

Estimación de orientación de superficies planares en escenas urbanas para la generación de panoramas.

Tesis de Licenciatura

Mario Gerardo Canul Ku



Licenciatura en Ciencias de la Computación

Facultad de Matemáticas

Universidad Autónoma de Yucatán

Mérida, Yucatán
México, 2010

Resumen

Esta tesis presenta una metodología para obtener información geométrica de superficies planares presentes en una escena mediante el análisis de múltiples imágenes. La orientación relativa de las superficies planares es un elemento importante de la representación tridimensional de la escena, obtenerla es el tema central de este trabajo de tesis. La solución propuesta consiste en el desarrollo de un sistema de software interactivo que realiza el análisis de múltiples imágenes digitales, permitiendo al usuario delimitar superficies planares del resto de la escena por medio de polígonos convexos. Podemos resumir las fases del proceso de obtención de orientaciones en cuatro etapas: relacionar las imágenes, delimitar las superficies en las imágenes de la escena, calcular homografías y estimar la orientación a partir de la descomposición de las homografías. Para probar la funcionalidad de la solución propuesta se realizaron diferentes experimentos, en los cuales se emplearon múltiples imágenes de escenas con condiciones controladas y escenas urbanas. Al final de este trabajo se discuten los resultados obtenidos de los experimentos, concluyendo que la descomposición de la homografía proporciona una buena aproximación de la orientación de la superficie planar.

Agradecimientos

El tiempo transitorio que cada persona tiene desde que nace hasta que muere le llamamos vida, en dicho tiempo podemos realizar muchas actividades, tener grandes logros, riquezas, pobreza, etc. En mi caso uno de estos logros que me ha tocado vivir, es el desarrollo de esta tesis con la cual obtengo el grado de Licenciado en Ciencias de la Computación.

Gracias al gran apoyo que he tenido de todas las personas que me han acompañado a largo de mi vida, he podido alcanzar un logro más, es por ello que expreso mis mayores agradecimientos a mis padres, que son los que me han acompañado desde mi nacimiento y me han crecido con muchos sacrificios, así que mi logro es también un logro de ellos; agradezco a mi hermana que me dio el ejemplo de seguir un ámbito profesional, a mi hermano que me dio el ejemplo de tener un gusto por el trabajo y que en cada uno de ellos se aprende algo nuevo. A mis amigos que me brindaron siempre alegría, consejos, apoyo y ánimos en momentos difíciles, y en especial a un gran profesional que depositó toda su confianza y que guió de una forma excepcional el desarrollo de esta tesis, me refiero a mi asesor el Dr. Arturo Espinosa Romero, pues siempre estuvo orientándome y metiendo nuevas ideas durante el desarrollo de la tesis, y como olvidar al M.C. Carlos Benito Mojica Ruiz que fungió como mi otro asesor, pues siempre dio aspectos importantes que me permitieron mejorar el sistema que diseñé como parte del trabajo de tesis.

Para terminar quiero agradecer a mis compañeros de estudio con los cuales trabajé de una forma muy amena y me brindaron su amistad, por lo cual les tengo un gran respeto y admiración. A los maestros que siempre entregaron su esfuerzo y dedicación en las aulas para que yo tuviese la formación necesaria para desarrollarme como un gran profesional; es por ello que agradezco a la L.M María Inés Panti Santiago que fue un parte aguas cuando cursaba el tercero de bachillerato, ya que empleó técnicas de enseñanza que me permitieron descubrir y disfrutar el fascinante mundo de las matemáticas, con las cuales se han desarrollado las diferentes tecnologías de la vida cotidiana.

Declaración

Esta tesis está dirigida a la Facultad de Matemáticas de la Universidad Autónoma de Yucatán como cumplimiento a uno de los requerimientos para obtener el grado de Licenciado en Ciencias de la Computación. Por este medio declaro que esta tesis fue realizada enteramente por mí y describe mi propio trabajo de investigación con excepción de las partes que así se indiquen.

Mario Gerardo Canul Ku
Mérida, Yucatán
México
11 de enero de 2011

Índice general

Resumen	II
Agradecimientos	III
Declaración	IV
Lista de Figuras	IX
Lista de Cuadros	X
1. Introducción	1
1.1. Objetivo de la tesis	5
Objetivos Particulares	5
1.2. Descripción del documento	5
2. Marco Teórico	7
2.1. Introducción	7
2.2. Características de las imágenes	8
Representación	8
Formación de la imagen	10
2.3. Relacionando las imágenes	12
2.4. Polígono Convexo	14
Algoritmo de cerco convexo	17
2.5. Homografía	18
Estimación de la homografía	20

Estimación robusta	22
Estimación optimizada	23
2.6. Estimación de orientación de una superficie planar en la escena . .	26
Estimación de la orientación de manera ideal	27
Estimación de la orientación usando calibración	30
2.7. Resumen	32
3. Metodología	33
3.1. Introducción	33
3.2. Interfaz gráfica de usuario	35
3.3. Estimación de orientaciones desde la GUI	39
Selección de imágenes de prueba	40
Obtención de Esquinas	42
Búsqueda de correspondencias	43
Delimitación de las superficies	45
Obtención de orientaciones	47
3.4. Resumen	49
4. Experimentos	50
4.1. Introducción	50
4.2. Escena con condiciones controladas	51
Discusión	55
4.3. Escena urbana (Centro Histórico)	55
Discusión	58
5. Conclusiones	59
5.1. Sumario	59
5.2. Trabajo futuro	61
Referencias	62

A. Implementación del sistema	65
A.1. Cola bidireccional	65
A.2. Modelo Vista Controlador	66
A.3. Código de coplanaridad	67

Índice de figuras

1.1. Planos en escenas urbanas	1
1.2. Panoramas	2
1.3. Ambientes virtuales	3
1.4. Homografía	4
2.1. Representación de una imagen como gráfica bidimensional	9
2.2. Representación icónica de una imagen	10
2.3. Proyección de un punto al plano Imagen	11
2.4. Intersección de polígonos	15
2.5. Polígono Convexo	16
2.6. Área de un triangulo	17
2.7. Error simétrico	23
2.8. Plano epipolar	26
3.1. Ventana principal del sistema	35
3.2. Menú principal del sistema	36
3.3. Menú para seleccionar secuencia de imágenes	36
3.4. Sección de objetos geométricos a dibujar	37
3.5. Botón para dibujo de puntos	37
3.6. Botón para dibujo de líneas	37
3.7. Botón para dibujo de trayectorias	38
3.8. Botón para dibujo de polígonos convexos	38
3.9. Botón para ir a imagen previa	38

3.10. Botón para ir a imagen siguiente	38
3.11. Botón para calcular de orientación de superficie	39
3.12. Botón para calcular esquinas	39
3.13. Botón para búsqueda de correspondencias	39
3.14. Botón para aplicar algoritmo del cerco convexo	39
3.15. Botón de desplazamiento	40
3.16. Secuencia de imágenes	40
3.17. Ventana con imágenes visualizadas	41
3.18. Esquinas de la imagen	42
3.19. Correspondencias entre imágenes	44
3.20. Botón para dibujo de puntos desde la GUI	44
3.21. Región de una superficie planar	45
3.22. Acotación de la superficie planar	46
3.23. Botón para aplicar algoritmo del cerco convexo	47
3.24. Puntos típicos y atípicos visualizados	47
3.25. Diagrama de flujo	49
4.1. Escena con condiciones controladas	52
4.2. Orientaciones relativas de la superficie planar a diferentes perspectivas	54
4.3. Escena urbana experimental	56
4.4. Orientación de la superficie planar, experimento dos	58

Índice de cuadros

2.1. Representación de una imagen como matriz con entradas enteras	9
2.2. Algoritmo cerco convexo	19
2.3. Algoritmo DLT Normalizado	22
2.4. Algoritmo de RANSAC y <i>Levenberg – Marquardt</i> para estimar homografías	25
2.5. Tabla de Soluciones de la descomposición homográfica	29
2.6. Algoritmo de estimación de orientación	31
4.1. Cuadro de porcentajes del primer experimento	53
4.2. Cuadro de porcentajes del segundo experimento	57

Capítulo 1

Introducción

El desarrollo de nuevas tecnologías tanto en el área de cómputo como en el de la electrónica, ha permitido que en los últimos años los investigadores diseñen sistemas capaces de procesar y analizar imágenes capturadas con cámaras digitales y relacionar éstas con información posicional, lo que provee a investigadores y desarrolladores de una invaluable fuente de información para diferentes tareas que involucran el modelado tridimensional del entorno. Esto es de utilidad en diversas aplicaciones como la robótica móvil, la generación de ambientes virtuales, entre otras.



Figura 1.1: Los ambientes urbanos consisten en gran parte de superficies planas, como son las paredes de los edificios, ventanas, calles, etc.

Al estar en un ambiente urbano nos encontramos con paredes de edificios, banquetas, señalamientos, etc., los cuales pueden ser considerados como superficies planas, en la figura 1.1 se muestra un ambiente urbano típico y en ella se puede

observar múltiples superficies planas que conforman la escena proyectada en la imagen, por ejemplo las paredes de los edificios.

Cuando una persona camina por las calles de una ciudad, el sistema visual humano percibe imágenes de las calles y edificios que lo rodean, a partir de la información de las imágenes el cerebro se encarga de obtener la orientación de cada superficie y de esa manera ubicar a la persona dentro de la escena.

Si en lugar de una persona colocáramos a un robot móvil dentro de un ambiente urbano, el robot requiere conocer las orientaciones de las superficies planares con respecto a sí mismo a fin de conocer su ubicación, por lo cual requiere percibir información de la escena para encontrar dichas orientaciones, dicha información se puede obtener al utilizar cámaras digitales como medio de percepción de imágenes.



(a) Panorama generado con dos imágenes



(b) Panorama generado con múltiples imágenes

Figura 1.2: Panoramas

Ahora consideremos el problema de la generación de panoramas, donde un panorama es una imagen de la proyección en una superficie (usualmente plana) de un conjunto de imágenes capturadas por una cámara que rota alrededor de su centro óptico, ver figura 1.2. Este tipo de representación contiene información de las texturas de la escena con respecto a un punto geográfico dado, simulando una cámara omnidireccional, sin embargo no es posible estimar profundidad en escenas a partir de únicamente rotaciones, véase en [Hartley y Zisserman \(2004\)](#), y por tanto no es evidente cómo relacionar entre sí la información de texturas contenida en el panorama con la estructura tridimensional de la escena que representa

el panorama.

El uso de panoramas es ampliamente utilizado para la generación de ambientes virtuales, por ejemplo, Google StreetView emplea los panoramas para generar un ambiente virtual de las calles fotografiadas, por medio de un arreglo de cámaras que simula a una cámara en rotación, ver figura 1.3; dichas cámaras son montadas sobre un vehículo que recorre las principales ciudades del mundo. Al movernos dentro del ambiente virtual al no conocer la estructura tridimensional de la escena, la orientación de las superficies planas nos provee información que permite identificar el piso de las paredes.

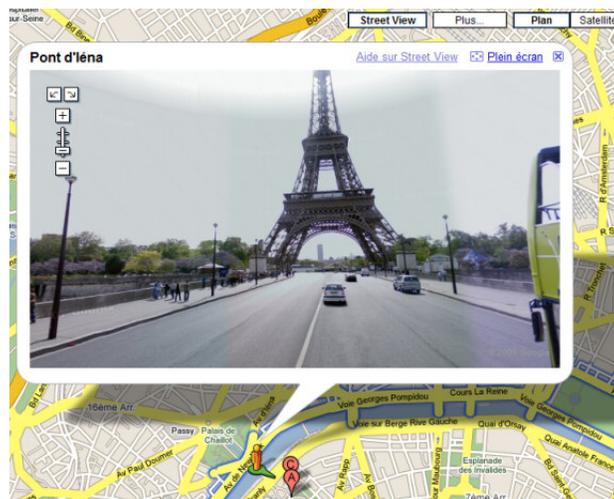


Figura 1.3: Google StreetView emplea los panoramas para generar ambientes virtuales de las calles de las principales ciudades del mundo.

Los ambientes urbanos consisten en gran parte de superficies planas y es posible, sin perder mucha información, simplificar la representación 3D como un conjunto de planos. La orientación relativa de las superficies planas presentes en una escena urbana, es un elemento importante de la representación tridimensional de la misma, por lo cual se decidió diseñar un sistema que permita estimar las orientaciones de las superficies a partir de múltiples imágenes, donde cada una contenga una proyección diferente de la escena urbana.

Es posible representar el cambio de la proyección en dos imágenes de una super-

ficie plana en la escena capturadas con la misma cámara, que difieran en su pose únicamente, esto es su rotación y translación, por medio de una transformación proyectiva, y con base en esto estimar la orientación relativa entre las diferentes superficies presentes en una escena.

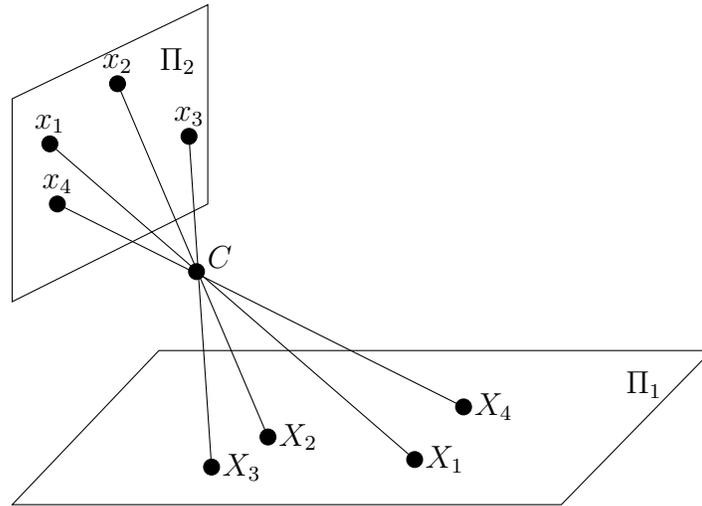


Figura 1.4: Los planos Π_1 y Π_2 se relacionan por medio de una transformación lineal denominada homografía.

La homografía es una transformación lineal que relaciona dos planos distintos, ver figura 1.4, con la cual se puede representar el cambio de la proyección entre dos imágenes. La homografía encapsula la orientación de una superficie plana de la escena, también nos permite generar panoramas, por lo cual el tener una estimación de la homografía juega un papel importante en este documento de tesis.

En el presente trabajo se presenta una solución que combina la percepción geométrica de una persona, con las capacidades de cómputo intensivo que proporciona una computadora. Esta solución consiste en el desarrollo de un sistema de software que realiza el análisis de múltiples imágenes digitales, permitiendo al usuario delimitar superficies planares del resto de la escena por medio de polígonos convexos, de forma que el sistema se encargue de estimar las homografías que encapsulan las orientaciones de las superficies. Un ejemplo de un sistema similar

en espíritu, y que sirvió de inspiración para el desarrollo de este documento es el sistema VideoTrace, véase en [Hengel et al. \(2007\)](#).

1.1. Objetivo de la tesis

El objetivo principal de la tesis es el desarrollar un sistema de software interactivo para la estimación de orientación de superficies presentes en una escena urbana a partir del análisis de múltiples imágenes.

Objetivos Particulares

1. Desarrollar un sistema de software interactivo para el análisis de conjuntos de imágenes.
 - 1.1. Diseñar una interfaz gráfica de usuario para la interacción del usuario con el sistema.
2. Implementar métodos para la estimación robusta de homografías.
3. Desarrollar una metodología para el desarrollo de diferentes experimentos que involucren escenas con superficies planares.
4. Elaborar experimentos que involucren superficies planares para la obtención de sus orientaciones, tanto en escenas urbanas como escenas con condiciones controladas.

1.2. Descripción del documento

Este documento esta conformado de la siguiente manera: en el capítulo 2 se da una explicación teórica del proceso de formación de una imagen, la manera en que relacionamos las imágenes, se explica el concepto de homografía y se detallan las ecuaciones empleadas para la estimación de orientación de superficies planares. El capítulo 3 presenta la metodología que se sigue para la obtención de las orientaciones de las superficies planares, haciendo uso del sistema desarrollado como parte del trabajo de tesis. El capítulo 4 presenta la realización de los

experimentos y muestra los resultados obtenidos mediante el uso de gráficas. En el último capítulo se presentan las conclusiones y discusiones derivadas de los resultados de los experimentos, realizando un análisis de las relaciones geométricas de la escena de prueba.

Capítulo 2

Marco Teórico

2.1. Introducción

La estimación de modelos tridimensionales de una escena es uno de los principales problemas de la visión computacional. En años recientes han habido importantes avances en el desarrollo de métodos geométricos que describen las relaciones entre varias imágenes de una escena estática capturadas por una o más cámaras, o bien una sola cámara en movimiento.

En la primera parte de este capítulo se explica la relación existente entre la escena y las imágenes mediante el modelo de cámara pinhole. Posteriormente se explica el registro entre las imágenes, el cual consiste en encontrar correspondencias entre dos o más imágenes con base en rasgos geométricos comunes, por ejemplo esquinas. Las esquinas son sencillas de localizar, son identificables de manera única, son invariantes a cambios de escala y de rotación, y son robustas ante los cambios de iluminación, por lo que es posible implementar algoritmos de búsqueda de correspondencias para relacionar las imágenes.

En nuestro caso particular se considera que la escena es estática, y en ella hay superficies planas, y se considera que la cámara es la que está en movimiento libre, lo cual produce transformaciones complejas en las imágenes capturadas, aunque las relaciones entre las superficies planares proyectadas en las imágenes pueden

ser descritas por medio de una relación lineal mucho más simple, la homografía. Para concluir este capítulo se explica cómo a partir del cálculo de homografías entre imágenes es posible estimar las orientaciones absolutas en la escena.

2.2. Características de las imágenes

El proceso de formación de una imagen de una manera ideal es conocido como modelo de cámara pinhole, el cual describe la relación existente entre los puntos de la escena y la imagen generada al proyectar dichos puntos a un plano, conocido como plano imagen.

Representación

Cuando se trabaja con imágenes digitales como en el presente trabajo, existen tres maneras de representarlas, estas son como función, matriz y de manera icónica. La primera representación describe a la imagen como una función bidimensional de intensidad de luz $I(x, y)$, donde x y y representan las coordenadas espaciales que definen puntos en la imagen, los cuales se encuentran contenidos en una región compacta Ω que toma valores en los números reales positivos, donde el valor de $I(x, y)$ en forma ideal es proporcional al brillo de la imagen en ese punto, ver figura 2.1. Una definición formal de una imagen I como función se describe como:

$$I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}_+; (x, y) \rightarrow I(x, y). \quad (2.1)$$

Cuando la imagen es generada por una cámara pinhole, Ω es el plano imagen, en el caso de imágenes digitales el dominio Ω y rango \mathbb{R}_+ son discretos, donde $\Omega = [1, r] \times [1, c]$ es subconjunto de \mathbb{Z}_+^2 y $\mathbb{R}_+ = [0, \phi]$ es subconjunto de \mathbb{Z}_+ . Por ejemplo si se tiene una imagen a escala de grises de 640×480 píxeles, donde cada píxel es almacenado en un byte $\Omega = [1, 640] \times [1, 480]$ y $\mathbb{R}_+ = [0, 255]$. En

la figura 2.1 puede observarse con detalle la representación de una imagen como una función.

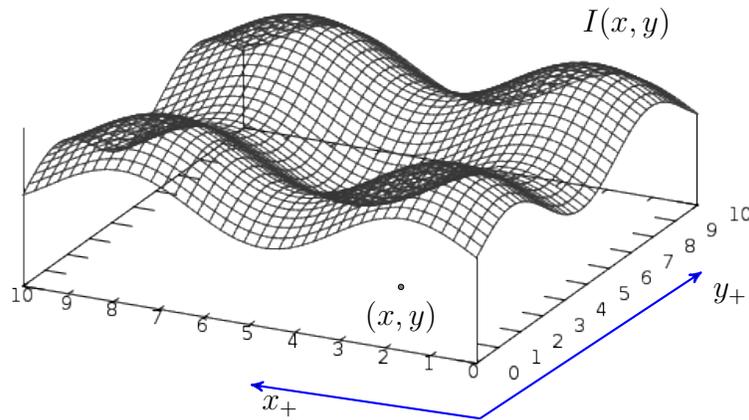


Figura 2.1: Representación de una imagen como gráfica bidimensional, cuyo dominio se encuentra en una región plana contenida en el plano xy .

La segunda representación se describe como una matriz de números enteros, cuyos índices de fila y columna identifican una región de la imagen, tal como se muestra en el cuadro 2.1, donde las entradas de la matriz dependen de las propiedades físicas de la escena proyectada en la imagen. Este tipo de representación facilita el procesamiento y análisis de la imagen a través de un ordenador mediante una serie de instrucciones o algoritmos.

128	45	214	12	45	232	75	123	89	96
23	78	145	95	66	66	128	84	45	50
98	54	54	45	78	245	128	248	45	125
87	95	198	198	66	66	12	56	248	111

Cuadro 2.1: Representación de una imagen como una matriz cuyos índices fila y columna definen una región de la imagen, donde las entradas dependen de las propiedades físicas de la escena.

Por último cuando la imagen es representada icónicamente el sistema visual humano es capaz de entender e interpretar de manera sencilla la información contenida en la imagen, este tipo de representación puede observarse en la figu-

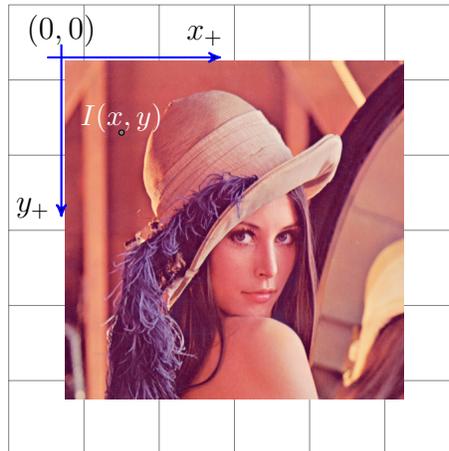


Figura 2.2: La representación icónica de una imagen es la representación que mejor interpretamos y entendemos. El origen de la imagen $(0,0)$ es fijado en su esquina superior izquierda, los ejes x_+ y y_+ indican el sentido de sus coordenadas espaciales, y el valor de $I(x,y)$ corresponde al brillo de la imagen en un punto p con coordenadas x y y .

ra 2.2. Por tal motivo una imagen I contiene la misma información que puede ser interpretada desde distintas maneras.

Formación de la imagen

Para comprender la relación que establece el modelo de cámara pinhole entre la escena y la imagen, considérese que el punto F_0 es el origen de un sistema coordenado o marco de referencia, el cual está situado en el centro de la lente de la cámara. Desde F_0 a una distancia f en dirección del vector \vec{k}_0 se encuentra el plano imagen $\mathbf{\Pi}$, también llamado plano de imagen real, el cual corresponde al lugar físico donde se ubica la película filmadora o la matriz de sensores donde se forma la imagen de la escena, ver figura 2.3.

El eje óptico es la línea perpendicular al plano de la imagen que pasa por F_0 . La intersección del eje óptico y el plano de la imagen es el punto F_1 que también se le conoce como punto principal o centro de la imagen. El punto F_2 es el origen de un marco de referencia situado en la esquina superior izquierda del plano imagen, donde el punto $x \in \mathbb{R}^2$ en $\mathbf{\Pi}$ es representado desde el marco de

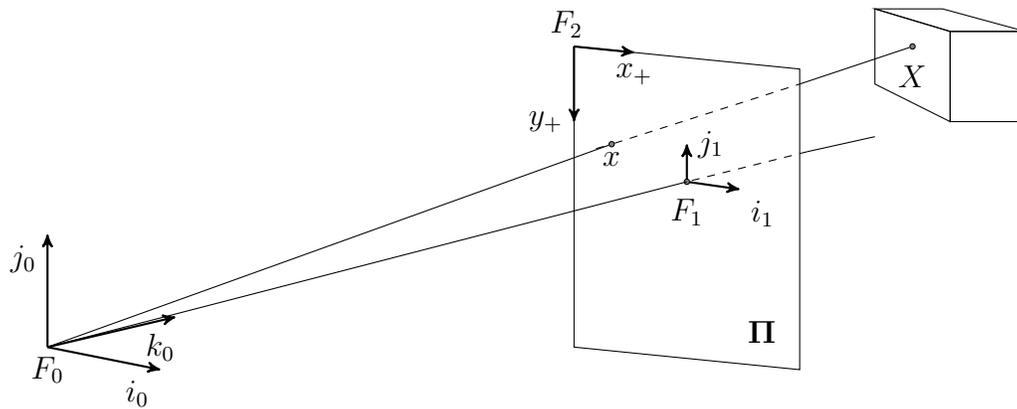


Figura 2.3: Modelo de cámara pinhole, donde el plano imagen Π es el lugar físico donde se encuentra el sensor digital, el punto F_0 es el marco de referencia situado en la lente de la cámara. El eje óptico es la línea perpendicular al plano imagen. El punto x es representado desde el marco de referencia F_2 y representa la proyección en el plano imagen del punto X de la escena. El punto F_1 es conocido como punto principal.

referencia F_2 . Supóngase que se tiene un punto $X \in \mathbb{R}^3$ de la escena, tal que una recta intersecta a los puntos x , F_0 y X , que al ser colineales existe una relación entre los puntos x y X , la cual se define a partir de los vectores homogéneos \mathbf{x} y \mathbf{X} que representan a cada punto.

Para representar a un punto x como un vector homogéneo \mathbf{x} basta con anexarle una componente extra, por lo general la componente agregada es igual a uno. Por ejemplo para el punto $x = (x_1, x_2)$ su representación homogénea sería $\mathbf{x} = [x_1, x_2, 1]^T$. Los vectores homogéneos definen al espacio proyectivo, el cual se define como:

$$\mathbb{P}^n = \{\mathbb{R}^{n+1}\} \setminus \{0\}.$$

La relación existente entre los vectores homogéneos $\mathbf{X} \in \mathbb{P}^3$ y $\mathbf{x} \in \mathbb{P}^2$, es llamada transformación de proyección, la cual se ajusta a la formación real de una imagen usando una cámara pinhole. La transformación de proyección puede ser expresada en términos de una transformación lineal entre \mathbf{X} y \mathbf{x} , la cual se

define como:

$$\lambda \mathbf{x} = KP\mathbf{X}, \quad \lambda \neq 0, \quad K = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

donde P es conocida como matriz de proyección de la cámara cuya dimensión es de 3×4 y $\mathbf{X} = [x_1, x_2, x_3, 1]^T$ es el vector homogéneo asociado al punto X . La matriz K es conocida como matriz de calibración cuya dimensión es de 3×3 , la cual encapsula los parámetros intrínsecos de la cámara, los cuales definen la geometría interna y óptica de la cámara, por lo general estos parámetros se mantienen constantes en tanto no varíen las características y posiciones relativas al centro de la lente de la cámara y la matriz de sensores.

Si P es la matriz canónica de proyección es decir

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

se tiene que $X = P\mathbf{X}$, por lo tanto se deduce que

$$\lambda \mathbf{x} = KX \tag{2.2}$$

donde las componentes x_1 y x_2 de \mathbf{x} toman importancia, debido a que es la representación cartesiana del punto x en la imagen digital I , por consiguiente también representan a los índices fila y columna de I cuando es representada mediante una matriz de números enteros. Para un mayor detalle sobre el proceso de formación y representación de una imagen puede verse en [Woods y Gonzalez \(2002\)](#); [Soatto *et al.* \(2004\)](#); [Forsyth y Ponce \(2004\)](#).

2.3. Relacionando las imágenes

En la sección anterior se estableció la relación existente entre la escena y una imagen, antes de iniciar la estimación de la orientación de una superficie planar es

necesario establecer relaciones entre las imágenes de la escena estática. Es posible encontrar diferentes métodos geométricos que describen las relaciones entre las imágenes, a partir de elementos geométricos de la imagen, y los cambios de brillo que se producen cercanos a ellos.

Entre los elementos geométricos se encuentran los puntos característicos, los cuales son puntos que tienen una posición identificable en una imagen dada. Un tipo de punto característico que se suele usar son las esquinas, donde una esquina se puede definir como la intersección de dos o más bordes de la imagen contenidos dentro de una ventana. Por lo tanto el problema de establecer correspondencias entre las imágenes mediante un conjunto de puntos característicos, consiste en determinar cual punto en una imagen corresponde a otro punto en otra imagen, en el sentido de que ambos puntos corresponden a proyecciones de un mismo punto tridimensional en la escena.

El trabajo que se documenta en esta tesis, las correspondencias se establecerán mediante puntos característicos, utilizando métodos automáticos y con ayuda de una interfaz gráfica, donde el usuario tiene la posibilidad de establecer manualmente algunas correspondencias existentes en las imágenes.

Para establecer correspondencias de forma automática primero se requiere detectar esquinas presentes en cada una de las imágenes, para ello existen diversos métodos para la detección de esquinas; uno de ellos es el método de detección de esquinas de [Harris y Stephens \(1988\)](#), también conocido como el algoritmo de Plessey, el cual es un método de auto-correlación que ofrece una respuesta isotrópica, es robusto respecto al ruido y ofrece una gran respuesta en los bordes. Por lo cual es el que se utiliza en éste trabajo. El método de Harris-Stephens utiliza una métrica para determinar si una ventana contiene o no una esquina, esta métrica se le conoce como *esquinidad*. Las esquinas cuya *esquinidad* sea grande se pueden considerar como esquinas verdaderas y con ellas establecer correspondencias.

La búsqueda de correspondencias compara esquinas entre ambas imágenes y de-

termina que una correspondencia ocurre cuando una esquina es lo suficientemente similar a una esquina en la otra imagen. La métrica utilizada para determinar si una esquina es similar a otra se conoce como correlación cruzada de media cero, cuyas siglas en inglés es ZNCC, la cual se basa en comparar vecindades de las esquinas mediante la correlación cruzada de sus intensidades. Considerando la vecindad como una ventana de dimensiones $(2N + 1) \times (2M + 1)$ píxeles centrada en una esquina, entonces la medida de similitud entre los dos esquinas $x = (x_1, x_2)$ y $x' = (x'_1, x'_2)$ se obtiene mediante el coeficiente de correlación, que en su forma discreta se define como:

$$\gamma = \frac{\sum_{s=-M}^M \sum_{t=-N}^N [I(x_1 - s, x_2 - t) - \bar{I}] [I'(x'_1 - s, x'_2 - t) - \bar{I}']}{\left\{ \sum_{s=-M}^M \sum_{t=-N}^N [I(x_1 - s, x_2 - t) - \bar{I}]^2 [I'(x'_1 - s, x'_2 - t) - \bar{I}']^2 \right\}^{1/2}}$$

donde \bar{I}' es la intensidad media de los píxeles en la vecindad considerada de la imagen $I'(x', y')$, \bar{I} es la intensidad media de los píxeles en la vecindad considerada de la imagen $I(x, y)$. El coeficiente de correlación γ se encuentra normalizado en el rango de -1 a 1 , y es independiente de los cambios de escala aplicados a $I(x, y)$ e $I'(x', y')$. El método utilizado para la búsqueda de correspondencias usando ZNCC puede verse en [Briceño Coronado \(2005\)](#).

En las secciones siguientes al establecer una correspondencia entre los puntos $x = (x_1, x_2)$ y $x' = (x'_1, x'_2)$ se expresara mediante sus vectores homogéneos como $(\mathbf{x}, \mathbf{x}')$. Para un análisis más detallado sobre el cálculo de esquinas y correspondencias puede verse en [Harris y Stephens \(1988\)](#); [Soatto *et al.* \(2004\)](#); [Woods y Gonzalez \(2002\)](#).

2.4. Polígono Convexo

Uno de los objetivos del trabajo que se documenta en esta tesis, es encontrar la orientación absoluta de una superficie planar Π_E presente en la escena. Durante el proceso de formación de las imágenes la superficie planar es proyectada

hacia regiones delimitadas por los polígonos \mathcal{P} y \mathcal{P}' de las imágenes I e I' , ver figura 2.4. Por lo consiguiente los polígonos \mathcal{P} y \mathcal{P}' representan las proyecciones de la superficie planar a la cual se le desea encontrar su orientación, y es a través de ellos que se realiza el cálculo de la orientación.

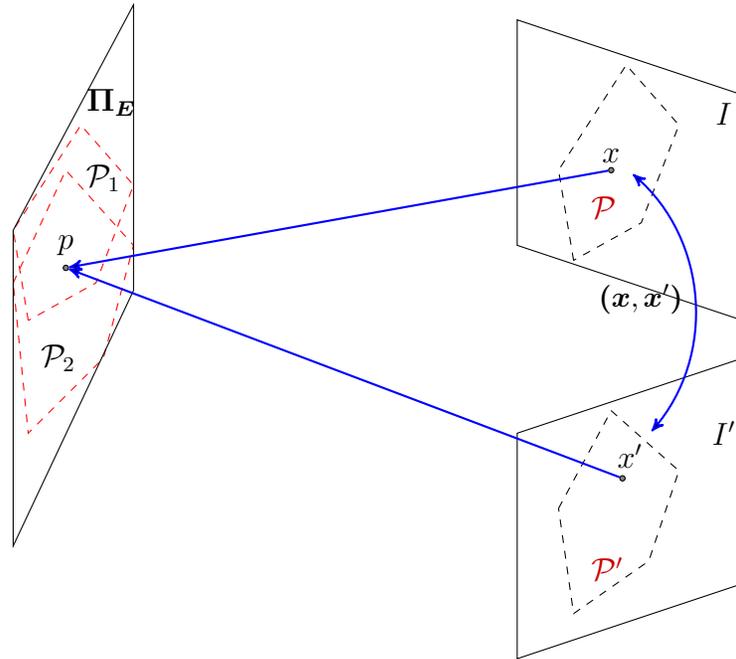


Figura 2.4: Esta imagen muestra la proyección de la superficie planar Π_E hacia regiones delimitadas por los polígonos \mathcal{P} y \mathcal{P}' . El polígono \mathcal{P} también representa la proyección del polígono \mathcal{P}_1 en I , de igual manera el polígono \mathcal{P}' representa la proyección en I' , los polígonos \mathcal{P}_1 y \mathcal{P}_2 son coplanares a Π_E . Si $p \in \mathcal{P}_1 \cap \mathcal{P}_2$ entonces $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$.

En las secciones anteriores se mencionó que para relacionar las imágenes se utilizan correspondencias de puntos característicos, las cuales se obtienen utilizando algoritmos de búsqueda de correspondencias, dichos algoritmos obtienen un determinado número de correspondencias que no pertenecen a los polígonos \mathcal{P} y \mathcal{P}' , por lo tanto para seleccionar un conjunto de correspondencias de manera estratégica, se plantea tomar únicamente aquellas correspondencias que se encuentren en los polígonos \mathcal{P} y \mathcal{P}' de las imágenes relacionadas.

El usuario juega un papel importante durante el procesamiento de las imá-

genes, ya que participa al seleccionar o delimitar las regiones de las imágenes que corresponden a superficies planares en la escena, y añadiendo o quitando, correspondencias en las imágenes. El usuario delimita las regiones por medio de polígonos y el sistema es el encargado de seleccionar las correspondencias que se encuentren dentro de cada polígono.

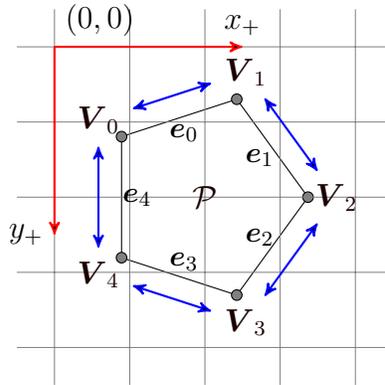


Figura 2.5: Esta imagen muestra los sentidos del recorrido de los aristas e_i del polígono convexo \mathcal{P} , el cual cuenta con cinco vértices V_i e igual número de aristas.

Un polígono \mathcal{P} es una figura geométrica plana delimitada por una colección finita de segmentos de recta formando una curva cerrada o circuito. Los segmentos e_i son llamados aristas o lados, y los puntos V_i donde dos segmentos adyacentes se intersectan son llamados vértices del polígono o esquinas. Un polígono se dice que es convexo si cumple que para cualesquiera par de puntos p_1 y p_2 en \mathcal{P} , el segmento de recta $\overline{p_1 p_2}$ queda enteramente contenida en \mathcal{P} .

Los vértices del polígono se pueden recorrer en dos sentidos diferentes, con respecto al sentido de las manecillas del reloj, es decir del eje x_+ hacia el eje y_+ , y contrario al sentido de las manecillas, de y_+ hacia x_+ , el recorrido en ambos sentidos puede verse en la figura 2.5

Supóngase que existe un punto $p = (x, y)$ en el mismo plano que \mathcal{P} , si se toman los vértices $V_1 = (x_1, y_1)$ y $V_0 = (x_0, y_0)$, de tal manera que formen un triángulo con p , tal y como se muestra en la figura 2.6, por álgebra lineal se sabe

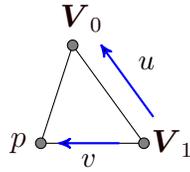


Figura 2.6: Esta imagen muestra el triángulo formado por los puntos V_0 , V_1 y p , donde V_0 y V_1 son vértices del polígono \mathcal{P} . En sentido contrario a las manecillas del reloj p se encuentra a la izquierda de los vértices.

que el área del paralelogramo definido por los vectores $u = [x_0 - x_1, y_0 - y_1]$ y $v = [x - x_1, y - y_1]$, es igual al determinante de los mismos, esta implicación se puede expresar como:

$$\det(B) = \begin{vmatrix} (x_0 - x_1) & (y_0 - y_1) \\ (x - x_1) & (y - y_1) \end{vmatrix} = (x_0 - x_1)(y - y_1) - (x - x_1)(y_0 - y_1) \quad (2.3)$$

con base al signo del valor del determinante se puede saber si el punto p se encuentra o no en \mathcal{P} , si el recorrido de los vértices es con respecto a las manecillas del reloj, p se encontrará del lado derecho de los aristas del polígono y el valor del determinante es mayor que cero, si el recorrido es contrario a las manecillas del reloj p se encontrará del lado izquierdo y su determinante es menor que cero, cuando $\det(B) = 0$ indica que p se encuentra en el perímetro de \mathcal{P} . Para un análisis mas detallado del cálculo del área del triangulo puede verse en [O'Rourke \(1998\)](#).

Algoritmo de cerco convexo

Supóngase que los polígonos \mathcal{P}_1 y \mathcal{P}_2 se encuentran en Π_E , entonces sus proyecciones \mathcal{P} y \mathcal{P}' se encuentran en las imágenes I e I' , respectivamente, si un punto $p \in \Pi_E$ tiene una correspondencia $(\mathbf{x}, \mathbf{x}')$ tal que p se encuentre en la intersección $\mathcal{P}_1 \cap \mathcal{P}_2$, entonces $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$, la intersección de los polígonos puede verse en la figura 2.4. Por lo tanto si algún punto del par puntos de una correspondencia $(\mathbf{x}_i, \mathbf{x}'_i)$ no se encuentra dentro algún polígono, entonces dicha correspondencia

no aporta información relevante de la superficie planar, por consiguiente si $x \notin \mathcal{P}$ o $x' \notin \mathcal{P}'$ la correspondencia entre los puntos puede ser descartada.

Por lo tanto si se tiene un conjunto de n correspondencias de puntos y al menos una correspondencia $(\mathcal{P}, \mathcal{P}')$ de polígonos fijados por el usuario, es posible descartar aquellas correspondencias que no aportan la suficiente información de la superficie, esto se logra empleando el algoritmo de cerco convexo, el cual se basa en la relación geométrica que existe entre los polígonos y los puntos, la cual se define en la ecuación 2.3. El algoritmo de cerco convexo se muestra en el cuadro 2.2.

2.5. Homografía

El concepto de homografía o transformación proyectiva, se define como una transformación invertible h que mapea elementos de un espacio proyectivo \mathbb{P}^2 hacia otro espacio \mathbb{P}^2 , solo si se cumple que para cualesquiera tres puntos x_1, x_2, x_3 colineales, al aplicarles h los puntos resultantes $h(x_1), h(x_2), h(x_3)$ también son colineales. Este concepto tiene una definición algebraica que se presenta bajo el siguiente teorema.

Teorema 2.5.1 *Una transformación $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ es una homografía sí y solo sí existe una matriz H no singular de 3×3 , tal que para cualquier punto en \mathbb{P}^2 representado por un vector \mathbf{x} se cumple que $h(\mathbf{x}) = H\mathbf{x}$, véase en [Hartley y Zisserman \(2004\)](#).*

Por lo tanto para referirse a una homografía basta con escribir la relación algebraica:

$$\mathbf{x}' = H\mathbf{x}, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (2.4)$$

Objetivo

Dadas n correspondencias $(\mathbf{x}, \mathbf{x}')$ y $m \geq 1$ correspondencias $(\mathcal{P}, \mathcal{P}')$, determinar las correspondencias cuyos puntos cumplen que $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$.

Algoritmo

1. **Paso 1:** Por cada correspondencia de polígonos generar una lista \mathcal{L} para almacenar las correspondencias en común.
2. **Paso 2:** Por cada correspondencia de polígonos se elige una correspondencia de puntos.
3. **Paso 3:** Por cada punto $x = (x, y)$ realizar el recorrido de los vértices \mathbf{V}_i del polígono \mathcal{P} , en el sentido de las manecillas del reloj.
4. **Paso 4:** Por cada par de vértices $\mathbf{V}_i = (x_i, y_i)$ e $\mathbf{V}_{i+1} = (x_{i+1}, y_{i+1})$ calcular

$$(x_i - x_{i+1})(y - y_{i+1}) - (x - x_{i+1})(y_i - y_{i+1})$$

Si al finalizar el recorrido de \mathcal{P} , todos los vértices cumplen que $(x_i - x_{i+1})(y - y_{i+1}) - (x - x_{i+1})(y_i - y_{i+1}) \geq 0$, entonces indicar que $x \in \mathcal{P}$, en caso contrario indicar que $x \notin \mathcal{P}$.

5. **Paso 5:** Repetir los pasos 2 y 3 para el punto $x' = (x, y)$ y el polígono \mathcal{P}' .
6. **Paso 6:** Si se cumple que $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$, agregar la correspondencia $(\mathbf{x}, \mathbf{x}')$ a \mathcal{L} , en caso contrario omitirla.
7. **Paso 7:** Repetir los pasos del dos al seis m veces.

Cuadro 2.2: Algoritmo de cerco convexo para la obtención de correspondencias cuyos puntos x y x' asocian a un punto $p \in \mathcal{P}_1 \cap \mathcal{P}_2$, tal que $\mathcal{P}_1, \mathcal{P}_2 \in \Pi_E$

donde los vectores \mathbf{x}' y \mathbf{x} establecen una correspondencia $(\mathbf{x}, \mathbf{x}')$ entre los planos, en nuestro caso planos presentes en las imágenes y H define la homografía entre ellos, en la práctica H se desconoce pero se puede estimar a partir de un conjunto de cuatro o más correspondencias.

La matriz H es una matriz homogénea, esto es, es posible multiplicarla por alguna constante diferente de cero y seguirá representando la misma transformación. Es por ello que podemos determinar una homografía a partir de ocho parámetros, o bien se dice que una homografía tiene 8 grados de libertad.

Estimación de la homografía

La estimación de una homografía se obtiene a partir de un conjunto de cuatro o más correspondencias de puntos entre las imágenes, que para fines prácticos los puntos x'_i y x_i son puntos característicos entre dos imágenes. Dicha estimación se obtiene a partir del algoritmo de Transformación Lineal Directa, cuyas siglas en inglés son DLT, el cual se obtiene a partir del cálculo del producto cruz entre los vectores homogéneos \mathbf{x}'_i y $H\mathbf{x}_i$, que al ser colineales da como resultado:

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{bmatrix} y'_i \mathbf{h}^{3T} \mathbf{x}_i - \mathbf{h}^{2T} \mathbf{x}_i \\ \mathbf{h}^{1T} \mathbf{x}_i - x'_i \mathbf{h}^{3T} \mathbf{x}_i \\ x'_i \mathbf{h}^{2T} \mathbf{x}_i - y'_i \mathbf{h}^{1T} \mathbf{x}_i \end{bmatrix} = 0$$

donde \mathbf{h}^{jT} representa la j -ésima fila de la homografía H , que al factorizarla en forma matricial la expresión anterior se obtiene que:

$$\begin{bmatrix} 0^T & -\mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ \mathbf{x}_i^T & 0^T & -x'_i \mathbf{x}_i^T \\ -y'_i \mathbf{x}_i^T & x'_i \mathbf{x}_i^T & 0^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = 0 \quad (2.5)$$

como se puede observar la ecuación tiene la forma $A_i \mathbf{h}$, donde A_i es una matriz de dimensión de 3×9 y \mathbf{h} es un vector de dimensión 9×1 ; como la tercera fila es

una combinación lineal de las primeras dos filas, la ecuación se puede reducir a:

$$\begin{bmatrix} 0^T & -\mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ \mathbf{x}_i^T & 0^T & -x'_i \mathbf{x}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = 0 \quad (2.6)$$

por definición la homografía es una transformación que mapea los puntos x_i de una plano hacia otro plano definidos en un espacio proyectivo \mathbb{P}^2 , esto implica que cada correspondencia debe cumplir la ecuación $A_i \mathbf{h} = 0$, por lo tanto dado un conjunto de n correspondencias $(\mathbf{x}_i, \mathbf{x}'_i)$, con $n \geq 4$, se define una matriz $A \in \mathbb{R}^{2n \times 9}$ como:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \quad (2.7)$$

donde cada A_i es la matriz representada en la ecuación 2.6, para resolver el sistema sobre-determinado $A\mathbf{h} = 0$ se opta por realizar la descomposición de valores singulares (SVD) de la matriz A y así obtener una estimación de la homografía que realice el mapeo entre las correspondencias del conjunto definido. Para una descripción detallada sobre la resolución de un sistema sobre-determinado véase en [Teukolsky et al. \(1992\)](#).

El algoritmo DLT consiste en calcular el valor de todas las entradas de la matriz A , a partir del conjunto de correspondencias, posteriormente resolver el sistema $A\mathbf{h} = 0$ y así obtener los valores de las entradas de la homografía H . Sin embargo este algoritmo no es lo suficientemente robusto, pues depende del marco de referencia en el cual expresamos los puntos, por consiguiente no es invariante ante transformaciones de similitud en la imagen. Para solucionar dicha situación Hartley propuso aplicar una transformación de normalización al conjunto de correspondencias, véase en [Hartley \(1997\)](#); esto es que los vectores \mathbf{x}_i son remplazados por $\tilde{\mathbf{x}}_i = T\mathbf{x}_i$ y los vectores \mathbf{x}'_i en la otra imagen son remplazados por $\tilde{\mathbf{x}}'_i = T'\mathbf{x}'_i$, donde T y T' son matrices de transformación de similitud correspondientes a los vectores homogéneos de cada imagen de tal manera que

su centro de masa sea el origen coordenado y cuya distancia promedio del origen sea $\sqrt{2}$. Estas matrices están dadas por:

$$T = \begin{bmatrix} k & 0 & -k\bar{x}_i \\ 0 & k & -k\bar{y}_i \\ 0 & 0 & 1 \end{bmatrix} \quad T' = \begin{bmatrix} k' & 0 & -k'\bar{x}'_i \\ 0 & k' & -k'\bar{y}'_i \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

a partir de las correspondencias $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i)$ se calcula una homografía \tilde{H} mediante el algoritmo DLT mencionado anteriormente, posteriormente se obtiene la homografía H mediante la ecuación $H = T'^{-1}\tilde{H}T$, que se deriva de la ecuación $\tilde{\mathbf{x}}'_i = \tilde{H}\tilde{\mathbf{x}}_i$ con $\tilde{H} = T'HT^{-1}$. El cálculo de la homografía por medio de la normalización de las correspondencias se le conoce como algoritmo DLT normalizado, el cual puede verse en el cuadro 2.3.

Objetivo

Dadas $n \geq 4$ correspondencias $(\mathbf{x}, \mathbf{x}')$ tal que $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$, determinar la homografía 2D entre dos imágenes.

Algoritmo

1. **Normalización de correspondencias:** Encontrar las transformaciones de similitud T, T' y fijar un nuevo conjunto de correspondencias $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i)$
2. **DLT:** Calcular la matriz \tilde{A} a partir de $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i)$ y resolver el sistema $\tilde{A}\tilde{\mathbf{h}}$ para obtener la homografía \tilde{H} .
3. **Des-normalización:** Calcular $H = T'^{-1}\tilde{H}T$.

Cuadro 2.3: Algoritmo DLT Normalizado para el cálculo de Homografías 2D, véase en [Hartley y Zisserman \(2004\)](#).

Estimación robusta

El algoritmo DLT normalizado utiliza un conjunto de correspondencias para estimar la homografía, algunas de las correspondencias pueden ser correspondencias erróneas, a las cuales se les denomina puntos atípicos, por lo tanto la

estimación puede ser afectada por el error que introduce la presencia de puntos atípicos.

Una solución para reducir el número de correspondencias erróneas es aplicar un algoritmo denominado RANSAC. El principio del algoritmo consiste en tomar muestras aleatorias de cuatro correspondencias que no presenten colinealidad entre si y calcular sus soluciones, se requieren de cuatro correspondencias ya que es la mínima cantidad de correspondencias, que se requieren para calcular una homografía. En nuestro caso las soluciones son homografías que permitan agrupar el conjunto de correspondencias en puntos típicos (correspondencias correctas) y atípicos, los puntos típicos son aquellas correspondencia que se ajustan mejor a la ecuación 2.5. El proceso de agrupar los puntos típicos y atípicos se repite N número de veces y se escoge la homografía \hat{H} que proporcione el mayor número de puntos típicos; para una descripción detallada del método original véase en Fischler y Bolles (1981).

Estimación optimizada

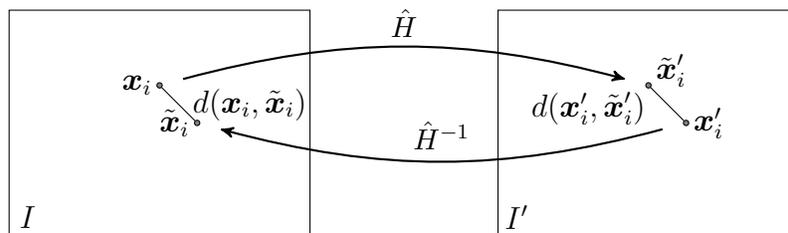


Figura 2.7: Error simétrico que se produce al realizar la estimación de la homografía \hat{H} , a partir de la correspondencia de puntos $(\mathbf{x}_i, \mathbf{x}'_i)$ de las imágenes I e I' . La notación $d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$ denota la distancia euclidiana entre \mathbf{x}_i y $\tilde{\mathbf{x}}_i$, donde $\tilde{\mathbf{x}}_i = \hat{H}^{-1}\mathbf{x}'_i$.

El conjunto de correspondencias correctas obtenidas mediante el algoritmo de RANSAC, estima una homografía \hat{H} que introduce cierto error al momento de realizar el mapeo entre las imágenes, por lo cual es necesario cuantificar el error que ocurre en ambas imágenes por cada correspondencia, el error se conoce como

error simétrico o error de reproyección y se define como:

$$d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \tilde{\mathbf{x}}'_i)^2 = (x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 + (x'_i - \tilde{x}'_i)^2 + (y'_i - \tilde{y}'_i)^2 \quad (2.9)$$

donde \mathbf{x}_i y $\tilde{\mathbf{x}}_i$ son vectores homogéneos que representan a puntos de la primera imagen, con $\tilde{\mathbf{x}}_i = \hat{H}^{-1}\mathbf{x}'_i$, cuya distancia euclidiana al cuadrado $d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)^2$ define el error que introduce la homografía, de manera similar se define el error que ocurre con la segunda imagen mediante la relación $d(\mathbf{x}'_i, \tilde{\mathbf{x}}'_i)^2$ donde $\tilde{\mathbf{x}}'_i = \hat{H}\mathbf{x}_i$, la relación de errores puede observarse con detalle en la figura 2.7.

Entonces si la homografía fuese exacta la distancia entre los puntos x_i y \tilde{x}_i de la primera imagen debe ser cero, de igual manera debe ocurrir con los puntos x'_i y \tilde{x}'_i de la segunda imagen. El error total que ocurre durante el mapeo se conoce error de reproyección o error residual, el cual es la suma de los errores ocurridos en ambas imágenes y se define como:

$$\|\vec{r}(\hat{H})\|^2 = \sum_{i=1}^n d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \tilde{\mathbf{x}}'_i)^2, \quad \text{tal que } \tilde{\mathbf{x}}'_i = \hat{H}\mathbf{x}_i \quad \forall i \quad (2.10)$$

en el presente caso se está buscando una homografía \hat{H}_k que minimice el error residual, esto significa que el error residual se aproxime a cero. Por lo tanto dado un conjunto de n correspondencias correctas, se desea encontrar una homografía \hat{H}_k que proporcione el mejor ajuste al mapeo entre las correspondencias, para ello se aplica el método iterativo de *Levenberg – Marquardt*.

Entonces para estimar una homografía que minimice el error de reproyección mediante un método robusto se añade el algoritmo de RANSAC y el método de *Levenberg – Marquardt* al algoritmo DLT normalizado, esta adaptación se muestra en el cuadro 2.4, donde el valor de $5.99\sigma^2$ utilizado para la obtención de puntos típicos, corresponde al umbral ideal para el cálculo de homografías, esto puede verse en [Hartley y Zisserman \(2004\)](#). Para un análisis más detallado sobre el método *Levenberg – Marquardt* véase en [Heath \(1997\)](#).

Objetivo

Dadas $n \geq 4$ correspondencias $(\mathbf{x}, \mathbf{x}')$ tal que $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$, determinar la homografía 2D entre dos imágenes cuyo error sea mínimo.

Algoritmo

1. **RANSAC, estimación robusta:** Repetir para N número de veces, donde N se calcula dinámicamente.

- Seleccionar una muestra aleatoria de 4 correspondencias $(\mathbf{x}'_i, \mathbf{x}_i)$, de las cuales 3 sean no colineales, posteriormente calcular mediante el algoritmo DLT normalizado la homografía H_n .
- Calcular el número de puntos atípicos mediante H_n tal que:

$$d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \tilde{\mathbf{x}}'_i)^2 < 5.99\sigma^2$$

Escoger una homografía H_0 que proporcione el mayor número de puntos típicos. En caso de empate escoger la que proporcione la menor desviación estándar de los puntos típicos.

2. **Estimación óptima :** Aplicar el método *Levenberg – Marquardt* cuyo parámetro inicial sea H_0 , para estimar la homografía \hat{H}_k que proporcione el menor error de reproyección.

Cuadro 2.4: Algoritmo de RANSAC y *Levenberg – Marquardt* para el cálculo de Homografías 2D con el menor error de reproyección sujeto a correspondencias erróneas, véase en [Hartley y Zisserman \(2004\)](#).

2.6. Estimación de orientación de una superficie planar en la escena

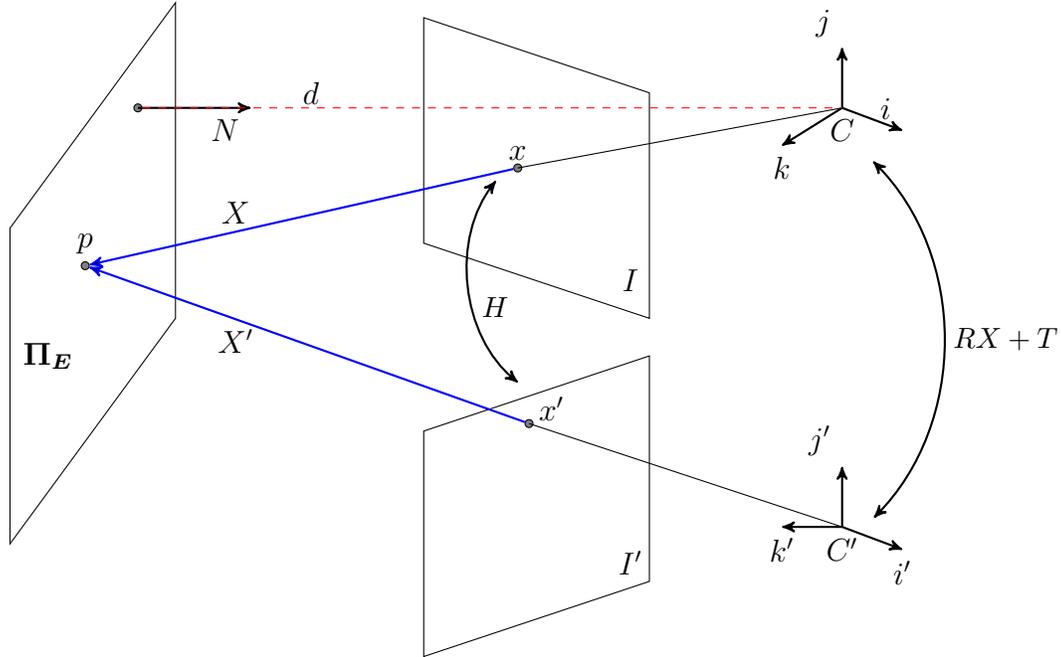


Figura 2.8: El punto p se encuentra en un plano Π_E de la escena, que junto con los centros C, C' de las cámaras forman un triángulo, los vectores X, X' representan al punto p , donde $\lambda \mathbf{x} = KX$ y $\lambda' \mathbf{x}' = KX'$. Dichos vectores están relacionados por medio de una homografía H . El vector N es normal al Π_E y proporciona la orientación del mismo, a su vez N se encuentra a una distancia d del punto C .

En esta sección se explica a detalle los conceptos teóricos descritos en [Soatto et al. \(2004\)](#), para el cálculo de la orientación de una superficie planar.

Considérese que se tienen dos imágenes I e I' de una misma escena estática, las cuales fueron tomadas desde dos marcos de referencia diferentes C y C' ; si se tiene un punto p de la escena tal como se muestra en la figura 2.8, este mismo punto es representado de manera diferente en cada marco, esto quiere decir que el punto p visto desde C es representado por el punto X y visto desde C' es representado por el punto X' . Los puntos X y X' se encuentran relacionados por

medio de una transformación rígida, la cual se expresa como:

$$X' = RX + T \quad (2.11)$$

donde R es una matriz de rotación y T es un vector de traslación. Si se considera que el mismo punto p se encuentra situado en una superficie planar Π_E de la escena, entonces existe un vector unitario $N^T = [n_1, n_2, n_3]$ normal a Π_E y existe una distancia $d > 0$ hacia el marco C tal que:

$$N^T X = n_1x + n_2y + n_3z = d \iff \frac{1}{d}N^T X = 1 \quad (2.12)$$

entonces si se multiplica por $\frac{1}{d}N^T X$ al vector de traslación T de la ecuación 2.11 se tiene que:

$$X' = (R + T\frac{1}{d}N^T)X, \quad H = (R + T\frac{1}{d}N^T) \quad (2.13)$$

donde H es la homografía que relaciona los puntos entre los marcos de referencia, al fijar N al marco C proporciona la orientación de la superficie planar Π_E .

Mediante la ecuación 2.2 se tiene que $\lambda\mathbf{x} = KX$ y que $\lambda'\mathbf{x}' = KX'$, entonces si sustituimos $\lambda K^{-1}\mathbf{x} = X$ y $\lambda'K^{-1}\mathbf{x}' = X'$ en la ecuación 2.13 se obtiene que:

$$\mathbf{x}' = \lambda_L KHK^{-1}\mathbf{x}, \quad \lambda_L = \frac{\lambda}{\lambda'} \quad (2.14)$$

suponiendo que la matriz $H_L = \lambda_L KHK^{-1}$ define una homografía tal y como se muestra en la ecuación 2.4, entonces $(\mathbf{x}', \mathbf{x})$ definen una correspondencia entre las imágenes I e I' .

Estimación de la orientación de manera ideal

Cuando se trabaja con imágenes generadas de manera ideal, significa que la matriz de calibración K es igual a la matriz identidad, entonces de la ecuación 2.15

se obtiene que:

$$\mathbf{x}' = H_L \mathbf{x}, \quad H_L = \lambda_L H, \quad \lambda_L = \frac{\lambda}{\lambda'}$$
 (2.15)

el escalar λ_L corresponde al segundo valor más grande de la descomposición de valor singular (SVD) de la matriz H_L , calculando dicho escalar es fácil obtener la homografía H mediante un simple despeje. Por lo tanto si estima una homografía H_L mediante el algoritmo definido en el cuadro 2.4, a partir de ella se puede calcular la homografía H , esto implica realizar la descomposición de valor singular de H_L , la cual se expresa como:

$$H_L^T H_L = V \Sigma V^T, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}$$
 (2.16)

donde los valores singulares σ_i^2 , $i \in [1, 2, 3]$ están ordenados en forma descendente, y por tanto la entrada σ_2^2 de la matriz Σ corresponde al segundo valor singular más grande de H_L . El valor de λ_L se obtiene mediante la relación $\sqrt{\sigma_2^2} = |\lambda_L|$, por tanto al dividir H_L entre $|\lambda_L|$ se tiene que:

$$H = \frac{1}{|\lambda_L|} H_L$$

para calcular N de la ecuación 2.13 se requiere calcular la matriz simétrica $H^T H = V \Sigma V^T$, la cual tiene tres eigenvalores $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq 0$, donde $\sigma_2 = 1$. Si $[v_1, v_2, v_3]$ son los vectores columna de V , se tiene que:

$$H^T H v_1 = \sigma_1^2 v_1, \quad H^T H v_2 = v_2, \quad H^T H v_3 = \sigma_3^2 v_3.$$
 (2.17)

donde v_2 es ortogonal a N y T , además su longitud es preservada cuando se realiza el mapeo mediante la homografía H . Si el determinante de V es negativo se toma la matriz $-V$. A partir de V o $-V$ podemos calcular dos vectores unitarios u_1 y

u_2 los cuales se definen como:

$$u_1 = \frac{\sqrt{1 - \sigma_3^2}v_1 + \sqrt{\sigma_1^2 - 1}v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}}, \quad u_2 = \frac{\sqrt{1 - \sigma_3^2}v_1 - \sqrt{\sigma_1^2 - 1}v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad (2.18)$$

a partir de los vectores v_2 , u_1 , u_2 se definen las matrices:

$$U_1 = [v_2, u_1, \widehat{v_2}u_1], \quad W_1 = [Hv_2, Hu_1, \widehat{Hv_2}Hu_1];$$

$$U_2 = [v_2, u_2, \widehat{v_2}u_2], \quad W_2 = [Hv_2, Hu_2, \widehat{Hv_2}Hu_2].$$

donde $\widehat{Hv_2}Hu_1$ define el producto cruz entre los vectores Hv_2 y Hu_1 . Con las matrices se obtiene que:

$$RU_1 = W_1, \quad RU_2 = W_2$$

donde R es la matriz de rotación de la descomposición de H definida en la ecuación 2.13, analizando R se puede ver que tiene dos valores distintos y cada valor tiene un valor simétrico. Por tanto H tiene cuatro soluciones, donde cada solución se presentan en el cuadro 2.5.

Soluciones					
Solución 1	$R_1 =$	$W_1U_1^T$	Solución 2	$R_2 =$	$W_2U_2^T$
	$N_1 =$	$\widehat{v_2}u_1$		$N_2 =$	$\widehat{v_2}u_2$
	$\frac{1}{d}T_1 =$	$(H - R_1)N_1$		$\frac{1}{d}T_2 =$	$(H - R_2)N_2$
Solución 3	$R_3 =$	R_1	Solución 4	$R_4 =$	R_2
	$N_3 =$	N_1		$N_4 =$	$-N_2$
	$\frac{1}{d}T_3 =$	$-\frac{1}{d}T_1$		$\frac{1}{d}T_4 =$	$-\frac{1}{d}T_2$

Cuadro 2.5: Tabla de Soluciones de la descomposición de H , se elige la solución que cumpla que $N^T\epsilon > 0$, véase en [Soatto et al. \(2004\)](#).

La solución que nos proporcione la orientación del plano con respecto a la primera cámara es aquella que cumpla la restricción de profundidad $N^T\epsilon = n_3 > 0$, de esa manera se asegura que el vector N este orientado hacia a la cámara.

Estimación de la orientación usando calibración

En el caso ideal la matriz K es igual a la identidad, en la práctica debido a los diferentes procesos de fabricación de las cámaras, K es diferente de la identidad, por lo cual existen diferentes métodos para obtener una estimación de la matriz de calibración. Supóngase que la matriz de calibración K es conocida, entonces despejando H de la ecuación 2.15 se obtiene que:

$$\lambda_L H = K^{-1} H_L K = H' \quad (2.19)$$

por lo que es posible estimar el vector normal al plano Π_E de igual manera que en el caso ideal, a partir de un conjunto de correspondencias entre las imágenes.

Para finalizar esta sección se presenta el cuadro 2.6, el cual incluye de manera implícita el cálculo de la homografía mediante el algoritmo DLT, ver cuadro 2.4, la cual es utilizada para estimar la orientación de una superficie planar.

En el siguiente capítulo se explicará cómo el acoplamiento entre los conceptos teóricos y los algoritmos mencionados anteriormente, nos permiten obtener información geométrica, a partir de una secuencia de imágenes digitales de la escena de prueba, con el fin de comprender la serie de procedimientos que se involucran, para la estimación de las orientaciones de las superficies planares presentes en la escena.

Objetivo

Dadas $n \geq 4$ correspondencias $(\mathbf{x}, \mathbf{x}')$ tal que $x \in \mathcal{P}$ y $x' \in \mathcal{P}'$, con la matriz de calibración K conocida, determinar la orientación de una superficie planar Π_E .

Algoritmo

1. **Estimación óptima:** Estimar una homografía H_L cuyo error sea mínimo.
2. **Estimación de la homografía :** Calcular la homografía H' mediante la expresión

$$H' = K^{-1}H_LK$$

3. **Normalización de la homografía :** Calcular los eigenvalores $\{\sigma_1, \sigma_2, \sigma_3\}$ de la homografía H' y obtener H mediante la expresión

$$H = H'/\sigma_2$$

4. **Descomposición de la homografía :** Calcular la descomposición de valor singular de

$$H^T H = V \Sigma V^T$$

Posteriormente calcular las cuatro soluciones para la descomposición $\{R, \frac{1}{d}T, N\}$, finalmente seleccionar las dos soluciones físicamente posibles que cumplen la restricción de profundidad $N^T \epsilon_3 > 0$, tal y como se muestra en el cuadro 2.5.

Cuadro 2.6: Algoritmo de estimación de la orientación de una superficie planar Π_E , basado en algoritmos de estimación de homografía y estimación de orientación de superficies a partir de homografías que aparecen en [Hartley y Zisserman \(2004\)](#) y [Soatto *et al.* \(2004\)](#) respectivamente.

2.7. Resumen

En este capítulo se explicó el proceso de formación de una imagen digital, la cual nos provee de información geométrica de la escena. La información de los elementos geométricos de la imagen nos permitió relacionar imágenes a partir de puntos característicos, como son las esquinas. El uso de esquinas facilitó el uso de algoritmos de búsqueda de correspondencias para relacionar las imágenes y a partir de ellas estimar una homografía que nos relacione elementos entre diferentes planos de las imágenes. La homografía al estimarse por medio de correspondencias, los algoritmos que llevan a cabo la búsqueda de correspondencias nos proporcionan ciertas correspondencias que no proveen información relevante del plano, por lo cual el usuario tiene la posibilidad de delimitar las superficies mediante polígonos convexos, a fin de eliminar aquellas correspondencias que no se requieran para el cálculo de la homografía.

Por último se explicó cómo la homografía encapsula información geométrica de la relación existente entre el plano de la escena y las imágenes donde es proyectado, como lo es el vector normal al plano. Encontrar el vector normal al plano es el tema central de este trabajo de tesis ya que nos proporciona la orientación del plano con respecto a un marco de referencia, como se mencionó anteriormente, el vector normal al plano puede ser estimado al descomponer la homografía.

Capítulo 3

Metodología

3.1. Introducción

Uno de los objetivos de la tesis fue desarrollar un sistema de software con el cual se llevaron a cabo los experimentos para probar tanto que el sistema interactivo como el método de estimación sirven. El sistema de software hace uso de técnicas de procesamiento de imágenes y visión computacional e implementa los algoritmos definidos en el marco teórico.

Se diseñó una interfaz gráfica de usuario (GUI, por sus siglas en inglés) para permitir la interacción entre el usuario y el sistema de software. La interfaz permite al usuario visualizar una o más imágenes de una secuencia, y sirve como el espacio en donde el sistema de software muestra resultados parciales de los experimentos. La interacción entre el usuario y el sistema de software principalmente se da en dos aspectos diferentes: primero como medio de validación del cómputo del registro entre imágenes, realizado de manera automática por el sistema de software; y segundo, la segmentación de las superficies planares en las imágenes. Las imágenes de la escena de prueba, consisten en una secuencia de imágenes generadas por una cámara en movimiento libre, sin presencia de oclusiones.

Al ser el sistema de software una herramienta importante al momento de efectuar los experimentos, se desarrolló utilizando una arquitectura de software que

facilité su diseño, su desarrollo y mantenimiento del mismo, es por ello que se eligió la arquitectura Modelo Vista Controlador (MVC), la cual separa los datos del sistema, la interfaz gráfica de usuario, y la lógica de control en tres componentes diferentes. La vista se encarga de interactuar con el usuario, en nuestro caso corresponde con la GUI. El modelo se encarga de acceder a los datos con los cuales el sistema opera. El controlador es el que se encarga de gestionar los eventos de la interfaz y de acceder al modelo.

Al trabajar con varias imágenes a la vez, se presenta la situación de la capacidad de almacenamiento de la memoria de la computadora, por lo cual es ineficiente almacenar en memoria toda la secuencia de imágenes, para solucionar dicha situación se hace uso de una estructura de datos llamada cola doble circular o bicola, la cual se encarga de cargar a memoria las imágenes desde un dispositivo de almacenamiento, conforme el sistema lo solicite. En nuestro caso la bicola es el componente que representa al modelo de datos en la arquitectura MVC.

El desarrollo del sistema se realizó utilizando el lenguaje de programación C++. Los controladores del sistema hacen uso de la biblioteca de funciones VisionLibs, la cual fue desarrollada en la Facultad de Matemáticas de la UADY para las operaciones de lectura, escritura, filtrado de imágenes, entre otras. Para el desarrollo de la GUI se usó la biblioteca WxWidgets 2.8, la cual provee de un gran número de clases y metodos que simplifican la creación de interfaces gráficas para los sistemas Windows, OS X, Linux y UNIX en arquitecturas de 32 y 64 bits. Para tener un mejor entendimiento de la arquitectura MVC y de la cola doble circular puede verse en el apéndice [A](#) de este documento.

3.2. Interfaz gráfica de usuario

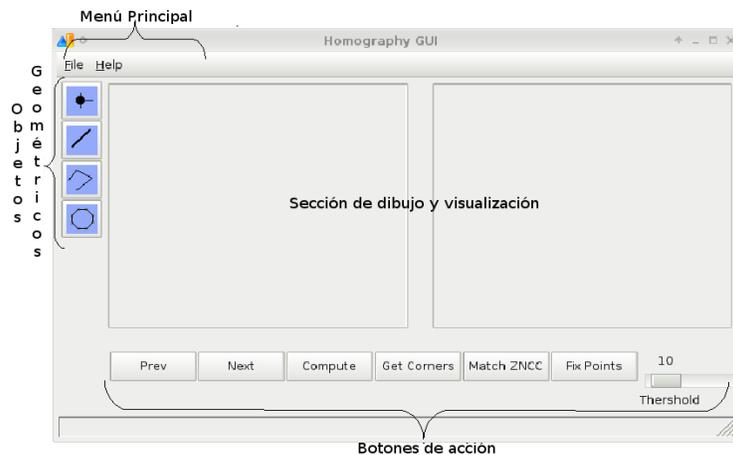


Figura 3.1: Ventana principal del sistema esta compuesta de cuatro secciones, sección de menú principal, objetos geométricos, botones de acción y área de dibujo y visualización.

La interfaz gráfica de usuario es un componente importante del sistema, ya que por medio de ella se desarrollan los experimentos que se plantean en este documento, por lo que es necesario conocer los componentes gráficos que la integran. La GUI se compone de una ventana principal que proporciona una serie de botones, que responden a las acciones que realiza el usuario, las cuales son:

- Abrir la secuencia de imágenes almacenadas en un dispositivo de almacenamiento.
- Dibujar sobre las imágenes.
- Seleccionar los objetos geométricos a dibujar.
- Seleccionar las imágenes de prueba.
- Obtener esquinas de las imágenes.
- Establecer correspondencias de manera automática o manual.
- Delimitar las superficies planares de las imágenes.

- Obtener las orientaciones de las superficies.
- Guardar en archivos los objetos geométricos dibujados en el área de dibujo.

La ventana principal se divide en cuatro secciones, la sección de menú principal, objetos geométricos, botones de acción y sección de dibujo y visualización, estas secciones se pueden observar en la figura 3.1.

La sección de dibujo y visualización se compone de dos áreas de dibujo, donde cada área tiene la funcionalidad de visualizar en pantalla una imagen de la secuencia y es donde el usuario puede dibujar los objetos geométricos que requiera, esto implica que solamente se trabaja con un par de imágenes a la vez.

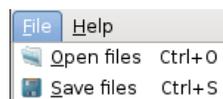


Figura 3.2: El menú principal está dividido en dos secciones File y Help. La sección File posee los botones Open y Save, y la sección Help muestra la versión del sistema.

El menú principal está dividido en dos secciones tal como se muestra en la figura 3.2. La sección File tiene los botones Open y Save. El botón Open tiene como función abrir una ventana de diálogo, donde el usuario elige la secuencia de imágenes con la que trabajará, tal y como se muestra en la figura 3.3.

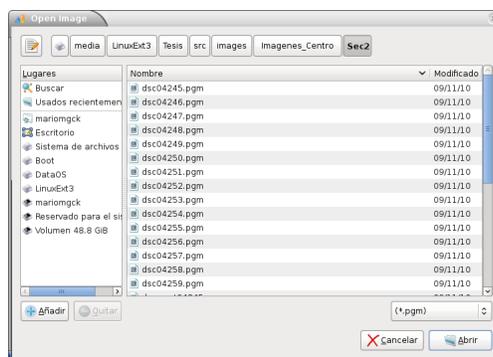


Figura 3.3: En esta imagen se muestra el menú que el sistema de software proporciona al usuario para seleccionar la secuencia de imágenes, desde un directorio de archivos.

El botón Save tiene como función guardar en un archivo de texto, los objetos geométricos dibujados en el área de dibujo. La sección Help del menú tiene como función mostrar al usuario la versión del sistema.



Figura 3.4: Sección de objetos geométricos, permite al usuario seleccionar los objetos geométricos a dibujar, estos pueden ser puntos, líneas, trayectorias abiertas y polígonos convexos.

La sección de objetos geométricos está compuesta de cuatro botones tal como se muestra en la figura 3.4, esta sección permite al usuario elegir el elemento geométrico a dibujar en el área de dibujo. El orden de los botones va de arriba hacia abajo y se presentan a continuación:

- El primer botón tiene como función habilitar la opción de dibujo de puntos característicos.



Figura 3.5: Botón para habilitar la opción de dibujo de puntos característicos.

- El segundo botón tiene como función habilitar la opción de dibujo de líneas rectas sobre la imagen.



Figura 3.6: Botón para habilitar la opción de dibujo de líneas rectas.

- El tercer botón tiene como función habilitar la opción de dibujo de trayectorias abiertas de líneas rectas.



Figura 3.7: Botón para habilitar la opción de dibujo de trayectorias abiertas de líneas rectas.

- El cuarto botón tiene como función habilitar la opción de dibujo de polígonos convexos.



Figura 3.8: Botón para habilitar la opción de dibujo de polígonos convexos.

La sección de botones de acción está compuesta de seis botones estáticos y un botón de desplazamiento. Las funciones que realiza cada botón estático son las siguientes:

- El botón “Prev” es utilizado para navegar entre la secuencia de imágenes, y se encarga de solicitar a la bicola las imágenes anteriores.



Figura 3.9: Botón para mostrar imagen previa de la secuencia.

- El botón “Next” solicita a la bicola las siguientes imágenes de la secuencia.



Figura 3.10: Botón para mostrar imagen siguiente de la secuencia.

- El botón “Compute” inicia el cálculo de las orientaciones de las superficies planares delimitadas por el usuario.



Figura 3.11: Botón para iniciar cálculo de orientaciones.

- El botón “Get Corners” inicia el cálculo de esquinas del par de imágenes visualizadas.

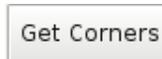


Figura 3.12: Botón para calcular esquinas.

- El botón “Match ZNCC” inicia la búsqueda de correspondencias a partir de las esquinas encontradas.

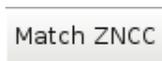


Figura 3.13: Botón para iniciar búsqueda de correspondencias.

- El botón “Fix Points” elimina todas las correspondencias que se encuentren fuera del perímetro de los polígonos convexos dibujados por el usuario.



Figura 3.14: Botón para iniciar algoritmo de cerco convexo.

El botón de desplazamiento tiene como función establecer el valor del umbral, con el cual se descartarán aquellas esquinas que no provean información de las superficies planares. Este botón puede verse en la figura 3.15.

3.3. Estimación de orientaciones desde la GUI

El procedimiento general de los experimentos, consiste en obtener un conjunto de imágenes de una escena estática que presente superficies planares, dichas imá-

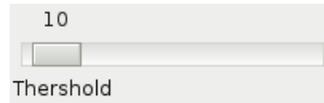


Figura 3.15: El botón de desplazamiento tiene la funcionalidad de fijar el valor del umbral para el filtrado de esquinas.

genes deben ser tomadas por una cámara en movimiento, tal que cada imagen sea una vista diferente de la escena, ver figura 3.16.



Figura 3.16: El arreglo de imágenes pertenece a una secuencia de imágenes, donde cada imagen es una vista diferente de una escena.

Esta secuencia se procede a almacenar en un dispositivo de almacenamiento para su posterior procesamiento. El movimiento de la cámara debe evitar oclusiones durante su trayectoria, es decir, las superficies planares presentes en una imagen de interés, deben tener sus correspondencias en otra imagen. La estimación de orientaciones de las superficies planares de la escena, conlleva una serie de pasos que describen a detalle en las subsecciones de este capítulo.

Selección de imágenes de prueba

Como se mencionó anteriormente, la interacción del usuario con el sistema es importante ya que es él quien determina las áreas de interés en las imágenes para que el sistema realice el cómputo correspondiente. Para ello se requiere abrir la secuencia de imágenes con la que el sistema trabajará, esto se realiza por medio

del botón Open del menú principal, ver figura 3.2, el cual permite seleccionar la secuencia de imágenes de interés, desde el directorio de archivos del dispositivo de almacenamiento. Eligiendo la primera imagen de la secuencia, el sistema de manera transparente al usuario, se encarga de cargar a memoria la secuencia de imágenes, esto lo realiza al descomponer el nombre de archivo de la primera imagen. El patrón del nombre de archivo se compone de *prefijo_índice*, el prefijo corresponde al nombre de la secuencia y el índice corresponde al número de la imagen. El sistema de software se encarga de validar el número de imágenes al obtener e incrementar el índice de la primera imagen y con ello desplegar en la ventana principal los pares de imágenes conforme se vaya requiriendo, tal y como se muestra en la figura 3.17.

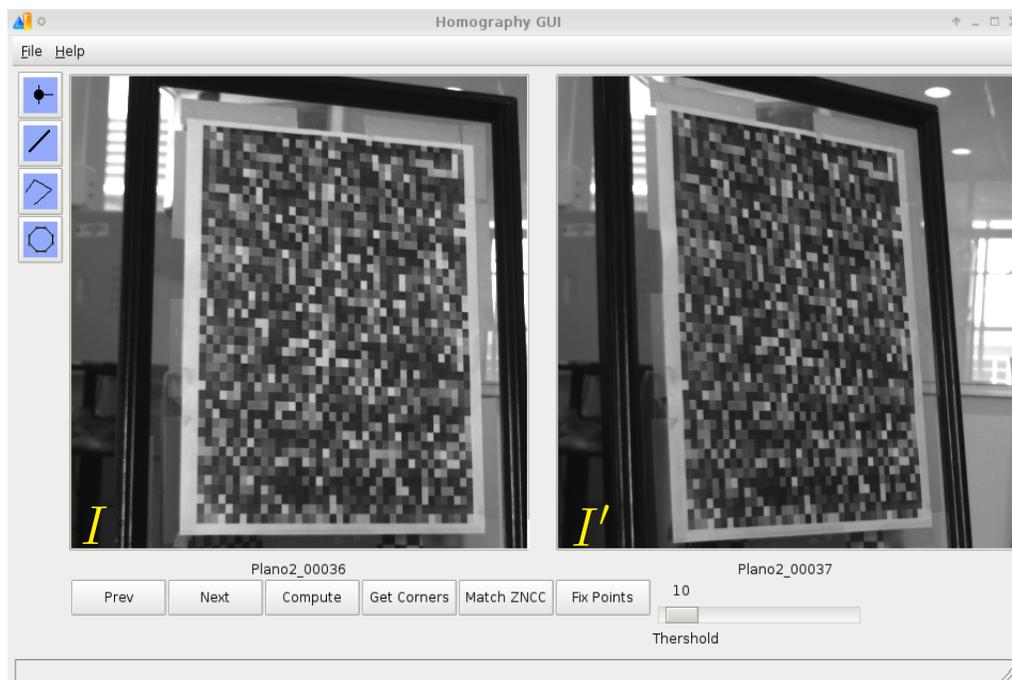


Figura 3.17: Ventana principal al cargar el primer par de imágenes de la secuencia. La imagen I corresponde a la imagen izquierda y la imagen I' corresponde a la imagen derecha.

Utilizando los botones “Next” y “Prev”, el sistema va desplegando el siguiente y el anterior par de imágenes de la secuencia, de tal forma que el usuario puede ir procesando las imágenes de par en par, se debe tener en cuenta que la imagen I

que aparece a la izquierda tiene como imagen correspondiente la imagen derecha I' .

Obtención de Esquinas

El procesamiento de las imágenes consiste en obtener un conjunto de puntos característicos, en nuestro caso esquinas, ya que el uso de esquinas proporciona una mayor robustez al momento de establecer correspondencias entre las imágenes. En nuestro caso el conjunto de esquinas se obtiene utilizando el método de Harris-Stephens, ver sección 2.3, para ello se requiere el uso de la interfaz gráfica de usuario.

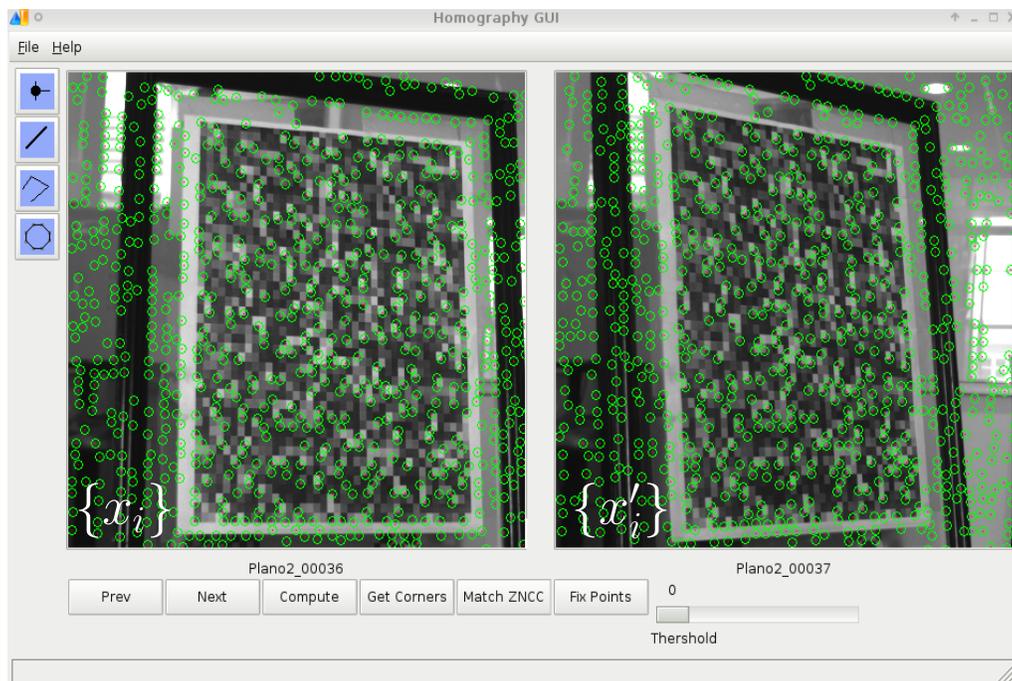


Figura 3.18: Conjunto de esquinas $\{x_i\}$ y $\{x'_i\}$ obtenidas para cada imagen, aplicando el método de Harris.

Una vez que se seleccionaron las imágenes de prueba, tal como se describió anteriormente, el siguiente paso consiste en obtener un conjunto de esquinas $\{x_i\}$ y $\{x'_i\}$ para las imágenes I e I' , respectivamente. Para obtener las esquinas basta con realizar un clic sobre el botón “Get Corners” de la GUI, el cual se

encarga de ejecutar los algoritmos encargados de aplicar el método de Harris a las imágenes I e I' , posteriormente de manera automática, el sistema se encarga de dibujar sobre las imágenes las esquinas obtenidas por los algoritmos, tal y como se muestra en la figura 3.18.

Teniendo las esquinas de las imágenes I e I' , el usuario puede validar la existencia de esquinas que no aporten información de las superficies planares de interés, a manera de ejemplo, para la figura 3.18 la superficie planar de interés Π_E , es el patrón de cuadros con valores de intensidad aleatorios, de manera clara se puede observar que las esquinas del marco y el fondo de la imagen no aportan información relevante.

Para hacer una búsqueda de correspondencias robusta, se hace uso de un umbral t_e para descartar aquellas esquinas que no aporten información de la superficie de interés. La validación la realiza el usuario al determinar el valor del umbral al ir cambiando de posición el botón de desplazamiento, esto hace que conforme vaya variando el umbral, el sistema oculte aquellas esquinas x_i y x'_i , cuya medida de *esquinidad* se encuentren debajo del valor del umbral, el botón de desplazamiento puede verse en la figura 3.15.

Búsqueda de correspondencias

Al tener un valor fijo del umbral t_e el siguiente paso de la estimación consiste en realizar la búsqueda de correspondencias con aquellas esquinas que no se encuentran ocultas, esto se realiza al hacer clic sobre el botón “Match ZNCC”, el cual se encarga de ejecutar los algoritmos de búsqueda que implementan el método ZNCC. Al finalizar la búsqueda el sistema descarta aquellas esquinas que no tuvieron correspondencias, esto se puede ver en la figura 3.19.

Al terminar el proceso de búsqueda de correspondencias (x_i, x'_i) , es posible encontrar ciertas regiones en la superficie fácilmente identificables en las cuales haya poca densidad de correspondencias, y por consiguiente hay menor informa-

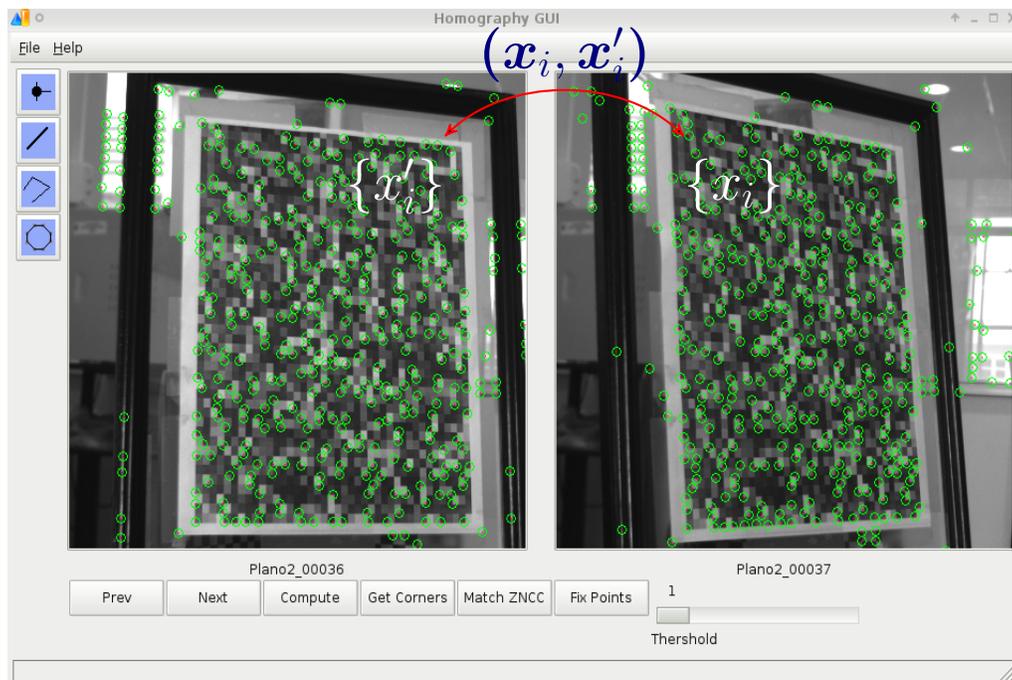


Figura 3.19: Conjunto de correspondencias establecidas mediante el método ZNCC.

ción de la superficie planar, ante esta situación el usuario puede manualmente agregar correspondencias que le sean fácil de identificar, esto lo puede realizar utilizando la sección de objetos geométricos de la GUI.



Figura 3.20: Botón para habilitar la opción de dibujo de puntos característicos.

Para agregar correspondencias manualmente se hace clic sobre el botón que habilita la opción de dibujo de puntos, ver figura 3.20, posteriormente se hace clic al botón izquierdo del cursor sobre algún punto x de la imagen I , ante esta acción el sistema dibuja un punto característico en las coordenadas de punto x , seguidamente se hace clic sobre la imagen I' para dibujar su punto correspondiente x' , es en ese instante cuando el sistema agrega una correspondencia (x, x') al conjunto de correspondencias previamente obtenido. El usuario puede incrementar la densidad de correspondencias según lo considere necesario, a fin

de tener mayor información de la superficie planar.

Delimitación de las superficies

El siguiente paso de la estimación consiste en dibujar polígonos convexos \mathcal{P}_j y \mathcal{P}'_j en cada imagen, $j \in [1, 2, \dots, n]$, donde cada polígono delimita una región que contenga una superficie planar, por consiguiente al momento de dibujar un polígono \mathcal{P}_j en la imagen izquierda, es necesario que tenga su correspondiente polígono \mathcal{P}'_j en la imagen derecha. Para este ejemplo se trata de una sola superficie planar, por lo cual cada imagen tiene un polígono, tal y como se muestra en la figura 3.21. Por lo tanto, el siguiente paso consiste en delimitar la superficie planar por medio de polígonos en las imágenes, para ello se hace clic sobre el botón que habilita la opción de dibujo de polígonos convexos, lo cual le permitirá al usuario dibujar libremente uno o más polígonos convexos.

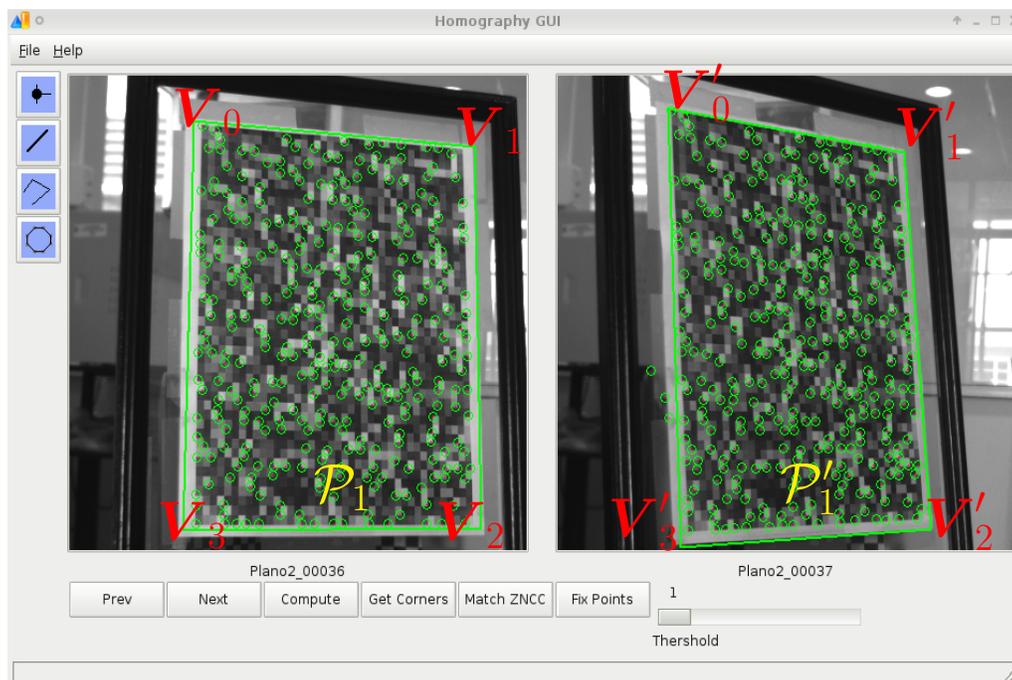


Figura 3.21: Cada una de las imágenes muestra un polígono convexo \mathcal{P} definido por un conjunto de vértices V_i , \mathcal{P} delimita la superficie planar de interés.

El proceso de dibujo tiene como primer paso hacer clic con el botón izquierdo

del cursor sobre algún punto x_0 en la imagen I , ante esta acción del usuario, el sistema toma las coordenadas del punto x_0 para fijar el primer vértice V_0 del polígono, posteriormente conforme el usuario mueve el cursor el sistema va dibujando una trayectoria rectilínea, y hasta que nuevamente haga clic con el botón izquierdo sobre otro punto x_1 , el sistema agrega un nuevo vértice V_1 en el punto x_1 , y es cuando se dibuja un segmento de línea recta que pasa por el primer y segundo vértice, de esta manera se pueden ir agregando los vértices que definan al polígono que se esté dibujando. Por último para indicar al sistema que se han agregado todos los vértices del polígono, solo se requiere hacer clic con el botón derecho del cursor sobre cualquier punto de la imagen I . Los vértices deben ser fijados en el sentido de las manecillas del reloj tal y como se muestra en la figura 3.22. El mismo proceso se repite para dibujar el polígono \mathcal{P}'_j de la imagen I' .

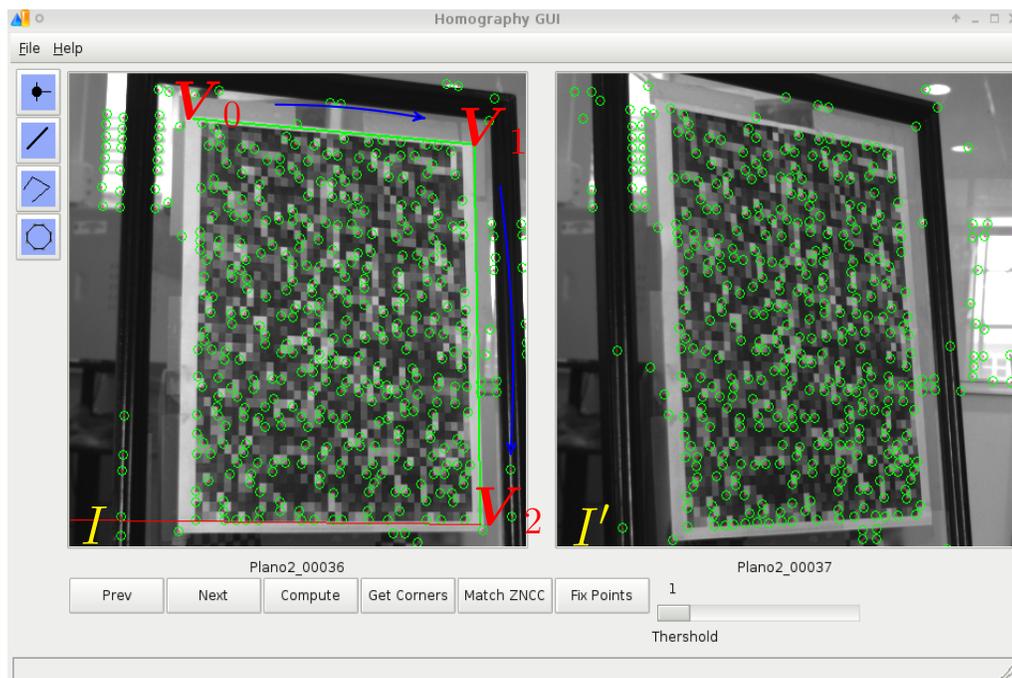


Figura 3.22: Esta imagen muestra el sentido en que se genera el polígono convexo que delimita la superficie planar.

Teniendo la superficie planar delimitada en ambas imágenes, entonces se pro-

cede a eliminar aquellas correspondencias que se encuentren fuera del perímetro de ambos polígonos, para ello se hace clic sobre el botón “Fix Points”, véase figura 3.23, el cual se encarga de ejecutar el algoritmo de cerco convexo definido en el cuadro 2.2. Al finalizar el sistema muestra en la ventana principal solo aquellas correspondencias (x_i, x'_i) cuyos puntos cumplen que $x_i \in \mathcal{P}_1$ y $x'_i \in \mathcal{P}'_1$, el resultado al aplicar el algoritmo puede verse en la figura 3.21.



Figura 3.23: Botón para iniciar algoritmo de cerco convexo.

Obtención de orientaciones

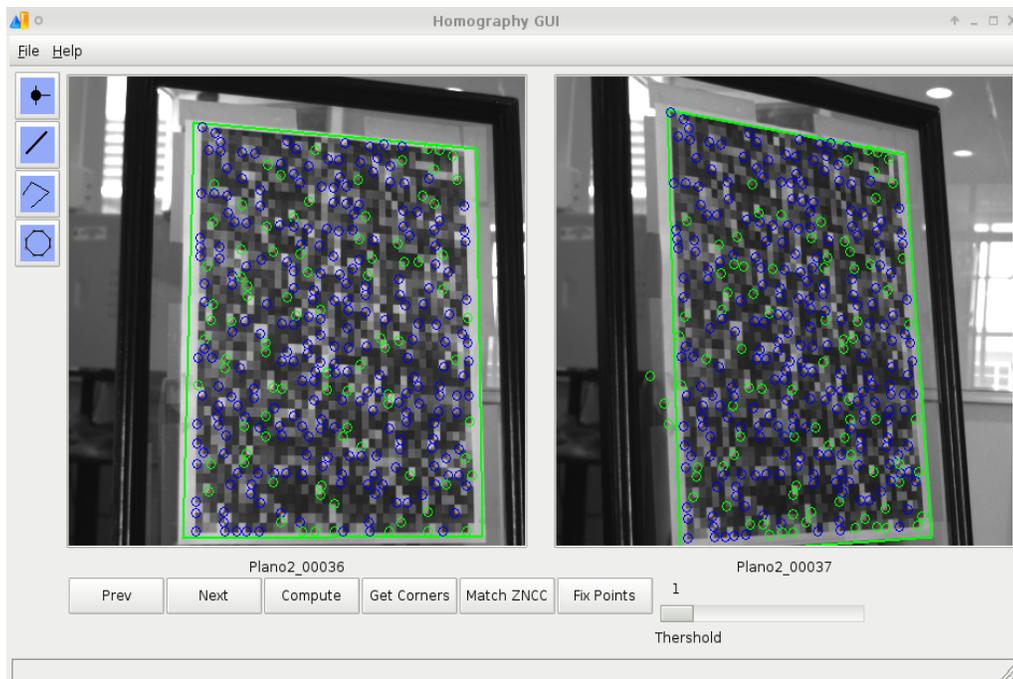


Figura 3.24: Imagen al finalizar la estimación del vector de orientación de la superficie planar, el sistema dibuja los puntos típicos de color azul y los puntos atípicos de color verde.

Al delimitar la superficie planar de interés se obtiene un conjunto de correspondencias asociadas a la superficie, con ello se procede a calcular la orientación, para

ello se hace clic sobre el botón “Compute”, el cual se encargará de ejecutar el algoritmo definido en el cuadro 2.6, el cual aplica los métodos *Levenberg–Marquardt* y RANSAC para la estimación de la homografía \hat{H}_L que relaciona los planos delimitados por los polígonos. Mediante la descomposición de la homografía, se estiman las dos posibles soluciones que nos proporcionan la orientación de la superficie planar.

Al concluir el algoritmo el sistema muestra en la ventana principal los puntos típicos y atípicos obtenidos. Los puntos típicos se muestran en color azul y los atípicos se muestran en color verde, tal y como se muestra en la figura 3.24, la cual corresponde al ejemplo planteado anteriormente.

Finalmente el sistema se encarga de guardar en un archivo de texto las orientaciones estimadas, para que posteriormente puedan ser visualizadas en algún software matemático, como puede ser Matlab o Gnuplot.

A manera de resumen para facilitar la comprensión de la metodología planteada, se presenta en la figura 3.25 un diagrama de flujo que muestra los pasos descritos anteriormente.

En el siguiente capítulo se muestran los experimentos realizados a diferentes superficies planares, siguiendo la metodología desarrollada en este capítulo.

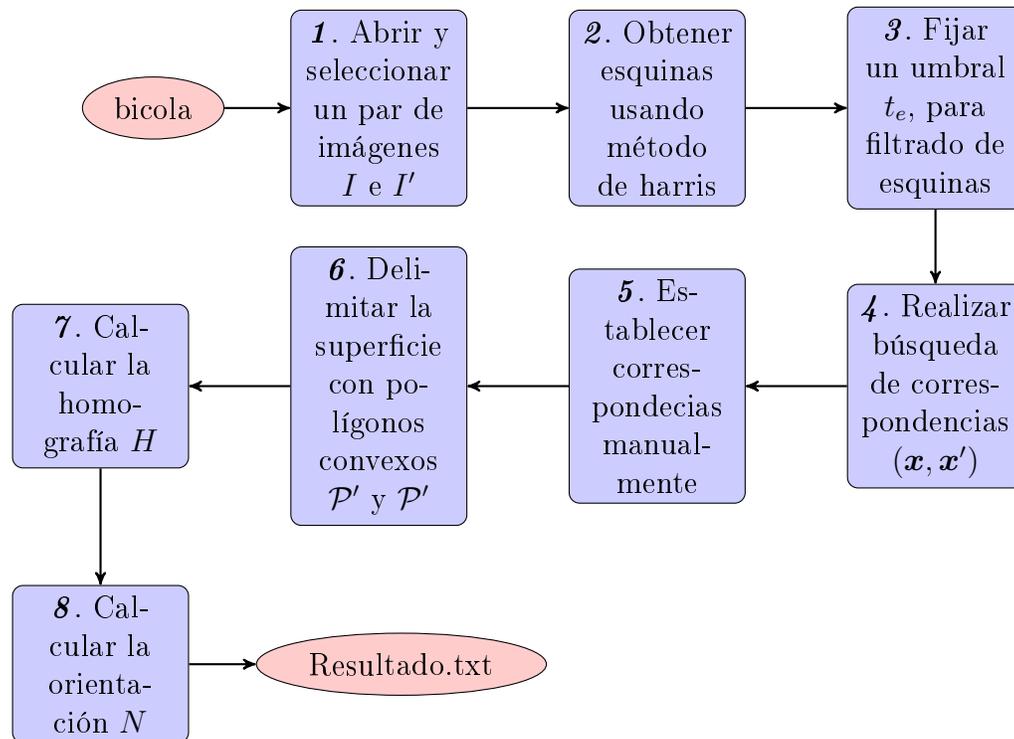


Figura 3.25: El proceso para obtener la orientación de una superficie planar, a partir de un par de imágenes, se muestra a través del diagrama de flujo, donde los pasos 2, 4, 7 y 8 son realizados por el sistema, los pasos 1, 3, 5 y 6 los realiza el usuario al interactuar con la GUI.

3.4. Resumen

En este capítulo se explicó la funcionalidad de la GUI, la cual permite al usuario interactuar con el sistema para realizar la delimitación de la superficie planar de interés, la cual estima las posibles orientaciones con respecto a un marco de referencia. También se explicó cómo los conceptos del marco teórico son empleados para el desarrollo de la metodología, la cual se emplea en los diferentes experimentos de este trabajo de tesis.

Capítulo 4

Experimentos

4.1. Introducción

En este capítulo se presentan los diferentes experimentos que se llevaron a cabo para probar el método de estimación de orientaciones de superficies planares ante diferentes escenarios de prueba. Para llevarlos a cabo se aplica la metodología definida en el capítulo 3, que de manera resumida se presenta en el cuadro 3.25.

En total se realizaron dos experimentos, el primer de ellos fue realizado bajo condiciones controladas y el segundo fue realizado en calles del centro de la ciudad de Mérida. Para la ejecución del sistema se usó una computadora Dell de 3 Ghz, con 4GB de memoria RAM, con sistema operativo Linux en su versión Debian 2.6.26-2-amd64. La matriz de calibración K se obtuvo mediante la herramienta [Camera Calibration Toolbox for Matlab](#) utilizando un patrón de calibración;

Para obtener esquinas el sistema emplea el método de [Hartley y Zisserman \(2004\)](#), utilizando una ventana Gaussiana de 5×5 píxeles y varianza $\sigma^2 = 1$, con la constante de Harris $k = 0.04$. Para filtrar las falsas esquinas se utilizó una ventana de supresión de valores no máximos de 7×7 píxeles, además se incluye el umbral que el usuario fija desde la GUI para filtrar un mayor número de esquinas.

Los valores de los parámetros utilizados para RANSAC para el cálculo de la homografía fueron los siguientes: un umbral $t = 5.99\sigma^2$ con $\sigma = 1$, un error de

$\epsilon = 0.5$ y una probabilidad $p = 0.99$. El valor 5.99 es el factor de ajuste empleado para el cálculo de homografías mostrado en [Hartley y Zisserman \(2004\)](#). Los resultados de experimentación se muestran en las siguientes secciones de este capítulo.

4.2. Escena con condiciones controladas

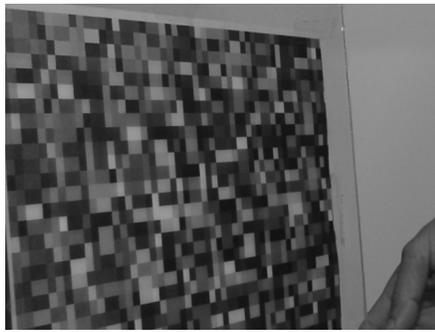
Este primer experimento se realizó en un laboratorio bajo condiciones controladas. Las condiciones que se controlaron fueron la luz ambiental y la posición de los elementos de la escena. Los elementos de la escena de prueba fueron: un plano montado sobre una base plana rotatoria y una cámara montada sobre un tripie.

La cámara se montó sobre el tripie a una altura del piso ligeramente mayor que la altura de la base rotatoria, la cual giraba sobre un eje ligeramente paralelo al eje y de la lente de la cámara. La base rotatoria se encontraba al frente de la lente de la cámara. La cámara se mantuvo estática durante el desarrollo de todo el experimento. Las imágenes fueron obtenidas por medio de una cámara de uso industrial SONY DFW-X710 con una lente de 8 mm, las cuales se almacenaron en formato PGM a una resolución de 640×480 píxeles.

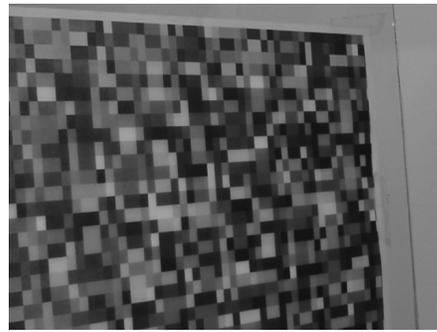
La superficie planar de la cual se deseaba obtener su orientación, fue montada sobre la base rotatoria presentando cierta inclinación hacia la cámara. El movimiento de la base fue regulado manualmente mientras se capturaban las imágenes de la superficie.

La textura de la superficie fue generada de manera artificial utilizando un patrón de cuadros, con valores de intensidad aleatorios, el cual permitió obtener una mayor homogeneidad de esquinas, aumentando de esta manera el número de correspondencias.

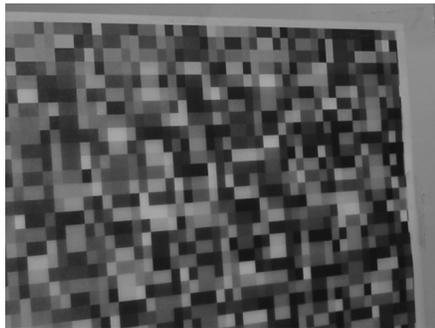
En total se seleccionaron once imágenes de toda la secuencia, en las cuales



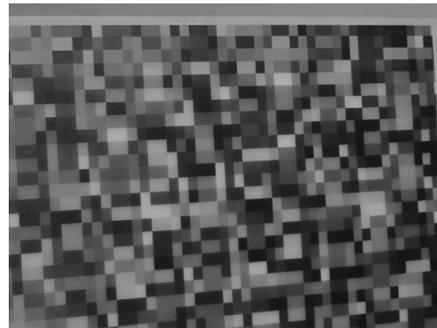
(a) Primera imagen de la secuencia



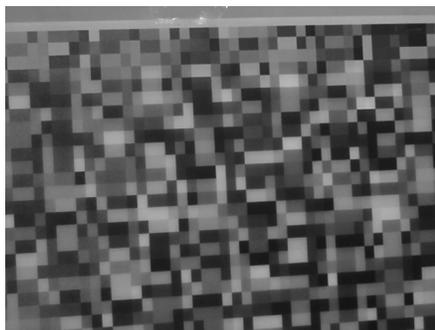
(b) Segunda imagen de la secuencia



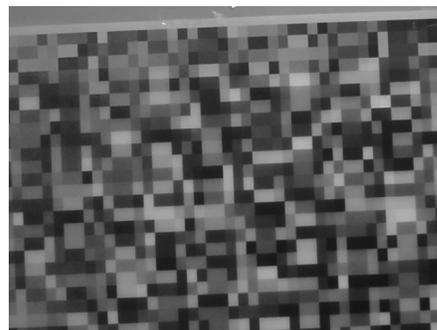
(c) Tercera imagen de la secuencia



(d) Cuarta imagen de la secuencia



(e) Quinta imagen de la secuencia



(f) Sexta imagen de la secuencia

Figura 4.1: En este arreglo de imágenes se muestran las seis primeras imágenes de las once seleccionadas para el experimento.

la superficie planar difiere en su orientación de manera considerable. En el la figura 4.1 se muestran las seis primeras imágenes de las once seleccionadas.

Para el filtrado de esquinas se utilizó un umbral $t_e = 0$, y se establecieron las correspondencias con los parámetros descritos al inicio del capítulo. Para la búsqueda de correspondencias utilizando la métrica ZNCC, la cual se mencionó en la sección 2.3, el umbral fijado para el filtrado de correspondencias fue de -1 y

1, la dimensión de la ventana de comparación, fue de 1/30 del ancho de la imagen por 1/30 del alto de la imagen; la dimensión del área de búsqueda, fue de 1/3 del ancho de la imagen por 1/3 del alto de la imagen.

Un resumen de los resultados obtenidos se muestran en el cuadro 4.1, donde para cada par de imágenes se proporciona el porcentaje de puntos típicos y atípicos, el número de correspondencias(NC) obtenidas y se muestra el error simétrico(ES) que produce la homografía que relaciona la superficie planar entre las imágenes.

Imágenes	Puntos típicos	Puntos atípicos	NC	ES
1 - 2	50.64 %	49.35 %	847	0.81
2 - 3	50.82 %	49.17 %	964	1.22
3 - 4	59.34 %	40.65 %	979	1.17
4 - 5	49.53 %	50.46 %	961	0.34
5 - 6	60.66 %	39.33 %	999	0.43
6 - 7	57.17 %	42.83 %	1053	0.45
7 - 8	56.18 %	43.81 %	1100	0.46
8 - 9	52.17 %	47.83 %	1106	0.62
9 - 10	45.97 %	54.02 %	931	0.22
10 - 11	45.07 %	54.92 %	741	0.75

Cuadro 4.1: Para cada par de imágenes se muestra el porcentaje de puntos típicos y atípicos obtenidos, se muestra el total de correspondencias(NC) obtenidas, y se muestra el error simétrico(ES) que presenta la homografía que relaciona a la superficie planar entre cada par de imágenes.

La matriz de calibración utilizada en este experimento para obtener la homografía definida en la ecuación 2.19, se muestra a continuación:

$$K = \begin{bmatrix} 1795.65861 & 0 & 266.23073 \\ 0 & 1790.93743 & 217.39366 \\ 0 & 0 & 1 \end{bmatrix}.$$

Debido al movimiento de la superficie, la trayectoria de su vector normal N presenta cierta coplanaridad, por lo cual se puede considerar que el vector normal se mueve sobre un plano, es por ello que las soluciones que presenten mayor coplanaridad son las soluciones que mejor se ajustan al movimiento del vector

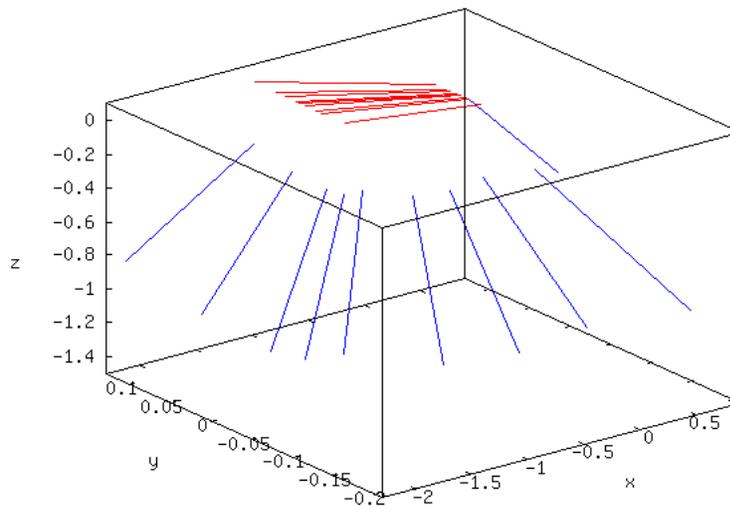


Figura 4.2: Orientaciones relativas de la superficie planar a diferentes perspectivas conforme se produjo el movimiento de la superficie.

normal a la superficie, el cual nos proporciona la orientación de la superficie conforme cambiaba su posición al frente de la cámara. En total se obtuvieron veinte soluciones relativas de las diferentes orientaciones de la superficie, dos soluciones relativas al marco de referencia de la primera de imagen del par, por cada par de imágenes. Para determinar cuáles soluciones presentan coplanaridad se utilizó el algoritmo de coplanaridad descrito en el apéndice A.3 y así generar la gráfica de las mismas, donde la vista de la gráfica fue centrada en el observador.

Las orientaciones de la superficie se muestran en la figura 4.2, las soluciones que presentaron una mayor coplanaridad aparecen en rojo y las que presentaron menor coplanaridad se muestran en color azul. Sin embargo las soluciones con mayor coplanaridad no presentaron una orientación significativa hacia la lente de la cámara, es decir hacia el eje z , y las que tuvieron menor coplanaridad son las mejor se ajustaron al movimiento que tuvo la superficie.

Discusión

Al realizar la descomposición de la homografía no se conoce el vector de traslación T , que nos indica que tanto se movió la cámara de una imagen a otra. En el caso de este experimento la cámara se mantuvo estática y únicamente la superficie planar es la que presenta movimiento de rotación, es por ello que se asume que el vector normal a la superficie se mueve sobre un plano, y por ello las soluciones presentan cierta coplanaridad. La adaptación del algoritmo de RANSAC nos proporcionó buenos resultados al dividir en dos subconjuntos las soluciones obtenidas, así se determina cuáles soluciones son las que mejor se ajustan al movimiento de la superficie planar. Es importante recordar que la orientación que se obtiene al descomponer la homografía es relativa entre las vistas, esto es, la orientación del vector normal calculado entre un par de imágenes, esta descrito en el marco de referencia de la cámara cuando se capturó la primera imagen. Aún así, en éste caso la coplanaridad se debe observar.

4.3. Escena urbana (Centro Histórico)

Este segundo experimento se realizó en las calles del centro histórico de la ciudad de Mérida, Yucatán. El experimento consistió en elegir una calle que presentara edificios ricos en textura, los cuales se fotografiaron mientras la cámara se encontraba en movimiento, manteniendo la orientación de la lente de la cámara hacia las paredes de los edificios.

Las imágenes fueron obtenidas por medio de una cámara de uso profesional SONY Cybershot DSC-F828 con una lente de 7.1 mm, las cuales se almacenaron en formato JPG, y posteriormente se convirtió a formato PGM a una resolución de 2592×1944 píxeles.

En total se seleccionaron las siete primeras imágenes de toda la secuencia, a las cuales se les corrigió su distorsión radial. En la figura 4.3 se muestran las



(a) Primera imagen de la secuencia



(b) Segunda imagen de la secuencia



(c) Tercera imagen de la secuencia



(d) Cuarta imagen de la secuencia



(e) Quinta imagen de la secuencia



(f) Sexta imagen de la secuencia

Figura 4.3: En este arreglo de imágenes se muestran las seis primeras imágenes de la secuencia de la escena urbana experimental.

seis primeras imágenes de la secuencia, donde la superficie planar de interés corresponde al plano sobre el cual se encuentran las ventanas del edificio. A diferencia del experimento anterior, en este experimento las correspondencias se realizaron de forma manual, estableciendo en total veinte y ocho correspondencias por cada par de imágenes. Esto se realizó debido a que la métrica ZNCC no es lo suficiente robusta para los cambios de perspectiva entre las imágenes, los cuales

están presentes en las imágenes de este experimento.

La matriz de calibración utilizada en este experimento para obtener la homografía definida en la ecuación 2.19, se muestra a continuación:

$$K = \begin{bmatrix} 2162.62150 & 0 & 1251.28059 \\ 0 & 2167.51028 & 968.06341 \\ 0 & 0 & 1 \end{bmatrix}.$$

Debido a que la cámara se encontraba en movimiento, el cálculo de las orientaciones se realizó con respecto a la primera imagen, con el fin de tener un marco de referencia en común entre las imágenes.

Imágenes	Puntos típicos	Puntos atípicos	NC	ES
1 - 2	85.71 %	14.28 %	28	0.60
1 - 3	75 %	25 %	28	0.26
1 - 4	60.71 %	39.28 %	28	0.48
1 - 5	53.57 %	46.42 %	28	0.64
1 - 6	46.42 %	53.57 %	28	0.01
1 - 7	35.71 %	64.28 %	28	0.02

Cuadro 4.2: Para cada par de imágenes se muestra el porcentaje de puntos típicos y atípicos obtenidos, se muestra el total de correspondencias(NC) obtenidas, y se muestra el error simétrico(ES) que presenta la homografía que relaciona a la superficie planar entre cada par de imágenes.

Un resumen de los resultados obtenidos se muestran en el cuadro 4.2, donde para cada par de imágenes se proporciona el porcentaje de puntos típicos y atípicos, el número de correspondencias(NC) obtenidas y se muestra el error simétrico(ES) que produce la homografía que relaciona la superficie planar entre las imágenes.

En total se obtuvieron doce soluciones de las diferentes orientaciones de la superficie, dos por cada par de imágenes; mismas que se graficaron y se muestran en la figura 4.3.

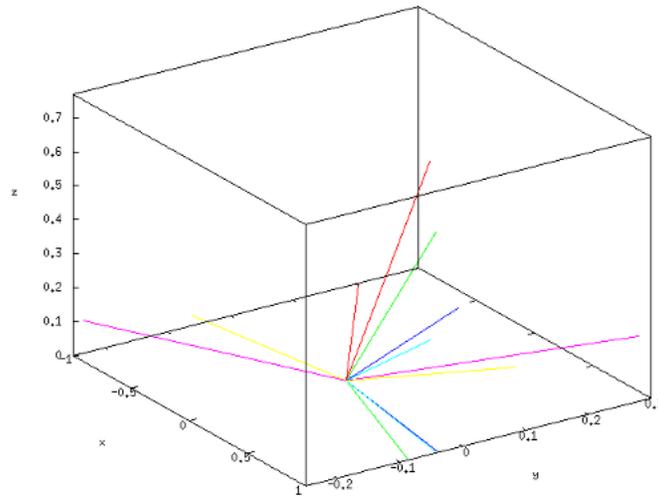


Figura 4.4: Orientaciones relacionadas al marco de referencia de la primera imagen, vistas desde la superficie planar, donde el vector que aparece en color rojo corresponde a la primera orientación de la superficie.

Discusión

El desarrollo de este experimento involucró imágenes generadas por una cámara en movimiento en una escena urbana, con las cuales se obtuvieron las soluciones de las orientaciones de la pared del edificio conforme la cámara se movía. Al desconocer el vector de traslación y al estar la cámara en movimiento se puede observar que las soluciones no presentan una coplanaridad significativa, por lo cual el código de coplanaridad no ofrece buenos resultados, una posible solución se presenta en [Faugeras y Lustman \(1988\)](#), pero su implementación queda fuera de los objetivos de este trabajo de tesis. Una observación importante del experimento es que conforme aumenta la diferencia en perspectiva de la primera imagen con las demás, el número de puntos típicos disminuye, por consiguiente es posible que mientras mayor sea la diferencia entre las imágenes el número de puntos atípicos sea menor.

Capítulo 5

Conclusiones

5.1. Sumario

El trabajo presentado en esta tesis trató sobre la obtención de información geométrica de superficies planares presentes en una escena urbana, a partir de una secuencia de imágenes, combinando técnicas de visión computacional con la percepción geométrica de una persona. La orientación relativa de las superficies planares es el elemento importante que nos proporciona información geométrica de la superficie y la obtención de la orientación fue el tema central de este trabajo de tesis.

La revisión de la literatura nos permitió seleccionar un conjunto de métodos y algoritmos para llevar a cabo la estimación de orientación de superficies planares, en escenas urbanas y controladas, estableciendo con ello una metodología para llevar a cabo el cálculo de la orientación de las superficies planares.

Uno de los objetivos de la tesis involucró el desarrollo de un sistema de software interactivo, el cual fue empleado en la realización de los experimentos. En él se implementaron los algoritmos definidos en el marco teórico, anexando otras rutinas para el diseño y manejo de eventos de la interfaz gráfica de usuario. Este sistema de software además de permitir interactividad con el usuario, logró integrar adecuadamente las bibliotecas VisionLibs y WxWidgets 2.8, obteniendo

con esto una aplicación de software que puede servir como base para el desarrollo de otros sistemas de software.

El patrón de diseño MVC, permitió ajustar los parámetros para el cálculo de esquinas y búsqueda de correspondencias, sin afectar al resto del sistema, por lo cual el sistema de software ganó robustez y flexibilidad ante los cambios que se realizaron en los dos experimentos. La herramienta Camera Calibration Toolbox for Matlab simplificó el cálculo de las matrices de calibración, cuyo error de reproyección fue pequeño, las cuales fueron utilizadas en los experimentos realizados.

El análisis realizado al primer experimento, se concluye que al tener una cámara estática y una superficie planar en rotación, las soluciones de las orientaciones presentan coplanaridad; utilizando esa característica geométrica se pueden emplear métodos geométricos que empleen alguna métrica para determinar que tan coplanares son las soluciones, ya que al desconocer el vector de traslación que relaciona a los pares de imágenes, nos es posible determinar de manera directa cuáles son las soluciones que mejor proporcionan las orientaciones de la superficie, por tal razón empleando una métrica de coplanaridad nos permite tener una primera aproximación para obtener las mejores soluciones.

En el segundo experimento, se pudo observar que sí la cámara se encuentra en movimiento las soluciones obtenidas por medio de la descomposición homográfica, no presentan una coplanaridad deseable y al conocer únicamente la dirección del vector de traslación, no es posible establecer una relación geométrica únicamente con alguna métrica de coplanaridad, ya que se requiere conocer la posición que tuvieron las cámaras. Por lo cual se requiere de una metodología diferente para obtener las mejores soluciones que nos proporciona la orientación de la superficie planar, conforme la cámara se encontraba en movimiento.

En conclusión la descomposición homográfica nos proporciona una buena aproximación de la orientación de la superficie planar, sin embargo al tener múltiples soluciones se requiere encontrar aquellas que mejor se ajusten a las orientaciones

de la superficie planar, por lo cual es necesario establecer una nueva metodología que nos permita determinar dichas soluciones.

5.2. Trabajo futuro

Concluimos indicando algunas de las áreas más importantes de este trabajo donde consideramos se pueden realizar futuras modificaciones que ayudarían a mejorar los resultados y la eficiencia del proceso de obtención de orientaciones de superficies planares.

- Primero, realizar una investigación acerca de la relación geométrica que exista entre las diferentes soluciones, con el fin de establecer una metodología para determinar cuáles de las soluciones estiman con mayor precisión la orientación de la superficie planar. La revisión realizada a la literatura se observó que no hay muchos trabajos que discutan este problema con claridad, por lo cual realizar una investigación sobre dicho problema es ampliamente enriquecedor.
- Segundo, los experimentos realizados se realizaron con una escena estática, empleando una cámara cuya matriz de calibración se conoce, valdría la pena realizar una investigación en la cual se involucren escenas en movimiento, en las cuales puedan haber oclusiones y no sea necesario conocer la matriz de calibración de la cámara.
- El resultado de los experimentos se obtuvieron a posteriori de la realización de los mismos, por lo cual realizar una investigación para obtener orientaciones de superficies en tiempo real, resultaría ampliamente enriquecedor para resolver problemas de navegación en ambientes urbanos, empleando algún robot móvil.
- Por último, un área interesante que va teniendo auge en los últimos años, es la computación paralela. Debido a que los algoritmos empleados en este

trabajo de tesis en su mayoría son paralelizables, lo cual conllevaría un gran aumento en la eficiencia de todo el proceso de la obtención de orientaciones.

Bibliografía

- Briceño Coronado, A. A. (2005). Reconstrucción métrica a través de la recuperación de la geometría epipolar. Tesis de Licenciatura, Facultad de Matemáticas, UADY.
- Faugeras, O. y Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. Technical report.
- Fischler, M. y Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Forsyth y Ponce (2004). *Computer Vision A Modern Approach*. Springer.
- Harris, C. y Stephens, M. (1988). A combined corner and edge detector. En *Proceedings of the Alvey Vision Conference, University of Manchester*, páginas 147–151. The Plessey Company.
- Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.
- Hartley, R. I. y Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN:0521549518.
- Heath, M. T. (1997). *SCIENTIFIC COMPUTING An Introductory Survey*. McGraw-Hill, Inc.
- Hengel, A. V. D., Dick, A. R., Thormählen, T., Ward, B., y Torr, P. H. S. (2007). Videotrace: rapid interactive scene modelling from video. *ACM Trans. Graph.*, 26(3).
- Kanatani, K. (1993). *Geometric Computation for Machine Vision*. Number 37 en Oxford engineering Science Series. Oxford Science Publications.
- O’Rourke, J. (1998). *COMPUTATIONAL GEOMETRY IN C*. Cambridge.
- Soatto, S., Kosecka, J., Ma, Y., y Sastry, S. S. (2004). *An Invitation to 3-D Vision*. Springer.
- Teukolsky, S. A., Vetterling, W. T., Press, W. H., y Flannerly, B. P. (1992). *Numerical Recipes in C*. Cambridge.

Woods, R. E. y Gonzalez, R. C. (2002). *Digital Image Processing*. Prentice Hall.

Apéndice A

Implementación del sistema

A.1. Cola bidireccional

Al trabajar con una secuencia de imágenes resulta impráctico cargar en memoria toda la secuencia, ya que el espacio en memoria es finito y solo se trabaja con un determinado número de imágenes en un instante de tiempo, por lo cual es necesario tener una estructura de datos que permita almacenar aquellos objetos con los que se estará trabajando, en este caso imágenes. Una cola bidireccional o bicola es una estructura de datos, caracterizada por ser una secuencia de elementos, donde se permiten la inserción y eliminación de elementos en ambos extremos de la cola. A menos que se indique lo contrario las bicolas se mantienen en un arreglo circular, con apuntadores IZQ y DER, que apuntan a los dos extremos de la bicola, asumiendo que los elementos se encuentran en el extremo izquierdo y el extremo derecho.

Las operaciones básicas de la bicola se encuentran:

- **Insertar al final** Almacena al final de la cola el elemento que se recibe como parámetro.
- **Insertar al frente** Almacena al frente de la cola el elemento que se recibe como parámetro.

- **Eliminar al frente** Saca de la cola el elemento que se encuentra al frente.
- **Eliminar al final** Saca de la cola el elemento que se encuentra al final.
- **Vacía** Regresa un valor booleano indicando si la cola tiene o no elementos.
- **Llena** Regresa un valor booleano indicando si la cola tiene espacio disponible para insertar nuevos elementos.

El uso de colas circulares como la bicola, ofrece un tiempo de recorrido menor de $\Omega(n)$ si la bicola tiene n elementos, por lo cual resulta eficiente al recorrer la secuencia de elementos.

A.2. Modelo Vista Controlador

La arquitectura de software Modelo-Vista-Controlador, actualmente es considerada como un patrón de diseño usado en la ingeniería de software. El patrón separa la lógica de aplicación(datos del sistema), de la interfaz de usuario(entradas y presentación de datos), permitiendo desarrollarlas, aplicarles pruebas de calidad y dar mantenimiento independiente a cada una de ellas.

El modelo es una capa abstracta que se encarga de gestionar la información y de atender notificaciones cuando cambia la información. El modelo es la representación de un dominio específico de los datos con los que la aplicación funciona. La lógica de dominio añade significado a los datos en bruto (por ejemplo, el cálculo de impuestos y gastos de envío para los artículos comprados). Cuando un modelo cambia su estado(modificación de datos), se notifica a los dominios asociados para que puedan actualizar su información.

Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente como una base de datos para almacenar datos o una estructura de datos. MVC no menciona específicamente la capa de acceso a datos, ya que se entiende que se encuentra encapsulado en el modelo. Para el caso la cola circular esta ignora

el tipo de datos que debe almacenar, solamente tiene la lógica de como debe ir accediendo a ellos.

La vista es la capa que se encarga de presentar los datos de una forma adecuada para su interacción, por lo general se utiliza una interfaz gráfica de usuario para presentar los datos. Pueden existir múltiples maneras de presentar los datos de un modelo único para diferentes propósitos. Una vista general, tiene una correspondencia uno a uno con una superficie de pantalla y sabe cómo representar a la misma.

El controlador es el encargado de procesar las entradas provenientes de la vista, y responde ante dichas entradas mediante objetos que obtiene del modelo. Un controlador se encarga de dar lógica a los eventos que se producen en la vista, para dar instrucciones al modelo con base al evento producido. Por ejemplo si ocurre el evento de hacer clic sobre un botón de la GUI, el controlador tiene la lógica de qué debe realizar ante ese evento, como puede ser iniciar un cálculo numérico con ciertos datos que el modelo posee.

A.3. Código de coplanaridad

Al tener el conjunto de soluciones de las orientaciones de la superficie, se tuvo el problema de determinar cuáles son las soluciones que presentan coplanaridad. Para resolver el problema se realizó una adaptación del algoritmo de RANSAC [Fischler y Bolles \(1981\)](#), a fin de particionar en dos subconjuntos el conjunto de soluciones, según la coplanaridad que presenten las soluciones. El algoritmo elige aleatoriamente tres soluciones a las cuales se les aplica el modelo comparación, el cual consiste en realizar el cálculo de la matriz de momentos para obtener una métrica que mida la coplanaridad de las soluciones. La métrica consiste en obtener la razón entre el segundo y primer eigenvalor de la matriz de momentos, si esta razón es mayor que cierto umbral, entonces las soluciones elegidas son coplanares, véase en [Kanatani \(1993\)](#). El algoritmo fue implementado en Octave

y se presenta a continuación.

```

2  N = [N1_1,N1_2,N1_3,N1_4,N1_5,N1_6,N1_7,N1_8,N1_9,N1_10,N2_1,
3      N2_2,N2_3,N2_4,N2_5,N2_6,N2_7,N2_8,N2_9,N2_10];
4
5  nv = length(N);
6
7  %%Umbral superior indica el valor minimo que debe tener la
8      razon entre el eigenvalor menor de todos y el segundo
9      eigenvalor menor
10
11  UmSup = 1000000; %%Umbral superior
12  UmDot = 0.01; %%Umbral del producto punto
13  cont = 0;
14  isCoplanar = 0;
15
16  while (cont < nv/2)
17
18      isCoplanar = 0;
19
20      while isCoplanar == 0
21          %%Calculamos matriz de momento de tres vectores
22          escogidos aleatoriamente
23
24          a=rand(nv,1);
25          [a, I] = sort(a);
26          m = N(:,I(1)) * N(:,I(1))' + N(:,I(2)) * N(:,I(2))' + N
27              (:,I(3)) * N(:,I(3))';
28          [V,L]=eig(m);
29          [L, I1] = sort(diag(L));
30
31          if L(2)/L(1) > UmSup
32
33              isCoplanar=1;
34              n = V(:,I1(1));
35              n = n / norm(n);
36          end
37      end
38
39      cont = 0;
40      Sol=zeros(nv,1);
41
42      for i=1:nv
43
44          v=N(:,i) / norm(N(:,i));
45
46          if abs(dot(v,n)) < UmDot

```

```
42         cont = cont + 1;  
          Sol(i) = 1;  
44     end  
      end  
46 end
```

Código A.1: Implementación de RANSAC para agrupar vector coplanares