaciones proyectivas nos permiten trabajar con puntos en el infinito de igual forma que con cualquier otro punto (Zisserman, 2000).

### **2.3. Generación de imágenes panorámicas**

La unión entre imágenes ha sido un problema clásico en el procesamiento digital de imágenes. Las técnicas para el alineamiento geométrico se utilizan en aplicaciones tales como interpretación de imágenes médicas o la creación de imágenes panorámicas (Wang S. X., 2011); es esta última aplicación la que resulta relevante para el presente proyecto.

Al hablar sobre el tema es necesario mencionar el método *Stitching* que tiene como finalidad crear una imagen de tipo panorámica de alta calidad a partir de imágenes extraídas de una misma escena (Hua, Li, & Li, 2010). Este método tiene una amplia gama de aplicaciones, desde imágenes satelitales (Horlent, Lamaro, & Lanfri, 2014), mapas digitales (Rejas Ayuga, 2008) y entornos virtuales (Huang, Lin, & Chen, 2015), entre otros. La composición de imágenes para formar una panorámica ha sido ampliamente estudiada, un ejemplo de este tipo de trabajos es el método propuesto por Hua, Li, & Li (2010), que consiste en utilizar el método SIFT (G, 1999) y el método MVSC (clonación transparente de valor medio) para encontrar las coincidencias entre un par de imágenes. El método MVSC utiliza más de un punto de vista, a diferencia del método tradicional cuyo único punto de vista es el centro de la imagen.

En la implementación propuesta por Tang & Jia (2008), se analiza la deformación de las estructuras presentes en las imágenes capturadas, con el fin de minimizarla y obtener una imagen

panorámica más coherente, utilizando el método de *Stitching* para encontrar las zonas coincidentes, corregir la intensidad de la imagen en estas y darles continuidad a las estructuras presentes en dichas zonas.

También se encuentran aplicaciones en el campo de la medicina. Un ejemplo directo es el trabajo realizado por Wang & Wang (2010), en donde se aplica el método *Stitching* en imágenes biomédicas con el fin de obtener mayor información y realizar un análisis más amplio a estas. En esta implementación se propone el método *fade-in and fade-out*, con el cual se elimina las diferencias de textura en la imagen resultante para que sea más uniforme. A la hora de hablar de imágenes de 360 grados se han comenzado a implementar algoritmos. Uno de ellos es generado para el tipo de cámara llamada “ojo de pez” (fisheye) la cual obtiene una captura en coordenadas esféricas, y la función de este algoritmo es transformar la captura de coordenadas esféricas a coordenadas cartesianas, dando como resultado una imagen de 360 grados (A.Zisserman, 2004).

### 3. Metodología

A continuación, se presenta el procedimiento para obtener una imagen panorámica de 360° a partir de N imágenes de entrada. Inicialmente se controla la cámara por medio de la herramienta Wget utilizando el protocolo TCP para acceder a esta a través de una conexión a internet; en este apartado se toman las imágenes necesarias de la escena; también es necesario realizar una serie de capturas a un patrón de ajedrez con el objetivo de calibrar la cámara PTZ y extraer sus parámetros.

Una vez obtenidas las imágenes de la escena completa se procede a llevarlas de coordenadas rectangulares a coordenadas cilíndricas. Luego se buscan los puntos principales, para ello se utiliza el método SIFT. A continuación, se comparan los puntos principales de las imágenes adyacentes para determinar cuáles son coincidentes y poder utilizarlos en el cálculo de la matriz de homografía. Para calcular dicha matriz se implementa el método RANSAC.

Una vez obtenida la homografía se aplica a las imágenes de entrada y finalmente se realiza la concatenación de estas utilizando el método *Blending piramidal*, para obtener una imagen panorámica a partir de las imágenes capturadas con la cámara rotatoria.

#### 3.1. Adquisición de imágenes

En este trabajo, el objetivo es realizar la captura de imágenes con la cámara modelo SNC-EP520. Este es un tipo de cámara PTZ (pan/tilt/zoom) ideal para el monitoreo por video de

zonas amplias (ver datos de la cámara Anexo B). Esta cámara ha sido seleccionada por la potencia de su óptica y por permitir el control de su dinámica (rotación, inclinación y configuración de la óptica) desde el computador, lo cual facilita la automatización de procesos, tales como la creación de imágenes panorámicas.



Figura 4. Cámara PTZ modelo SNC-EP520

La SNC-EP520 captura imágenes de definición estándar (NTSC) en tres alternativas de estándares muy usados de codificación de video. Las imágenes pueden grabarse opcionalmente en una tarjeta de memoria integrada para obtener copias de seguridad a prueba de fallas. La cámara puede controlarse desde un teléfono inteligente o una tableta, para que se pueda visualizar imágenes en forma remota. En esta etapa se implementó una función en MATLAB descrita en el Algoritmo 1.

Algoritmo 1. Pseudocódigo para realizar el ajuste de los parámetros de la cámara como, traslación, rotación y zoom, de forma remota y para realizar las capturas de las imágenes.

---

```

1  Inicio
   Definir la IP de la cámara
   Definir la ruta donde se encuentra el programa Wget
   Definir la ruta donde se encuentra el reproductor de video compatible con la cámara
   Function executecomand
2  Case 'parameter' do
3      if el parámetro ingresado no es numérico do
4          Imprimir error
5      else do
6          Verificar que el parámetro ingresado este dentro del rango de libertad de la cámara
7      end
8          Enviar la orden a la cámara para editar el parámetro 'pan' por medio del programa
           Wget, utilizando la dirección IP
9  end

```

---

La configuración o control de la cámara se realiza mediante MATLAB a través Internet Explorer; se requiere la instalación del SNC Toolbox y de una conexión de red a la cámara que permita enviarle órdenes a través de MATLAB;

El Wget es un paquete de software de libre distribución con licencia GNU<sup>1</sup> que permite realizar conexión entre servidores web de forma simple (permite enviar comando de MATLAB a la cámara) y soporta múltiples protocolos de red, para esta aplicación se utilizó el protocolo TCP (Transmission Control Protocol), es un protocolo estándar que se utiliza para la conexión a internet.

Si se requiere variar por ejemplo la inclinación de la cámara, se ingresa un valor que representa la inclinación verificando que se encuentre dentro del rango que establece el fabricante, esto en MATLAB el cual lo envía a la cámara haciendo uso de la herramienta Wget

---

<sup>1</sup> <https://www.gnu.org/software/wget/>

mencionada. En la Figura 5, se muestra una representación gráfica de la forma en que se conecta la cámara IP al computador, mediante un puerto de red.



Figura 5. La forma en que se conecta la cámara IP al computador, mediante un puerto de red.

### 3.2. Calibración de la cámara

La calibración es el primer paso para generar una imagen panorámica, ya que permite determinar una relación entre lo que aparece en una imagen y lo que se encuentra en el mundo. Permitiendo calcular la matriz de calibración de la cámara, conocida como  $\mathbf{K}$ , útil para conocer la distancia focal que es un parámetro necesario en la transformación de las imágenes a coordenadas cilíndricas y en el cálculo de la matriz de homografía.

El proceso de calibración, se parte de diversas capturas de un patrón de ajedrez, tomadas cada una desde un ángulo diferente. Para mayor precisión se requiere un número de capturas este entre 15 y 20.

Luego de una búsqueda bibliográfica se encontró que Python cuenta dentro de sus librerías con una función que realiza esta tarea llamada *'calibrate'*, la cual requiere como entrada las imágenes tomadas a un patrón de ajedrez por la cámara a calibrar, y como salida arroja los parámetros intrínsecos y extrínsecos de la cámara.

### 3.3. Detección de puntos de interés

#### 3.3.1. Extracción de características

Para crear una imagen panorámica es necesario conocer las características de cada una de las imágenes que la componen. Estas características vienen dadas por parte de los píxeles más representativos de dichas imágenes, conocidos como puntos principales. Para encontrar los puntos principales de una imagen, se deben implementar métodos que sean invariantes a las rotaciones y a escala. El objetivo de estas invarianzas es asegurar que los puntos encontrados son estables frente a las perturbaciones y las transformaciones que pueda sufrir la imagen analizada como por ejemplo un cambio de escala o una rotación.

#### *Método SIFT*

En el año 2004 David Lowe propuso un nuevo algoritmo llamado *Scale-Invariant Feature Transform* o SIFT, para la detección de puntos de interés en una imagen, es invariante a la escala y a la rotación. El algoritmo SIFT se divide en 4 etapas:

#### *Detección de extremos en la escala-espacio*

Esta etapa consta de un filtrado que se realiza en la imagen para llevarla a diferentes escalas. Para llevar a cabo dicho filtrado se encuentra el “Laplaciano de Gauss”, pero es un poco costoso desde el punto de vista computacional, por ello el algoritmo SIFT realiza una diferencia gaussiana como aproximación al Laplaciano de Gauss. La diferencia gaussiana se muestra en la siguiente ecuación:

$$\nabla^2 L(x, y; t) = \frac{1}{2\Delta t} (L(x, y; t + \Delta t) - L(x, y; t - \Delta t)) \quad (22)$$

En la ecuación anterior,  $t$  representa la escala de la imagen,  $\Delta t$  representa el nivel donde se encuentra la escala que se desea comparar y  $L(x, y, t)$  es la representación espacio-escala de una imagen.

Este proceso se realiza para diferentes octavas dentro de la pirámide gaussiana, como se muestra en la Figura 6.

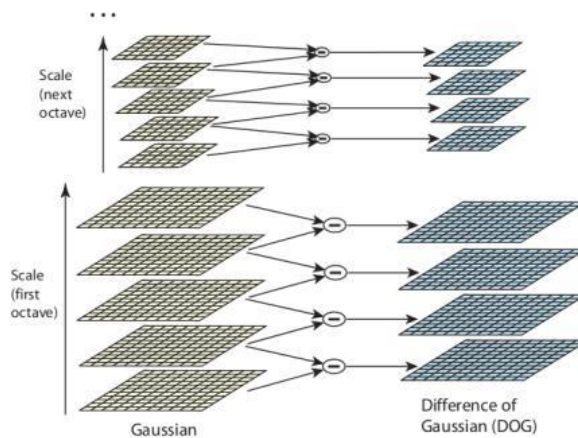


Figura 6. Ilustración del procedimiento para encontrar la diferencia gaussiana a diferentes escalas. (Lowe, 2004).

Una vez obtenida la diferencia gaussiana, se compara la respuesta de cada pixel con los 8 pixeles a su alrededor.

#### *Localización de puntos principales*

Se utiliza la expansión en series de Taylor en la escala espacio de la ecuación (23), para obtener una localización más precisa del máximo extremo

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n \quad (23)$$

Se eliminan tanto el borde como los puntos de bajo contraste entre los seleccionados en la etapa anterior.

#### *Asignación de orientaciones*

Ahora se busca la invariancia a la rotación y para ello se asigna la orientación a cada punto de interés seleccionado. Para esto se toma una vecindad de punto alrededor del punto a analizar, y se calcula la magnitud y la dirección del vector gradiente como se puede apreciar en las ecuaciones (24) y (25).

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y - 1) - L(x, y + 1))^2} \quad (24)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{(L(x + 1, y) - L(x - 1, y))^2}{(L(x, y - 1) - L(x, y + 1))^2} \right) \quad (25)$$

Luego se realiza un histograma para compararlos y el mayor pico es el que dicta la orientación del punto analizado.

#### *Descriptor de puntos de interés*

Finalmente se crea el vector descriptor para cada punto de interés. Para esto, se toma una vecindad de puntos de  $16 \times 16$  alrededor del punto principal, luego esta vecindad se divide en sub-grupos de tamaño  $4 \times 4$ . Para cada sub-grupo se crea un histograma de 8 direcciones, dando como resultado 128 direcciones en total por cada sub-grupo en la vecindad de puntos. La implementación de este algoritmo hace parte de la biblioteca VLFeat.

### ***Método SURF***

Es un método de detección de puntos principales, basado en el algoritmo SIFT, presentado por Herbert Bay, Tinne Tuytelaars y Luc Van Gool en el año 2006, el cual emplea la matriz heseana, pero utiliza un detector basado en el laplaciano. A diferencia del método SIFT que tiene 128 dimensiones en su vector descriptor; esto reduce el tiempo de comparación de coincidencias e incrementa simultáneamente la robustez del vector descriptor. Este método tiene 2 partes principales:

#### 1. *Detección:*

Para encontrar la localización y la escala se utiliza el determinante de dicha matriz. Dado un punto  $X = (x, y)$  en una imagen  $I$ , la matriz heseana  $H(x, \sigma)$  en  $x$  para una escala  $\sigma$  es:

$$\mathbf{H}(x, \sigma) = \begin{pmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{pmatrix} \quad (26)$$

donde  $L_{xx}(x, \sigma)$  es la convolucion de la derivada gaussiana de segundo orden  $\frac{\partial^2}{\partial x^2} g(\sigma)$  con la imagen  $I$  en el punto  $x$ , de forma similar para  $L_{xy}(x, \sigma)$  y  $L_{yy}(x, \sigma)$ . El método SURF (H Bay, 2008) no realiza una aproximación al laplaciano de gauss con la diferencia gaussiana sino con filtros de caja.

Las imágenes integrales permiten una rápida implementación de los filtros de convolución tipo caja. La entrada de una imagen integral  $I \sum(X)$  en un lugar  $X = (x, y)$  representa la suma de todos los pixeles en la primera imagen de entrada dentro de un rectángulo formado por el punto  $X$  y el origen:

$$I(x) = \sum_{i=0}^{i < x} \sum_{j=0}^{j < y} I(i, j) \quad (27)$$

## 2. Descriptor:

Para obtener un punto invariante a las rotaciones, iluminación y orientación se utiliza el wavelet de Haar sobre las direcciones de  $x$  y  $y$  en una región circular de radio  $6s$ , siendo  $s$  es la escala del punto de interés. Una vez calculados para todos los vecinos, se estima la orientación dominante calculando la suma de todos los resultados dentro de una ventana deslizante. El vector más largo de las ventanas es lo que define la orientación del punto de interés. La región de interés se divide regularmente en  $4 \times 4$  subregiones cuadradas, y para cada una de ellas se calculan unas características simples, el Wavelet de Haar para  $x$  y  $y$ , y se suavizan los resultados mediante un filtro Gaussiano (para ofrecer una mayor robustez a deformaciones, ruido y traslaciones), obteniendo  $dx$  y  $dy$ .

Para cada sub-región se suman los resultados  $dy$  y  $dx$ , además de calcularse su valor absoluto  $|dx|$  y  $|dy|$ . De esta manera, cada subregión proporciona un vector, que está compuesto por:

$$v = \left( \sum dx, \sum dy, \sum |dx|, \sum |dy| \right) \quad (28)$$

Finalmente, el descriptor del SURF se obtiene mediante la unión de los vectores de las subregiones. El algoritmo para detectar los puntos de interés se encuentra dentro de las librerías con las que cuenta MATLAB.

### *Método Harris*

Este operador fue desarrollado en 1988 por Mike Stephens y Chris Harris, interesados en utilizar técnicas de análisis de movimiento para interpretar el entorno basándose en imágenes de una sola cámara móvil (C. D. Kuglin, 1975). El algoritmo detecta la diferencia de intensidad entre pixeles adyacentes y selecciona los pixeles de mayor valor en toda la imagen. Para esto, dicho algoritmo analiza una vecindad de puntos, la comparación de intensidades se hace de la siguiente forma:

$$e(u, v) = \sum_{x,y} w(x, y) * [I(x + u, y + u) - I(x, y)]^2 \quad (29)$$

$w(x, y)$  es la función de ventana,  $I(x + u, y + u)$  el cambio de intensidad y  $I(x, y)$  es la intensidad del pixel. La función de ventana es una ventana gaussiana la cual da el peso de los pixeles. Para realizar la detección de esquinas se le aplica la expansión en series de Taylor a la ecuación (29) y se obtiene:

$$e(u, v) = (uv) * \mathbf{M} * \begin{pmatrix} u \\ v \end{pmatrix} \quad (30)$$

en la ecuación anterior  $\mathbf{M}$  es,

$$\mathbf{M} = \sum_{x,y} w(x, y) * \begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix} \quad (31)$$

$I_x$  y  $I_y$  son derivadas aplicadas a la imagen tanto en la dirección x como en y respectivamente.

Finalmente llegamos a la siguiente expresión:

$$\mathbf{R} = \det(\mathbf{M}) - k(\text{trace}(\mathbf{M}))^2 \quad (32)$$

donde

$$\det(\mathbf{M}) = \lambda_1 * \lambda_2 \quad (33)$$

$$\text{trace}(\mathbf{M}) = \lambda_1 + \lambda_2 \quad (34)$$

$\lambda_1$  y  $\lambda_2$  son los valores propios de la matriz  $\mathbf{M}$ . Para decidir si el punto analizado es una esquina se realiza el siguiente análisis

- Cuando  $|R|$  es pequeña,  $\lambda_1$  y  $\lambda_2$  son y el punto analizado pertenece a un plano.
- Cuando  $R < 0$ , significa que  $\lambda_1 \gg \lambda_2$  o viceversa y el punto analizado pertenece a un borde.
- Cuando  $|R|$  es grande, significa que  $\lambda_1$  y  $\lambda_2$  y son aproximadamente parecidos, entonces el punto analizado es una esquina.

Para la implementación de este algoritmo se utiliza las librerías de MATLAB.

### 3.3.2. Detección de coincidencias

Una vez obtenidos los puntos principales de una imagen, junto a sus vectores descriptores, es necesario saber cuáles de estos puntos principales son puntos coincidentes entre las dos imágenes. Para obtener los puntos coincidentes se utiliza el método Matching (Rubin, 1973). Dicho método dice que dados un conjunto de puntos  $P = \{p_1, p_2, p_3, \dots, p_n\}$  en un espacio vectorial  $X$ , estos deben ser procesados de tal forma que si se ingresa un punto  $q$  en el espacio  $X$ , el punto  $p$  con la menor distancia al punto  $q$  corresponde a un punto coincidente. En este proyecto, se asume que el espacio vectorial  $X$  corresponde al espacio euclídeo y para encontrar la

distancia entre los puntos  $p$  y  $q$  se utiliza la expresión de la distancia euclidiana (35) (Marius Muja, 2009).

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (35)$$

El cálculo de correspondencias mediante el método descrito anteriormente se conoce como método Match (ver Algoritmo 2) y su uso para el cálculo de homografías se ilustra en la Figura 7

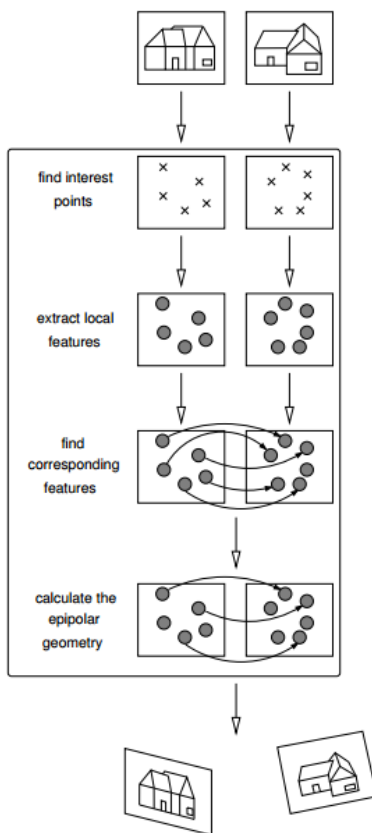


Figura 7. Descripción gráfica de pasos necesarios para encontrar la geometría epipolar. La entrada es un par de imágenes que muestran una escena común (Jacobus & Hollander, 2007)

A continuación, se tiene el algoritmo del método Match:

Algoritmo 2. Pseudocódigo del método *Match*

---

```
1  Inicio
   leer las matrices que contienen los vectores descriptores
   Extraer la cantidad de vectores descriptores por cada matriz de entrada
   Definir la variable contador
2  For j=1 hasta el tamaño de la matriz 2DO
3      for i=1 hasta el tamaño de la matriz 1 DO
4          guardar la posición del vector en la matriz 2 que se va a comparar
          comparar 1 vector de la matriz 2 con todos los vectores de la
          matriz 1
5      end
6      Organizar los resultados de menor a mayor
7      if la división entre los valores más pequeños es menor a 1.5 DO
8          Aumentar en 1 la variable contador
          guardar la posición del vector de la matriz 1 junto con la posición
          del vector de la matriz 2 con el valor más pequeño en la comparación
9      end
10 end
```

---

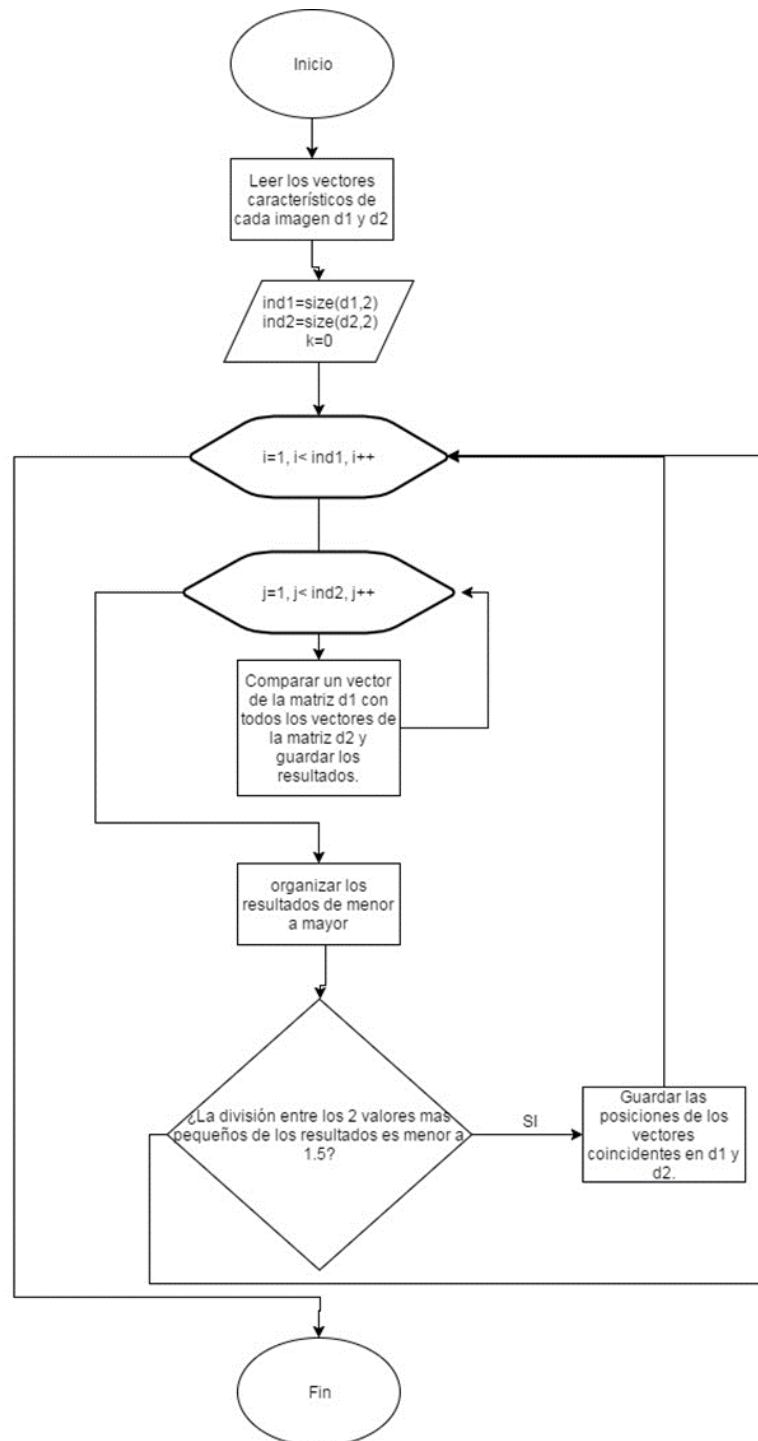


Figura 8. Diagrama de flujo del método Match

### 3.4. Estimación de homografías

Dado que una homografía puede relacionar una imagen de una escena plana en el mundo con dicha escena plana, y puede relacionar varias imágenes de la misma escena formadas a partir de distintas posiciones del centro de cámara, entonces hay muchas situaciones en visión por computador donde la estimación de una homografía puede ser requerida.

El objetivo del algoritmo de cálculo automático de homografía 2D implementado en este proyecto es estimar la -matriz de homografía 2D  $\mathbf{H}$  que relaciona un par de este tipo de imágenes (Dubrofsky, 2009).

Considerando un conjunto de  $N$  puntos de una imagen, conociendo sus correspondencias en una segunda imagen  $x_i \leftrightarrow x'_i$ . Entonces se debe determinar la matriz  $\mathbf{H}$  que relaciona los dos conjuntos de puntos entre sí:

$$x'_i = \mathbf{H}x_i \tag{36}$$

Lo primero que hay que determinar es el número mínimo de puntos necesarios para estimar la matriz  $\mathbf{H}$ . Este número depende del tipo de transformación proyectiva pues se necesitan tantas ecuaciones como grados de libertad tenga la matriz  $\mathbf{H}$ . Además, de cada par de correspondencias  $x_i \leftrightarrow x'_i$  se obtienen 2 ecuaciones linealmente independientes (un punto 2D tiene 2 grados de libertad, uno por la componente  $x$  y otro por la  $y$ ) (Basagoitia, 2004). Por tanto, el número mínimo de puntos necesarios de acuerdo al tipo de transformación son:

- Afinidad: 6 grados de libertad requiere 3 puntos.
- Homografía: 8 grados de libertad requiere 4 puntos

La homografía se puede calcular fácilmente con el algoritmo DLT (Direct Linear Transformation) normalizado (Hartley & Zisserman, 2001). Sin embargo, los emparejamientos obtenidos, independientemente del método empleado, suelen incluir emparejamientos espurios. Para evitar que estos emparejamientos afecten al cálculo de los parámetros geométricos se utilizan procedimientos robustos.

### 3.4.1. Transformación lineal directa (DLT)

En las secciones anteriores se propusieron las transformaciones que modelan las distintas deformaciones que presentan las imágenes. El proceso operativo es el mismo, con la única diferencia es referente al número mínimo de emparejamientos requeridos para su cálculo. La homografía se puede calcular fácilmente con el algoritmo DLT (Direct Linear Transformation) normalizado (Hartley & Zisserman, 2001). Esta es una forma alterna de encontrar la matriz de transformación de imágenes, consiste en el uso de mínimos cuadrados. Se toma la transformación causada por la cámara giratoria, es decir, se tendrá en cuenta la matriz de parámetros extrínsecos para calcular la homografía (Hoff, 2011).

#### *Ajustes de mínimos cuadrados a una línea*

Teniendo los datos  $(x_i, y_i)$ , se pretende ajustar los datos a una recta de la forma

$$y = f(x) = mx + b \quad (37)$$

y de esta manera encontrar los parámetros  $(m, b)$ , que minimizan la función

$$e = \sum_i |y_i - f(x_i)|^2 \quad (38)$$

Escribiendo una función lineal  $\mathbf{Ax} = \mathbf{b}$

Los parámetros a ajustarse están en el vector  $x$  y los datos de entrada son  $\mathbf{A}$  y  $\mathbf{b}$ , y los parámetros del vector son:

$$x = \begin{pmatrix} m \\ \mathbf{b} \end{pmatrix} \quad (39)$$

entonces la ecuación lineal está dada por

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{pmatrix} \begin{pmatrix} m \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad (40)$$

así que para una línea

$$\mathbf{A} = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_3 \end{pmatrix} \quad (41)$$

minimizando

$$\mathbf{e} = |\mathbf{Ax} - \mathbf{b}|^2 \quad (42)$$

Expandiendo la ecuación (42)

$$\mathbf{e} = x^T(\mathbf{A}^T\mathbf{A})x - 2x^T(\mathbf{A}^T\mathbf{b}) + |\mathbf{b}|^2 \quad (43)$$

derivando en término de  $x$  e igualando a cero y despejando la variable  $x$  se obtiene:

$$x = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \quad (44)$$

en MATLAB la forma de implementar la ecuación (44) es:

$$x = \mathbf{A}/\mathbf{b} \quad (45)$$

Para encontrar la matriz de transformación de imágenes se requiere conocer un conjunto de puntos de correspondencias y una estimación de los parámetros de transformación (matriz de

parámetros de la cámara). La matriz de rotación y traslación de la cámara en el eje y se describe a continuación:

$$\begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix} = \begin{pmatrix} c & -s & t_x \\ s & c & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix} \quad (46)$$

$$\mathbf{R} = \begin{pmatrix} c & -s & t_x \\ s & c & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

donde

$$c = \cos\theta, s = \sin\theta \quad (47)$$

$$x_B = cx_A - sy_A + t_x \quad (48)$$

$$y_B = cx_A + sy_A + t_y \quad (49)$$

Puesto en la forma  $Ax = b$ ,

$$\mathbf{A} = \begin{pmatrix} x_A^{(1)} & -y_A^{(1)} & 1 & 0 \\ y_A^{(1)} & x_A^{(1)} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ y_A^{(N)} & x_A^{(N)} & 0 & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} c \\ s \\ t_x \\ t_y \end{pmatrix}, \mathbf{b} = \begin{pmatrix} x_B^{(1)} \\ y_B^{(1)} \\ \vdots \\ y_B^{(1)} \end{pmatrix} \quad (50)$$

El algoritmo DLT es sensible a estos factores y el resultado es variante (Harley, 2003). Para resolver esto, se recomienda hacer una normalización a los datos antes de aplicar el algoritmo y aplicar transformaciones de similitud, pues así el resultado no se ve afectado por la escala y orientación escogidas (Harley, 2003).

En el Algoritmo 3, propuesto por Zisserman (Harley, 2003), se representa una manera de calcular de forma automática una homografía entre dos imágenes a partir del método anterior:

OBJETIVO: dadas  $n \geq 4$  correspondencias 2D-2D, determine la matriz de la homografía  $H$ , tal que  $x'_i = Hx_i$ .

Algoritmo 3. Algoritmo del método DLT normalizado

---

```

1  Inicio
   Aplicar la transformada de similitud a los puntos de entrada
2  For i=1 hasta el número de puntos DO
3      Normalizar los puntos de entrada, multiplicándolos por su transformada de similitud y guardar el
       resultado como sus matrices de similitud  $T_i$  y  $T_w$ .
4  end
5  For i=1 hasta el número de puntos DO
6      Armar el sistema de ecuaciones para encontrar los vectores fila de la matriz  $A$ .
7  end
8      Aplicar el método de descomposición en valores singulares a la matriz  $A$  y obtener  $U$ ,  $D$ , y  $V$ .
       Tomar los coeficientes de la matriz  $H$  de las componentes del vector  $V$ 
       Normalizar la matriz  $H$ 
       Multiplicar la matriz  $H$  por la inversa de  $T_i$  por izquierda y por  $T_w$  por derecha

```

---

Enseguida se presenta el diagrama de bloques del método DLT normalizado

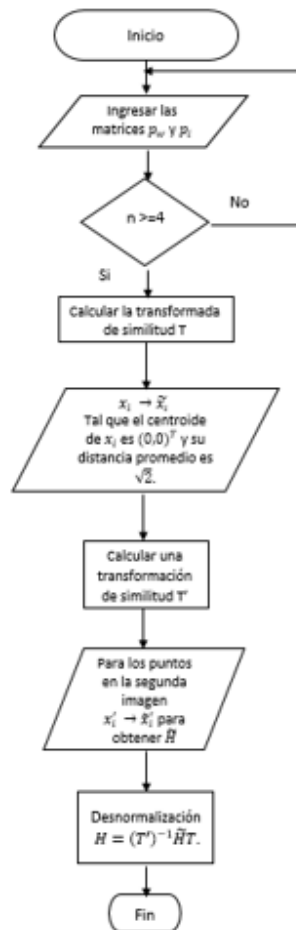


Figura 9. Diagrama de flujo del método DLT normalizado (Harley, 2003)

### 3.4.2. Método RANSAC

El método RANSAC (Fischler & Bolles, 1981) es el más conocido y se basa en ir tomando conjuntos de puntos (el menor número permitido), calcular el modelo a partir de esos puntos, y estimar cuántos puntos del conjunto completo están dentro de una distancia mínima con respecto a dicho modelo.

El procedimiento de este método es diferente a las técnicas convencionales, ya que en vez de utilizar tantos datos como sea posible y obtener una respuesta inicial y, a partir de esta, eliminar los datos erróneos, RANSAC usa una cantidad de datos tan pequeña como sea posible y luego se amplía este conjunto de datos. En otras palabras, a la hora de encontrar un modelo en específico, este método toma aleatoriamente una cantidad mínima de datos para estimarlo y luego de esto comprueba el modelo encontrado con los datos restantes (A.Zisserman, 2004).

En este proyecto se desea obtener el modelo de la matriz de transformación de perspectiva, conocida también como matriz de homografía (**H**); con el objetivo de llevar las imágenes capturadas a un mismo sistema de coordenadas. El sistema que se plantea es el siguiente:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} * \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad (51)$$

Teniendo dos imágenes como entrada, donde  $x_2$  y  $y_2$  son la posición de un punto coincidente en la *imagen 2*, y al mismo tiempo  $x_1$  y  $y_1$  es la posición de su correspondencia en la *imagen 1*; básicamente al realizar esta multiplicación se lleva el punto  $p_1$  de la *imagen 1* a su correspondencia en la *imagen 2* que es el vector  $p_2$ . Para encontrar los coeficientes de la matriz **H** es necesario normalizar las coordenadas  $x$  y  $y$  como se muestra a continuación:

$$x_2 = \frac{h_{11} * x_1 + h_{12} * y_1 + h_{13}}{h_{31} * x_1 + h_{32} * y_1 + h_{33}} \quad (52)$$

$$y_2 = \frac{h_{21} * x_1 + h_{22} * y_1 + h_{23}}{h_{31} * x_1 + h_{32} * y_1 + h_{33}} \quad (53)$$

Para la presente aplicación el coeficiente  $h_{33}$  no puede ser cercano a cero ya que dichos puntos no pueden estar en líneas o planos en el infinito.

Teniendo en cuenta lo anterior nuestro sistema de ecuaciones queda de la siguiente forma:

$$x_2 = \frac{h_{11} * x_1 + h_{12} * y_1 + h_{13}}{h_{31} * x_1 + h_{32} * y_1 + 1} \quad (54)$$

$$y_2 = \frac{h_{21} * x_1 + h_{22} * y_1 + h_{23}}{h_{31} * x_1 + h_{32} * y_1 + 1} \quad (55)$$

Eliminando la fracción y llevando las ecuaciones a la forma matricial obtenemos:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 * x_2 & -y_1 * x_2 \end{bmatrix} * \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} \end{bmatrix}^T = x_2 \quad (56)$$

$$\begin{bmatrix} 0 & 0 & 0 & x_2 & y_2 & 1 & -x_1 * y_2 & -y_1 * y_2 \end{bmatrix} * \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} \end{bmatrix}^T = y_2 \quad (57)$$

La ecuación (56) corresponde a la multiplicación entre el vector fila, que resulta al despejar la ecuación (52) y el vector columna que contiene los coeficientes de la matriz de homografía H y en la ecuación (56) es igual, se obtiene al despejar la ecuación (53). Ya que el sistema a resolver es el siguiente:

$$\mathbf{A} * \mathbf{h} = \mathbf{b} \quad (58)$$

La matriz  $\mathbf{A}$  se define como:

$$\mathbf{A} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (59)$$

Donde  $a_x$  y  $a_y$  son:

$$a_x = (x_1 \quad y_1 \quad 1 \quad 0 \quad 0 \quad 0 \quad -x_1 * x_2 \quad -y_1 * x_2) \quad (60)$$

$$a_y = (0 \quad 0 \quad 0 \quad x_2 \quad y_2 \quad 1 \quad -x_1 * y_2 \quad -y_1 * y_2) \quad (61)$$

Y los vectores  $\mathbf{b}$  y  $\mathbf{h}$  son:

$$\mathbf{h} = \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (62)$$

Tal como se plantea el sistema hasta ahora, con una pareja de puntos coincidentes obtenemos un sistema de 2 ecuaciones y 8 incógnitas, como se explicó anteriormente el parámetro  $h_{33}$  se definió con un valor igual a 1.

Una vez resuelto el sistema de la ecuación (58), se prueba la matriz de homografía encontrada. Para realizar dicha prueba se aplica la homografía a los puntos coincidentes de la *imagen 1* y al resultado se le calcula el error con el punto correspondiente en la *imagen 2* utilizando el mismo método que se implementó para comparar vectores descriptores, la diferencia euclidiana. Si el error resultante es menor a cierto umbral entonces la homografía es correcta y es guardada. Este procedimiento se repite una cantidad N de veces con el fin de encontrar la homografía que más se ajuste a las imágenes analizadas (A.Zisserman, 2004), al final se selecciona la homografía con mayores aciertos al ser aplicada.

El Pseudocódigo y el diagrama de flujo para el cálculo de la homografía mediante RANSAC, tal como se describió anteriormente, se encuentra en el Algoritmo 4 y la Figura 10, respectivamente.

Algoritmo 4. Algoritmo para el cálculo de la matriz de homografía mediante el método

RANSAC

---

```

1  Inicio
  Leer los puntos principales de las imágenes de entrada
  Leer lista de coincidencias
  N= número de puntos coincidentes
  Definir las matrices A y R para resolver el sistema de ecuaciones
  Definir el umbral de error
2  For f=1 hasta 100 DO
3      Seleccionar 4 pares de puntos coincidentes de forma aleatoria
4          For i=1 hasta 4 DO
5              Plantear e sistema de ecuaciones con 1 par de puntos seleccionados
6          end
7          Resolver el sistema de ecuaciones y encontrar la matriz de homografía
8  For k=1 hasta el numero de coincidencias DO
9      Tomar 4 puntos coincidentes de forma aleatoria
      Aplicar la matriz de homografía encontrada a uno de los puntos seleccionados y
      compararlos con su punto coincidente
      guardar el error encontrado en la comparación
10     if el error encontrado es menor que el error definido DO
11         Aumentar en 1 la variable contador
12     end

```

---

En la siguiente figura se presenta el diagrama de bloques para el método RANSAC

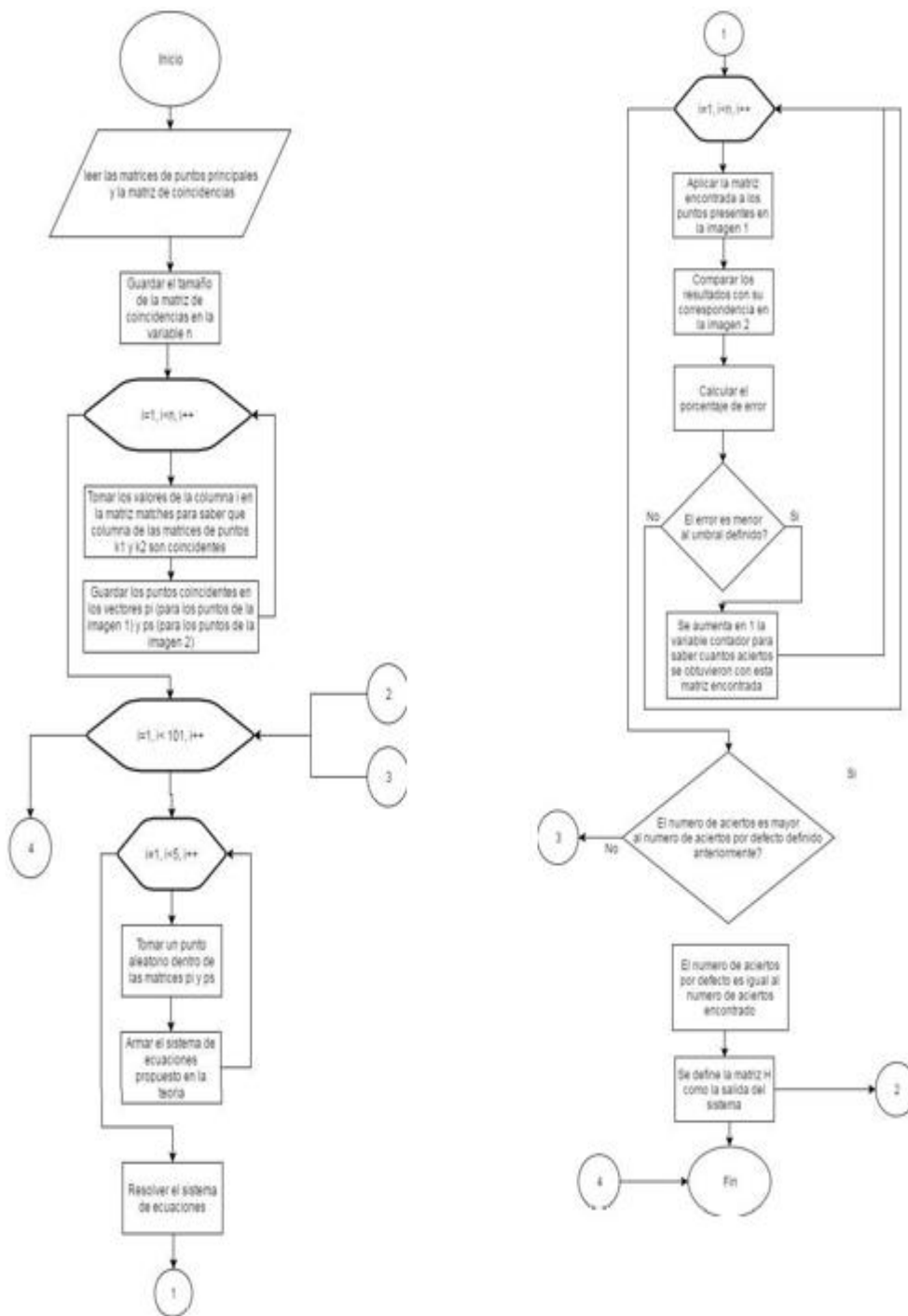


Figura 10. Diagrama de bloques de método RANSAC

### 3.5. Generación de imágenes panorámicas

#### 3.5.1. Método *Stitching* y *Blending*

Basados en los resultados obtenidos en la etapa anterior, se procede aplicar el método de *Stitching* a un grupo de imágenes para concatenarlas y llegar a una imagen final que contenga el 100% de la escena que se pretende capturar. También se hace necesario refinar la imagen resultante para minimizar los errores de continuidad, intensidad de color y corregir también los bordes de la imagen, mediante la técnica *Blending* (combinar), para emparejar las tonalidades de los colores a lo largo de la misma. Todo esto con el fin de hacer que la imagen sea lo más uniforme posible en todas sus características.

Para eliminar dichos defectos, se pueden implementar dos algoritmos, *Feathering* y *Pyramid Blending* (Verma, s.f.).

#### ***Feathering (mezcla de imágenes a partir de un promedio ponderado)***

Los valores de los píxeles en las regiones mezcladas son el promedio ponderado de las dos imágenes superpuestas como se representa en la Figura 11. A veces este enfoque simple no funciona (por ejemplo, en presencia de diferencias de exposición).

La ecuación (63) corresponde a la máscara o promedio ponderado de dos imágenes

$$PB(i, j) = (1 - w) * PA(i, j) + w * PB(i, j) \quad (63)$$

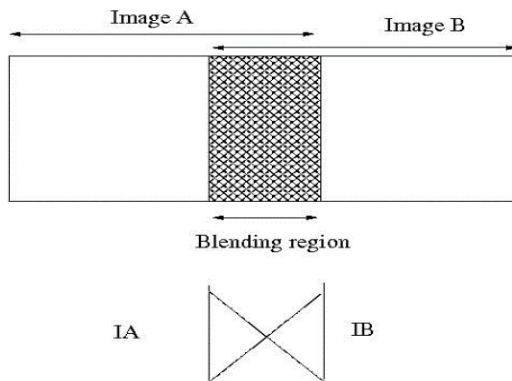


Figura 11. Feathering or center weighting image *Blending* (Shum, 1997)

### ***Pyramid Blending***

Algoritmo que utiliza la pirámide Gaussiana para mezclar las imágenes mientras que mantiene la característica significativa en el tiempo medio. Reduce la imagen en diferentes niveles (tamaños). Posteriormente expande el gaussiano hacia la palanca inferior y resta de la imagen en esa palanca para adquirir la imagen.

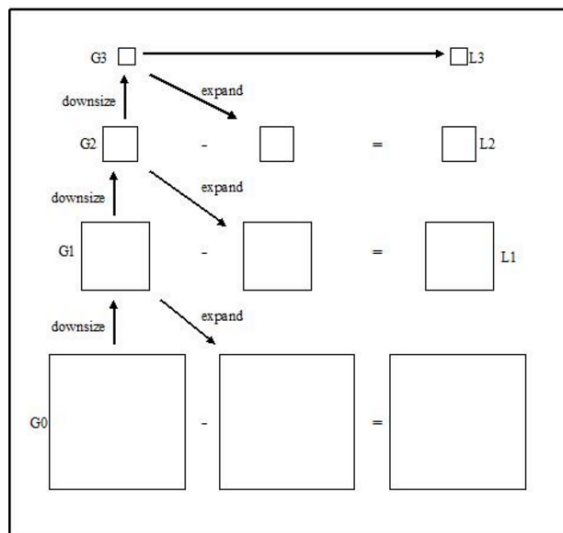


Figura 12. *Pyramid Blending* (Shum, 1997)

Al realizar varias pruebas se encontró que el método *Blending piramidal* hace una transición más suave entre una imagen y la otra, mientras que en el *Feathering Blending*, es notable a simple vista los límites entre la primera y la segunda imagen. Por este motivo se implementó el primer método.

En el Algoritmo 5 se resume la implementación del método *Blending piramidal*

Algoritmo 5. Algoritmo del método *Blending piramidal*

---

```

1  Inicio
  Leer las imágenes de entrada y el número n de niveles
  Crear una máscara para la unión de las imágenes
  Pasar las imágenes a formato de doble precisión
  Crear una pirámide laplaciana de n niveles ingresados, para cada imagen de entrada
  Crear una pirámide gaussiana para la máscara creada, con los n niveles ingresados
  Crear un arreglo de n celdas
2  For i=1 hasta el número de niveles DO
3    Se recorta el nivel i de la pirámide gaussiana para que tenga el mismo tamaño del nivel i de las
    pirámides laplacianas y se guarda como la variable I.
    se multiplica el nivel i de la pirámide gaussiana de la primera imagen con la variable I
    Se multiplica el nivel i de la pirámide laplaciana de la imagen 2 con la variable I invertida
    Se suman los resultados de las multiplicaciones anteriores y se guardan en el nivel i del arreglo
    de celdas
4  end
5  Se reconstruye la imagen de salida a partir de los resultados guardados en los niveles del arreglo
    de celdas

```

---

La siguiente Figura 14, tiene el diagrama de flujo del método anterior

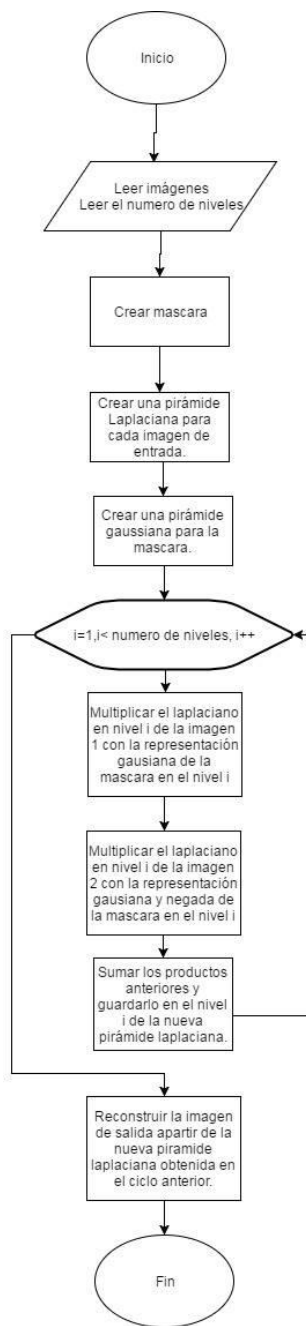


Figura 13. Diagrama de flujo del método Blending piramidal

## 4. Resultados

En este capítulo se explica el desarrollo del objetivo final del proyecto, la obtención de imágenes panorámicas a partir de las imágenes capturadas por una serie de cámaras con especial atención en las cámaras rotatorias, y los resultados obtenidos en cada paso del algoritmo.

Los pasos son el registro de imágenes, la deformación de imágenes y la mezcla de imágenes. El registro de imágenes implica la detección de características, el descriptor de características, la comparación de características y la estimación del modelo de transformación. Todo esto se discute en los capítulos anteriores (ver diagrama de proceso para obtener una imagen panorámica en **Anexo A**).

El sistema desarrollado se implementó en MATLAB, a excepción de la calibración de las cámaras que se realizó en Python, haciendo uso de las librerías VLFeat versión 0.9.20 (Fulkerson, 2008).

Los resultados se dividen en dos partes, en la primera se realizan pruebas con una cámara digital y en la segunda con una cámara PTZ, estas a su vez se subdividen en tres unidades en las cuales se utilizan cada uno de los tres métodos de estimación de características.

### 4.1. Comparación de algoritmos de detección de puntos de interés

Al momento de unir un par de imágenes se necesita información de cada una de estas, para determinar cuáles son los puntos coincidentes y posteriormente realizar el cálculo de la

homografía. Dicha información se obtiene al extraer los puntos principales de cada imagen. En esta sección se comparan 3 métodos diferentes de extracción de parámetros, mediante una serie de pruebas que se realizarán con un número determinado de pares de imágenes. El objetivo de esta comparación es determinar cuál de estos métodos es el más adecuado, al momento de encontrar los puntos principales de una imagen, y cuál de ellos arroja el menor porcentaje de error al final del proceso.

Inicialmente se toma un banco de pares de imágenes a las cuales se les aplican los algoritmos SIFT, SURF y Harris. Posteriormente, se aplican diferentes transformaciones a la primera imagen de cada par y se determina el número de puntos principales en cada una de estas.

Finalmente se calcula el porcentaje de error entre la cantidad de puntos principales detectados por cada uno de los tres métodos y los puntos coincidentes encontrados. Los criterios a tener en cuenta en estas pruebas son la cantidad de puntos principales encontrados y la cantidad de puntos coincidentes encontrados, entre la imagen principal y las transformaciones realizadas en la misma.

Para iniciar las pruebas se aplican 4 transformaciones diferentes, dos de ellas son transformaciones de escala, la primera para reducir la imagen a la mitad de su tamaño original y la segunda para aumentarla de tamaño; como tercera y cuarta transformación se aplica ruido blanco y una rotación de 45 grados en el eje Y a la *imagen 1*. Para el análisis también se tiene en cuenta una segunda imagen de la misma escena presente en la imagen principal pero tomada desde un punto de vista diferente, para que solo una parte de dicha imagen sea coincidente con la

imagen principal. Estas transformaciones se realizan para comparar el desempeño de los diferentes métodos de extracción de características en la imagen principal y las diferentes versiones encontradas de la misma.

Se inicia el análisis comparando la cantidad de puntos principales extraídos con cada método para cada una de las imágenes utilizadas. Como se puede ver en la Tabla 1, el método SIFT obtiene en promedio un número mayor de puntos principales en cada una de las imágenes analizadas. También se observa que el método Harris tiene el desempeño más bajo en las imágenes con rotación de 45 grados frente a la imagen normal, omitiendo puntos principales que había tomado en cuenta en la primera imagen. Finalmente podemos ver que el algoritmo SURF aumenta el número de puntos encontrados al agregar ruido blanco a la *imagen 1*, esta cantidad de puntos extra pueden el resultado de dicho ruido blanco y por lo tanto se les considera como errores de estimación del método.

Tabla 1

*Número promedio de puntos principales encontrados para cada una de las imágenes analizadas con los métodos a comparar.*

	Imagen 1	Imagen 2	Imagen a 45 grados	Imagen con ruido blanco	Imagen a escala de 0.5	Imagen a escala de 1.5
<b>SIFT</b>	3183	3146	3333	2263	697	9124
<b>SURF</b>	738	808	799	943	304	1079
<b>HARRIS</b>	503	488	247	326	274	625

*Nota:* \* Número promedio de puntos principales encontrados en la imagen a analizar y sus diferentes transformaciones, en cada una de las pruebas realizadas.

A continuación, se extraen los puntos principales de las imágenes transformadas, y se realiza el *Match* para determinar las coincidencias de la *imagen 1* con las imágenes transformadas y la *imagen 2*.

En la Tabla 2 se tiene el promedio de puntos coincidentes entre la *imagen 1* y las diferentes imágenes transformadas. Al igual que en la tabla anterior, el método SIFT al tener un número mayor de coincidencias en cada una de las pruebas, esto se debe a que dicho método arroja un número mayor de puntos principales. También se puede observar que al comparar la *imagen 1* con la imagen a 45 grados el 70% de los puntos comparados son coincidentes, mientras que para el método SURF y Harris solo se tienen 30% y 25%; ilustrando que el método SIFT tiene un mejor comportamiento ante una rotación, las cuales suelen presentarse al comparar 2 imágenes de una misma escena. Al comparar las coincidencias entre las imágenes 1 y 2 nos encontramos con porcentajes del 11% para el método SIFT, 20% para el método SURF, y 28% para el método Harris aproximadamente. A pesar de tener un menor porcentaje de coincidencias frente a la cantidad de puntos detectados, el número de coincidencias sigue siendo mayor por parte del método SIFT con respecto a los otros dos métodos; de igual forma al comparar los demás resultados es notorio que los Matches entre la *imagen 1* y las imágenes transformadas tienen un mayor acierto utilizando el método SIFT.

Tabla 2

*Porcentaje de Matches entre la imagen 1 y las imágenes transformadas, para cada uno de los métodos analizados*

	imagen 2	Imagen a 45 grados	Imagen con ruido blanco	Imagen a escala de 0.5	Imagen a escala de 1.5
<b>SIFT</b>	11.78%	69.78%	21.11%	19.48%	81%
<b>SURF</b>	19.64%	31.44%	46.34%	27.51%	58.67%
<b>HARRIS</b>	28.43%	25.05%	31.61%	27.63%	31.61%

*Nota:* \*Porcentaje de aciertos entre la imagen analizada y las diferentes transformaciones de este, con cada uno de los métodos a validar.

El algoritmo Harris a pesar de tener un desempeño comparable con los demás métodos, al visualizar los resultados obtenidos después de comparar los puntos detectados con dichos métodos se puede observar que las coincidencias encontradas son erróneas, ya que los puntos encontrados en la *imagen 1* no son correspondientes con ningún punto encontrado en las demás pruebas. Esto se debe a que el método Harris está diseñado para encontrar diferencias de tonalidades en una imagen, es decir, límites entre zonas de diferentes colores. Por lo tanto, dicho método es utilizado en la práctica para la detección de bordes y esquinas en una imagen, pero no en el cálculo de la homografía. Esto se debe a que los puntos encontrados en la *imagen 1* pueden ser coincidentes con un punto, o varios, de forma errónea en la *imagen 2*, ya que el criterio de selección de este método solo tiene en cuenta los píxeles según los cambios de tonalidad entre el mismo y sus píxeles adyacentes y los toma en cuenta dependiendo las condiciones en las que se utilice (detección de borde o de esquinas).

Finalmente, los métodos SIFT y SURF en donde se puede apreciar, según las dos tablas expuestas anteriormente, el segundo tiene un desempeño más bajo, especialmente en imágenes con rotaciones y ruido. Al momento de encontrar la homografía se necesita la mayor cantidad posible de puntos coincidentes, al mismo tiempo con un margen de error pequeño; el algoritmo SURF a pesar de tener una cantidad considerable de detecciones presenta un porcentaje de error más grande frente al algoritmo SIFT lo cual puede perjudicar futuros cálculos.

## 4.2. Desempeño en la estimación de homografías

Como se mencionó, alternativamente al método RANSAC, existe otra forma para calcular la matriz de homografía, el método DLT (transformación lineal directa). Los algoritmos presentados para el cálculo de la matriz de homografía se describieron anteriormente, Para cada uno de los métodos de extracción de características se calcula su matriz de homografía.

Para realizar el cálculo de la homografía se tienen en cuenta los puntos principales entre dos imágenes, para ello se comparan los puntos detectados de la *imagen 1* e *imagen 2*, y se determina cuáles de estos son coincidentes. Dentro de tal conjunto pueden existir errores que también serán tomados en cuenta a la hora de calcular la matriz de homografía.

Después de diversas pruebas realizadas se encontró que el método DLT es menos exacto en comparación con el método RANSAC, debido a que este toma solo cuatro correspondencias al azar dentro de los cuales puntos erróneos y calcula la matriz de homografía a partir de estos, arrojando resultados inexactos. Mientras que el método RANSAC, al ser iterativo, va seleccionando de forma aleatoria cuatro puntos diferentes y descarta los puntos para los cuales la homografía no cumple con cierto umbral.

Para evaluar los dos métodos primero se calculan las matrices de homografía, luego se aplica cada homografía a los puntos coincidentes encontrados en la imagen 1 y se comparan con sus correspondencias en la imagen 2. Esta comparación se realiza utilizando el concepto de la distancia euclidiana, que consiste en tomar dos puntos y extraer la distancia entre ellos. Si esta distancia encontrada es menor a cierto umbral que se ha definido previamente, entonces el par de

puntos analizados son coincidentes y serán tomados como un acierto, para la matriz que se aplicó en esta prueba; al finalizar el análisis con todos los puntos coincidentes, se extrae el porcentaje error utilizando el número de aciertos frente al número total de puntos coincidentes. El método seleccionado corresponde al método con un mayor porcentaje de aciertos.

A continuación, se puede apreciar los porcentajes de error obtenidos con cada uno de los métodos anteriormente descritos, esta prueba se realizó con seis pares distintos de imágenes:

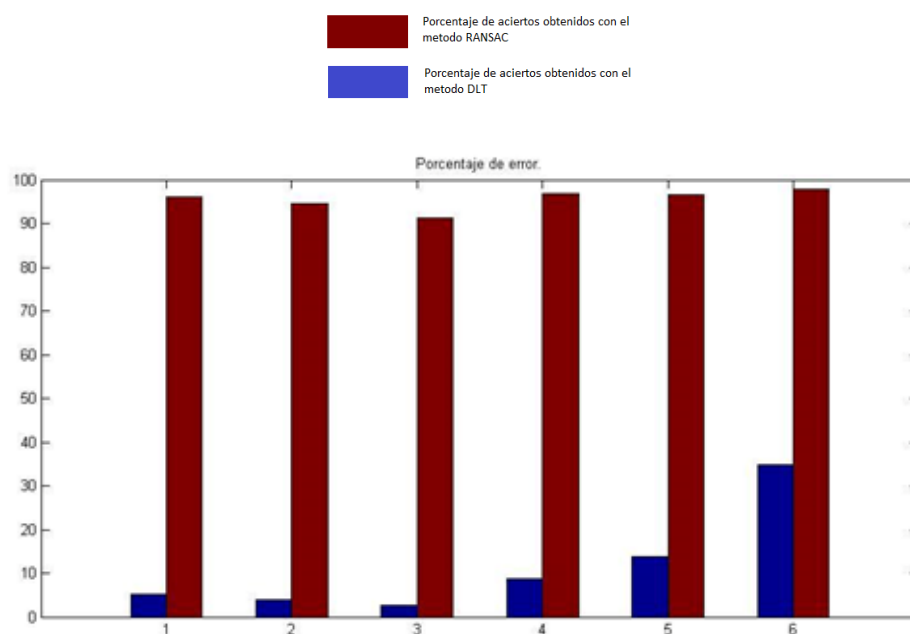


Figura 14. Diagrama de barras que contiene el desempeño para el cálculo de la matriz de homografía de dos métodos.

El diagrama de barras de la Figura 14, demuestra la eficiencia del método RANSAC frente al método de transformación lineal directa, las barras en color azul corresponden al vector de error generado por la matriz de homografía encontrada con el primer método cuya media es de

11.5119 %, mientras que las barras en color rojo son equivalentes al error generado con el segundo método y la media es de 95.5325 %.

Para ilustrar los resultados de la tabla anterior se tiene un ejemplo particular, en la Figura 15a se tiene una imagen a la cual no se le ha aplicado ninguna homografía.

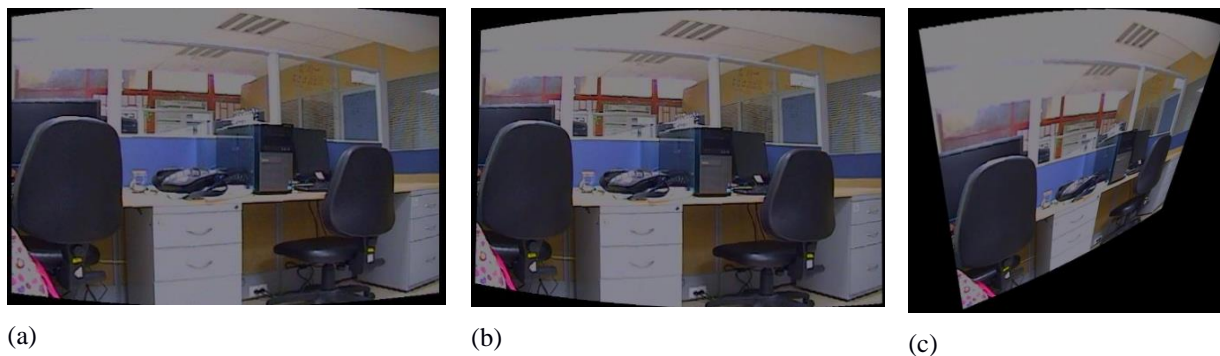


Figura 15. (a) Imagen de entrada en coordenadas cilíndricas. (b) Imagen de muestra transformada con la homografía encontrada con el método RANSAC. (c) Imagen de entrada transformada con la matriz de homografía encontrada con el método DLT.

Ahora transformamos esta imagen utilizando la homografía encontrada con el método RANSAC obteniendo la imagen de la Figura 15b.

Finalmente se aplica a la imagen de la Figura 15a la homografía encontrada con el método DLT esto se representa en la Figura 15c. Al comparar las Figuras 15b y 15c se puede ver que el método DLT es erróneo frente al método RANSAC ya que dicha transformación no corresponde a una proyección y al comparar los puntos coincidentes de la imagen de entrada con una segunda imagen, estos también nos conducen a un error, a diferencia del método RANSAC.

En la Figura 16 ilustran los resultados obtenidos en la Figura 14, en donde se encuentran un par de imágenes y 5 puntos coincidentes entre ellas, y en la Figura 17, se puede ver como estos

quedan ubicados en la misma posición después de aplicar la homografía a las imágenes de entrada.



Figura 16. Puntos coincidentes entre un par de imágenes pertenecientes a la misma escena.



Figura 17. Par de imágenes de la Figura 16, transformadas con la homografía calculada con el método RANSAC y superpuestas para evidenciar la coincidencia entre los 5 puntos de muestras

### 4.3. Panorámicas de 2 imágenes

#### Experimento con imágenes tomadas por una cámara digital

Aplicando los métodos anteriormente descritos, llamados en una sola función, con dos imágenes de entrada.

Inicialmente se realiza la calibración de la cámara con un algoritmo que se encuentra dentro de las librerías OpenCV de Python llamado ‘*calibrate*’ de esta manera se obtienen los parámetros de la misma, la siguiente matriz contiene los parámetros intrínsecos de la cámara de un teléfono celular, con el cual se tomaron las imágenes

$$\mathbf{K} = \begin{pmatrix} 782.1864 & 0 & 338.0369 \\ 0 & 669.8502 & 229.7495 \\ 0 & 0 & 1 \end{pmatrix} \quad (64)$$

Los parámetros de la diagonal de la matriz  $\mathbf{K}$  representan la distancia focal, en este caso es alrededor de 782 mm.

Se transforma las imágenes a coordenadas cilíndricas (ver Anexo C) mediante la función `deformacion_inversa (image, f)`, en donde el primer término corresponde a las imágenes de entrada, y el segundo es la distancia focal, anteriormente encontrada. Una vez se tienen las imágenes en coordenadas cilíndricas, se procede a cambiar el formato de la imagen RGB a escala de grises con la función `rgb2gray(A1)`, esto para poder encontrar las coincidencias.

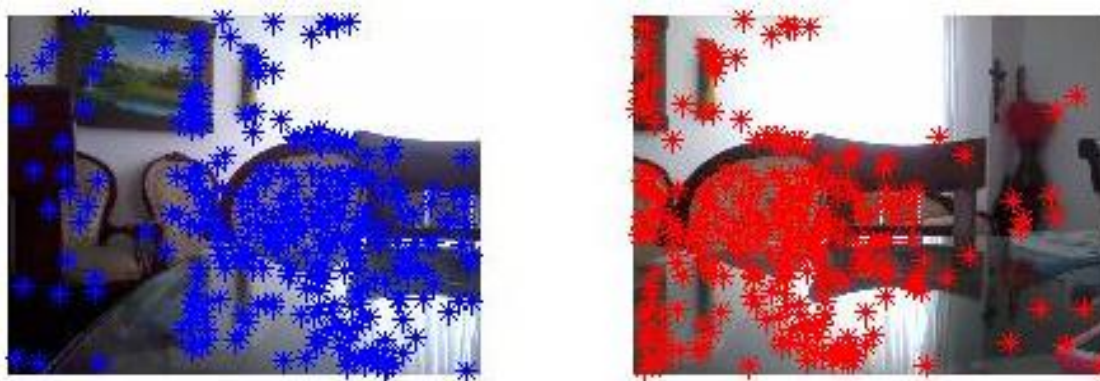


Figura 18. Extracción de características de un par de imágenes tomadas por un teléfono celular

Utilizando los vectores descriptores de las imágenes, a partir del método SIFT, se comparan estos descriptores para buscar correspondencias mediante el algoritmo Match. Realizando una serie de operaciones entre cada descriptor de la primera imagen con cada uno de la segunda, se decide cuáles son correspondientes con mayor probabilidad, eligiendo el que tenga una diferencia euclídea total menor.

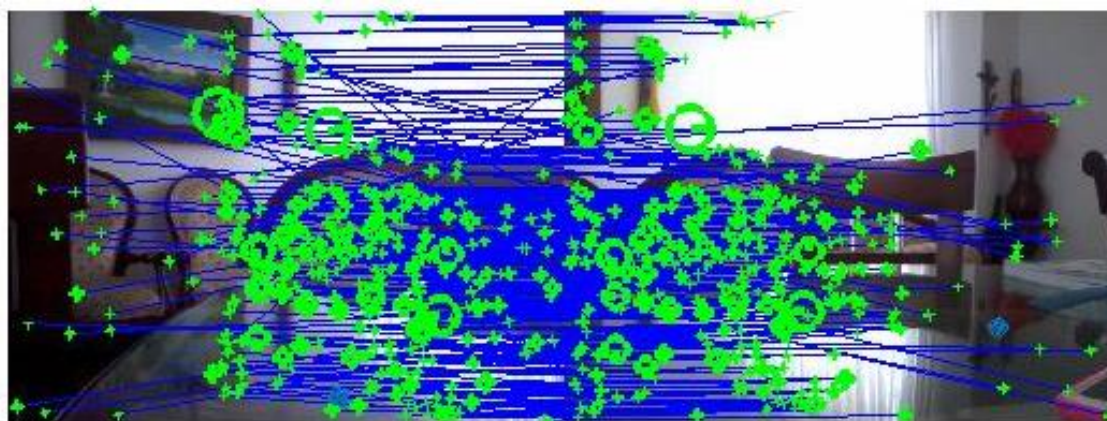


Figura 19. Correspondencias entre las imágenes de la Figura 18

Con las coincidencias anteriormente encontradas, se realiza el cálculo de la homografía con la función  $RANSAC(k1, k2, Matches)$  y con esta se aplica una transformación proyectiva a la segunda imagen, luego se unen las imágenes a partir de la técnica *Stitching*, y finalmente se usa el método *Blending*, estos últimos calculan de forma exacta la distancia con la cual se deben combinar dichas imágenes de forma correcta. Con lo anterior se obtiene la imagen panorámica de la Figura 20.



Figura 20. Imagen panorámica obtenida con dos imágenes tomadas a una misma escena, tomadas por otro tipo de cámara

Como se estima en la Figura 20, la imagen resultante contiene pixeles negros en su contorno, esto a causa del proceso; por tanto, se hace necesario implementar una función, que defina los límites con los cuales se imprime la panorámica de tal manera que se obtenga solo la información concerniente a la escena.

### Experimento con imágenes tomadas por la cámara PTZ

Se realiza el mismo proceso que en el experimento anterior, ahora para las imágenes de la misma escena de la Figura 16; la calibración de la cámara rotatoria arroja la siguiente matriz de parámetros intrínsecos:

$$\mathbf{K} = \begin{pmatrix} 777.9459 & 0 & 375.2552 \\ 0 & 778.8819 & 309.2947 \\ 0 & 0 & 1 \end{pmatrix} \quad (65)$$

De la matriz anterior la distancia focal, es aproximadamente 778 mm, teniendo este parametro se deforman las imágenes en coordenadas cilíndricas, es proceso se puede visualizar en la Figura 15a. Con el cálculo de la homografía, se aplica una transformación proyectiva a la segunda imagen. Luego se une la primera imagen con la segunda transformada (Figura 15b), después de aplicar las funciones *Blending* y *Stitching*, se obtiene la siguiente imagen panorámica:



Figura 21. Imagen panorámica obtenida con dos imágenes tomadas a una misma escena, tomadas con la cámara PTZ

Para realizar los experimentos anteriores con más imágenes, se editó el código implementado, de tal modo que se ingresen  $N$  imágenes, y se obtenga una panorámica a partir de las mismas.

#### 4.4. Panorámicas con $n$ imágenes

Para realizar este experimento se tomó como base el experimento anterior de unión de dos imágenes. A diferencia de la sección anterior, los mosaicos obtenidos a partir de 2 imágenes serán tomados en cuenta en las siguientes iteraciones hasta obtener el resultado final. Para ilustrar el procedimiento se tomarán 3 imágenes de entrada (Figuras 22a, 22b y 22c).

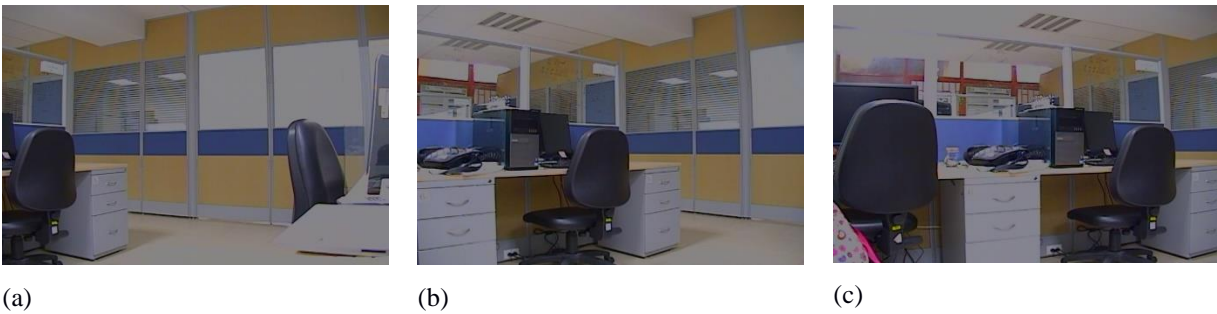


Figura 22. (a) Primera imagen de entrada. (b) Segunda imagen de entrada. (c) Tercera imagen de entrada

Con las dos primeras imágenes se obtiene un primer mosaico (Figura 23), luego, se obtiene un segundo mosaico utilizando la segunda y tercera imagen (Figura 24).



Figura 23. Mosaico obtenido con la primera y segunda imagen



Figura 24. Mosaico obtenido a partir de la segunda y tercera imagen

Cada mosaico contiene dos imágenes; la primera imagen corresponde a la segunda imagen con la que se obtuvo el mosaico anterior y la segunda imagen es la nueva imagen de entrada, por ejemplo, en el segundo mosaico la primera imagen corresponde a la imagen 2 del primer mosaico, la cual corresponde a la segunda imagen de entrada (Figura 22b); mientras que la segunda imagen del segundo mosaico es la nueva imagen de entrada que en este caso es la imagen 3 (Figura 22c). De esta forma en mosaico1 está compuesto por las imágenes 1 y 2, y en mosaico2 están presentes las imágenes 2 y 3.

El objetivo de realizar estas uniones es aumentar el área de puntos coincidentes entre los mosaicos obtenidos, para minimizar el porcentaje de error que se presenta en a la hora de calcular la homografía. Se repite el mismo procedimiento, de unir 2 imágenes, pero esta vez las entradas son los dos mosaicos obtenidos en el paso anterior. De esta forma se obtiene una imagen panorámica.

La siguiente figura es la unión de los mosaicos anteriormente encontrados:

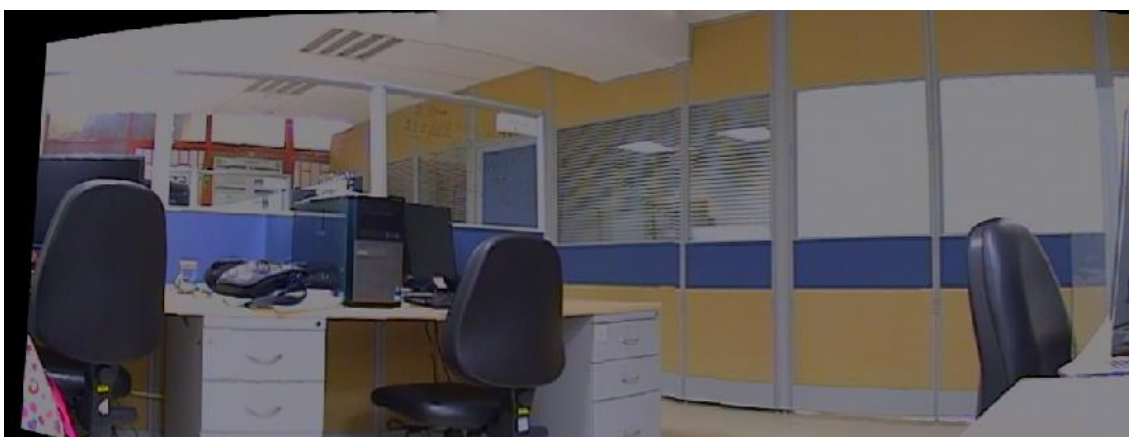


Figura 25. Imagen final a partir de 3 imágenes de entrada.

Como se pudo observar en el ejemplo anterior se necesitan 2 iteraciones para obtener un mosaico a partir de 3 imágenes, con este procedimiento, en la primera iteración se obtienen 2 mosaicos de 2 imágenes cada uno y en la segunda iteración se obtiene 1 mosaico de 3 imágenes.

Al realizar varias pruebas con un número mayor de imágenes se concluye que para un número  $N$  de imágenes se necesita un número de  $n-1$  iteraciones y en cada una de estas el número de imágenes de entrada se reduce en 1 y el número de imágenes que componen cada mosaico se aumenta en 1 y al finalizar el proceso se obtiene una panorámica con un número  $N$  de imágenes.

Después de terminar el proceso y obtener una panorámica de  $n$  imágenes se puede observar que los bordes de esta nueva imagen presentan un tamaño menor al resto de la imagen y que al hacer un zoom para observar los bordes de la imagen es posible que estos no se muestren correctamente, debido a que esta parte de la imagen ha sido transformada en las diferentes iteraciones al punto de llegar a deformarse. Esto se puede apreciar en la figura 26 donde tenemos una imagen panorámica de 360 grados creada a partir de 18 imágenes de entrada, que fueron tomadas teniendo en cuenta el barrido horizontal de la cámara, que según la hoja de datos de la cámara es de 340 grados y el Angulo de visión de la cámara que es de 85 grados.



Figura 26. Imagen panorámica de 360 grados a partir de 18 imágenes de entrada.

## 5. Conclusiones

En la práctica, los emparejamientos obtenidos por transformación lineal directa (DLT), método que toma solo cuatro puntos de forma aleatoria, suelen incluir emparejamientos espurios. Para evitar que estos emparejamientos afecten al cálculo de los parámetros geométricos se utilizan procedimientos robustos de regresión. Los métodos robustos de regresión tratan de explotar la redundancia de emparejamientos para obtener una solución consistente con la mayoría de ellos (por ejemplo: el error de proyección en ambas imágenes debe estar por debajo de un determinado umbral  $t$ ). Como se puede evidenciar en el diagrama de la Figura 14, se optó por emplear el método RANSAC, debido a que de forma iterativa toma puntos de forma aleatoria y descarta aquellos que se encuentren fuera de cierto umbral. Haciendo de este un método muy exacto, con un error medio aproximado del 11 %, obtenido en el experimento realizado con seis pares de imágenes; dicho error representa la distancia euclidiana de los puntos en la *imagen 1* transformada con respecto a sus correspondientes en la *imagen 2*.

Para el cálculo de la matriz proyectiva se tomó el algoritmo RANSAC, debido a que arrojó un porcentaje de acierto elevado, gracias a que utiliza todas las coincidencias entre dos imágenes.

Al transformar las imágenes a un mismo sistema 2D, se debe tener en cuenta los valores extremos de los ejes coordenados,  $x$  y  $y$ , para poder realizar la unión entre estas de forma correcta.

Para identificar la calidad visual de la imagen panorámica obtenida, no existe un criterio u objetivo que lo permita, puesto que el concepto de “calidad” implica diferentes aspectos cognitivos. Sin embargo, lo que sí es cierto es que dos imágenes adquiridas con la misma distancia focal, formaran una panorámica con la misma secuencia, que aquella que haya sido adquiridas a diferentes distancias, debido a que se presentará un nivel de detalle más elevado que la que haya sido adquirida desde mas lejos. Con lo anterior, la distancia a la escena se utiliza como medida para determinar la calidad de las imágenes adquiridas.

Para el cálculo de correspondencias se realiza una serie de operaciones entre cada descriptor de la primera imagen con cada uno de la segunda, en donde se decide quienes son correspondientes con mayor probabilidad, eligiendo aquel con una diferencia euclídea total menor. El coste computacional de las comparaciones será muy elevado si no se tiene en cuenta ciertas situaciones, por ejemplo, si se nota a simple vista que ciertos puntos de la primera imagen no aparecen en determinada zona de la segunda imagen, podríamos descartar los descriptores de esa zona y así ahorrar el coste de las comparaciones. Otra restricción a tener en cuenta es el tamaño de las imágenes, lo cual equivale a una estimación extensa de correspondencias. Por tanto, en función del tipo de aplicación que se quiera implementar y de la información que se tenga a priori, se podrá optimizar el código de una forma u otra.

## 6. Recomendaciones

El proyecto desarrollado a lo largo de este documento, representa una base sólida para el desarrollo de un algoritmo que, a partir de las imágenes panorámicas encontradas, realice una reconstrucción 3D de la escena.

Otro aspecto que se podría mejorar, es la obtención de un algoritmo que permita corregir de forma automática la distorsión que se genera en las imágenes en forma de barril, a causa del tipo de cámara (rotatoria).

A pesar de obtener buenos resultados al aplicar el método *Blending piramidal* la imagen final podía llegar a presentar errores, en la zona donde se produce el cambio de una imagen a otra. Para minimizar estos errores se recomienda trabajar la imagen final aplicando métodos de corrección como por ejemplo el algoritmo de Dijkstra, el cual encuentra el camino más corto entre 2 vértices y podría ser utilizado para corregir la superposición de dos puntos.

## Referencias Bibliográficas

- A. Zisserman, R. a. (2004). *Projective Geometry and Transformations of 2D, Multiple View Geometry in Computer Vision*. New York: Cambridge university press.
- Basagoitia, E. P. (2004). *Multiple camera calibration using point correspondences oriented to intelligent spaces*.
- blamethemonkey. (2014). Retrieved from blamethemonkey: <http://blamethemonkey.com/hdr-photography-panorama-tutorial>
- C. D. Kuglin, D. C. (1975). The phase correlation image alignment method. *Proceedings of IEEE International Conference Cybernet Society*, (pp. 163-165). New York.
- C. M. Huang, S. W. (1998). Efficient Image Stitching of Continuous Image Sequence With Image and Seam Selections. *IEEE*.
- Chavarría, J. F. (2013). *Diseño, implementación y prueba de un algoritmo de estimación de los 8 parámetros de la transformación proyectiva (homografía) entre dos imágenes a partir del rastreo de esquinas*. Ciudad Universitaria Rodrigo Facio: Universidad de Costa Rica.
- Dubrofsky, E. (2009). *Homography Estimation*. Vancouver.
- F., J. D., Vargas Jiménez, D. S., & Otero Foliaco, J. E. (2015, marzo). *Desarrollo e Implementación de Recorridos 360° en portales*. pereira: Scientia et Technica. Retrieved from Dialnet: [file:///C:/Users/Ghissela/Downloads/Dialnet-DesarrolloEImplementacionDeRecorridos360EnPortales-5026900%20\(5\).pdf](file:///C:/Users/Ghissela/Downloads/Dialnet-DesarrolloEImplementacionDeRecorridos360EnPortales-5026900%20(5).pdf)

Fischler, M., & Bolles, R. (1981). *Random sample*. New York: Communications of the ACM.

Fulkerson, A. V. (2008). *An Open and Portable Library of Computer Vision Algorithms*.

Retrieved from An Open and Portable Library of Computer Vision Algorithms:

[www.vlfeat.org](http://www.vlfeat.org)

G, L. D. (1999). *Object Recognition from Local Scale-Invariant Features*. British Columbia :

Association for Computing Machinery .

García, A. M. (2014). *Algoritmo de Calibración en MATLAB para Cámaras Digitales*. Quito.

González, J. I. (2003). *Estudio experimental de métodos de calibración y autocalibración de cámaras*.

H Bay, A. E. (2008). *Speeded Robust features*. New York: Springer Berlin Heidelberg.

H.Shum, M. R. (1998). Interactive Construction of 3D Models from Panoramic Mosaics. *IEEE Sensors Journal*.

Harley, R. &. (2003). *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge

University Press.

Hartley, R., & Zisserman, A. (2001). *Multiple View Geometry in Computer Vision*. New York:

Cambridge University.

Herbert Bay, A. E. (2008). Speeded Up Robust Features. In *Computer Vision and Image*

*Understanding* (pp. 346--359).

Hoff, W. A. (2011). *Computer Vision* . Golden, Colorado .

- Horlent, N., Lamaro, A., & Lanfri, s. (2014, julio 7). *sistemas satelitales y teledetección*. Retrieved from sistemas satelitales y teledetección: [http://epidemiologia-panoramica\\_MinSal\\_CONAE.pdf](http://epidemiologia-panoramica_MinSal_CONAE.pdf)
- Hua, Z., Li, Y., & Li, J. (2010). Image stitch algorithm based on SIFT and MVSC. *IEEE*.
- Huang , C. M., Lin, S. W., & Chen, J. H. (2015). *Efficient Image Stitching of ContinuousImage Sequence With Image and Seam Selections* (Vol. 15). China: IEEE Sensors Journal.
- ITREBA. (n.d.). Retrieved from ITREBA: [http://atcproyectos.ugr.es/itreba/index.php?option=com\\_content&view=article&id=50&Itemid=58](http://atcproyectos.ugr.es/itreba/index.php?option=com_content&view=article&id=50&Itemid=58)
- Jacobus, R., & Hollander, M. D. (2007). *Random Sampling Methods for*. Holanda: Universidad Técnica de Delft.
- Levin, A. Z. (2003). *Seamless image stitching in the gradient domain*. Jerusalem, Israel.
- Loop, C. &. (1999). *Computing Rectifying Homographies for Stereo Vision* . Colorado, USA: Fort Collins.
- Lowe, D. G. (2004). *Distinctive image features from scale-invariant keypoints*. Canada.
- Marius Muja, D. G. (2009). *FAST APPROXIMATE NEAREST NEIGHBORS*. Vancouver Canada: VISAPP International Conference on Computer Vision Theory and Applications.
- Marsden, J. E., & Tromba, A. J. (2004). *Cálculo vectorial*. Madrid: Pearson Educación .
- Mordvintsev, A., & Abid, K. (2013). *OpenCV-Python tutorials*. Retrieved from OpenCV-Python tutorials: <http://opencv-python->

tutroals.readthedocs.io/en/latest/py\_tutorials/py\_calib3d/py\_calibration/py\_calibration.html

Pé, A. S. (2007). *Geintra*. Retrieved from <http://www.geintra-uah.org/system/files/private/Proyecto.pdf>

R., S. (2010). *image stitching, Computer Vision: Algorithms and Applications*. Washington: Springer. Retrieved from <http://szeliski.org/Book/>

Rejas Ayuga, J. G. (2008). *Tratamiento Digital Previo De Las Imágenes*. Retrieved from *Tratamiento Digital Previo De Las Imágenes:*  
[http://api.eoi.es/api\\_v1\\_dev.php/fedora/asset/eoi:45518/componente45516.pdf](http://api.eoi.es/api_v1_dev.php/fedora/asset/eoi:45518/componente45516.pdf)

Rejas Ayuga, J. G. (2008). *Tratamiento Digital Previo De Las Imágenes*. Retrieved from *Tratamiento Digital Previo De Las Imágenes:*  
[http://api.eoi.es/api\\_v1\\_dev.php/fedora/asset/eoi:45518/componente45516.pdf](http://api.eoi.es/api_v1_dev.php/fedora/asset/eoi:45518/componente45516.pdf)

Rubin, D. B. (1973). *Matching to remove bias in observational studies. Biometrics*. Massachusetts: International Biometric Society.

Shum, R. S.-Y. (1997). *Creating full view panoramic image mosaics and texture-mapped models*. New York: ACM Press/Addison-Wesley Publishing Co.

Stephens, H. a. (1980). A combined corner and edge detector . *Proceedings of the 4th Alvey Vision Conference*, (pp. 147–151).

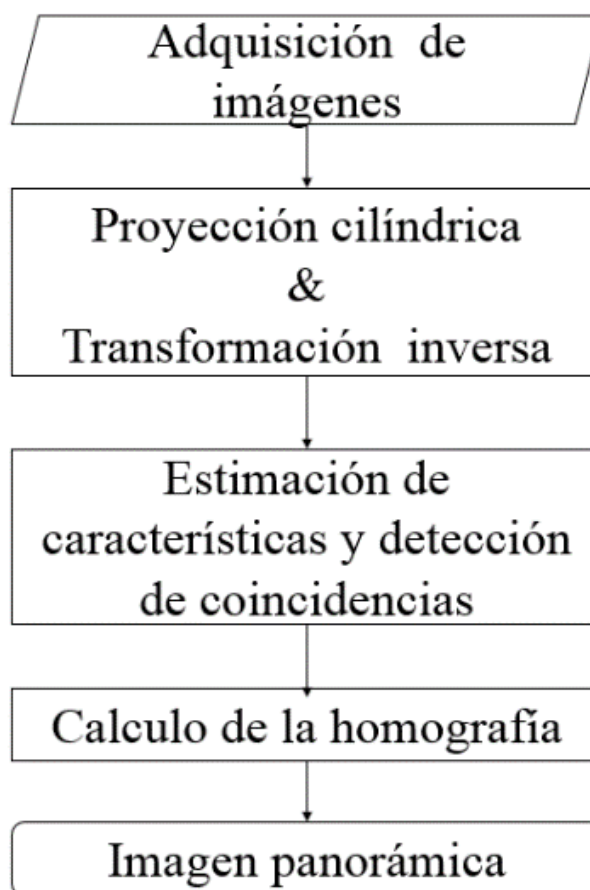
Szeliski, R. (2004). *Image alignment and stitching: A tutorial*. Washington: Technical Report MSRTR-2004-92,.

- Szeliski, R. (2010). *image stitching, Computer Vision: Algorithms and Applications*. Washington: Springer.
- Tang, C. K., & Jia, J. (2008). Image Stitching Using Structure Deformation. *IEEE transactions on pattern analysis and machine intelligence*.
- Usman . (2012, 08 22). *stackoverflow*. Retrieved from stackoverflow:  
<http://stackoverflow.com/questions/12017790/warp-image-to-appear-in-cylindrical-projection/12029538#12029538>
- Verma, C. S. (n.d.). *Panoramic Image Mosaic*. Retrieved from Panoramic Image Mosaic:  
[http://pages.cs.wisc.edu/~csverma/CS766\\_09/ImageMosaic/imagemosaic.html](http://pages.cs.wisc.edu/~csverma/CS766_09/ImageMosaic/imagemosaic.html)
- Wang, S. X. (2011). Generation of Panoramic View from 360 ° Fisheye Images Based on Angular Fisheye Projection. *International Symposium . Hebei Province,China: Computing and Applications to Business, Engineering and Science*.
- Wang, Y., & Wang, M. (2010). Research on Stitching Technique of Médical Infrared Images. *IEEE*.
- Yewei, H. Z. (2010). Image stitch algorithm based on SIFT and MVSC. *IEEE*.
- Yewei, H. Z. (2010). Image stitch algorithm based on SIFT and MVSC . *IEEE*.
- Zhang, Z. (1998, diciembre 2). A Flexible New Technique for Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,. Retrieved from  
Calibration.<http://research.microsoft.com/enus/um/people/zhang/Papers/TR98-71.pdf>

Zisserman, R. H. (2000). *Multiple View Geometry in Computer Vision*. New York: The Pitt Building, Trumpington Street, Cambridge, United Kingdom .

## Apéndices

Apéndice A. Diagrama del proceso que genera de imágenes panorámicas



## Apéndice B. Parámetros característicos de la cámara PTZ SNC-EP520

Cámara	
Sensor de imagen	Sensor CCD EXview HAD de 1/4"
Número de píxeles efectivos	Aprox. 0,38 megapíxeles
Sistema de sincronización	Sincronización interna
Iluminación mínima (50 IRE)	Color: 1,4 lx (F1.6, AGC encendido, 1/60 s, 50 IRE)
	Blanco y negro: 0,15 lx (F1.6, AGC encendido, 1/60 s, 50 IRE, Modo nocturno)
Iluminación mínima (30 IRE)	Color: 0,9 lx (F1.6, AGC encendido, 1/60 s, 30 IRE)
	Blanco y negro: 0,1 lx (F1.6, AGC encendido, 1/60 s, 30 IRE, Modo nocturno)
Rango dinámico	92 dB teóricos con tecnología DynaView
Relación señal/ruido (ganancia de 0 dB)	50 dB (Ganancia 0 dB)
Ganancia	Automático/Manual (de -3 dB a +28 dB)
Velocidad del obturador	De 1/1 s a 1/10.000 s
Control de exposición	Totalmente automática, prioridad de obturación, prioridad de iris, manual
Balace de blancos	Automático, ATW, interior, exterior, de un toque, manual, lámpara de vapor de sodio
Lente	Lente con zoom de enfoque automático
Zoom electrónico	Si
Relación de zoom	Zoom óptico 36x
	Zoom digital 12x
	Zoom total 432x
Enfoque electrónico	Si
Enfoque automático	Si
Ángulo de visión horizontal	De 57,8° a 1,7°
Ángulo de visión vertical	De 43,4° a 1,3°
Distancia focal	f = de 3,4 mm a 122,4 mm
Número F	F1.6 (gran angular) a F4.5 (teleobjetivo)
Distancia mínima del objeto	De 10 mm (gran angular) a 1500 mm (teleobjetivo)
Ángulo de giro/tilt (electrónico)	Rotación sin fin de 360°/210°
Velocidad de giro	300°/s (máx.)
Velocidad de inclinación	300°/s (máx.)
Número de posiciones de preajuste	256

Apéndice C. Deformación de imágenes a coordenadas cilíndricas

Considerando un cilindro de radio unidad centrado en  $O(0,0,0)$ , en donde cada punto de este cilindro se puede parametrizar por medio de un ángulo  $\theta$  y una altura  $h$ .  $\theta$  es el ángulo que la proyección que el vector hace con el eje  $x$ , y  $h$  es la componente del vector en la dirección del eje  $y$ , tal como se observa en la Figura 30.

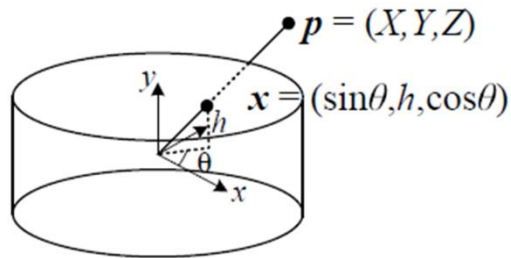


Figura 27. Cilindro de radio unitario (Szeliski R. , 2010)

Suponiendo un punto en el espacio  $P(x, y, z)$ , este punto se desea proyectar sobre la superficie cilíndrica, con las coordenadas cilíndricas 3D correspondientes a  $(\theta, h)$  dadas por:

$$(\sin \theta, h, \cos \theta) \sim (x, y, f) \quad (66)$$

a partir de la anterior correspondencia, se pueden calcular las coordenadas deformadas:

$$x' = s\theta = s \tan^{-1} \frac{x}{f} \quad (67)$$

$$y' = sh = s \frac{y}{\sqrt{x^2 + f^2}} \quad (68)$$

la variable  $s$  es un factor de escala arbitrario (llamado radio del cilindro), que se puede establecer como  $s = f$ , para minimizar la distorsión cerca del centro de la imagen.

Deformaciones de una imagen directamente podría crear agujeros en la imagen deformada, por lo que se prefiere la deformación inversa (Szeliski R. , 2010). Las ecuaciones para la deformación inversa son las siguientes:

$$x = f \tan \theta = f \tan \frac{x}{s} \quad (69)$$

$$y = h \sqrt{x^2 + f^2} = \frac{y'}{s} f \sqrt{1 + \tan^2 \left( \frac{x'}{s} \right)^2} \quad (70)$$

En el siguiente algoritmo se propone una manera de generar de forma automática la deformación de imágenes en coordenadas cilíndricas:

Algoritmo 6. Algoritmo para la deformación de imágenes en coordenadas cilíndricas

---

```

1  Leer las imágenes de entrada
   Definir el punto central de la imagen
   Crear una matriz de ceros del mismo tamaño que las imágenes de entrada
2  For filas= 1 hasta llegar al tamaño de las filas. do
3      For columna=1 hasta el tamaño de las columnas. do
4          Centrar las coordenadas en el plano de la imagen
           Proyectar las coordenadas en un cilindro
           Calcular las coordenadas cilíndricas dentro de la imagen
5      end
6  end
7  Definir la imagen de salida
8  end

```

---

A continuación, se presenta el diagrama de flujo del algoritmo anterior

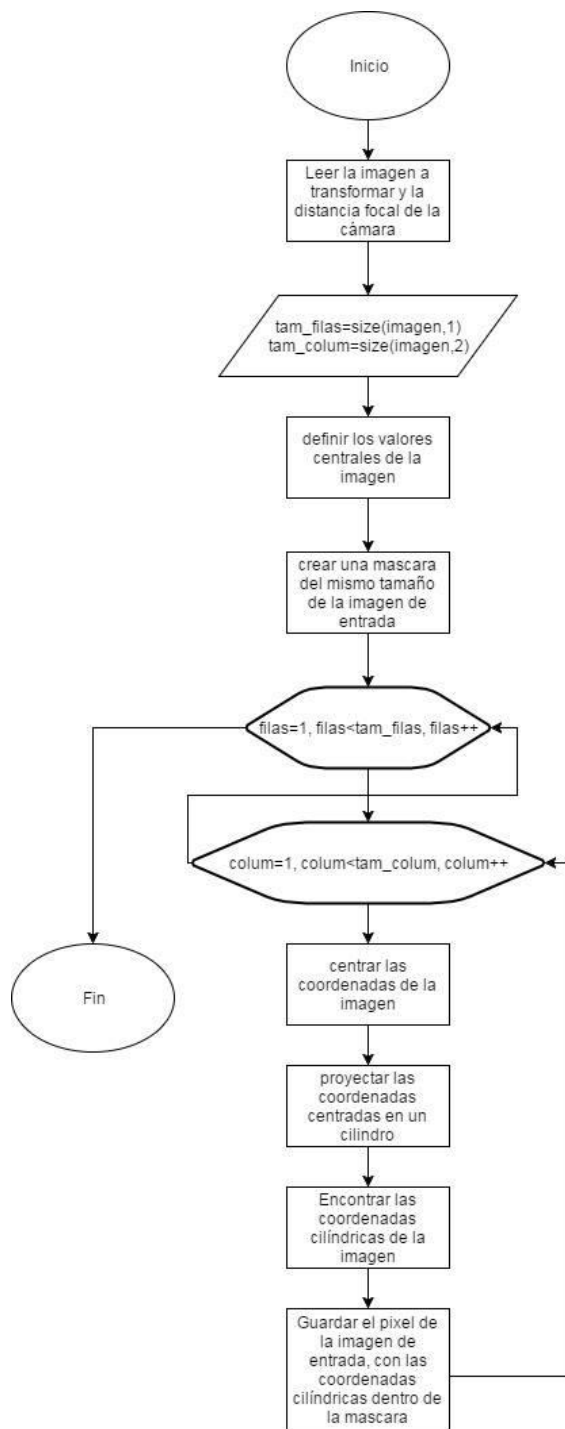


Figura 28. Diagrama de flujo del algoritmo de deformación de imágenes en coordenadas cilíndricas