



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“GENERACIÓN DE MOSAICOS DE 3 O MÁS  
IMÁGENES DE FONDO DE OJO PARA LA  
VISUALIZACIÓN DE VASOS SANGUÍNEOS EN 2D”**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRA EN CIENCIAS  
(COMPUTACIÓN)**

**P R E S E N T A:**

**ANA LAURA ÁVILA MAYO**

**DIRECTOR DE TESIS  
DRA. MARÍA ELENA MARTÍNEZ PÉREZ**

**CODIRECTOR  
DR. ARTURO ESPINOSA ROMERO**

México, D.F.

2007.



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Resumen

Esta tesis presenta una manera de obtener un mosaico en 2D a partir de una secuencia de imágenes de fondo de ojo. El proceso descrito a lo largo del documento tiene un enfoque modular, basado en la metodología descrita en Zhang *et al.* [1994] y en Capel [2004]. Podemos resumir las fases del proceso de reconstrucción desarrollado en este trabajo en tres etapas principales: relacionar las imágenes, encontrar la transformación adecuada y por último creación del mosaico. El objetivo principal de este documento es obtener una escena plana de mosaico a partir de una secuencia de imágenes médicas. Los métodos probados en las diferentes fases son descritas ampliamente en el documento. Cada una de las fases es descrita independientemente pero manteniéndolas dentro del contexto del objetivo principal.

# Dedicatoria

*A mis padres a quienes les debo todo lo que soy.*

# Agradecimientos

Quiero agradecer a mis asesores, la Dra. Ma. Elena Martínez Pérez y el Dr. Arturo Espinosa Romero, por su apoyo e invuable ayuda en esta tesis. A CONACYT y a la UAP por el apoyo financiero otorgado en los dos años de maestría. Al IIMAS por su apoyo y ayuda cuando la he necesitado.

Gracias a mi familia por su comprensión, paciencia y apoyo que me animan a seguir adelante. Gracias a mis compañeros que han hecho de la maestría una grata experiencia y un lugar agradable. Por último gracias a mi novio y a mis amigos y a todos aquellos que de alguna manera ayudaron con sus comentarios o recomendaciones para programar e hilar ideas en la elaboración de esta tesis.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	3
1.1.1. Mosaicos . . . . .	3
1.1.2. Anatomía del Ojo . . . . .	4
1.1.3. Técnicas para Capturar las Imágenes . . . . .	8
1.1.4. Planteamiento del Problema . . . . .	9
1.1.5. Estado del Arte . . . . .	10
1.1.6. Objetivos . . . . .	12
1.1.7. Imágen Digital . . . . .	12
1.1.8. Organización del Documento . . . . .	13
<b>2. Conceptos Generales</b>	<b>14</b>
2.1. Geometría Proyectiva . . . . .	14
2.1.1. Transformaciones Proyectivas 2D . . . . .	15
2.1.2. Categorización de las Transformaciones Proyectivas 2D . . . . .	16
2.2. Visión Estéreo . . . . .	19
2.2.1. Análisis Bifocal . . . . .	19
2.3. Capítulo Siguiete . . . . .	23
<b>3. Relacionando las Imágenes</b>	<b>25</b>
3.1. Extracción de Puntos Característicos . . . . .	25
3.1.1. Detector de Bifurcaciones y Cruces RISA . . . . .	26
3.1.2. Resultados . . . . .	28
3.2. Cálculo de Correspondencias . . . . .	29
3.2.1. Correlación Cruzada Normalizada de Media Cero (ZNCC) . . . . .	31
3.2.2. Resultados . . . . .	33
3.3. Eliminando Correspondencias Erróneas . . . . .	34
3.3.1. Proceso de Relajación . . . . .	35
3.3.2. Resultados . . . . .	38
3.3.3. Algoritmo Robusto: Least Median of Squares, LMedS . . . . .	38
3.3.4. Resultados . . . . .	42
3.4. Otro Algoritmo Robusto: RANSAC . . . . .	42
3.4.1. Resultado . . . . .	46

3.5. Experimentos . . . . .	47
3.6. Resumen . . . . .	52
<b>4. Creación de mosaicos</b>	<b>53</b>
4.1. Transformación Geométrica de una Imagen . . . . .	53
4.1.1. Función de Transformación . . . . .	54
4.2. Dimensiones de la Imagen Nueva . . . . .	58
4.3. Mezcla de Imágenes . . . . .	58
4.4. Mosaico de tres o más Imágenes de Fondo de Ojo . . . . .	60
4.5. Experimentos . . . . .	63
4.6. Introducción a la Reconstrucción de un Mosaico de Imágenes en 3D. . .	65
<b>5. Conclusiones y Trabajo Futuro</b>	<b>77</b>
5.1. Conclusiones . . . . .	77
5.2. Trabajo Futuro . . . . .	79

# Índice de figuras

1.1.	Pinturas. (a) Pintura pre-renacentista: Pintura circa, Grecia. (b) Pintura renacentista: Capilla Sixtina, Italia. Fuentes tomadas de imágenes de <a href="http://www.culturageneral.net">www.culturageneral.net</a> .	2
1.2.	Imagen 3D. (a) Imagen bidimensional. (b) nube de puntos para la reconstrucción tridimensional. Imágenes obtenidas de la página del laboratorio de gráficas por computadora de la Universidad de Standford. . . . .	2
1.3.	Vista panorámica. Mannheim, Alemania. Vista panorámica formada por seis imágenes. Hecho en Adobe Photoshop CS2 ver. 9.0. . . . .	3
1.4.	Construcción de un mosaico. (a)(b)(c) secuencia de imágenes tomadas de una misma escena; (d) transformación 2D, la imagen de referencia es la segunda imagen de izquierda a derecha; (e) mezcla de las imágenes formando el mosaico. . . . .	5
1.5.	Construcción de mosaico a partir de dos imágenes. (a)(b) Secuencia de imágenes de una misma escena. (c)(d) Puntos característicos en ambas imágenes. (e) Mosaico contruido. Fuentes tomadas de imágenes de <a href="http://www.pages.drexel.edu">http://www.pages.drexel.edu</a> . . . . .	6
1.6.	Ojo humano normal, (a) vista de frente. Fuente tomada de Duane's Clinical Ophthalmolog, versión digital. (b) Anatomía del ojo. Fuente tomada de <a href="http://www.wikipedia.org">www.wikipedia.org</a> . . . . .	7
1.7.	Vista del ojo de perfil, mostrando los vasos sanguíneos y los componentes que están antes de llegar a ellos. Fuente tomada de <a href="http://www.wikipedia.org">wikipedia.org</a> . . . . .	8
1.8.	(a) Ángulos de vista; fuente tomada de Ophthalmic Photography. (b) fotografía de fondo de ojo. . . . .	9
1.9.	Construcción de un mosaico. (a) Enumera la forma arbitraria en que fueron escogidas las imágenes. (b) Mosaico construido. Imágenes tomadas de Ryan <i>et al.</i> [2004]. . . . .	11
1.10.	Construcción de un mosaico. (a) Gráfica de correspondencia. (b) Mosaico construido. Imagen tomada de Can <i>et al.</i> [2002]. . . . .	11
1.11.	Imagen Digital. Psyché ranimeé le baiser de l'Amour. Museo de Louvre, Paris. . . . .	13
2.1.	Plano proyectivo. Puntos y líneas de $\mathcal{P}^2$ son representados por rayos y planos respectivamente, a través del origen en $\mathcal{R}^3$ . Las líneas que están en el plano $x_1x_2$ representan puntos ideales, y el plano $x_1x_2$ representa $\mathbf{l}_\infty$ . . . . .	15
2.2.	Geometría Proyectiva 2D. Clasificación de las Transformaciones proyectivas 2D . . . . .	18

2.3.	Geometría de puntos correspondientes. (a) Los centros de las cámaras junto con un punto tridimensional $X$ forman el plano $\pi$ en donde también se encuentran los puntos correspondientes. (b) Se muestran los epipolos localizados en la intersección de la línea que une los centros de las cámaras y los planos de la imagen, también se observa la línea epipolar correspondiente al punto $x$ . Imágenes tomadas de [Hartley y Zisserman, 2004]. . . . .	20
2.4.	Esquema de líneas epipolares. (a) y (c) Dos vistas epipolares con líneas epipolares sobrepuestas. (b) Geometría epipolar de dos vistas. Fuente tomada [Hartley y Zisserman, 2004] . . . . .	20
2.5.	Esquema de líneas epipolares. (a) Líneas epipolares. (b) Dos vistas con líneas epipolares sobrepuestas, las cuales están en el infinito. Fuente tomada [Hartley y Zisserman, 2004]. . . . .	21
3.1.	(a) Imagen tomada de un sujeto. (b) Imagen segmentada. Imágenes tomadas de [Martínez-Pérez, 2000]. . . . .	27
3.2.	Imágenes del proceso de segmentación. (a) Una pequeña sección de la imagen original. (b) El árbol segmentado en gris y el esqueleto en negro. Imágenes tomadas de [Martínez-Pérez, 2000]. . . . .	27
3.3.	(a) Puntos candidatos marcados con un círculo. (b) La primer columna ('Skeleton') muestra los puntos candidatos obtenidos en el primer filtro, la segunda columna ('Intersections with feature') muestra el tipo de punto, la cuarta columna ('Output') muestra el resultado del segundo filtro, bifurcaciones y cruces verdaderos. Imágenes tomadas de [Martínez-Pérez, 2000]. . . . .	28
3.4.	Resultado de RISA. (a)(b) Dos imágenes de fondo de ojo del mismo sujeto, (c)(d) resultados del primer filtro de RISA (328 y 330 puntos respectivamente), (e)(f) resultados del segundo filtro de RISA (59 y 52 puntos respectivamente). . . . .	30
3.5.	Correlación. Correlación del punto $m_1$ contra todos los puntos que caen en la ventana de búsqueda de la imagen 2. . . . .	32
3.6.	Rango de ángulo $\theta$ . Donde, $L$ es el vector formado por un punto $m_1$ a un punto $m_2$ ; $\alpha$ es el ángulo obtenido de la moda; $\theta$ es el rango entre $-\beta$ y $+\beta$ . . . . .	33
3.7.	(a y b) Cruces y bifurcaciones encontrados por RISA en un par de vistas, (c) Conjunto de puntos obtenidos por el método de ZNCC. . . . .	34
3.8.	Proceso de relajación. Los círculos negros son los puntos de correspondencias candidatas $m_{1i}, m_{2j}$ , los círculos vacíos son los vecinos que se encuentran dentro de una vecindad definida por un radio $R$ . . . . .	36
3.9.	(a) La tabla $T_{SM}$ guarda el valor de la <i>fuerza de correlación</i> para cada par de correspondencia de puntos. (b) La tabla $T_{UA}$ guarda el valor de ambigüedad para cada par de correspondencia de puntos. Ambas tablas son ordenadas en forma descendiente. . . . .	38
3.10.	(a)(b) Cruces y bifurcaciones encontrados por RISA en un par de vistas, (c) Conjunto de puntos obtenidos por el método de ZNCC. (d) Conjunto de puntos obtenidos por el proceso de relajación. . . . .	39

3.11. Ilustración de la técnica de cajas (bucketing). Fuente tomada de Zhang <i>et al.</i> [1994]. . . . .	42
3.12. Selección de cajas. Cada caja tiene la misma probabilidad a ser seleccionada. Fuente tomada de Zhang <i>et al.</i> [1994]. . . . .	43
3.13. LMedS. (a)(b) Cruces y bifurcaciones encontrados por RISA en un par de vistas. (c) Conjunto de puntos obtenidos por el método de ZNCC. (d) Conjunto de puntos obtenidos por el proceso de relajación. (e) Conjunto de puntos obtenidos por el proceso de LMedS. . . . .	44
3.14. Resultado de correr RANSAC sobre las imágenes de fondo de ojo. (a),(b) Bifurcaciones y cruces obtenidas por RISA. (c) Resultado de correr (a) y (b) sobre el programa de Kovesi [2004], el cual utiliza RANSAC. . . . .	46
3.15. Relación de un par de imágenes. (a)(b) Imágenes a relacionar con 66 y 55 puntos respectivamente. (c) Resultado de la correlación con ZNCC. (d) Resultado de la relajación. (e) Resultado de aplicar LMedS. . . . .	48
3.16. Relación de un par de imágenes. (a)(b) Imágenes a relacionar con 346 y 369 puntos respectivos. (c) Resultado de la correlación con ZNCC. (d) Resultado de la relajación. (e) Resultado de aplicar LMedS. . . . .	51
4.1. Transformación de una imagen. Los pixeles de la imagen origen son reorganizados para coincidir con los pixeles de la imagen destino, por medio de un mapeo entre la imagen origen y la imagen destino. Fuente tomada de Heckbert [1989]. . . . .	54
4.2. Ejemplo de distorsiones geométricas. a) y b) muestran las deformaciones causadas por la lente de una cámara en la vida real, lente ojo de pez y omnidireccional respectivamente. c) d) e) muestra el esquema partiendo de la imagen origen, y las deformaciones de cojín y barril que son originados por las cámaras b) y a) respectivamente. Fuente tomada de Researchers [2006]. . . . .	56
4.3. Una ilustración esquemática de la transformación polinomial de segundo orden. En general todas las líneas rectas son mapeadas a curvas. Fuente tomada de Ryan <i>et al.</i> [2004] . . . . .	57
4.4. (a) Imagen original. (b) Imagen destino. (c) Rejilla con las dimensiones de la <i>imagen original</i> pero separada a 0,5 pixeles en ambas direcciones $x$ y $y$ . (d) Rejilla transformada sobre las coordenadas de la <i>imagen destino</i> . (d) Dimensiones de un marco rectangular que contendrá el mosaico formado por la imagen original e imagen destino. . . . .	59
4.5. Mezcla. (a) Rejilla de origen transformada. (b) Imagen destino. (c) Ejemplo de colocar las imágenes dentro de $I$ sin utilizar un desplazamiento. (d) Ejemplo de colocar las imágenes dentro de $I$ utilizando un desplazamiento. . . . .	61
4.6. Interpolación del vecino más cercano. (a) Si los puntos no coinciden toma el valor de intensidad del vecino más próximo. (b) y (c) se describe esquemáticamente la interpolación del vecino más cercano. Fuente tomada de Researchers [2006]. . . . .	62

4.7. Mosaico formado por dos imágenes de fondo de ojo. (a) y (b) Muestra la imagen origen y destino respectivamente, con los puntos correspondientes entre ellas, arrojados por el método LMedS (sección 3.3.3) y mostrados en la figura 3.13. (c) Muestra la imagen origen-transformada. (d) Mezcla la imagen origen-transformada con la imagen destino para crear el mosaico. . . . .	67
4.8. Mosaico formado por tres imágenes de fondo de ojo. (a) y (b) Muestra la imagen origen y destino respectivamente, con los puntos correspondientes. (c) Mezcla la imagen origen-transformada con la imagen destino para crear el mosaico. . . . .	68
4.9. (a) y (b) muestran las imágenes origen y destino respectivamente. c) muestra el mosaico construido por cuatro imágenes. . . . .	69
4.10. (a) Muestra el mosaico esquemático formado por 4 imágenes. (b) Muestra la forma en que fueron seleccionadas e incorporadas al mosaico. . . . .	70
4.11. Mosaico formado por dos imágenes de fondo de ojo. Tomando los puntos de correspondencias de la etapa de correlación (ZNCC) 3.5. (a) y (b) Muestran el conjunto de puntos correspondientes. (c) Conjunto de puntos correspondientes en forma de vectores, donde se pueden apreciar los puntos erróneos a simple vista. (d) Mosaico mal alineado. . . . .	71
4.12. (a) y (b) Imagen origen y destino respectivamente con el conjunto de puntos correspondientes entre ambas imágenes. (c) muestra el conjunto de puntos correspondientes de las imágenes en forma de vectores. (d) muestra el mosaico bien alineado. . . . .	72
4.13. Mosaico formado por 4 imágenes de fondo de ojo. (a)-(d) Muestra la secuencia de imágenes en el orden en que fueron escogidas. (e) Muestra el mosaico construido por la secuencia de imágenes de la (a)-(b). . . . .	73
4.14. Mosaico formado por una secuencia de 5 imágenes de fondo de ojo. (a) - (e) Muestra la secuencia de imágenes en la forma en que fueron escogidas aleatoriamente. f) Muestra el mosaico construido por las imágenes de la (a)-(e) y las áreas mal alineadas son resaltadas por un rectángulo. . . . .	74
4.15. Triangulación de puntos correspondientes. $C$ y $C'$ son los centros de las cámaras y $X$ es un punto tridimensional. Fuente tomada de Pollefeys [2000]. . . . .	75
4.16. (a) y (b) muestran dos imágenes de fondo de ojo. (c) y (d) muestran los esqueletos de los vasos sanguíneos extraídos por RISA. (e) Muestra el resultado del proceso de correlación hecho manualmente. (f) Muestra los vasos sanguíneos de (a) y (b) en 3D. . . . .	76

# Índice de cuadros

2.1.	Algoritmo de los 8 puntos normalizados. Tomado de Hartley y Zisserman [2004].	24
3.1.	Algoritmo para obtener el rango de $\theta$ .	33
3.2.	Proceso de relajación. Tomado de Zhang <i>et al.</i> [1994].	37
3.3.	RANSAC(Random Sample Consensus). Algoritmo de RANSAC. Método descrito en Hartley y Zisserman [2004].	45
3.4.	Experimento 1. Muestra el resultado en cada una de las etapas de nuestra implementación. <i>1er columna</i> , muestra el par de imágenes a evaluar. <i>2da. columna</i> , conjunto de puntos entrantes obtenidos por RISA. <i>3er. columna</i> , puntos buenos detectados manualmente. <i>4ta. columna</i> , resultado de la etapa ZNCC. <i>5ta. columna</i> , resultado de la etapa de relajación. <i>6ta. columna</i> , resultado de la etapa de LMedS. <i>7ma. columna</i> son los falsos negativos.	49
3.5.	Experimento 2. Muestra el resultado en cada una de las etapas de nuestra implementación al aplicar un conjunto de puntos con ruido del 20%,40% y 60% generados aleatoriamente sobre un par de imágenes.	49
3.6.	Experimento 3. Muestra el resultado en cada una de la etapas de nuestra implementación con una imagen de una oficina.	50
3.7.	Experimento 4. Se corrieron en Kovese [2004], los mismos experimentos realizados para nuestra implementación con LMedS y se comparan los resultados de ambos. La columna 6 muestra los falsos positivos de nuestra implementación y la columna 8 muestra los falsos positivos de RANSAC.	50
4.1.	Experimento 3. La <i>1er. columna</i> , muestra la forma aleatoria en que fueron seleccionados las imágenes de la figura 4.14 (a)-(e). La <i>2da. columna</i> , muestra la imagen origen que será transformada en las coordenadas de la imagen destino. La <i>3er columna</i> , muestra la imagen destino.	64
4.2.	Experimento 4. La <i>1er. columna</i> , muestra la forma aleatoria en que fueron seleccionados las imágenes de la figura 4.14 (a)-(e). La <i>2da. columna</i> , muestra la imagen origen que será transformada en las coordenadas de la imagen destino. La <i>3er columna</i> , muestra la imagen destino.	64

# Capítulo 1

## Introducción

Desde la antigüedad, el hombre ha sentido siempre la necesidad de representar gráficamente el entorno que le rodea, como lo demuestran los dibujos encontrados en las cuevas prehistóricas. Sin embargo, las pinturas prehistóricas son planas, no muestran la profundidad de los objetos como existen en nuestro mundo tridimensional.

Fue hasta el renacimiento cuando los pintores de esta época intentaron representar la profundidad. Ellos fueron los primeros en entender la formación de las imágenes y avocarse en el estudio de la geometría para reproducir los efectos de la perspectiva en las imágenes del mundo que observaban. En las pinturas planas el tamaño de los objetos está relacionado más con la importancia dentro de la obra que con la ubicación espacial. Se muestra un ejemplo en la figura 1.1 (a). En las pinturas renacentistas, se aprecia claramente la profundidad producida por las líneas que convergen en un punto de fuga, de esta manera se le hace creer al observador que está frente a una escena tridimensional. Se muestra un ejemplo en la figura 1.1 (b).

A partir del plano cartesiano introducido por Descartes (1596-1650) se empieza a concebir la geometría desde un punto de vista algebraico. Así las entidades geométricas son descritas como coordenadas y entidades algebraicas.

En el año 1826 el químico francés Niepce llevó a cabo la primer fotografía, colocando una superficie fotosensible dentro de una cámara oscura para fijar la imagen, plasmando por primera vez una escena del mundo tridimensional en una imagen bidimensional. Una cámara produce imágenes bidimensional de un mundo físico percibido como tridimensional.

Ahora, en la actualidad con el avance de la tecnología, se intenta reconstruir los objetos del mundo tridimensional a partir de imágenes bidimensionales. Esto es un tema que forma parte de visión computacional.

La visión computacional es una gran herramienta para establecer la relación entre

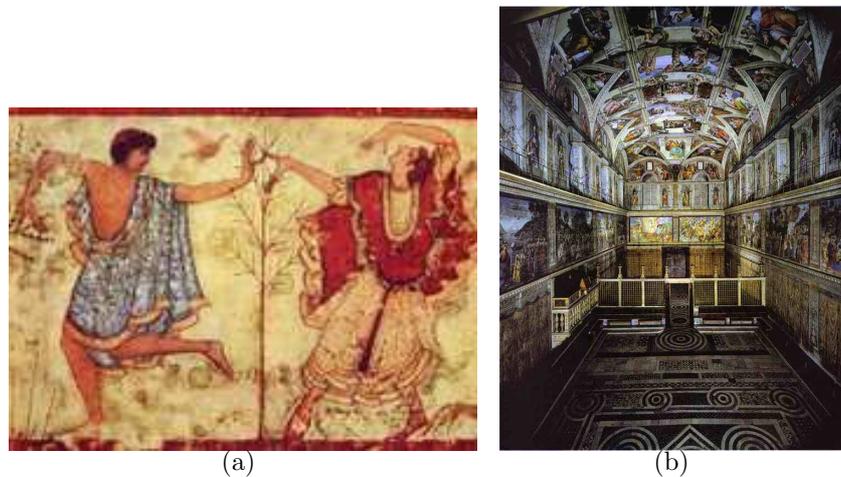


Figura 1.1: Pinturas. (a) Pintura pre-renacentista: Pintura circa, Grecia. (b) Pintura renacentista: Capilla Sixtina, Italia. Fuentes tomadas de imágenes de [www.culturageneral.net](http://www.culturageneral.net).

el mundo tridimensional y las vistas bidimensionales tomadas de él. Por medio de esta teoría se puede hacer por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas (figura 1.2) y por otra parte llevar a cabo una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

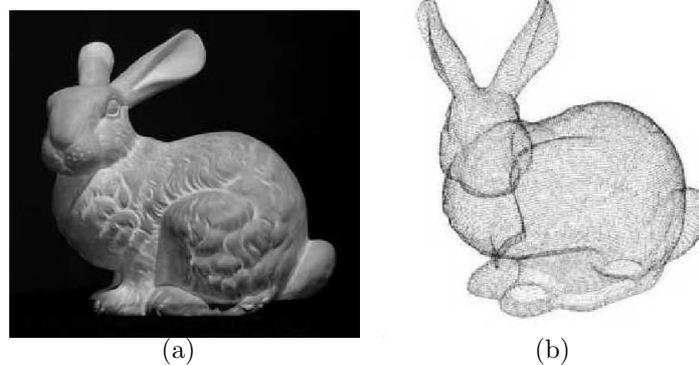


Figura 1.2: Imagen 3D. (a) Imagen bidimensional. (b) nube de puntos para la reconstrucción tridimensional. Imágenes obtenidas de la página del laboratorio de gráficas por computadora de la Universidad de Standford.

La proyección en un plano bidimensional de varias imágenes para formar una vista panorámica (también llamado mosaico) de una escena tridimensional es un tema de visión computacional. El objetivo de crear una vista panorámica es tener una visión general de la escena, poder apreciar el tamaño de un objeto con respecto al resto de



Figura 1.3: Vista panorámica. Mannheim, Alemania. Vista panorámica formada por seis imágenes. Hecho en Adobe Photoshop CS2 ver. 9.0.

los objetos y a su posición relativa y tener una mejor idea de la escena que se está observando. Un ejemplo de una vista panorámica o mosaico de imágenes es mostrada en la figura 1.3.

La investigación sobre construcción de mosaicos se ha llevado a cabo por más de 30 años [Milgram, 1975], el rango de aplicaciones incluye formación de imágenes panorámicas, realidad virtual, compresión de imágenes, detección de cambios, superresolución, seguimiento (*tracking*), navegación, composición de documentos y cartografía.

Una aplicación en la que los mosaicos son particularmente valorables es en el diagnóstico y tratamiento de enfermedades de la retina, útil para monitorear la progresión de enfermedades y que puede ser usado como un mapa espacial durante tratamientos quirúrgicos [Can *et al.*, 2002].

En este trabajo de tesis nos enfocamos en la creación de un mosaico bidimensional a partir de una secuencia de imágenes médicas de vasos sanguíneos tomadas de un mismo sujeto. En un trabajo futuro estas técnicas pueden ser utilizadas para la reconstrucción 3D de imágenes de fondo de ojo.

## 1.1. Antecedentes

### 1.1.1. Mosaicos

Vamos a hablar en términos generales de los mosaicos. Un mosaico es la alineación de múltiples imágenes dentro de una composición más grande la cual representa una porción de una escena tridimensional. El mosaico es formado a partir de varias imágenes del mundo real que se sobreponen; en la literatura existen varias definiciones citadas [Can *et al.*, 2002; Zhang *et al.*, 1994]. Nosotros daremos una breve descripción sobre el método elemental. Hay cuatro pasos básicos: registro, marco, reproyección y mezcla, los cuales son descritos a continuación [Capel, 2004].

1. *Registro* (también llamado *cálculo de correspondencia*): primero, identificamos puntos característicos (ver 1.5 (c)-(d)) en ambas imágenes y mediante técnicas de correspondencia identificamos los puntos de la imagen 1 que también se encuentran en la imagen 2.
2. *Marco*: el objetivo es escoger un marco global, es decir, escoger la alineación con respecto a una imagen llamada imagen de referencia, así la transformación entre el marco global y la imagen de referencia es simplemente la identidad. Para una secuencia de imágenes, este proceso primero calcula la transformación geométrica entre un par de imágenes. En la figura 1.4 se escoge a (b) como la *imagen referencia* y las imágenes (a) y (c) son los que se alinearán a ella.
3. *Reproyección*: después de crear el marco global, cada punto en cada imagen puede ser transformado a un punto en el marco global. En algunos casos, este mapeo puede ser una transformación tan simple que sólo involucre escalamiento y traslación, o bien puede ser alguna transformación más complicada, tales como rotación de cámara. Habiendo escogido una transformación, cada imagen es incorporada en el marco de la imagen de referencia. En la figura 1.4 (d) las imágenes son alineadas a la *imagen referencia*.
4. *Mezcla*: es la etapa final, el objetivo es mezclar todas las imágenes por la porción que se superponen. En la figura 1.4 (e) se muestra cómo queda la mezcla de la secuencia de imágenes. La figura mostrada está completa y se puede ver cómo una sola imagen.

Un ejemplo de construcción de mosaico con dos imágenes tomadas de una misma escena puede verse en la figura 1.5, donde al crear el mosaico se puede apreciar una vista más amplia de la escena.

### 1.1.2. Anatomía del Ojo

Vamos a hablar brevemente sobre la anatomía del ojo humano, para poder entender claramente el tipo de imágenes sobre las que se trabajarán a lo largo de este documento.

El ojo humano es un órgano de visión que detecta la luz. El sistema de visión humano nos permite extraer información tridimensional a partir de las imágenes bidimensionales proyectadas en cada uno de los ojos. Como un objeto tridimensional, el globo del ojo se aproxima a una esfera irregular como se muestra en la figura 1.6 (a).

Se han registrado diferentes medidas en la dimensión del globo del ojo, sin embargo, se encuentra un rango de valor constante entre la población adulta y no tiene relación con el sexo o raza. En los adultos el globo pesa 7.5g y tiene un diámetro promedio de 23.48mm.

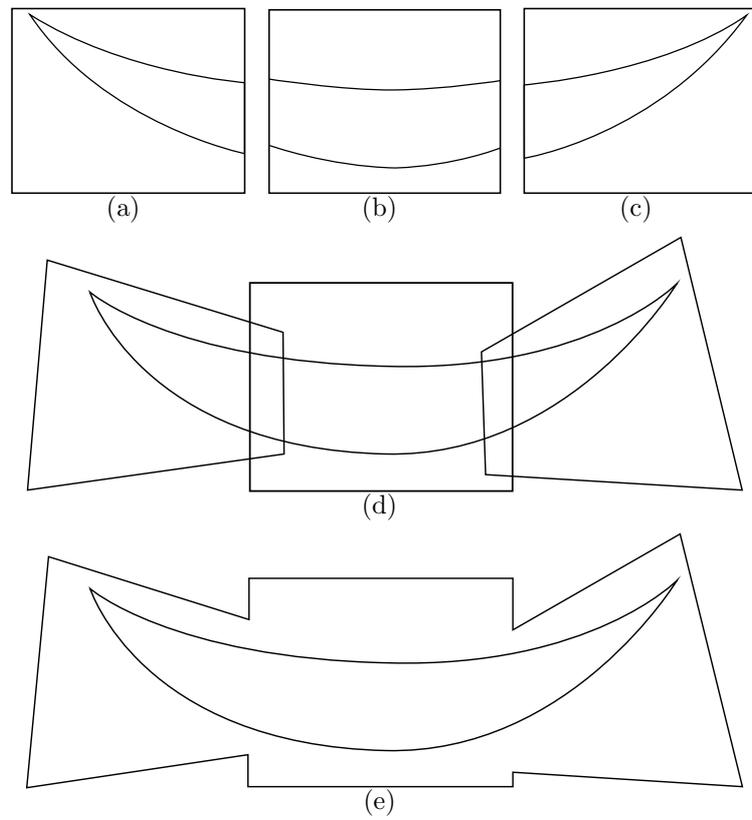


Figura 1.4: Construcción de un mosaico. (a)(b)(c) secuencia de imágenes tomadas de una misma escena; (d) transformación 2D, la imagen de referencia es la segunda imagen de izquierda a derecha; (e) mezcla de las imágenes formando el mosaico.



(a)



(b)



(c)



(d)



(e)

Figura 1.5: Construcción de mosaico a partir de dos imágenes. (a)(b) Secuencia de imágenes de una misma escena. (c)(d) Puntos característicos en ambas imágenes. (e) Mosaico contruido. Fuentes tomadas de imágenes de <http://www.pages.drexel.edu>.

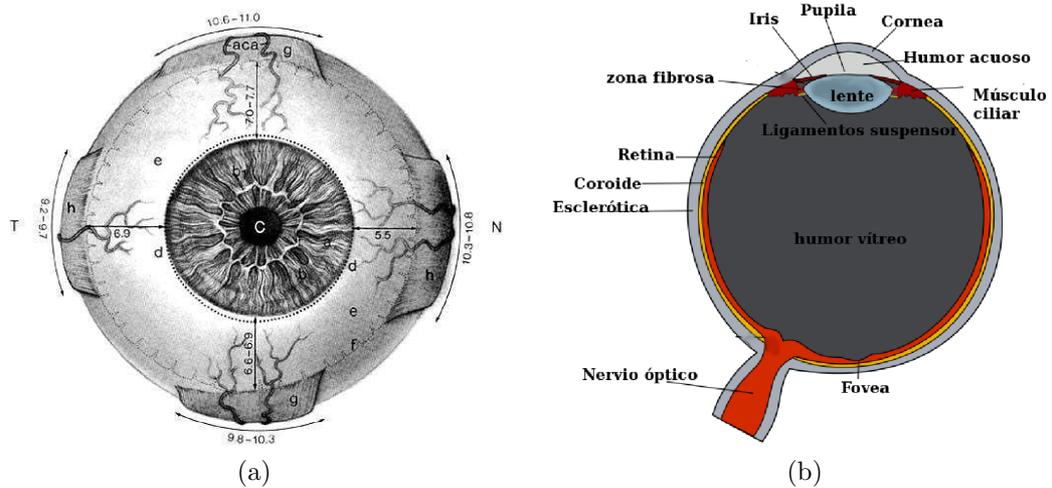


Figura 1.6: Ojo humano normal, (a) vista de frente. Fuente tomada de Duane's Clinical Ophthalmolog, versión digital. (b) Anatomía del ojo. Fuente tomada de [www.wikipedia.org](http://www.wikipedia.org).

La anatomía del ojo (figura 1.6 (b)) está estructurada para enfocar la luz sobre la retina, todos los componentes por los que pasa la luz antes de llegar a la retina son transparentes. La córnea y el lente ayuda a converger los rayos de luz sobre la retina. La luz causa cambios químicos en las células fotosensibles de la retina y con esto se disparan impulsos eléctricos que llegan al cerebro. La luz entra al ojo desde un medio externo, pasa a través de la córnea y de la pupila y entra al humor acuoso. La mayor refracción de luz ocurre en la córnea la cual tiene una curvatura fija. El humor acuoso es una masa clara la cual conecta la córnea con la lente del ojo y ayuda a mantener la forma convexa de la misma. La cámara vítrea es la zona entre el lente y la retina y está rellena de un gel transparente llamada humor vítreo.

El iris está entre el lente y el humor acuoso y es un anillo rojo de músculo fibroso. La luz debe pasar primero a través del centro del iris, la pupila. El tamaño de la pupila es activamente ajustado por un músculo radial y circular para mantener una constancia de luz que entra en el ojo. El lente detrás del iris es un disco convexo elástico, el cual enfoca la luz a través del humor vítreo sobre la retina.

Tres capas mantienen y dan forma al globo del ojo (figura 1.6 (b)), la esclerótica, la cual da el tono blanco al ojo. La coroides es la capa interior siguiente. Y la tercera capa es la retina; en la tercera capa se encuentran los vasos sanguíneos (figura 1.7).

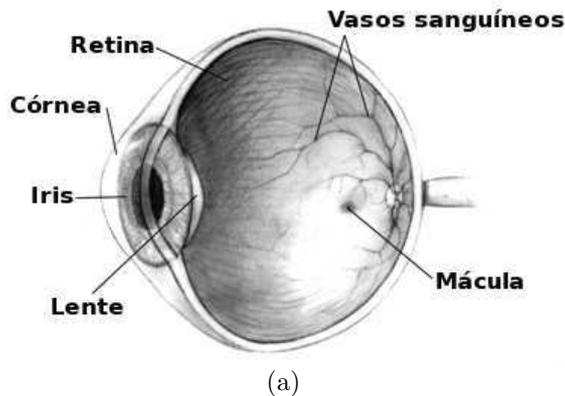


Figura 1.7: Vista del ojo de perfil, mostrando los vasos sanguíneos y los componentes que están antes de llegar a ellos. Fuente tomada de wikipedia.org.

### 1.1.3. Técnicas para Capturar las Imágenes

Las imágenes con las cuales se trabaja en esta tesis son imágenes de fondo de ojo. La técnica para tomar estas imágenes es descrita a continuación.

La retina es el tejido neurosensorial del ojo que traslada las imágenes ópticas en impulsos eléctricos al cerebro. Sobre la capa de la retina se encuentran los vasos sanguíneos que son de interés en nuestra investigación. Estos pueden ser fotografiados observando a través de la pupila. A estas tomas se les llama fotografía de fondo de ojo y el equipo con el cual son tomadas se llama cámara de fondo de ojo.

En el mercado existe una gran diversidad de cámaras de fondo de ojo y básicamente son identificadas por el ángulo de vista y el ángulo óptico de aceptación de la lente (figura 1.8 (a)). Existen cámaras con ángulo múltiple y cámaras con ángulo único. El ángulo de  $30^\circ$  se considera el ángulo normal de vista, crea una imagen 2.5 veces más grande que la realidad; una cámara de fondo de ojo con ángulo amplio captura imágenes entre 45 y 140 grados y provee proporcionalmente menos magnificación de la retina; un ángulo estrecho es de  $20^\circ$  o menos.

El sistema óptico de la cámara de fondo de ojo proyecta un anillo iluminado a través de la pupila dilatada; la luz refleja la retina sobre el fondo de la cámara formando una imagen plana (fotografía de fondo de ojo, figura 1.8 (b)). Las imágenes así tomadas pueden ser capturadas digitalmente a través de una cámara digital CCD (Charge Coupled Device) o bien a través de la digitalización de una película fotográfica.

El ángulo usado para la toma de las fotografías en esta tesis es de  $50^\circ$ , sin embargo, no se puede hacer una sola toma fotográfica abarcando toda la superficie del ojo, así que se hacen varias tomas para capturar la mayor parte de la superficie del ojo.



Figura 1.8: (a) Ángulos de vista; fuente tomada de Ophthalmic Photography. (b) fotografía de fondo de ojo.

#### 1.1.4. Planteamiento del Problema

El objetivo del presente trabajo es reconstruir la superficie del fondo de ojo en un plano bidimensional a partir de una secuencia de imágenes médicas de fondo de ojo, de manera que podamos visualizar la superficie completa en una sola imagen y poder tener una vista más amplia (también llamado mosaico). Para construir un mosaico con un mínimo de errores, debemos tomar en cuenta las distorsiones provocadas por factores internos y externos del ojo. Los factores internos son provocados por todos los componentes transparentes por los que tiene que pasar la luz antes de llegar a la retina (córnea, pupila, iris, lente, humor vítreo). Los factores externos son los provocados por la lente de la cámara de fondo de ojo y por la transformación proyectiva de un objeto 3D a un objeto plano.

El problema estará dividido en varias fases, las cuales se describen a continuación.

**Relacionando las imágenes.** En esta primera etapa se busca la manera de encontrar correspondencias entre las imágenes, así como establecer una geometría entre ellas. El resultado será un conjunto de puntos que indican la sobreposición entre las imágenes.

**Transformación 2D.** En esta fase debemos encontrar la forma de alinearlas, usando alguna transformación geométrica que se adapte al tipo de la imagen: es decir, tomando en cuenta que la fotografía al ser tomada sufre algún tipo de deformación .

**Creación del mosaico.** En esta última fase, se mezclan todas las imágenes para formar el mosaico de imágenes de fondo de ojo.

El objetivo del trabajo de tesis presentado puede servir como base para la recons-

trucción 3D de los vasos sanguíneos con más de una vista. Este tema se comentará como una introducción breve en esta tesis pero es continuación de otra.

### 1.1.5. Estado del Arte

La creación de mosaicos a partir de una secuencia de imágenes de fondo de ojo es un problema conocido en computación [Can *et al.*, 2002; Ryan *et al.*, 2004; Shen *et al.*, 2003; Becker *et al.*, 1998] el objetivo de estos trabajos es simplificar la interpretación clínica.

Ryan *et al.* [2004] utilizan de 5 a 7 fotografías para crear el mosaico. El método describe los siguientes pasos: primero se etiqueta una imagen de referencia arbitrariamente, las demás se toman como imágenes distorsionadas. Se identifica los puntos de control (en un par de imágenes). Este paso se realiza manualmente. Para relacionar las imágenes se hace primero una transformación afín, seguida por una transformación de similitud y por último una transformación polinomial de segundo orden para corregir la distorsión geométrica, y con esto obtiene la transformación necesaria para la creación del mosaico. Se crea el mosaico de la siguiente forma:

Del total de imágenes que se tienen:

- Inicialmente se toma una imagen arbitraria a la que se llama *imagen de referencia*, y el resto de las imágenes deberán alinearse a ella (a las demás imágenes se les llama *imágenes distorsionadas*).
- Se toma una *imagen distorsionada*, y se encuentran las correspondencias con la *imagen de referencia*.
- Cuando una imagen-nueva seleccionada (*distorsionada*) se intersecta con más de una imagen (la de referencia y otras más), entonces la imagen-nueva será comparada con todas aquellas que intersecta; es decir, la nueva imagen será la imagen distorsionada y con quien empata será la *imagen de referencia*.

El proceso de la creación del mosaico es ilustrado en la figura 1.9 (a) y en la 1.9 (b) muestra el mosaico construido.

Can *et al.* [2002] proponen una técnica lineal, que crea la correspondencia directa e indirectamente entre los pares de puntos de control (sobre una gráfica) por transformación de estimación conjunta de todas las imágenes dentro de la imagen de referencia del sistema de coordenadas. El algoritmo permite crear mosaicos a partir de  $n$ -imágenes. Primero se detectan los puntos de control automáticamente, y después se hace una transformación afín para encontrar la correspondencia entre imágenes y para refinar esta correspondencia se utiliza la suma de diferencia de cuadrados (SSD), teniendo esto se aplica una transformación cuadrática y con este resultado construye el mosaico. Se muestra un ejemplo de la construcción del mosaico en la figura 1.10.

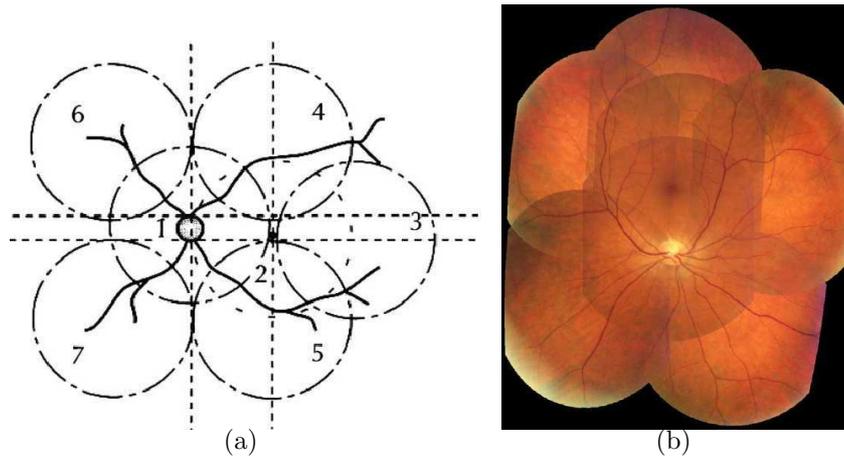


Figura 1.9: Construcción de un mosaico. (a) Enumera la forma arbitraria en que fueron escogidas las imágenes. (b) Mosaico construido. Imágenes tomadas de Ryan *et al.* [2004].

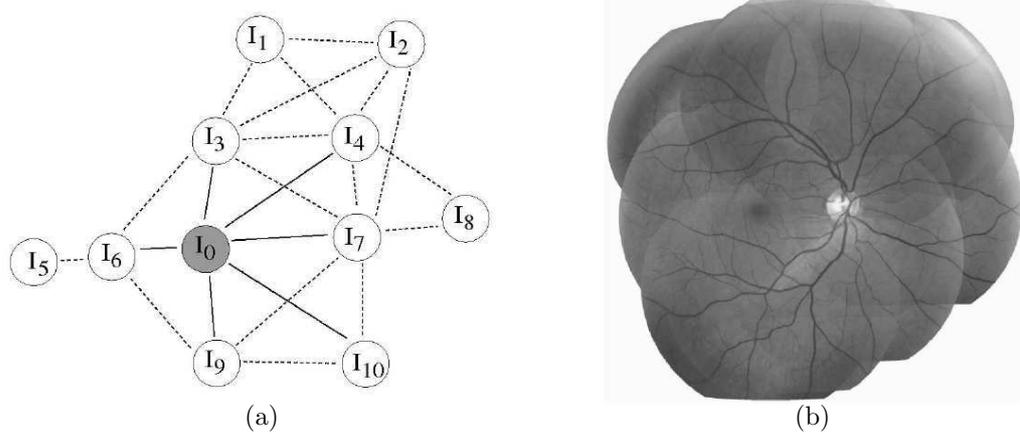


Figura 1.10: Construcción de un mosaico. (a) Gráfica de correspondencia. (b) Mosaico construido. Imagen tomada de Can *et al.* [2002].

Shen *et al.* [2003] crean un mosaico para ser utilizado en tiempo real, utilizando invariantes para identificar el mismo conjunto de puntos en diferentes imágenes usando la transformación cuadrática que define Can *et al.* [2002]. Las invariantes están basadas en la posición y orientación de un conjunto puntos seleccionados.

Existen otros autores que han implementado mosaicos de imágenes de fondo de ojo en 3D [Deguchi *et al.*, 2000; Martínez-Pérez y Espinosa-Romero, 2004]. Deguchi *et al.* [2000] contruyen un mosaico en 3D a partir de 3 imágenes: primero modelan el lente del ojo en forma esférica e identifican todos los parámetros ópticos de la cámara, y entonces re proyectan las imágenes a la superficie esférica reconstruida. Martínez-Pérez y Espinosa-Romero [2004] crean un mosaico en 3D a partir de dos imágenes de fondo de ojo utilizando una superficie cilíndrica para proyectar las imágenes.

### 1.1.6. Objetivos

En la presente tesis tenemos como objetivo hacer una extensión del trabajo de Martínez-Pérez y Espinosa-Romero [2004]. Nuestros objetivos se dividen en dos:

1. El cálculo automático de las características así como de las correspondencias para lo cual investigaremos y evaluaremos diferentes métodos.
2. La generación de mosaicos en el plano con tres o más vistas de imágenes tomadas con una abertura de  $50^\circ$ , de manera que podamos tener una vista panorámica mayor para su posterior reconstrucción en el espacio. Para ello probaremos transformaciones cuadráticas que corrijan la transformación geométrica necesaria para proyectar la composición de la esfera ocular en el plano.

### 1.1.7. Imágen Digital

Antes de empezar con la descripción para la creación de un mosaico bidimensional y los métodos que se han propuesto para obtenerla, existen algunos conceptos que deben ser comprendidos claramente ya que son necesarios para obtener las ideas más avanzadas descritas en capítulos posteriores de este trabajo. Podemos describir una imagen como '*Una función bidimensional de la intensidad de la luz  $I(x,y)$  donde  $x,y$  son coordenadas espaciales y cualquier punto  $(x,y)$  es proporcional a la brillantez de la imagen en ese punto*' [Gonzalez y Woods, 1993]. Siguiendo esta definición lo que ha sido discretizado es tanto las coordenadas espaciales como los niveles de brillantez. Podemos imaginar una imagen como una matriz cuyos índices de fila y columna identifican a un punto en la imagen y el valor del elemento de la matriz identifica el nivel de gris (en caso de imágenes en escala de grises) en este punto. Los elementos que forman esta matriz son llamados comúnmente *pixeles*. La figura 1.11 muestra una imagen digital.

El proceso necesario para la reconstrucción de un escena panorámica partiendo de la información bidimensional contenida en imágenes lleva varios años de investigación y en capítulos posteriores serán descritos algunos métodos.

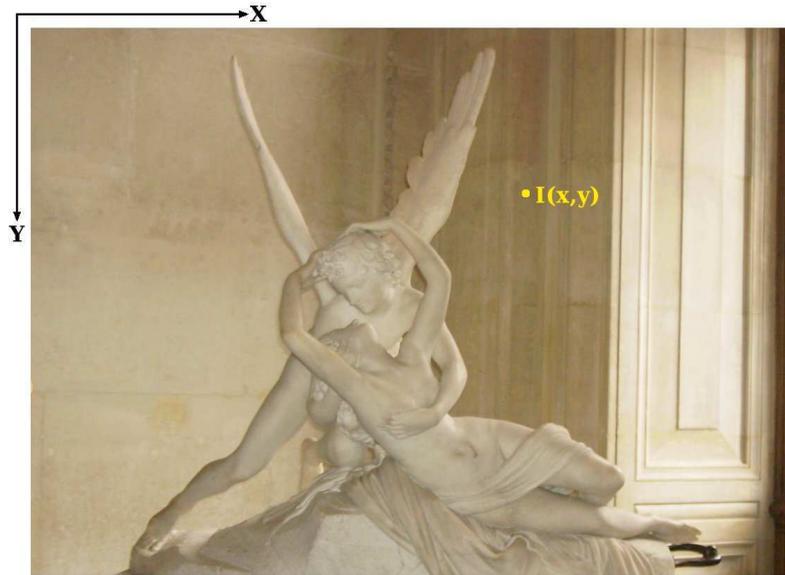


Figura 1.11: Imagen Digital. Psyché ranimee le baiser de l'Amour. Museo de Louvre, Paris.

### 1.1.8. Organización del Documento

Este documento está formado de la siguiente manera: en el capítulo dos trataremos los conceptos básicos de la geometría epipolar necesaria para la presentación de los métodos de correspondencia que utilizaremos. En el capítulo tres presentaremos algunos métodos para relacionar imágenes (cálculo de correspondencias), comenzando desde la obtención de puntos característicos hasta obtener un conjunto de correspondencias correctas que relacione a un par de imágenes. El capítulo cuatro describiremos la creación del mosaico, empezando por encontrar la transformación 2D necesaria para alinear las imágenes y por último construir el mosaico. En el capítulo cinco se representarán varios experimentos realizados con el método propuesto para construir mosaicos. En el capítulo seis analizaremos nuestra implementación e indicaremos los puntos en que se puede mejorar.

# Capítulo 2

## Conceptos Generales

En este capítulo se introducen los conceptos y las principales ideas geométricas y la notación que son requeridos para entender el resto del trabajo de la tesis presentada. En particular, este capítulo cubre: *i*) las transformaciones proyectivas de un plano. Estas transformaciones modelan la distorsión geométrica que le ocurre a un plano que es visto a través de la perspectiva de una cámara. Sin embargo, no todas las propiedades geométricas del plano son distorsionadas, algunas como la colinealidad son preservadas; por ejemplo, las líneas rectas son vistas como líneas rectas, mientras que otras propiedades no son preservadas, tales como las líneas paralelas que por lo general no son vistas como líneas paralelas. *ii*) Visión estéreo. En esta sección se estudiarán las relaciones algebraicas y geométricas que existen cuando se ha tomado más de una vista de una escena. Se pondrá especial énfasis en el análisis de dos, geometría *bifocal*. Para más detalle sobre las transformaciones proyectivas y sobre visión estéreo puede ser investigado en Hartley y Zisserman [2004].

### 2.1. Geometría Proyectiva

Vamos a entender bien el concepto de geometría proyectiva. La geometría proyectiva está en todos lados y todos estamos familiarizados con las transformaciones proyectivas: cuando miramos un fotografía vemos esquinas que no son esquinas, o círculos que no son círculos. La transformación que mapea a los objetos 3D en objetos planos, es un ejemplo de transformación proyectiva.

La mayoría de las propiedades geométricas no son preservadas por la transformación proyectiva, por ejemplo, el círculo que puede aparecer como una elipse, ni el ángulo, ni la distancia, ni el radio, son preservados. Sin embargo, una de las propiedades que es preservada es la colinealidad. Así podemos definir una transformación proyectiva de un plano como algún mapeo de puntos sobre el plano que preserva líneas rectas.

En el resto del capítulo hablaremos de un plano proyectivo, que definimos a continuación. Un plano proyectivo  $\mathcal{P}^2$  es un conjunto de rayos en  $\mathcal{R}^3$ . El conjunto de todos

los vectores  $k(x_1, x_2, x_3)^\top$  cuando  $k$  varía forma un rayo que pasa a través del origen. Tal rayo puede pensarse como la representación de un punto sobre  $\mathcal{P}^2$ . Las líneas proyectadas en  $\mathcal{P}^2$  corresponden a planos que cruzan el origen. Se verifica que dos rayos no idénticos inciden en un plano y que dos planos intersectan en una recta. Esto es la analogía de dos puntos únicos y diferentes que definen a una línea y de dos líneas que siempre se intersectan en dos puntos.

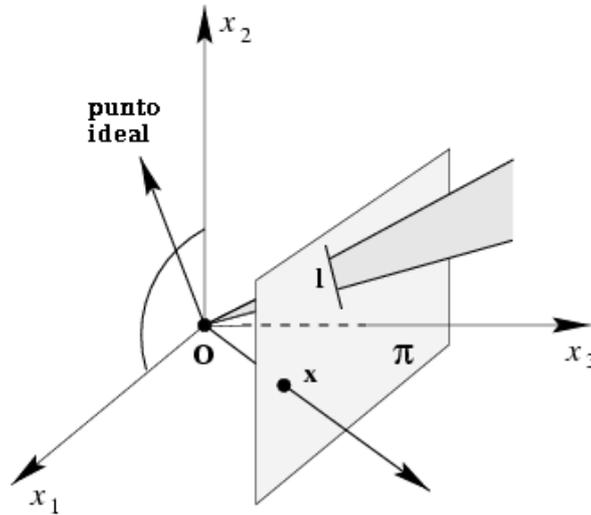


Figura 2.1: Plano proyectivo. Puntos y líneas de  $\mathcal{P}^2$  son representados por rayos y planos respectivamente, a través del origen en  $\mathcal{R}^3$ . Las líneas que están en el plano  $x_1x_2$  representan puntos ideales, y el plano  $x_1x_2$  representa  $l_\infty$

### 2.1.1. Transformaciones Proyectivas 2D

La geometría proyectiva 2D es el estudio de las propiedades del plano proyectivo  $\mathcal{P}^2$  que son invariantes bajo un grupo de transformaciones conocidas como *proyectividades*. Una proyectividad es también llamada *colinealidad*, *transformación proyectiva* o una *homografía*.

Una proyectividad es una transformación invertible dada por:  $h\mathcal{P}^2 \rightarrow \mathcal{P}^2$  de manera tal que una línea recta es transformada en otra línea recta. La proyectividad está definida como:

$$h(x) = x' = Hx \tag{2.1}$$

donde  $H$  es una matriz de 3x3 no singular y  $x$  es un punto de  $\mathcal{P}^2$   $(x,y,1)$  representado como un 3-vector homogéneo. Se dice entonces que  $x'$  es la transformación lineal  $H$  de

$x$ . Esta transformación es biunívoca entre dos planos 2D, cuyos puntos son representados homogéneamente por  $x$  y  $x'$ . Es decir, un punto en el plano 2D tiene una única correspondencia en un punto de otro plano 2D y cada punto en un plano tiene un solo punto correspondiente en el otro plano.

La condición invariante (una recta es transformada en una recta) puede comprobarse de la siguiente manera: Si  $x_1, x_2$  y  $x_3$  están en la misma línea recta  $l$  ( $l$  es un vector con los componentes de la ecuación de la recta), entonces  $l^\top x_i = 0$  para  $i = 1, 2, 3$ . Esta ecuación puede ser escrita como  $l^\top H^{-1} H x_i = 0$ . Así los puntos  $H x_i$  caen en una línea  $H^{-\top} l$  y la colinealidad es preservada por la transformación.

La ecuación 2.1 se puede escribir como:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (2.2)$$

Su multiplicamos  $H$  por una constante  $k$  diferente de cero, se observa que los resultados en la transformación de coordenadas no son afectadas para la condición de invariante (una recta es transformada en una recta).

### 2.1.2. Categorización de las Transformaciones Projectivas 2D

En esta sección describiremos la especial importancia de una transformación proyectiva y sus propiedades geométricas. Las transformaciones proyectivas forman un grupo, llamado el *grupo lineal proyectivo* y será visto que estas especializaciones son subgrupos de este grupo.

A continuación se presentan cuatro categorías existentes en las transformaciones proyectivas.

#### Clasificación I: Transformación Isométrica (Euclídea)

En la transformación isométrica se conserva la distancia euclídea, es decir, la distancia entre dos puntos es igual a la distancia entre los puntos transformados. Una transformación isométrica es representada como:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.3)$$

donde  $\theta$  es el ángulo de la rotación entre los ejes y  $t_x, t_y$  es el desplazamiento del origen y  $\epsilon = \pm 1$ . Si  $\epsilon = 1$  entonces se prevea la orientación. Si  $\epsilon = -1$  entonces la transformación isométrica invierte la orientación.

Una transformación euclídea puede ser escrita en forma sintetizada, de la siguiente manera:

$$x' = H_E x = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} x \quad (2.4)$$

La transformación Euclídea tiene 3 grados de libertad: uno para la rotación y dos para la translación.

### Clasificación II: Transformación de Similitud

Una transformación de similitud es una composición isométrica con una escala isotrópica. En esta transformación de similitud se conserva la forma de los objetos. Sin embargo, la distancia entre dos puntos ya no es igual a la distancia entre los puntos transformados. La transformación proyectiva de similitud se representa de la siguiente manera:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.5)$$

donde el escalar  $s$  representa la escala isotrópica, con  $s$  se obtiene la ampliación ( $s > 1$ ) o reducción de los objetos ( $0 < s < 1$ ). Las invariantes de esta transformación son : i) ángulos entre rectas, ii) líneas paralelas, iii) razón entre dos distancias y iv) razón entre dos áreas.

También puede ser escrita en forma sintetizada, de la siguiente manera:

$$x' = H_s x = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} x \quad (2.6)$$

Una transformación de similitud tiene 4 grados de libertad, una más que la transformación isométrica que es factor escala.

### Clasificación III: Transformación Afín

Una transformación afín es una transformación lineal no singular seguida por una traslación. En otras palabras, en una transformación afín se distorsiona la forma de los objetos introduciendo una matriz  $A$  de  $2 \times 2$  no ortonormal en lugar de la matriz de rotación  $R$ . La transformación afín, está representada por la siguiente matrix:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.7)$$

donde las  $a_{11} \dots a_{22}$  forman la matriz  $A$  no singular. Las invariantes de esta transformación son: i) líneas paralelas y ii) razón entre dos áreas.

También puede ser escrita en forma sintetizada, de la siguiente manera:

$$x' = H_A x \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x \quad (2.8)$$

Esta transformación tiene 6 grados de libertad, cuatro para la matriz  $A$  y dos para la matriz de traslación.

#### Clasificación IV: Transformación Projectiva General

Es una transformación lineal no singular de coordenadas homogéneas. Es la generalización de las transformaciones lineales  $\mathcal{R}^2 \rightarrow \mathcal{R}^2$  en la que las líneas paralelas no son transformadas necesariamente como tales. Las invariantes de esta transformación son la concurrencia y la colinealidad.

La matriz de proyección general está descrita en la fórmula 2.2 y se puede escribir en forma sintetizada de la siguiente manera:

$$x' = H_P x = \begin{bmatrix} A & t \\ v^T & u \end{bmatrix} \quad (2.9)$$

donde  $v = (v_1, v_2)^T$ , es un vector de valores arbitrarios y  $u$  es un escalar. Ya que lo importante es la proporción que existe entre los elementos de la matriz, ésta se puede homogeneizar con respecto de un valor de escala diferente de cero, quedando así 8 grados de libertad.

En la figura 2.2 se muestra un ejemplo de las cuatro clasificaciones de la transformación projectiva.

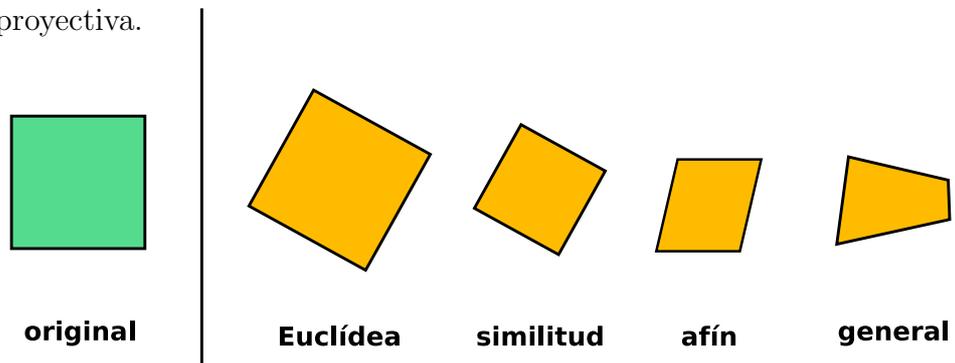


Figura 2.2: Geometría Projectiva 2D. Clasificación de las Transformaciones proyectivas 2D

## 2.2. Visión Estéreo

El término *estéreo* en visión computacional se utiliza cuando existe más de una vista en una escena. La relación que existe entre las diferentes vistas es importante para la reconstrucción de escena más grande (mosaico) y también se puede tener la idea de las características tridimensionales de la escena en estudio (reconstrucción 3D).

### 2.2.1. Análisis Bifocal

En el análisis bifocal se tiene un sistema de visión con dos cámaras, o bien una sola cámara que toma dos imágenes del objeto de estudio en dos tiempos distintos, suponiendo que en este tiempo la cámara o el objeto se han movido. Con la teoría expuesta en esta sección se puede deducir la solución al problema de dos vistas distintas con una sola cámara.

La geometría de dos vistas es conocida como la *Geometría Epipolar*. El término epipolar viene del griego *epi* que significa sobre, encima y *polos* cuyo significado es punto de atracción o uno de los dos puntos de una esfera que son intersectados por su eje de rotación. La Geometría Epipolar lleva este nombre porque, como se verá más adelante, a cada una de las dos imágenes se le asocia un *epipolo*. A continuación se describe la *Geometría Epipolar*.

### Geometría Epipolar

La geometría epipolar es la geometría proyectiva intrínseca entre dos vistas [Hartley y Zisserman, 2004]. Es independiente de la estructura de la escena y sólo depende de la posición y parámetros internos de la cámara.

La matriz fundamental encapsula esta geometría intrínseca y es independiente de la estructura de la escena. Sin embargo, puede ser calculada desde los puntos señalados en la imagen sin requerir conocimiento de los parámetros internos o la posición relativa de la cámara.

Vamos a entender bien el concepto de geometría epipolar; supongamos que un punto  $X$  en el espacio 3D es proyectado en dos vistas [Hartley y Zisserman, 2004], denominaremos  $x$  en la primer vista y  $x'$  en la segunda vista, como se muestra en la figura 2.3(a), los puntos de la imagen  $x$ ,  $x'$ , el punto en el espacio  $X$  y los centros de las cámaras ( $c$  y  $c'$ ) son coplanares (forman un plano  $\pi$ ). Si sólo conocemos a  $x$ , el plano de  $\pi$  está determinado, por la línea que que pasa por el centro de la primer cámara  $C$  y el punto  $x$  y por la línea que une los centros de las dos cámaras (llamada línea base). El punto  $x'$  cae sobre la línea de intersección del plano  $\pi$  con el plano de la segunda imagen (figura 2.3(b)).

Los términos utilizados en la geometría epipolar son los siguientes [Hartley y Zisserman, 2004]:

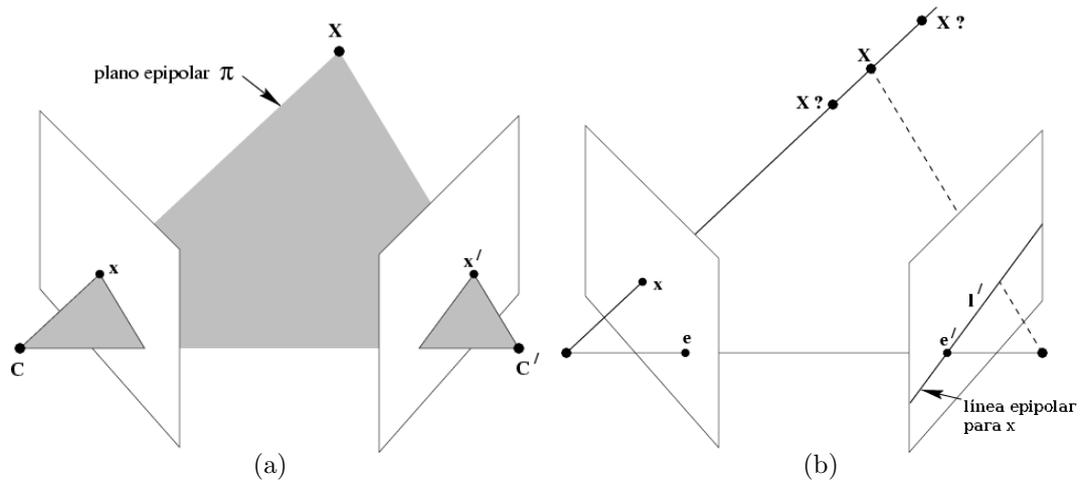


Figura 2.3: Geometría de puntos correspondientes. (a) Los centros de las cámaras junto con un punto tridimensional  $X$  forman el plano  $\pi$  en donde también se encuentran los puntos correspondientes. (b) Se muestran los epipolos localizados en la intersección de la línea que une los centros de las cámaras y los planos de la imagen, también se observa la línea epipolar correspondiente al punto  $x$ . Imágenes tomadas de [Hartley y Zisserman, 2004].

- **Epipolo.** Es el punto de intersección de la línea base con el plano de la imagen.
- **Plano epipolar.** Es un plano que contiene la línea base.
- **Línea epipolar.** Es la intersección de un plano epipolar con el plano de la imagen

En la figura 2.4 se muestra la geometría epipolar entre un par de imágenes.

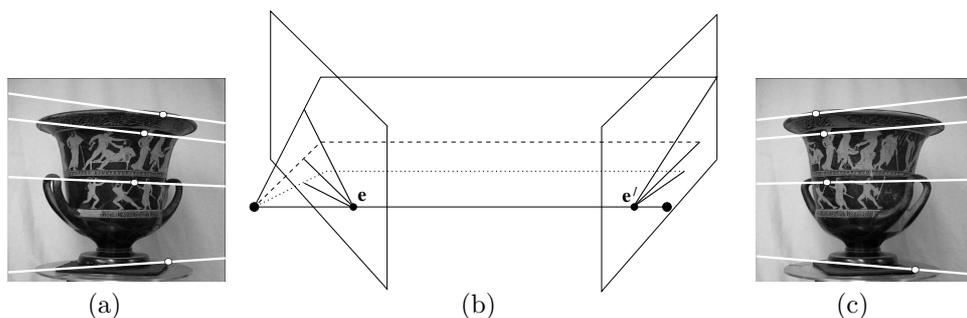


Figura 2.4: Esquema de líneas epipolares. (a) y (c) Dos vistas epipolares con líneas epipolares sobrepuestas. (b) Geometría epipolar de dos vistas. Fuente tomada [Hartley y Zisserman, 2004]

En la figura 2.5 muestra dos planos prácticamente paralelos de la misma escena.

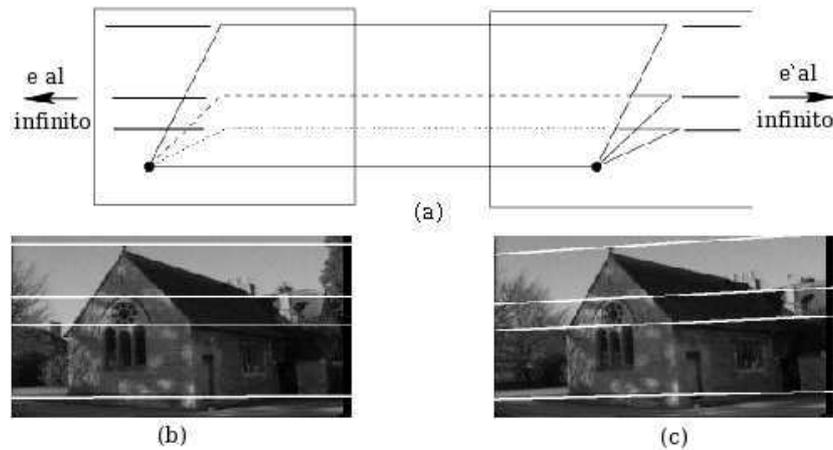


Figura 2.5: Esquema de líneas epipolares. (a) Líneas epipolares. (b) Dos vistas con líneas epipolares superpuestas, las cuales están en el infinito. Fuente tomada [Hartley y Zisserman, 2004].

Se ve claramente que las líneas epipolares son paralelas, lo que indica que están en el infinito.

### Estimación de la Matriz Fundamental

Como se describió anteriormente, existe una matriz que encapsula la geometría epipolar: la matriz fundamental, la cual denotaremos con  $F$ , es una matriz de  $3 \times 3$  y de rango 2 y es la representación algebraica de la geometría epipolar. Dado un par de imágenes, como en figura 2.3 la matriz fundamental relaciona a cada punto  $x$  en la primera imagen con su correspondiente punto  $x'$ , el cual cae en la línea epipolar ( $l'$ ) en la segunda imagen. La línea epipolar es la proyección en la segunda imagen del rayo que atraviesa del punto  $x$  al centro de la cámara. Ésto se puede escribir:

$$x \rightarrow l' \quad (2.10)$$

Los puntos que relaciona la matriz fundamental satisfacen la relación  $x'^T F x = 0$ . La matriz  $F$  puede encontrarse dada una serie de correspondencias entre dos imágenes.

Existen otros métodos para el cálculo de la matriz fundamental, la mayoría de ellos enfocados a la minimización de alguna función de distancia, ejemplos de estos métodos pueden ser encontrados en Hartley y Zisserman [2004]. El algoritmo más usado y sencillo para el cálculo de la matriz fundamental es presentado a continuación.

## El Algoritmo de los Ocho Puntos Normalizado

El algoritmo de los ocho puntos involucra la construcción y solución de un conjunto de ecuaciones lineales. El algoritmo original fue propuesto por Longuet-Higgins [1981] y aunque por mucho tiempo sufrió de críticas debido a su sensibilidad al ruido sólo fue necesaria una pequeña modificación [Hartley, 1997] para convertirlo en un algoritmo confiable y que se desempeñara bien, incluso mejor que varios algoritmos iterativos. Como la matriz fundamental es una matriz de  $3 \times 3$  determinada hasta un factor de escala, sólo son necesarias 8 ecuaciones para obtener una solución única. La forma más simple de calcular es utilizando la ecuación  $x'^T F x = 0$  que puede reescribirse de la siguiente manera:

$$\begin{bmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{bmatrix} f = 0 \quad (2.11)$$

donde  $f$  es un vector que contiene las 9 entradas de la matriz  $\mathbf{F}$ .

Agrupando ocho de estas ecuaciones (una por cada correspondencia) en una matriz  $\mathbf{A}$  se obtiene la ecuación:

$$\mathbf{A}f = 0 \quad (2.12)$$

Este sistema se resuelve de manera sencilla utilizando SVD (Singular Value Decomposition) [Burden y Faires, 1985]. Aplicando esta descomposición a la matriz  $\mathbf{A}$  obtenemos  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  con  $\mathbf{U}$  y  $\mathbf{V}$  matrices ortogonales y  $\mathbf{S}$  una matriz diagonal que contiene los valores singulares en orden decreciente. La solución de nuestro sistema es el vector singular correspondiente al menor valor singular el cual está dado por la última columna  $\mathbf{V}$ .

Una de las propiedades de la matriz fundamental es su singularidad, de hecho como ya se ha mencionado tiene un rango 2. Muchas aplicaciones de la matriz fundamental dependen de esa propiedad por lo cual debemos asegurarnos de que se cumpla. La matriz  $\mathbf{F}$  obtenida como se describió anteriormente, no tiene rango 2 en general. Una manera de forzar esta restricción es sustituyendo  $\mathbf{F}$  por  $\mathbf{F}'$  que minimice la norma de Frobenius  $\|\mathbf{F} - \mathbf{F}'\|$  sujeta a la condición  $\det \mathbf{F}' = 0$ . Para realizar esto Tsai y Huang [1984] propusieron un método que minimiza la norma de Frobenius  $\|\mathbf{F} - \mathbf{F}'\|$  y consiste en lo siguiente: dada  $\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  donde  $\mathbf{U}\mathbf{S}\mathbf{V}^T$  es la descomposición SVD de  $\mathbf{F}$  y  $\mathbf{S}$  es una matriz diagonal  $\mathbf{S} = \text{diag}(r, s, t)$  con  $r \geq s \geq t$ , entonces  $\mathbf{F}'$  está dada por  $\mathbf{F}' = \mathbf{U}\text{diag}(r, s, 0)\mathbf{V}^T$ .

La normalización fue propuesta por primera vez por Hartley [1997]. Esta normalización consiste en calcular transformaciones de similitud  $\mathbf{T}$  y  $\mathbf{T}'$  correspondientes a los puntos de cada imagen de tal manera que su centroide sea el origen coordenado y cuya distancia promedio del origen sea  $\sqrt{2}$ . Dichas transformaciones están dadas por:

$$T = \begin{bmatrix} k & 0 & -k\bar{x} \\ 0 & k & -k\bar{y} \\ 0 & 0 & 1 \end{bmatrix} \quad T' = \begin{bmatrix} k' & 0 & -k'\bar{x}' \\ 0 & k' & -k'\bar{y}' \\ 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

donde:

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N x_i, \quad k = \frac{\sqrt{2}}{\frac{1}{N} \sum_{i=1}^N \|x_i - \bar{\mathbf{x}}\|} \quad (2.14)$$

$$\bar{\mathbf{x}}' = \begin{bmatrix} \bar{x}' \\ \bar{y}' \\ \bar{z}' \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N x'_i, \quad k' = \frac{\sqrt{2}}{\frac{1}{N} \sum_{i=1}^N \|x'_i - \bar{\mathbf{x}}'\|} \quad (2.15)$$

Se aplican las transformaciones  $\mathbf{T}$  y  $\mathbf{T}'$  para obtener los puntos normalizados de la siguiente manera:  $\tilde{x}_i = T x_i$  y  $\tilde{x}'_i = T' x'_i$ .

Realizar la normalización antes de formular las ecuaciones lineales lleva a una mejora sustancial en el condicionamiento del problema y por lo tanto a la estabilidad del resultado. Una vez realizada la normalización de las correspondencias se sigue al cálculo de la matriz fundamental como se describió anteriormente. Después para obtener la matriz  $\mathbf{F}$  buscada tenemos que realizar la desnormalización de la matriz  $\hat{F}$  encontrada de la siguiente manera:

$$\mathbf{F} = \mathbf{T}'^T \hat{F} \mathbf{T}$$

En el cuadro 2.1 se muestra el algoritmo de los ocho puntos normalizados.

## 2.3. Capítulo Siguiete

Ya que hemos estudiado los conceptos generales para poder entender los capítulos que siguen, ahora podemos empezar a describir nuestro trabajo. En el capítulo siguiente empezamos por describir el método utilizado para la extracción de puntos característicos, seguido del método implementado para la correspondencia de puntos y por último se implementan técnicas para eliminar correspondencias erróneas.

Objetivo:

Dadas  $n \geq 8$  correspondencias de puntos  $x_i \leftrightarrow x'_i$ , determinar la matriz fundamental  $\mathbf{F}$  tal que  $x_i'^T \mathbf{F} x_i = 0$ .

Algoritmo:

1. **Normalización:** Transforma las coordenadas de acuerdo a  $\hat{x}_i = T x_i$  y  $\hat{x}'_i = T' x'_i$ , donde  $T$  y  $T'$  son transformaciones de normalización que consisten en una traslación y un escalamiento.
2. Encuentra la matriz fundamental  $\hat{F}'$  con las correspondencias  $\hat{x}_i \leftrightarrow \hat{x}'_i$ :
  - **Solución Lineal:** Determina  $\hat{F}$  del vector singular correspondiente al menor valor singular de la matriz  $\hat{A}$ , la cual está compuesta por las correspondencias  $\hat{x}_i \leftrightarrow \hat{x}'_i$  de la manera descrita en la fórmula 2.12.
  - **Restricción:** Sustituye  $\hat{F}$  por  $\hat{F}'$  que  $\det \hat{F}' = 0$  usando SVD.
3. **Denormalización:**  $F = T'^T \hat{F}' T$ . La matriz  $F$  es la matriz fundamental correspondiente a los datos originales  $x_i \leftrightarrow x'_i$ .

Cuadro 2.1: Algoritmo de los 8 puntos normalizados. Tomado de Hartley y Zisserman [2004].

# Capítulo 3

## Relacionando las Imágenes

En esta capítulo se analizarán distintas técnicas para establecer la correspondencia entre elementos que están presentes en varias imágenes. Para establecer una relación geométrica entre las imágenes no es necesario determinar la relación entre todos los puntos de la imagen, con un subconjunto de puntos de correspondencia es suficiente para determinar la relación geométrica entre imágenes. El primer paso consiste en seleccionar un número de puntos de interés o puntos característicos (sección 3.1). En el segundo paso, los puntos característicos son comparados y se obtiene un número de puntos de correspondencias (sección 3.2). En el tercer paso, el conjunto de puntos de correspondencia obtenidas generalmente contiene un número importante de correspondencias erróneas. Para filtrar y desaparecer el total o el mayor número de correspondencias erróneas se implementan técnicas para eliminar correspondencias erróneas (sección 3.3).

### 3.1. Extracción de Puntos Característicos

Hay dos requerimientos importantes para puntos característicos: primero, la correspondencia de puntos a la misma escena podrán ser extraídos consistentemente sobre las diferentes vistas. Si éste no es el caso, podría ser imposible encontrar suficiente correspondencia entre ellos. Segundo, debería existir suficiente información en los vecinos de los puntos de tal manera que la correspondencia se pueda hacer automáticamente. Pollefeys [2000] cita varios algoritmos para extraer puntos característicos que han sido propuestos para diferentes tipos de imágenes. En la literatura a estos puntos característicos se les llama *esquinas*; en nuestro caso particular por el tipo de imagen utilizadas les llamamos *bifurcaciones* y *cruces*.

Un buen detector de esquinas debe satisfacer varios criterios tales como: detectar todas las esquinas verdaderas, buena localización, robusto respecto al ruido, estable y ser computacionalmente eficiente. Los métodos encontrados en la literatura sobre técnicas para extraer puntos característicos generalmente hacen una mezcla de diferentes técnicas. Se puede encontrar un clasificación de técnicas para extraer puntos

característicos en Patrón Pérez [2006].

Algunos autores han clasificado estos métodos en tres categorías principales: los métodos relacionados con los bordes, los topológicos y los de auto-correlación. Existen métodos que utilizan otras características de la imagen, como líneas o curvas en lugar de puntos para establecer las correspondencias; una breve descripción de estos se puede encontrar en Hartley y Zisserman [2004]. En el caso de imágenes de fondo de ojo se hace una combinación de aplicaciones morfológicas, operaciones de convolución y topología.

A continuación se describe el método presentado en este trabajo de tesis, el cual es un detector creado para imágenes de fondo de ojo, basado en topología y detección morfológica.

### 3.1.1. Detector de Bifurcaciones y Cruces RISA

La extracción de puntos característicos en las imágenes de fondo de ojo que corresponden a cruces y bifurcaciones es realizado de manera automática con el programa RISA (**R**etinal **I**mage multi**S**cale **A**nalysis) [Martínez-Pérez *et al.*, 2007, 2002]. Para obtener los cruces y bifurcaciones RISA se llevan a cabo dos etapas.

En la primer etapa se lleva a cabo un proceso de segmentación de los vasos sanguíneos utilizando un algoritmo basado en análisis multiescala [Martínez-Pérez *et al.*, 2007] y a partir de dos características geométricas basadas en la primera y segunda derivada de la intensidad de la imagen (gradientes y curvatura principal máxima), utilizan un procedimiento de crecimiento de regiones de múltiples pasadas para segmentar los vasos sanguíneos utilizando la información anterior junto con información espacial sobre ocho píxeles vecinos. En la figura 3.1 se muestra la imagen que fue tomada a un sujeto y el resultado de la segmentación.

La segunda etapa consiste en extraer de la imagen binaria los árboles sanguíneos [Martínez-Pérez *et al.*, 2002]. Para esto se obtiene el esqueleto del árbol a través del proceso de adelgazamiento cuidando de no destruir la conectividad existente con un esquema de conectividad-ocho. La figura 3.2 muestra un ejemplo.

Basándose en el esqueleto se hacen dos filtros para obtener los puntos significativos, así como otros datos geométricos como son la longitud, área y ángulo. El resultado de estos filtros es una clasificación de los puntos significativos en tres tipos: terminales, bifurcaciones y cruces. Los filtros consisten en lo siguiente:

1. En el primer filtro, los píxeles con un solo vecino son etiquetados como puntos terminales, los que tengan tres vecinos, son etiquetados como bifurcaciones candidatas. En este filtro se pueden etiquetar puntos significativos que no son bifurcaciones y cruces.

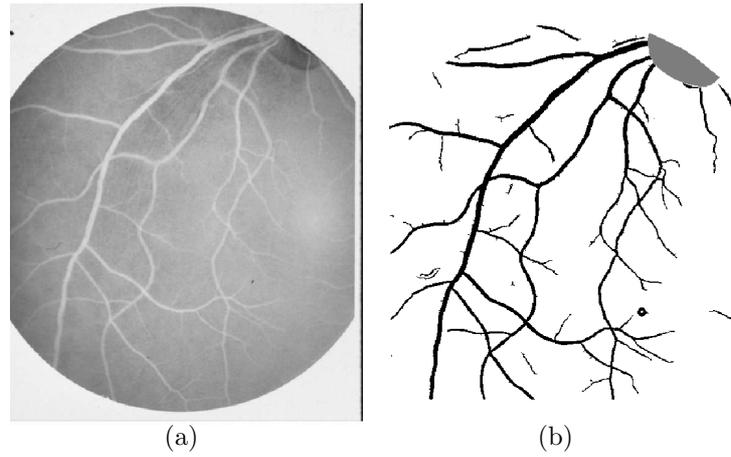


Figura 3.1: (a) Imagen tomada de un sujeto. (b) Imagen segmentada. Imágenes tomadas de [Martínez-Pérez, 2000].

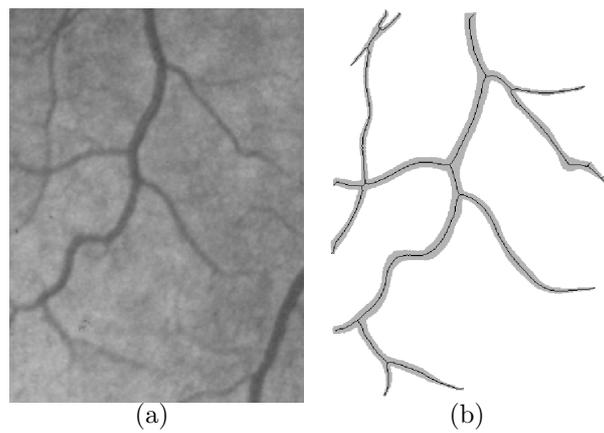


Figura 3.2: Imágenes del proceso de segmentación. (a) Una pequeña sección de la imagen original. (b) El árbol segmentado en gris y el esqueleto en negro. Imágenes tomadas de [Martínez-Pérez, 2000].

- En el segundo filtro, se toman los puntos característicos candidatos y se coloca una ventana circular de tamaño fijo centrada en el punto de bifurcación para detectar los puntos significativos “verdaderos”, los cuales aparecen en el esqueleto como dos puntos de bifurcación muy cercanos uno del otro.

En la figura 3.3 (a), se muestra con un círculo los puntos característicos marcados como candidatos y en la figura 3.3 (b) se muestra un resumen de los dos filtros hechos.

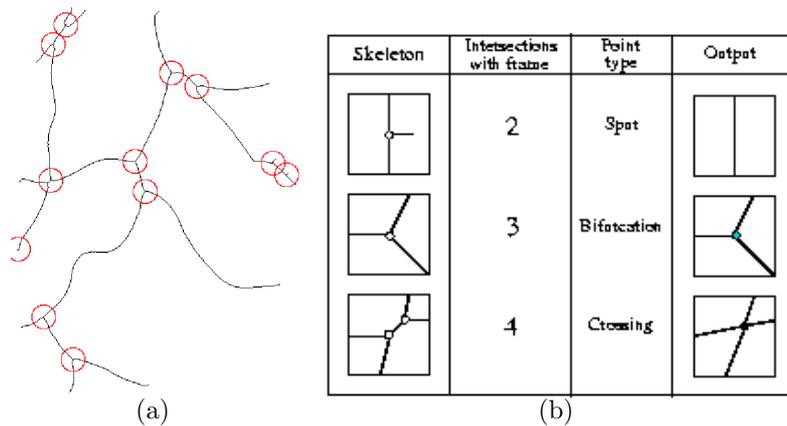


Figura 3.3: (a) Puntos candidatos marcados con un círculo. (b) La primer columna (‘Skeleton’) muestra los puntos candidatos obtenidos en el primer filtro, la segunda columna (‘Intersections with feature’) muestra el tipo de punto, la cuarta columna (‘Output’) muestra el resultado del segundo filtro, bifurcaciones y cruces verdaderos. Imágenes tomadas de [Martínez-Pérez, 2000].

Finalmente, se tiene el árbol completo con el esqueleto marcado. Los puntos característicos significativos marcados como bifurcaciones y cruces son los puntos que utilizaremos para poder hacer la correspondencia de puntos en la sección.

El conjunto de puntos de *bifurcaciones y cruces* que se utilizaran a lo largo del trabajo presentado serán los obtenidos en el segundo filtro de RISA.

### 3.1.2. Resultados

Un ejemplo del resultado del aplicar RISA a dos imágenes de fondo de ojo para extraer los puntos de *bifurcaciones y cruces* se muestra en la figura 3.4.

Para la figura 3.4 (a) RISA obtiene en el primer filtro 328 puntos candidatos (ver figura 3.4 (c)). En el segundo filtro sobre esta misma figura RISA obtiene 59 puntos (ver figura 3.4 (e)), el resultado es un número menor de puntos pero con una mayor confiabilidad de que son bifurcaciones y cruces “verdaderos”. Este mismo procedimiento se hace para la figura 3.4(b) donde se encuentran en el primer filtro 330 puntos candidatos (ver figura 3.4(d)) y en el segundo filtro se encuentra 52 bifurcaciones y cruces “verdaderos” (ver figura 3.4 (f)).

## 3.2. Cálculo de Correspondencias

Una vez obtenidos los puntos característicos en las imágenes debemos encontrar una manera de relacionarlos. Esto puede ser muy difícil debido a distintos problemas que se pueden presentar tales como oclusiones, es decir, algunos puntos tal vez no tengan una correspondencia debido a la presencia de algún objeto que obstruye la visibilidad del punto correspondiente. Otro problema lo constituyen las correspondencias erróneas, esto se da debido a la presencia de puntos similares en la vecindad de la correspondencia verdadera. También puede haber cambios en la intensidad en puntos correspondientes de una imagen a otra, debido al cambio de dirección de la luz, ruido o cambio de tamaño entre otros.

Un primer paso para encontrar correspondencias consiste en definir alguna medida que nos permita establecer cuándo dos puntos característicos en diferentes imágenes son similares. Se han propuesto varias medidas en la literatura de visión computacional [Pollefeys, 2000]. Sin embargo, las técnicas más comunes para encontrar correspondencias se pueden dividir de manera general en dos ramas: las basadas en correlación y las basadas en rasgos. Las primeras consisten en calcular la correlación entre la distribución de disparidad de una ventana centrada en un punto de la imagen y una ventana del mismo tamaño centrada en el punto a analizar de la otra imagen. Por otro lado las técnicas basadas en rasgos comparan primitivas de alto nivel, tales como bordes, segmentos, curvas y regiones, las cuales poseen un conjunto de características invariantes a la proyección en mayor o menor medida.

Algunas medidas típicas de correlación son: correlación directa, correlación de media normalizada, correlación de varianza normalizada, sumas de cuadrados de las diferencias entre píxeles correspondientes (SSD), suma de las magnitudes de las diferencias, correlación cruzada normalizada (ZNCC), entre otras. Un estudio comparativo de algunas de estas medidas se puede encontrar en Burt *et al.* [1982]. En este trabajo se utiliza un método basado en correlación, el método ZNCC que se describe a continuación.

Cabe recordar que seguimos los algoritmos propuestos por Zhang *et al.* [1994] para el registro de imágenes. El ZNCC es más costoso computacionalmente que el SSD, aunque es más estable. En nuestro caso particular el número de correspondencias son pequeñas

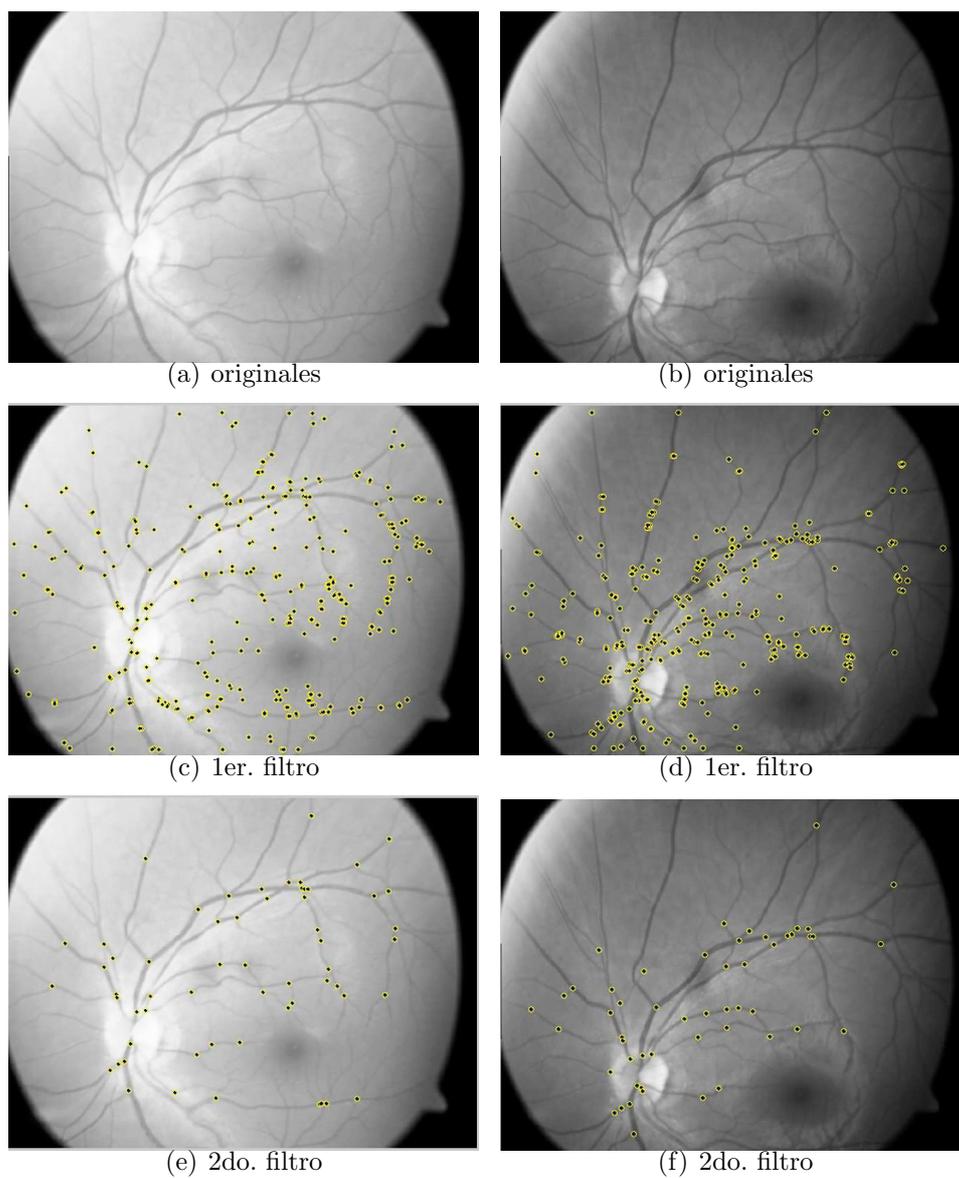


Figura 3.4: Resultado de RISA. (a)(b) Dos imágenes de fondo de ojo del mismo sujeto, (c)(d) resultados del primer filtro de RISA (328 y 330 puntos respectivamente), (e)(f) resultados del segundo filtro de RISA (59 y 52 puntos respectivamente).

(entre 30 y 50 puntos correspondientes aproximadamente) y no cuesta demasiado usar una medida de correlación cara como el ZNCC.

### 3.2.1. Correlación Cruzada Normalizada de Media Cero (ZNCC)

Antes de explicar el método de ZNCC (Zero Mean Normalized Cross Correlation), vamos a entender cómo funciona la técnica de correlación.

Un buen resultado de correspondencia podría ser alcanzado tomando la información de varios píxeles [Pollefeys, 2000]. Esto podría ser llevado a cabo comparando la vecindad a través de la correlación cruzada de sus intensidades. La vecindad es especificada como una pequeña ventana de  $(2N + 1) \times (2N + 1)$  píxeles centrada alrededor de un punto seleccionado. Para los puntos  $(x, y)$  y  $(x', y')$  la medida de similitud es obtenida de la manera siguiente:

$$C = \sum_{i=-N}^N \sum_{j=-N}^N \left( I(x - i, y - j) - \bar{I} \right) \left( I'(x' - i, y' - j) - \bar{I}' \right) \quad (3.1)$$

Donde  $I$  e  $I'$  son los valores de intensidad de cierto punto. Y los valores de  $\bar{I}$  y  $\bar{I}'$  representan la intensidad media (promedio de la intensidad) en la vecindad considerada.

El método de correlación cruzada es invariante a traslaciones y cambios en la intensidad de la imagen. Si existe una importante rotación o escalamiento, la medida de similitud no es la apropiada; lo mismo cuando las condiciones de luz difieren mucho. De cualquier forma el enfoque de la correlación cruzada se basa en que sólo puede ser usada en imágenes para las cuales la posición de la cámara no es trasladada demasiado.

Si se considera que el desplazamiento entre las imágenes es pequeño entonces se puede reducir la complejidad combinatoria del algoritmo; es decir, en lugar de comparar cada esquina encontrada en una imagen con todas las esquinas en la otra imagen, sólo calculamos la similitud con las esquinas que se encuentran dentro de una vecindad cercana. Esto es, de manera más precisa, dada una esquina localizada en la posición  $(x, y)$  en una imagen, ésta sólo se compara con esquinas en la otra imagen localizadas en el intervalo  $[x - w_i, x + w_i] \times [y - h_i, y + h_i]$ , donde usualmente el tamaño de  $w_i$  y  $h_i$  corresponden con el 10% o el 20% del tamaño de la imagen.

Existe una forma normalizada de la correlación cruzada, llamada **ZNCC** (Zero mean Normalized Cross Correlation) dada por:

$$C = \frac{\sum_{i=-N}^N \sum_{j=-N}^N \left( I(x - i, y - j) - \bar{I} \right) \left( I'(x' - i, y' - j) - \bar{I}' \right)}{\sqrt{\sum_{i=-N}^N \sum_{j=-N}^N \left( I(x - i, y - j) - \bar{I} \right)^2 \left( I'(x' - i, y' - j) - \bar{I}' \right)^2}} \quad (3.2)$$

Nosotros usamos dos ventanas para comparar las vecindades a través del ZNCC. En la primera imagen colocamos una ventana de tamaño  $(2n + 1) \times (2m + 1)$  centrada en un punto  $m_i$  (a esta ventana le llamamos *ventana de correlación*). En la segunda imagen seleccionamos un área de investigación de tamaño  $(2dx + 1) \times (2dy + 1)$  (a ésta le llamamos *ventana de búsqueda*) y realizamos la operación de correlación con el ZNCC barriendo la ventana de correlación dentro del área marcada por la ventana de búsqueda en la segunda imagen, comparando el punto  $m_i$  con todos los puntos  $m_j$  que caen dentro de la ventana de búsqueda. En la figura 3.5 se muestra lo anterior de forma gráfica.

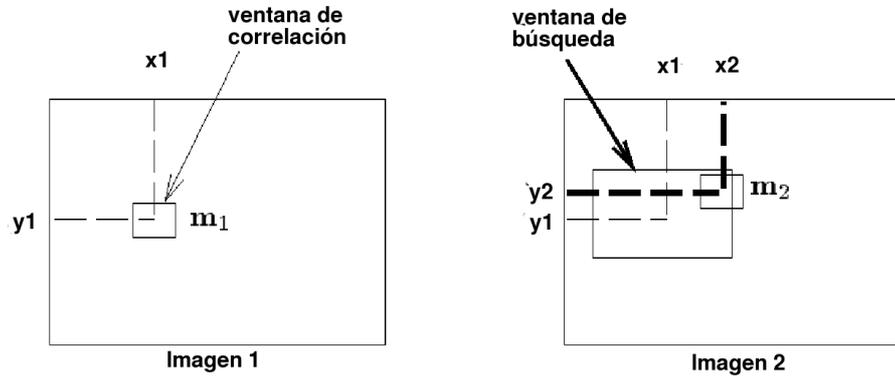


Figura 3.5: Correlación. Correlación del punto  $m_1$  contra todos los puntos que caen en la ventana de búsqueda de la imagen 2.

Para un par de puntos de correspondencias candidatas, el valor de correlación debe ser más alto que un umbral [Zhang *et al.*, 1994]. En nuestra implementación para las imágenes de fondo de ojo utilizadas, el umbral es definido a 0.5 (este valor se obtuvo de manera experimental). Los pares de puntos que tengan un valor menor al umbral son eliminados.

Otra restricción impuesta por nosotros es que después de hacer la correlación, analizamos el ángulo de rotación de cada vector que es formado por un punto en la imagen 1 a un punto en la imagen 2. Estos puntos deben estar entre un rango  $\theta$ .

El rango de  $\theta$  se establece siguiendo el algoritmo mostrado en el cuadro 3.1.

En la figura 3.6 se muestra gráficamente la obtención del rango de  $\theta$ .

La hipótesis de este ángulo está basado en que la rotación de la cámara es mínima, en el caso particular de las imágenes de fondo de ojo.

1. Se obtiene de cada vector el ángulo que forma con respecto al eje  $x$ .
2. Se obtiene el ángulo que más se repita (la moda).
3. Se establece un rango sumando  $30^\circ$  y restando  $30^\circ$  al ángulo obtenido en el paso anterior (moda). El ángulo que sumamos y restamos puede variar dependiendo de la imagen utilizada.

El rango  $\theta$  es establecido a no mayor de  $60^\circ$ .

Cuadro 3.1: Algoritmo para obtener el rango de  $\theta$ .

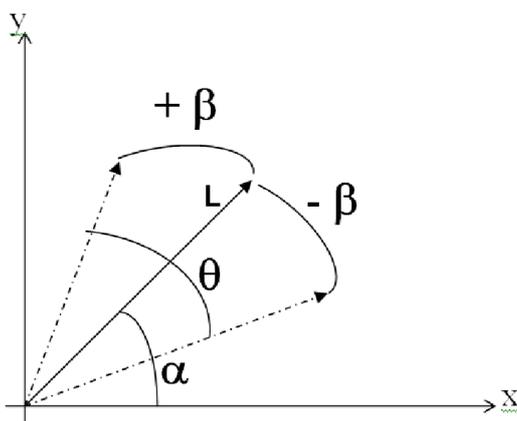


Figura 3.6: Rango de ángulo  $\theta$ . Donde,  $L$  es el vector formado por un punto  $m_1$  a un punto  $m_2$ ;  $\alpha$  es el ángulo obtenido de la moda;  $\theta$  es el rango entre  $-\beta$  y  $+\beta$ .

### 3.2.2. Resultados

Los resultados obtenidos con este método fueron buenos en general. Una ventaja del ZNCC es que el valor dado por esta medida siempre se encuentra entre un rango de  $[1, -1]$  lo cual facilita su uso. Sin embargo, aún existen *correspondencias erróneas* por lo que es necesario utilizar técnicas de eliminación para descartar estos puntos erróneos.

La figura 3.7 muestra el resultado de aplicar el proceso de correlación cruzada normalizada (ZNCC). Las imágenes utilizadas en esta prueba tienen una resolución de  $576 \times 768$  píxeles. Los puntos de *bifurcación y cruce* encontradas por RISA para la figura 3.7 (a) son 59 puntos y para la figura 3.7 (b) 52 puntos. El resultado del método implementado (ZNCC) arrojó 33 correspondencias (ver figura 3.7 (c)). El tamaño de la ventana de búsqueda fue de  $200 \times 200$  píxeles y el tamaño de la ventana de correlación fue de  $25 \times 25$  píxeles. El umbral utilizado fue de 0,5 y el rango del ángulo fue entre

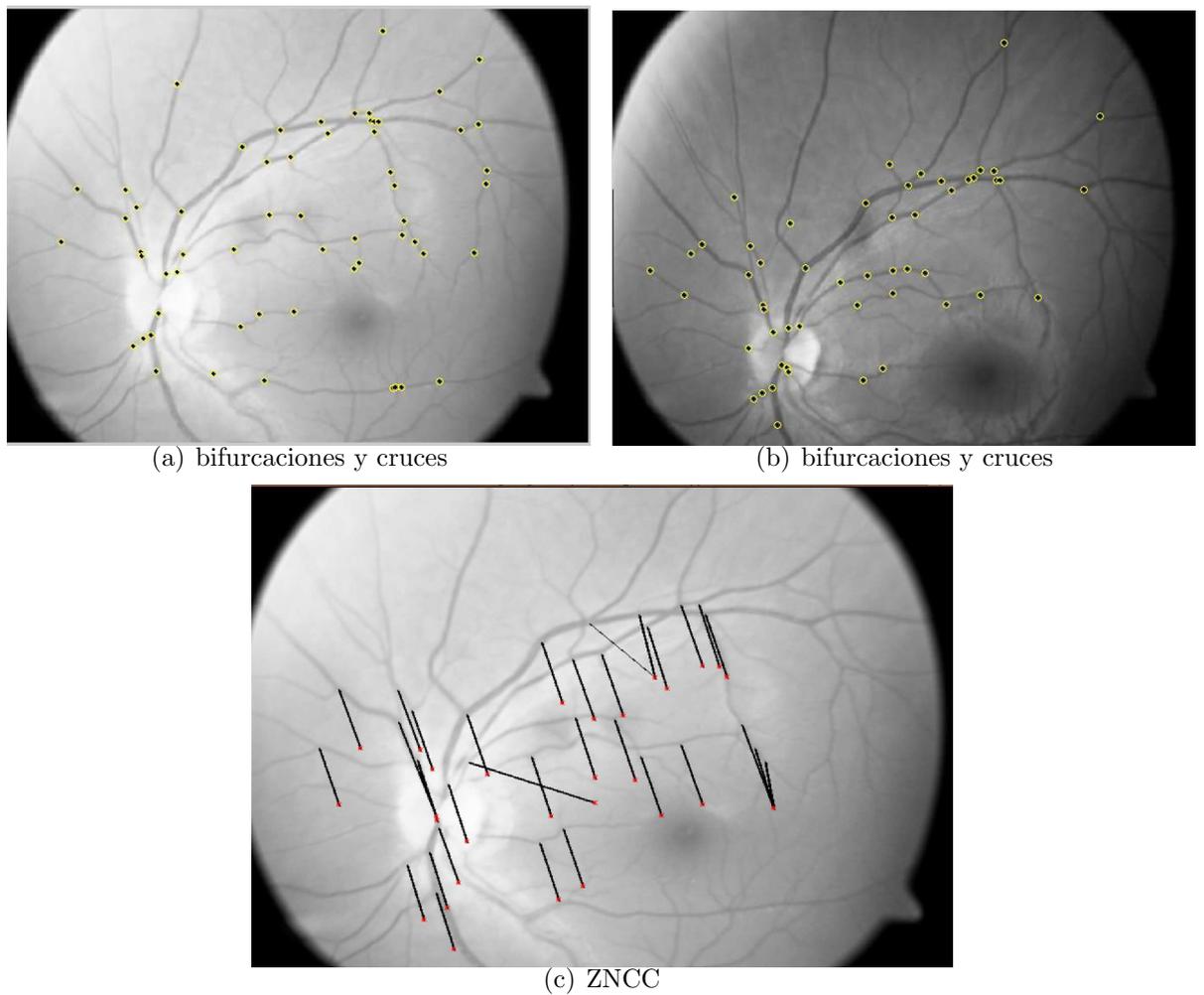


Figura 3.7: (a y b) Cruces y bifurcaciones encontrados por RISA en un par de vistas, (c) Conjunto de puntos obtenidos por el método de ZNCC.

$262^{\circ}$ - $322^{\circ}$ . El conjunto de correspondencia obtenido contiene un número de correspondencias erróneas. Se pueden verificar los puntos *buenos* y *erróneos* visualmente; en este resultado existen 5 correspondencias erróneas, las cuales tienen una dirección diferente al resto de los vectores. Para eliminar las correspondencias erróneas utilizaremos técnicas de eliminación de errores.

### 3.3. Eliminando Correspondencias Erróneas

En la sección 3.2 obtuvimos un conjunto de correspondencias en donde no todos los puntos son “buenos”. En la literatura podemos encontrar algunas técnicas para

resolver las correspondencias erróneas [Hartley y Zisserman, 2004; Pollefeys, 2000].

En este trabajo de tesis utilizamos dos metodos para eliminar correspondencias erróneas:

1. El primer método cae dentro de las técnicas de *relajación*. El nombre de ésta técnica es tomada de los métodos numéricos iterativos a los que se asemeja. La idea es permitir a los puntos de correspondencias reorganizarse por si mismos por propagación basados en algunas reglas, tales como continuidad y singularidad en referencia a sus vecinos. Esto se obtiene calculando un valor de similitud y posteriormente este valor de similitud es actualizado iterativamente hasta que converja o hasta encontrar suficientes correspondencias buenas. Sin embargo, la convergencia y la complejidad de este acercamiento no siempre está bien definida, y en la practica a menudo son necesarias técnicas robustas. Nosotros utilizamos esta técnica para eliminar correspondencias ambiguas; llamaremos ambiguas a la situación en que un punto en la imagen 1 está apareado con varios puntos en la imagen 2. Básicamente utilizamos esta técnica para obtener un conjunto de correspondencias erróneas menor antes de utilizar un método robusto.
2. *Métodos robustos clásicos*. Se les llama robusto por que son sensibles en la identificación de *correspondencias erróneas*. Hay muchos tipos de algoritmos robustos, los cuales se escogen dependiendo de la proporción de *correspondencias erróneas*.

En el trabajo de tesis presentado, se implementa un algoritmo del tipo de relajación para quitar correspondencias ambiguas (relación de uno a muchos entre un par de imágenes) y un algoritmo robusto, el LMedS (Least Median of Squares) para detectar correspondencias erróneas.

### 3.3.1. Proceso de Relajación

Para realizar el proceso de relajación primero debemos establecer una medida de soporte para las correspondencias, es decir, cómo determinamos el valor de correlación que nos indique que un par de puntos son correspondencias *buenas* o *erróneas*.

Considerando una correspondencia candidata  $(m_{1i}, m_{2j})$  donde  $m_{1i}$  es un punto en la primer imagen y  $m_{2j}$  es un punto en la segunda imagen, y  $N(m_{1i})$  y  $N(m_{2j})$  serán los vecinos de  $m_{1i}$  y  $m_{2j}$ , dentro de un disco de radio  $R$ , si  $(m_{1i}, m_{2j})$  es una buena correspondencia, entonces esperaremos ver muchas correspondencias en sus vecinos  $(n_{1k}, n_{2l})$ , donde  $n_{1k} \in N(m_{1i})$  y  $n_{2l} \in N(m_{2j})$ , tal que la posición de  $n_{1k}$  es relativa a  $m_{1i}$  de manera similar  $n_{2l}$  es relativo a  $m_{2j}$ . De otra manera si  $(m_{1i}, m_{2j})$  es una correspondencia errónea, entonces esperamos ver sólo algunas correspondencias o ninguna en sus vecinos. Se puede ver un ejemplo en la figura 3.8.

Se define una medida de soporte para la correspondencia, la cual llamaremos *fuerza de correspondencia* ( $S_M$ ). Básicamente esta medida se obtiene analizando la similitud

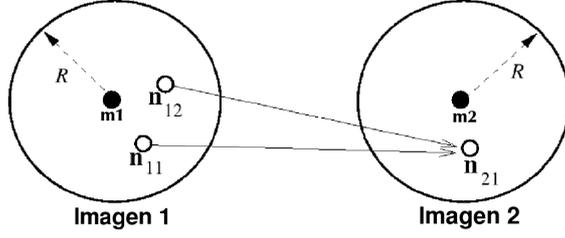


Figura 3.8: Proceso de relajación. Los círculos negros son los puntos de correspondencias candidatas  $m_{1i}, m_{2j}$ , los círculos vacíos son los vecinos que se encuentran dentro de una vecindad definida por un radio  $R$ .

que tiene un vector que va de un par de puntos seleccionado llamados  $(m_{1i}, m_{2j})$  con respecto a los vectores que forman sus vecinos  $(n_{1k}, n_{2l})$ , la similitud está basada en la posición relativa de  $n_{1k}$  con respecto a  $m_{1i}$  y lo mismo para  $n_{2k}$  con respecto a  $m_{2i}$ , esto está basado en la distancia relativa  $r$ , es decir. La similitud es justificada en la hipótesis de que una transformación afín puede aproximar el cambio entre un vecino y el punto candidato seleccionado. Ahora describimos algebraicamente la *fuerza de correspondencia* ( $S_M$ ) como:

$$S_M(m_{1i}, m_{2j}) = c_{ij} \sum_{n_{1k} \in N(m_{1i})} \left[ \max_{n_{2l} \in N(m_{2j})} \frac{c_{kl} \delta(m_{1i}, m_{2j}; n_{1k}, n_{2l})}{1 + \text{dist}(m_{1i}, m_{2j}; n_{1k}, n_{2l})} \right] \quad (3.3)$$

donde  $c_{ij}$  y  $c_{kl}$  son los valores de la correlación obtenidas con la técnica de ZNCC descrita en la sección 3.2.1,  $\text{dist}(m_{1i}, m_{2j}; n_{1k}, n_{2l})$  es la distancia promedio de dos pares de puntos.

$$\text{dist}(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = [d(m_{1i}, n_{1k}) + d(m_{2j}, n_{2l})] / 2 \quad (3.4)$$

con  $d(m, n) = \|m - n\|$ , es la distancia euclidiana entre  $\mathbf{m}$  y  $\mathbf{n}$ , y

$$\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = \begin{cases} e^{-r/\epsilon_r} & \text{si } (n_{1k}, n_{2l}) \text{ es una correspondencia candidata y } r < \epsilon_r \\ 0 & \text{de otra forma} \end{cases} \quad (3.5)$$

donde  $r$  es la diferencia de la distancia relativa dada por:

$$r = \frac{|d(m_{1i}, n_{1k}) - d(m_{2j}, n_{2l})|}{\text{dist}(m_{1i}, m_{2j}; n_{1k}, n_{2l})} \quad (3.6)$$

y  $\epsilon_r$  es un umbral sobre la diferencia de la distancia relativa.

Si definimos la función de energía como la suma de todas las fuerzas de correspondencias calculadas de la siguiente manera:

$$\vartheta = \sum_{(m_{1i}, m_{2j})} S_M(m_{1i}, m_{2j}) \quad (3.7)$$

entonces el problema de correspondencia ambigua es equivalente a la minimización de la función de energía  $\vartheta$ . El esquema de relajación es un enfoque a esto. Formulamos el procedimiento a continuación:

**ciclo**

{

- Calcula la *fuerza de correspondencia* ( $S_M$ ) para cada punto de correspondencia candidato.
- Actualiza las correspondencias minimizando el total de energía.

} hasta que converja la energía.

Cuadro 3.2: Proceso de relajación. Tomado de Zhang *et al.* [1994].

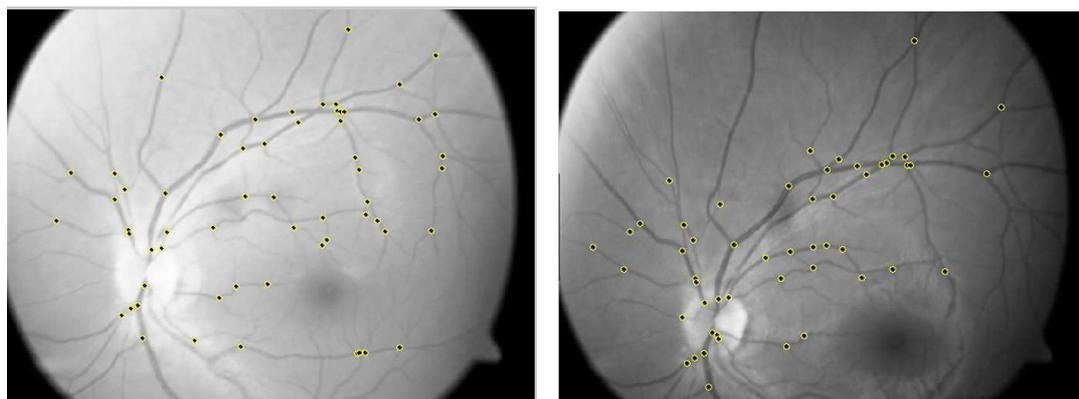
Se implementó una estrategia para actualizar las correspondencias y minimizar el total de energía la cual es propuesta por Zhang *et al.* [1994], algunos-ganadores-toman-todo (some-winners-take-all), que se describe a continuación. Consideran todos los puntos cuyos valores de *fuerza de correspondencia* es más alto. A tales pares de puntos de correspondencia les llaman *correspondencias potenciales* y es denotada por  $P_i$ . Para  $P_i$  se construyen dos tablas. La primera, denotada por  $T_{SM}$ , que guarda la *fuerza de correspondencia* de cada  $P_i$  y es ordenada en forma decreciente (figura 3.9 (a)). La segunda tabla es denotada por  $T_{UA}$ , que guarda valor el que indica la ambigüedad en cada  $P_i$  (figura 3.9 (b)). Este valor está dado por:

$$U_A = 1 - S_M^{(2)}/S_M^{(1)} \quad (3.8)$$

donde  $S_M^{(1)}$  es el valor de *fuerza de correspondencia* de  $P_i$  y  $S_M^{(2)}$  es la *fuerza de correspondencia* del segundo mejor par de correspondencia.

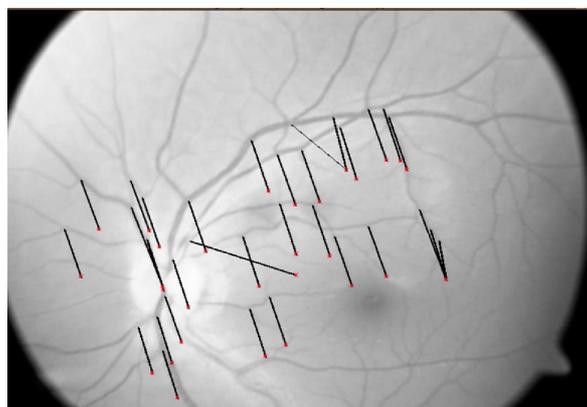
Así  $T_{UA}$  está en un rango entre 1 (no ambiguo) a 0 (ambiguo). La tabla  $T_{UA}$  es también ordenada en forma decreciente. Finalmente, aquellas correspondencias  $P_i$  las cuales están en los primeros  $q$  puntos de la tabla  $T_{SM}$  y entre los  $q$  primeros puntos en  $T_{UA}$  son seleccionados como correspondencias correctas (ver figura 3.9). Así, las correspondencias ambiguas no serán seleccionados aunque ellos tengan un  $S_M$  alto y aquellos que tengan un  $S_M$  débil no serán seleccionados aunque ellos no sean ambiguos. Si una correspondencia candidata no tiene soporte ( $S_M = 0$ ), será eliminada;  $q$  debe ser un valor mayor del al 50 % del total de elementos.



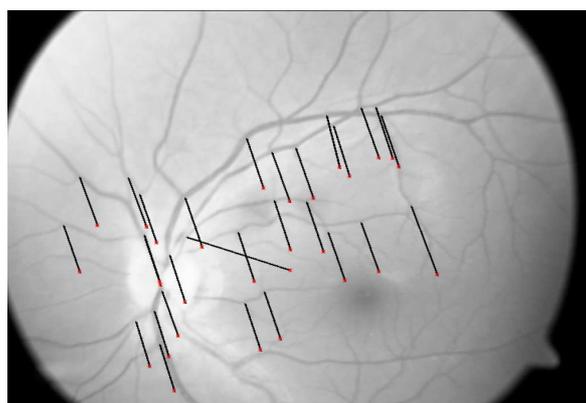


(a) bifurcaciones y cruces

(b) bifurcaciones y cruces



(c) ZNCC



(d) relajación

Figura 3.10: (a)(b) Cruces y bifurcaciones encontrados por RISA en un par de vistas, (c) Conjunto de puntos obtenidos por el método de ZNCC. (d) Conjunto de puntos obtenidos por el proceso de relajación.

por  $M_j$ , con respecto al conjunto total de correspondencias de puntos. Ésto se puede expresar como sigue:

$$M_j = \text{med}_{i=1, \dots, n} \left[ d^2 (\tilde{m}_{2i}, F_j \tilde{m}_{1i}) + d^2 (\tilde{m}_{1i}, F_j^\top \tilde{m}_{2i}) \right] \quad (3.9)$$

donde,  $d^2 (\tilde{m}_{2i}, F_j \tilde{m}_{1i}) + d^2 (\tilde{m}_{1i}, F_j^\top \tilde{m}_{2i})$ , es

$$\left( \frac{1}{(F \tilde{m}_{1i})_1^2 + (F \tilde{m}_{1i})_2^2} + \frac{1}{(F^\top \tilde{m}_{2i})_1^2 + (F^\top \tilde{m}_{2i})_2^2} \right) (\tilde{m}_{2i}^\top F \tilde{m}_{1i})^2 \quad (3.10)$$

Retenemos el  $F_j$  para el cual  $M_j$  es el mínimo de todos los  $M_j$ . La pregunta ahora es, ¿cómo determinamos a  $m$ ? Es computacionalmente ineficiente e innecesario obtener todos los posibles subconjuntos, entonces se toma  $m$  submuestras, las cuales son suficiente para asegurar una probabilidad  $p$  que asegure que al menos una de las muestras (de  $s$  puntos) seleccionada está libre de *correspondencias erróneas*. Suponiendo que  $\omega$  es la probabilidad de que un punto sea *correspondencia correcta*, entonces  $\epsilon = 1 - \omega$  es la probabilidad de que sea *correspondencia errónea*. Entonces, el número necesario de pruebas está dado por:

$$m = \log(1 - p) / \log(1 - (1 - \epsilon)^s) \quad (3.11)$$

donde  $s$  representa el número de datos que se toman en cada muestra.

Para mayor información sobre cómo encontrar a  $m$  referirse a Hartley y Zisserman [2004]; él muestra un tabla colocando a  $p = 0.99$  para valores de  $s$  y  $\epsilon$  dados.

La eficiencia del LMedS es pobre ante la presencia de ruido Gaussiano [Rousseeuw y Leroy, 1987]. La eficiencia de un método está definido como la razón entre la varianza de los parámetros estimados más baja y la varianza actual que provee el método dado. Para compensar esta deficiencia, se llevó a cabo el procedimiento de *mínimos-cuadrados de pesos*. La desviación estándar robusta estimada está dada por:

$$\hat{\sigma} = 1,4826 [1 + 5/(n - d)] \sqrt{M_j}, \quad (3.12)$$

donde  $d$  es la dimensión del espacio, en nuestro trabajo es igual a 2.  $n$  es el número total de elementos.  $M_j$  es la mediana mínima. La constante 1,4826 es un factor de corrección asintótica; la lectura de cómo se obtiene este número está en [Rousseeuw y Leroy, 1987]. Basándose en  $\sigma$ , se asigna un peso para cada correspondencia:

$$\omega_i = \begin{cases} 1 & \text{si } r_i^2 \leq (1,5\hat{\sigma})^2 \\ 0 & \text{de otra forma,} \end{cases} \quad (3.13)$$

donde

$$r_i^2 = d^2(\tilde{m}_{2i}, Fm_{1i}) + d^2(\tilde{m}_{1i}, F^T\tilde{m}_{2i}) \quad (3.14)$$

donde,  $d^2(\tilde{m}_{2i}, Fm_{1i}) + d^2(\tilde{m}_{1i}, F^T\tilde{m}_{2i})$ , se describe en la fórmula 3.10.

Las correspondencias que tengan  $\omega_i = 0$  son correspondencias erróneas y son eliminadas; obteniendo con ésto un conjunto de puntos libres de errores. La matriz fundamental  $F$  puede ser finalmente estimada resolviendo el problema de mínimos cuadrados con peso, que es encontrar la matriz fundamental  $F$  que minimize a (fórmula 3.15) :

$$F = \arg \min_F \sum_i \omega_i r_i^2 \quad (3.15)$$

En este trabajo de tesis no obtenemos la  $F$  final que minimize a la fórmula 3.15, ya que sólo nos interesa el conjunto de puntos correspondientes libres de errores.

Como se dijo anteriormente, para obtener los  $m$  subconjuntos se utiliza una técnica tipo Monte Carlo. Sin embargo, para una submuestra generada aleatoriamente los puntos pueden quedar muy cerca unos con otros. Tal situación puede ser evitada. Para alcanzar una distribución uniforme Zhang *et al.* [1994] propone un método al cual llaman *selección aleatoria uniforme* basada en técnicas de cajas (bucketing), las cuales trabajan como sigue. Primero se calculan las coordenadas mínima y máxima de los puntos en la primer imagen. La región es entonces dividida en cajas de  $b \times b$  (ver figura 3.11). Se identifican las cajas en las cuales caen puntos (e indirectamente un conjunto de puntos correspondientes en la segunda imagen) dentro de ellos; y las cajas en las cuales no caen ningún punto no son tomadas en cuenta. Para generar una submuestra de ocho puntos, primero se selecciona aleatoriamente ocho diferentes cajas y entonces se escoge aleatoriamente un punto de cada caja.

Sin embargo, el número de correspondencias en una caja es diferente una de la otra. Como resultado de esto, una caja que tiene muchos puntos tiene más posibilidad de ser seleccionado que una caja que tiene pocos puntos. Para que cada punto tenga la misma probabilidad a ser seleccionada, se propone el siguiente procedimiento. Si tenemos un total de  $l$  cajas, entonces la selección de las cajas serán en un intervalo  $[0-(l-1)]$ , de tal que todas las cajas tengan la misma probabilidad de ser seleccionada (ver figura 3.12). En nuestra implementación,  $b = 8$ .

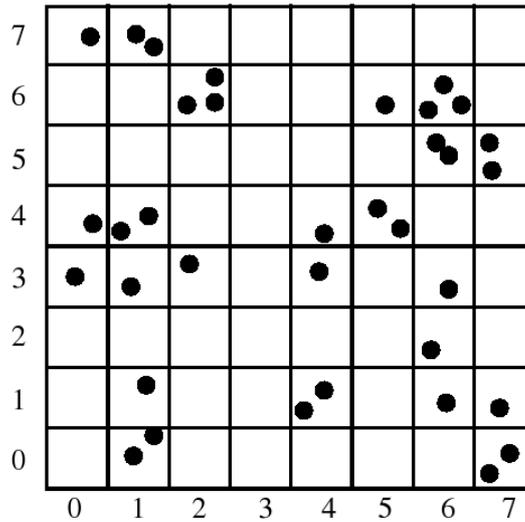


Figura 3.11: Ilustración de la técnica de cajas (bucketing). Fuente tomada de Zhang *et al.* [1994].

### 3.3.4. Resultados

Se corrió el proceso de LMedS con las imágenes de fondo de ojo de la figura 3.7 (a) y (b), se muestra uno de los resultado en la figura 3.13. En la etapa de *ZNCC* se obtuvieron 33 correspondencias (figura 3.13 (c)), en la etapa de *relajación* se obtuvieron 29 correspondencias (figura 3.13 (d)) y en la etapa de *LMedS* se obtuvieron 28 correspondencias (figura 3.13 (e)). Con la *relajación* se eliminaron 4 correspondencias erróneas y con el LMedS se elimina una correspondencia errónea, quedando un conjunto de correspondencias *buenas* entre una par de imágenes.

## 3.4. Otro Algoritmo Robusto: RANSAC

RANSAC(Random Sample Consensum) es otro algoritmo robusto clásico, el cual será probado en la sección de experimentos (sección 3.5) contra nuestra implementación.

La idea es muy simple, se toma un subconjunto aleatoriamente del total de datos, estos puntos definen un modelo, (el modelo es escogido según el problema y puede ser lineal o no lineal). El *soporte* para el modelo es medido por el número de puntos que caen dentro de un umbral  $t$ . El número de puntos buenos son los que caen dentro de un umbral  $t$  alrededor de esta línea.

Se calculan soluciones con estos subconjuntos, cada una de las soluciones parciales nos permiten dividir el total de datos en *correspondencias erróneas* y *correspondencias correctas*, este proceso se repite un  $N$  número de veces y se escoge la solución que pro-

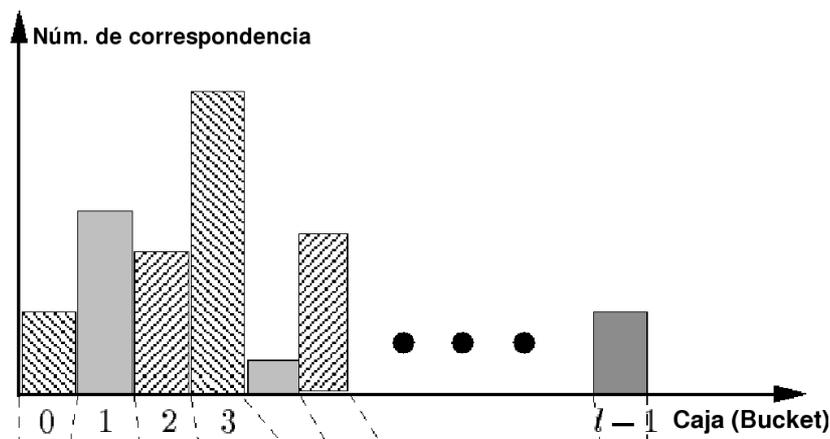


Figura 3.12: Selección de cajas. Cada caja tiene la misma probabilidad a ser seleccionada. Fuente tomada de Zhang *et al.* [1994].

porcione más *correspondencias correctas*. Esto se hace en base a un umbral  $T$ , el cual determina si el tamaño del conjunto es similar a el número de *correspondencias correctas* creíbles en un conjunto de datos, dada la proporción de *correspondencias erróneas*. Por ejemplo, para  $n$  puntos  $T = (1 - \epsilon)n$ , donde  $\epsilon$  es la proporción de correspondencias erróneas y  $n$  es el número total de datos.

Ahora las preguntas son, cuántas veces debemos iterar para obtener una buena solución, cómo calcular la distancia que decide cuándo un dato es considerado una *correspondencia correcta* o una *correspondencia errónea* y cuándo se ha alcanzado el número suficiente de *correspondencias buenas* que permita acortar el número de iteraciones del algoritmo.

Para determinar el número de  $N$  iteraciones necesarias, se emplea el mismo concepto que LMedS, utilizando la fórmula 3.11.

En la práctica la distancia del umbral  $t$  se escoge empíricamente de manera que cumpla con una probabilidad  $\sigma$ , la cual nos indica que el punto es una correspondencia correcta; sin embargo, si asumimos que la medida del error sigue una distribución Gaussiana con desviación estándar  $\sigma$ , entonces el valor de  $t$  puede ser calculado. Una revisión de cómo el valor es obtenido está en Zhang *et al.* [1994]. El algoritmo completo es descrito en el cuadro. 3.3

La aplicación para utilizar RANSAC se toma de un programa de uso libre para relacionar imágenes [Kovesi, 2004]. En la etapa del cálculo de correspondencia utiliza una técnica de correlación normalizada y en la etapa de eliminación de errores utiliza el método RANSAC con la matriz fundamental. El modelo que utiliza este programa

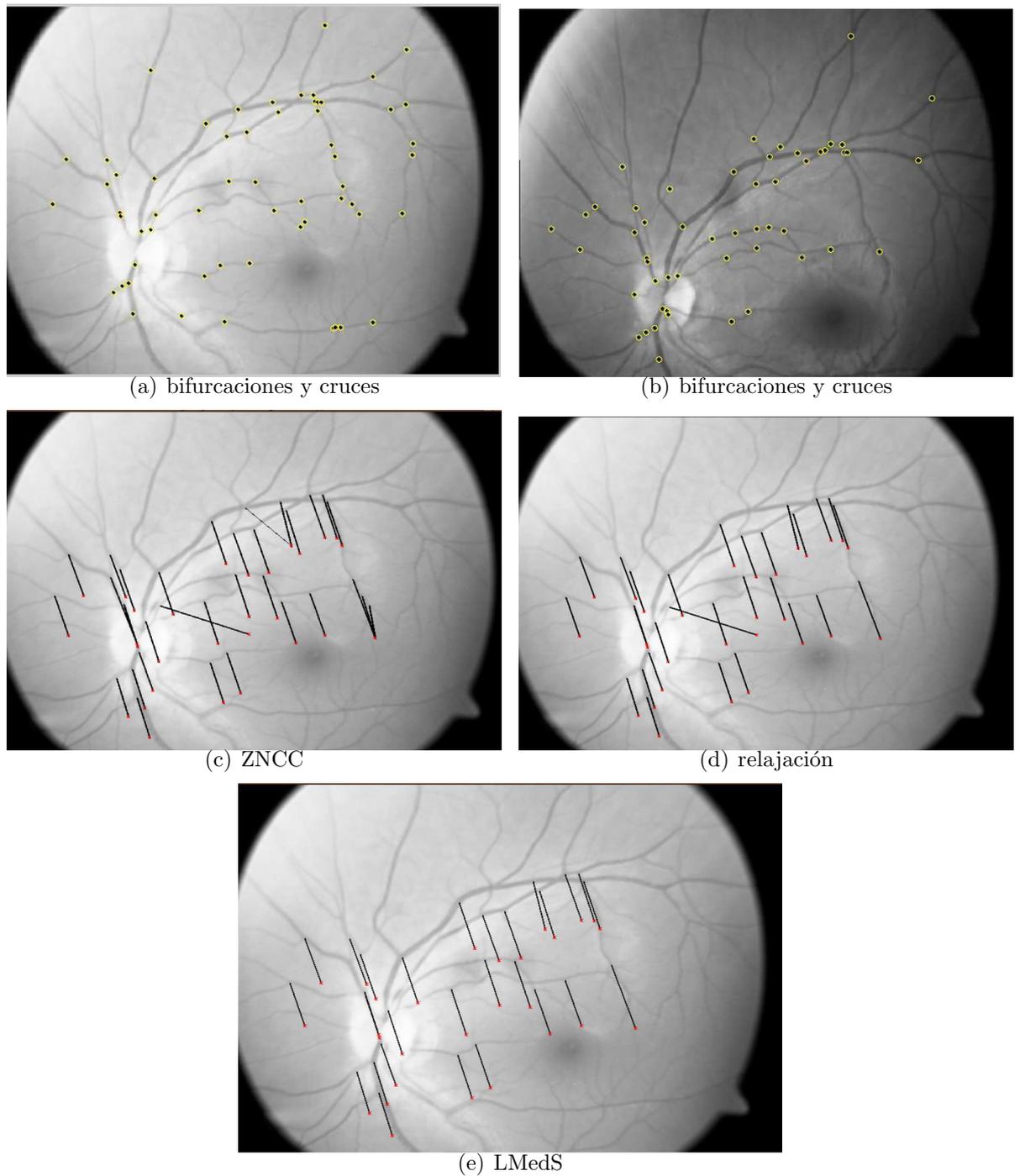


Figura 3.13: LMedS. (a)(b) Cruces y bifurcaciones encontrados por RISA en un par de vistas. (c) Conjunto de puntos obtenidos por el método de ZNCC. (d) Conjunto de puntos obtenidos por el proceso de relajación. (e) Conjunto de puntos obtenidos por el proceso de LMedS.

Objetivo: Ajuste robusto a un modelo desde un conjunto de datos  $S$  el cual contiene *correspondencias erróneas*.

Algoritmo:

1. Seleccionar aleatoriamente un conjunto de datos  $s$  desde un total de  $S$  e instanciar el modelo desde estos datos.
2. Determinar el conjunto de puntos  $S_i$ , los cuales están dentro de una distancia definida por un umbral  $t$  del modelo. El conjunto  $S_i$  define correspondencias buenas de  $S$
3. Si el tamaño de  $S_i$  (el número de *correspondencias buenas*) es más grande que algún umbral  $T$ , reestimar el modelo usando todos los puntos en  $S_i$  y terminar.
4. Si el tamaño de  $S_i$  es menor que  $T$ , selecciona un nuevo subconjunto y repite los pasos anteriores.
5. Después de  $N$  veces se selecciona el conjunto más grande de  $S_i$  y el modelo es re-estimado usando todos los puntos en el subconjunto  $S_i$ .

Cuadro 3.3: RANSAC(Random Sample Consensus). Algoritmo de RANSAC. Método descrito en Hartley y Zisserman [2004].

es ajustado automáticamente al tipo de datos que tenga como entrada. Se toma este programa como prueba por estar referenciado en Hartley y Zisserman [2004]).

### 3.4.1. Resultado

Se corrieron las imágenes médicas de fondo de ojo de la figura 3.7 (a) y (b), sobre el programa de [Kovesi, 2004], la extracción de puntos característicos fue hecho con RISA (sección 3.1) (figura 3.14 (a) y (b) ), el resultado arroja 32 correspondencias (figura 3.14 c) ). El tamaño de la ventana de búsqueda fue de  $200 \times 200$  pixeles y el tamaño de la ventana de correlación fue de  $25 \times 25$  pixeles. De las 32 correspondencias obtenidas por el programa de Kovesi [2004], existen 27 *correspondencias buenas* y 5 *correspondencias erróneas*, (figura 3.14 c)).

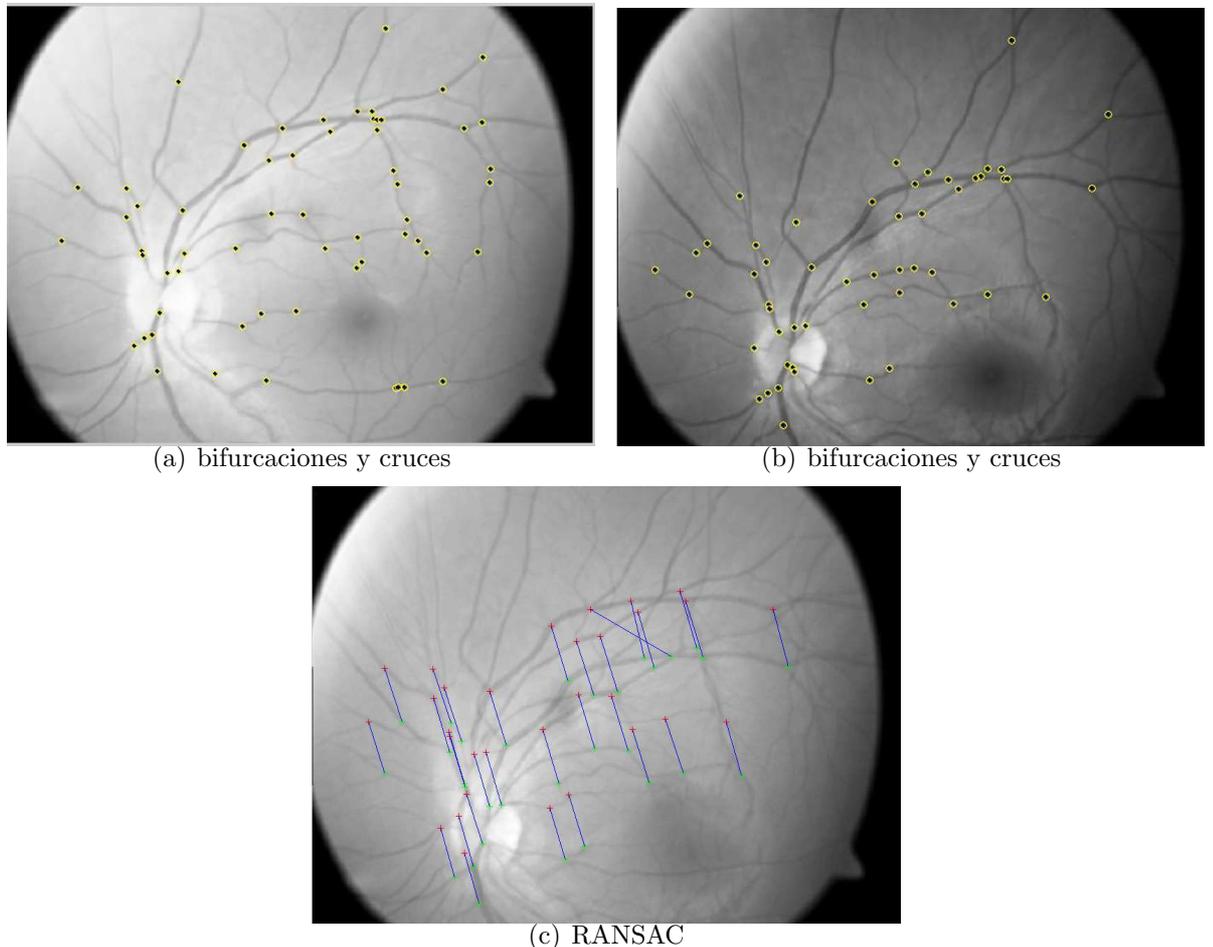


Figura 3.14: Resultado de correr RANSAC sobre las imágenes de fondo de ojo. (a),(b) Bifurcaciones y cruces obtenidas por RISA. (c) Resultado de correr (a) y (b) sobre el programa de Kovesi [2004], el cual utiliza RANSAC.

## 3.5. Experimentos

Se realizaron varios experimentos para probar el funcionamiento de nuestra implementación, cuyo objetivo fue comparar su capacidad para clasificar *correspondencias correctas* y *correspondencias erróneas* en cada una de sus etapas (correlación con ZNCC, relajación y LMedS) y por último comparar nuestra implementación contra alguna otra que use RANSAC para detectar puntos erróneos.

### ■ Experimento 1:

Para el primer experimento, se realizaron varias pruebas con puntos de *bifurcaciones y cruces* obtenidas por RISA (sección 3.1.1). Con este conjunto de puntos como entrada se ejecutó la implementación de nuestro algoritmo y se obtuvieron los resultados mostrados en el cuadro 3.4. El cuadro resume nueve pares de imágenes de fondo de ojo en cada una de las etapas de nuestra implementación. Como un paso previo, se contaron manualmente los puntos que a simple vista son *buenas correspondencias* (cuadro 3.4, 3er. columna) para tener un valor tentativo de comparación con el resultado que se obtiene en cada etapa (cuadro 3.4, columnas 4ta,5ta,6ta ). Finalmente en la 7ma. columna del cuadro 3.4 se muestran los *falsos negativos*, esta columna representa a los pares de puntos correspondientes que existen en la realidad como buenos pero no fueron encontrados por el método presentado en este trabajo de tesis; este valor se obtiene restando los puntos que a simple vista son buenos (3era. columna) menos los *puntos buenos obtenidos* en la última etapa del proceso (6ta. columna).

Uno de los experimentos (el par de imágenes 1 de la tabla 3.4) puede verse en la figura 3.15. Las imágenes utilizadas tienen una resolución de  $768 \times 576$  píxeles, las *bifurcaciones y cruces* encontradas por RISA son de 66 y 55 puntos respectivamente (ver figura 3.15(a)(b) y 2da. columna de la tabla 3.4), el número de correspondencias que a simple vista son buenos es de 40 puntos (3er. columna de la tabla 3.4), el resultado del método ZNCC arrojó 45 correspondencias (ver figura 3.15(c)), y 5 de ellas son erróneas. En el proceso de relajación fueron eliminadas 4 correspondencias ambiguas, así en el proceso de relajación obtuvimos 41 correspondencias y una de ellas errónea (ver figura 3.15(d)). En el proceso de LMedS se obtuvieron 40 correspondencias correctas (ver figura 3.15(e)). Los resultados obtenidos en este experimento en general son buenos, en la mayoría de las imágenes comparadas obtuvimos un conjunto de correspondencia final con un mínimo o nulo de *correspondencias erróneas*.

### ■ Experimento 2:

Para el segundo experimento, se introdujo ruido generando 20 %, 40 % y 60 % de puntos aleatorios sobre un par de imágenes. El objetivo de este experimento es

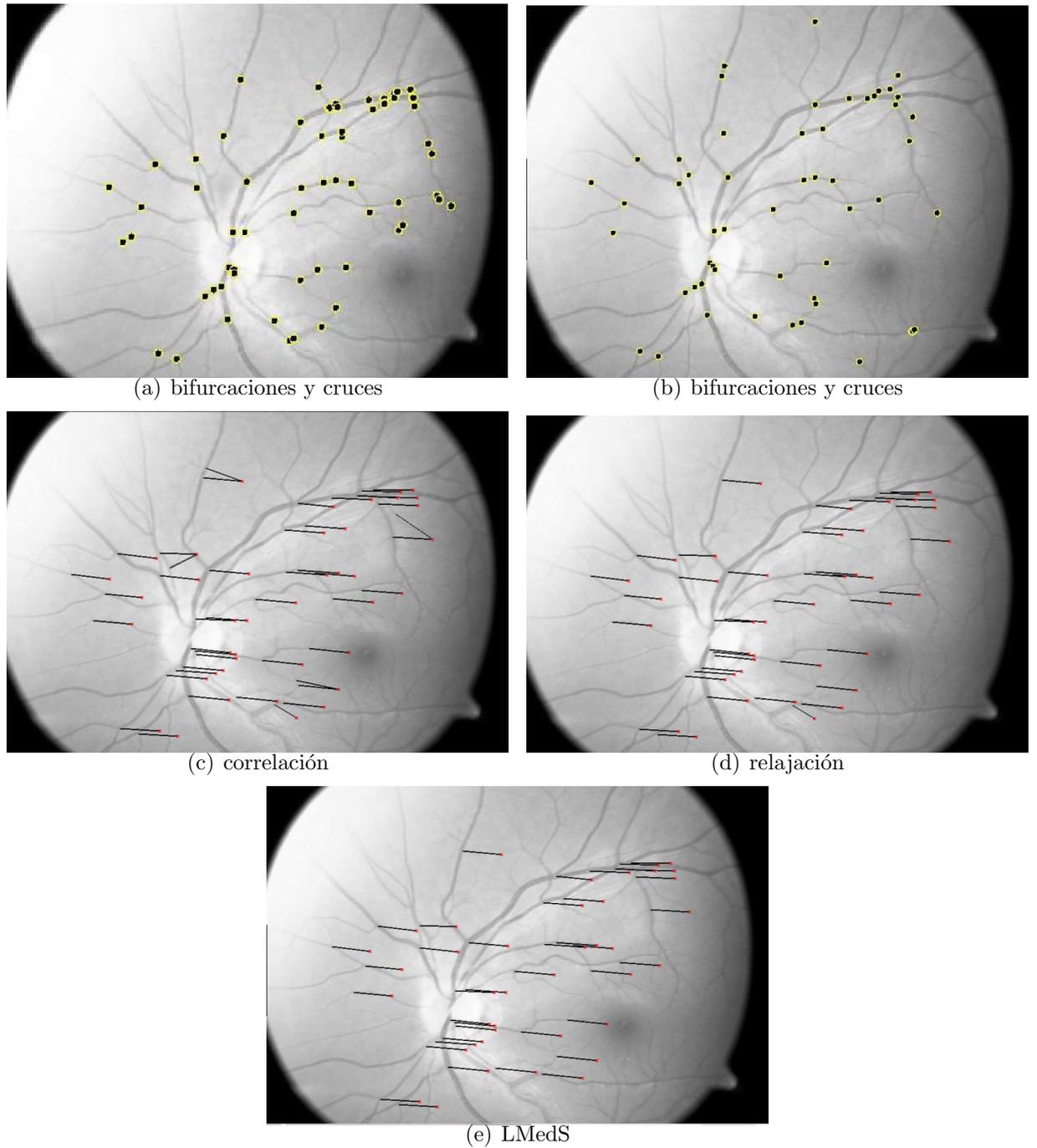


Figura 3.15: Relación de un par de imágenes. (a)(b) Imágenes a relacionar con 66 y 55 puntos respectivamente. (c) Resultado de la correlación con ZNCC. (d) Resultado de la relajación. (e) Resultado de aplicar LMedS.

Par de imágenes	Num. Pts. (izq.-der.)	Pts. Buenos (manual)	ZNCC (total/error)	Relajación (total/error)	LMedS (total/error)	Falsos (-)
1	(66, 55)	40	41/1	40/0	40/0	0
2	(66, 59)	41	45/9	36/0	36/0	5
3	(55, 59)	36	34/0	34/0	34/0	2
4	(55, 52)	36	33/1	33/1	32/0	4
5	(59, 59)	29	30/3	30/3	29/2	2
6	(59, 52)	36	33/5	29/1	28/0	8
7	(43, 59)	40	39/0	39/0	39/0	1
8	(43, 52)	22	17/0	17/0	17/0	5
9	(59, 52)	20	14/0	14/0	14/0	6

Cuadro 3.4: Experimento 1. Muestra el resultado en cada una de las etapas de nuestra implementación. *1er columna*, muestra el par de imágenes a evaluar. *2da. columna*, conjunto de puntos entrantes obtenidos por RISA. *3er. columna*, puntos buenos detectados manualmente. *4ta. columna*, resultado de la etapa ZNCC. *5ta. columna*, resultado de la etapa de relajación. *6ta. columna*, resultado de la etapa de LMedS. *7ma. columna* son los falsos negativos.

medir la robustez de nuestra implementación al ruido introducido. El resultado obtenido con este conjunto de puntos se muestra en el cuadro 3.5. En la tabla se indica la cantidad de puntos aumentadas por el porcentaje indicado (cuadro 3.5 1er. columna). Los resultados obtenidos son buenos a pesar del ruido introducido (ver cuadro 3.5 6ta. columna). La última columna de la tabla muestra los *falsos negativos* que se obtienen de la misma forma que en el experimento 1.

Par de imagen 1	Num. Pts. (izq.-der.)	Pts. Buenos (manual)	ZNCC (total/error)	Relajación. (total/error)	LMedS (total/error)	Falsos (-)
20%(11pts)	(77, 66)	40	43/3	41/2	40/1	1
40%(22pts.)	(88, 77)	40	45/5	42/2	40/0	0
60%(33pts.)	(99, 88)	42	58/16	54/12	42/0	0

Cuadro 3.5: Experimento 2. Muestra el resultado en cada una de las etapas de nuestra implementación al aplicar un conjunto de puntos con ruido del 20%,40% y 60% generados aleatoriamente sobre un par de imágenes.

### ■ Experimento 3:

Para el tercer experimento, se utilizó una imagen que no es una imagen médica de fondo de ojo, los puntos de interés (esquinas) fueron obtenidos con el detector de esquinas de Harris (una descripción del algoritmo puede encontrarse en [Patrón Pérez, 2006]). Los resultados obtenidos se muestran en el cuadro 3.6 y en la figura 3.16. Este experimento se realizó con el fin de saber cómo se comportaba nuestra implementación con imágenes que no tienen el mismo tipo de distorsión geométrica que las imágenes de fondo de ojo utilizadas en este trabajo de tesis.

Imagen	Num. Pts. (izq.-der.)	Pts. Buenos (manual)	ZNCC (total/error)	Relajación. (total/error)	LMedS (total/error)	Falsos (-)
Escritorio	(347, 368)	228	252/54	215/17	202/3	29

Cuadro 3.6: Experimento 3. Muestra el resultado en cada una de la etapas de nuestra implementación con una imagen de una oficina.

Los resultados son buenos en general, nos indican que tienen un buen comportamiento en la presencia de ruido en la mayoría de los casos experimentados.

#### ■ Experimento 4:

Se corrieron los 3 experimentos anteriores sobre el programa de Kovesi [2004], con el fin de comparar nuestra implementación que utiliza LMedS contra la de Kovesi [2004] que utiliza RANSAC, ambos métodos fueron utilizados para eliminar *correspondencias erróneas*. En el cuadro 3.7 columna 5, se muestran los resultados de la etapa final de cada uno de los experimentos anteriores y en el cuadro 3.7 columna 6, se muestra el resultado de correr cada uno de los experimentos anteriores con el programa de Kovesi [2004]. En términos generales la comparación de estas dos columnas no muestra una diferencia significativa. Sin embargo, el objetivo en estos métodos es obtener un conjunto de *correspondencias buenas* con un mínimo o nulo error, por lo que nuestra implementación se muestra ligeramente mejor.

Num. Exper.	imagen	Num. Pts. (izq-der)	Pts. Buenos	LMedS (total/error)	Falsos (-)	RANSAC (total/error)	Falsos (-)
1	1	(66, 55)	40	40/0	0	39/0	1
	2	(66, 29)	41	36/0	5	35/0	6
	3	(55, 59)	36	34/0	2	33/0	3
	4	(55, 52)	36	32/0	4	31/0	5
	5	(59, 59)	29	29/2	2	25/0	4
	6	(59, 52)	36	28/0	8	32/5	9
	7	(43, 59)	40	39/0	1	40/0	0
	8	(43, 52)	22	17/0	5	17/0	5
	9	(59, 52)	20	14/0	6	18/4	6
2	1 (20 %)	(77, 66)	40	40/1	1	40/0	0
	1 (40 %)	(88, 77)	40	40/0	0	41/1	0
	1 (60 %)	(99, 88)	42	42/0	0	43/3	2
3	Escritorio	(347, 368)	228	202/3	29	202/0	26

Cuadro 3.7: Experimento 4. Se corrieron en Kovesi [2004], los mismos experimentos realizados para nuestra implementación con LMedS y se comparan los resultados de ambos. La columna 6 muestra los falsos positivos de nuestra implementación y la columna 8 muestra los falsos positivos de RANSAC.

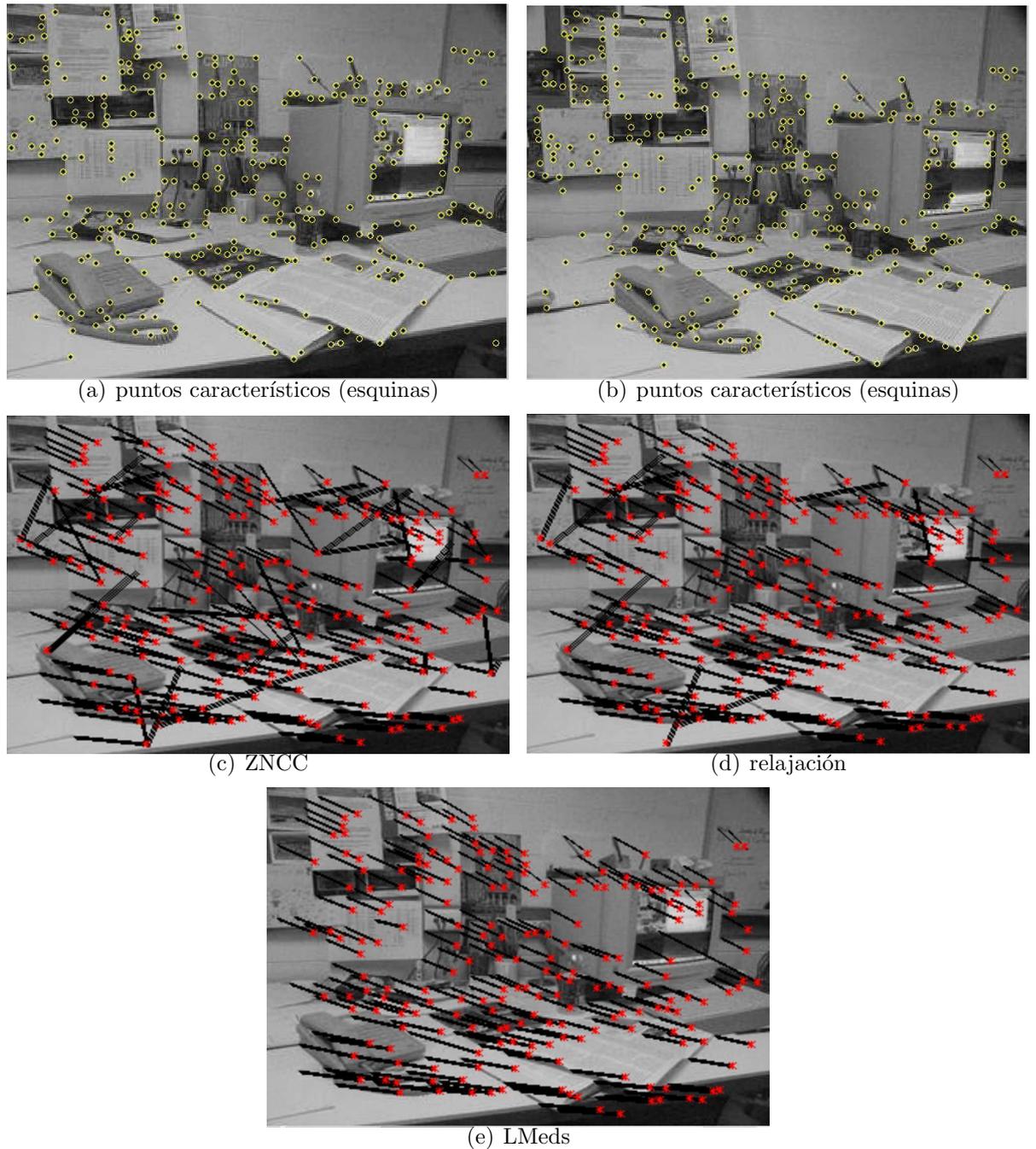


Figura 3.16: Relación de un par de imágenes. (a)(b) Imágenes a relacionar con 346 y 369 puntos respectivos. (c) Resultado de la correlación con ZNCC. (d) Resultado de la relajación. (e) Resultado de aplicar LMedS.

## 3.6. Resumen

El objetivo de este capítulo fue encontrar un conjunto de correspondencias con un mínimo de errores que relacionara a dos imágenes. Primero se describió la técnica de extracción de puntos a los que llamamos *bifurcaciones y cruces*. Luego las correspondencias candidatas se obtuvieron usando técnicas de correlación cruzada normalizada (ZNCC), por último se implementó LMedS para eliminar las correspondencias erróneas. En los experimentos presentados se compara nuestra implementación contra otra que utiliza RANSAC ([Kovesi, 2004]). Nuestra implementación arrojó mejores resultados en los diferentes experimentos realizados, obteniendo un conjunto de *correspondencias buenas* libres de errores en la mayoría de los pares de imágenes evaluadas.

En el siguiente capítulo se construye el mosaico de las imágenes de fondo de ojo, teniendo como entrada el conjunto de correspondencias encontradas en este capítulo por el método LMedS.

# Capítulo 4

## Creación de mosaicos

El objetivo de este capítulo es la creación de un mosaico a partir de una secuencia de imágenes. Habiendo establecido en la sección 3.3.3 el conjunto de correspondencias entre un par de imágenes, la siguiente tarea es identificar una transformación que nos permita alinear una imagen con otra.

Básicamente, la idea principal de construir un mosaico es el cálculo de la función de transformación (sección 4.1). El paso siguiente es calcular el tamaño de la nueva imagen que contendrá a las imágenes que formarán el mosaico (sección 4.2), y el último paso de la construcción del mosaico es realizar la mezcla de todas las imágenes en una sola imagen (sección 4.3). Finalizamos el capítulo con una breve introducción a la reconstrucción 3D de vasos sanguíneos con más de 3 imágenes.

### 4.1. Transformación Geométrica de una Imagen

La *transformación geométrica de una imagen* significa reorganizar los píxeles de la imagen. También es llamada *corrección geométrica*, *distorsión de la imagen* y *torcer la imagen* (*image warping*).

Para *transformar una imagen* necesitamos una función que mapee los puntos correspondientes entre la imagen origen y la imagen destino. En la literatura esta función es llamada *función de transformación* o *función de mapeo*. Un ejemplo de una *transformación de una imagen* se puede ver en la figura 4.1.

Ahora bien, nuestro objetivo es encontrar una *función de transformación* que nos permita reorganizar los píxeles de la imagen origen con respecto a los píxeles de la imagen destino.

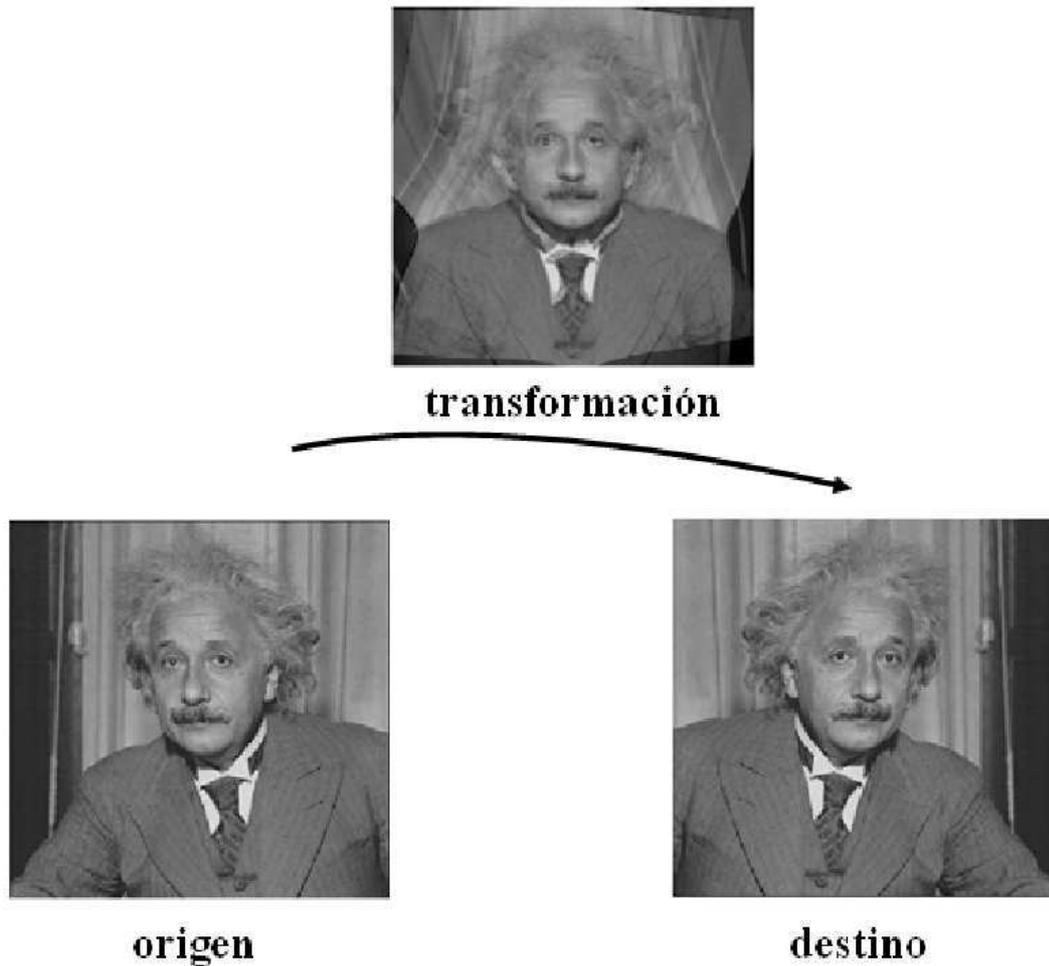


Figura 4.1: Transformación de una imagen. Los píxeles de la imagen origen son reorganizados para coincidir con los píxeles de la imagen destino, por medio de un mapeo entre la imagen origen y la imagen destino. Fuente tomada de Heckbert [1989].

#### 4.1.1. Función de Transformación

Existen diferentes *funciones de transformación*; el caso más simple en 2D son las transformaciones lineales (las cuales son: la transformación afín, similitud, euclídea y proyectiva. Ver sección 2.1.1). Otro tipo de transformaciones son las polinomiales ([Ryan *et al.*, 2004]). También pueden ser usadas transformaciones arbitrarias, por ejemplo la transformación Beier-Neely, que es una transformación cúbica ([Heckbert, 1989]).

La *transformación de una imagen* es usada principalmente para corregir distorsiones geométricas introducidas por un sistema de imperfección de imágenes [Heckbert, 1989]. La lente de la cámara algunas veces introduce distorsión de barril o distorsión de cojín, la perspectiva de la vista introduce una distorsión proyectiva y otros componentes ópticos no lineales pueden crear distorsiones más complejas. Una revisión sobre

técnicas de *transformación* puede ser revisado en [Heckbert, 1989]. Se muestran en la figura 4.2 ejemplos de distorsión provocada por la lente de la cámara.

Para encontrar la *función de transformación* que corrija la distorsión de las imágenes de fondo de ojo utilizadas en este trabajo de tesis, tomamos en cuenta que el objetivo de estas imágenes es escanear el área sobre la periferia del fondo de ojo, y que la distorsión entre imágenes puede ser compleja dada la distorsión perspectiva. Por esta razón fue asumido que usar una función de transformación que involucre traslación, rotación y escala no sería suficiente para corregir esta distorsión ([Ryan *et al.*, 2004]). Un modelo de transformación capaz de corregir un área completa es la transformación polinomial de segundo orden. Una transformación polinomial de grado dos permite flexibilidad en la distorsión geométrica modelada y es caracterizada por la fórmula 4.3. La fórmula es obtenida como sigue:

Un polinomio general de grado  $m$  en 2D está dado por:

$$\sum_{i=0}^m \sum_{j=0}^{m-i} a_{ij} x^i y^j \quad (4.1)$$

donde  $a_{ij}$  son los coeficientes a buscar y  $x, y$  es un punto en la imagen.

Desarrollándolo para  $m = 2$  obtenemos las siguientes ecuaciones :

$$\begin{aligned} a_{00}x^0y^0 + a_{01}x^0y^1 + a_{02}x^0y^2 \\ a_{10}x^1y^0 + a_{11}x^1y^1 \\ a_{20}x^2y^0 \end{aligned} \quad (4.2)$$

Ordenando en forma de matrices para un punto  $(x, y)$ .

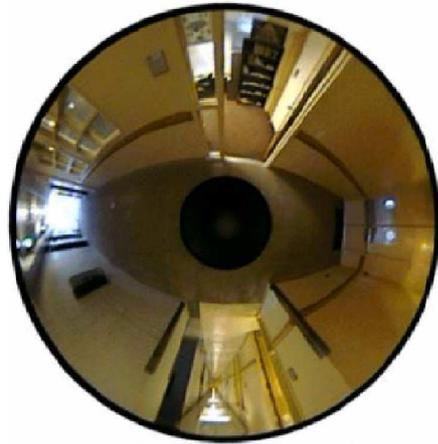
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{00} & a_{10} & a_{01} & a_{11} & a_{20} & a_{02} \\ b_{00} & b_{10} & b_{01} & b_{11} & b_{20} & b_{02} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix} \quad (4.3)$$

Los parámetros de este sistema de ecuaciones se encuentran de manera sencilla utilizando SVD (Singular Value Descompositon).

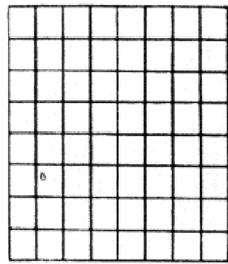
Básicamente la *transformación* es realizada desde un conjunto de puntos correspondientes entre la imagen origen y la imagen destino. Esta función nos permite mapear



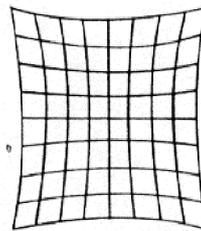
(a) lente ojo de pez



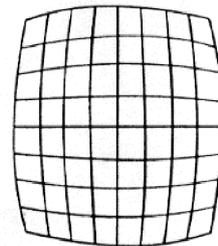
(b) omnidireccional



(c) original



(d) distorsión de cojín



(e) distorsión de barril

Figura 4.2: Ejemplo de distorsiones geométricas. a) y b) muestran las deformaciones causadas por la lente de una cámara en la vida real, lente ojo de pez y omnidireccional respectivamente. c) d) e) muestra el esquema partiendo de la imagen origen, y las deformaciones de cojín y barril que son originados por las cámaras b) y a) respectivamente. Fuente tomada de Researchers [2006].

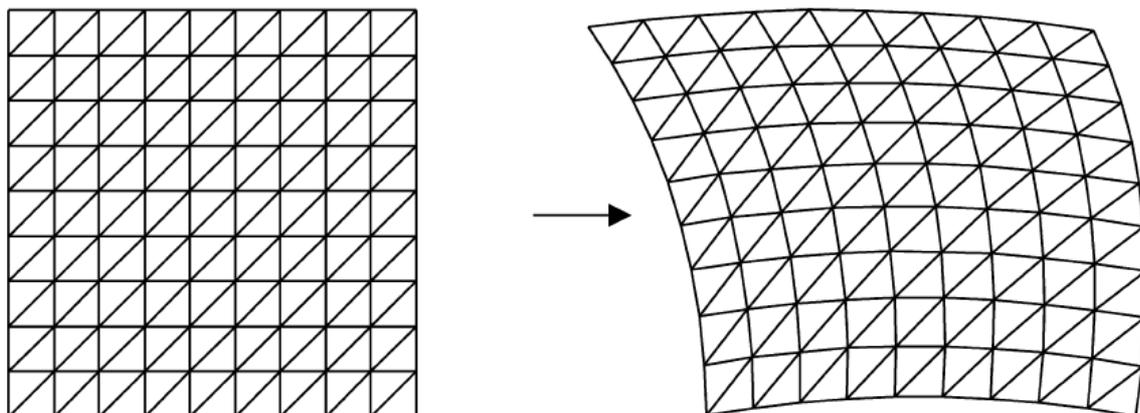


Figura 4.3: Una ilustración esquemática de la transformación polinomial de segundo orden. En general todas las líneas rectas son mapeadas a curvas. Fuente tomada de Ryan *et al.* [2004]

todas las líneas a curvas (ver figura 4.3), con un gran flexibilidad.

Otros autores que utilizan esta *función de transformación* para crear el mosaico con imágenes de fondo de ojo son Ryan *et al.* [2004] y Can *et al.* [2002].

En los capítulos siguientes vamos a hacer referencia a esta *función de transformación* como  $H$  (fórmula 4.3).

Expresamos la transformación de un punto  $(x, y)$  utilizando una función de transformación  $H$  de la siguiente manera:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = H \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix} \quad (4.4)$$

donde  $x'$  y  $y'$  son las coordenadas transformadas, y  $H$  es una matriz que representa la función de transformación.

Una vez calculada la *función de transformación*  $H$ , se calculan los nuevos límites de la imagen.

## 4.2. Dimensiones de la Imagen Nueva

Para calcular los límites de la *nueva imagen* (figura 4.4 (e)) que contendrá el mosaico formado por la imagen origen (figura 4.4 (a)) y la imagen destino (figura 4.4 (b)), la idea es la siguiente: se hace una multiplicación de la imagen a origen por la función de transformación  $H$ , como resultado tenemos la imagen origen-transformada. Con esto se obtiene la ubicación de la imagen origen con respecto al marco de coordenadas de la imagen destino; mediante las esquinas de la *imagen destino* y la *imagen transformada* se calcula el tamaño que tendrá la *nueva imagen*.

Sin embargo, si transformamos la *imagen origen* de esta manera, el resultado puede sufrir pixelación por la deformación de la imagen. Para evitar esto aumentamos la resolución haciendo una rejilla del mismo tamaño de la imagen origen pero con una separación de .5 pixeles de unidad (figura 4.4 (c)), a esta rejilla aplicamos la transformación  $H$  (figura 4.4 (d)) y posteriormente en la sección 4.3 la llenamos con los valores de intensidad de la imagen origen. Se ilustra el proceso completo en la figura 4.4.

En la figura 4.4 (e) se muestra la rejilla de la *nueva imagen* con las dimensiones correctas ( $R_y, R_x$ ) para contener el mosaico que formarán las imágenes (a) y (b). Vamos a llamar de aquí en adelante a la *nueva imagen* simplemente  $\mathbf{I}$ .

## 4.3. Mezcla de Imágenes

Obtenidas las dimensiones de la *nueva imagen*  $\mathbf{I}$ , procedemos a mezclarlas dentro de esta *nueva imagen* para crear el mosaico. El procedimiento es el siguiente.

Se utilizó una estructura de datos para representar a la nueva imagen  $\mathbf{I}$  de tres capas (bandas).

En la capa 1 de  $\mathbf{I}$ , se coloca la *imagen destino*, en la capa 2 de  $\mathbf{I}$ , colocamos la *rejilla transformada* y la llenamos con los valores de intensidad de la *imagen origen* (ahora tenemos la *imagen origen transformada*). Para colocar la *imagen origen transformada* y la *imagen destino* se debe hacer un corrimiento para que entren en las dimensiones del marco de la imagen  $\mathbf{I}$ . Vamos a entender esto con un ejemplo específico; supongamos que la *imagen origen transformada* tiene coordenadas en  $M_{min} = -40$  en el eje  $y$  y  $N_{min} = -251$  en el eje  $x$ , y que la *imagen destino* tiene coordenadas en  $M_y = 768$  y  $M_x = 576$ ; si colocamos las imágenes con las coordenadas mencionadas, alguna de las imágenes podría quedar fuera del marco de  $\mathbf{I}$  (ver figura 4.5 (c)). Si aplicamos un corrimiento a ambas imágenes de tal manera que el corrimiento esté dado por el mínimo de las coordenadas entre las imágenes.

$$\text{corrimiento} = -(\text{mín}(\text{Imagen origen transformada}, \text{imagen destino}))$$

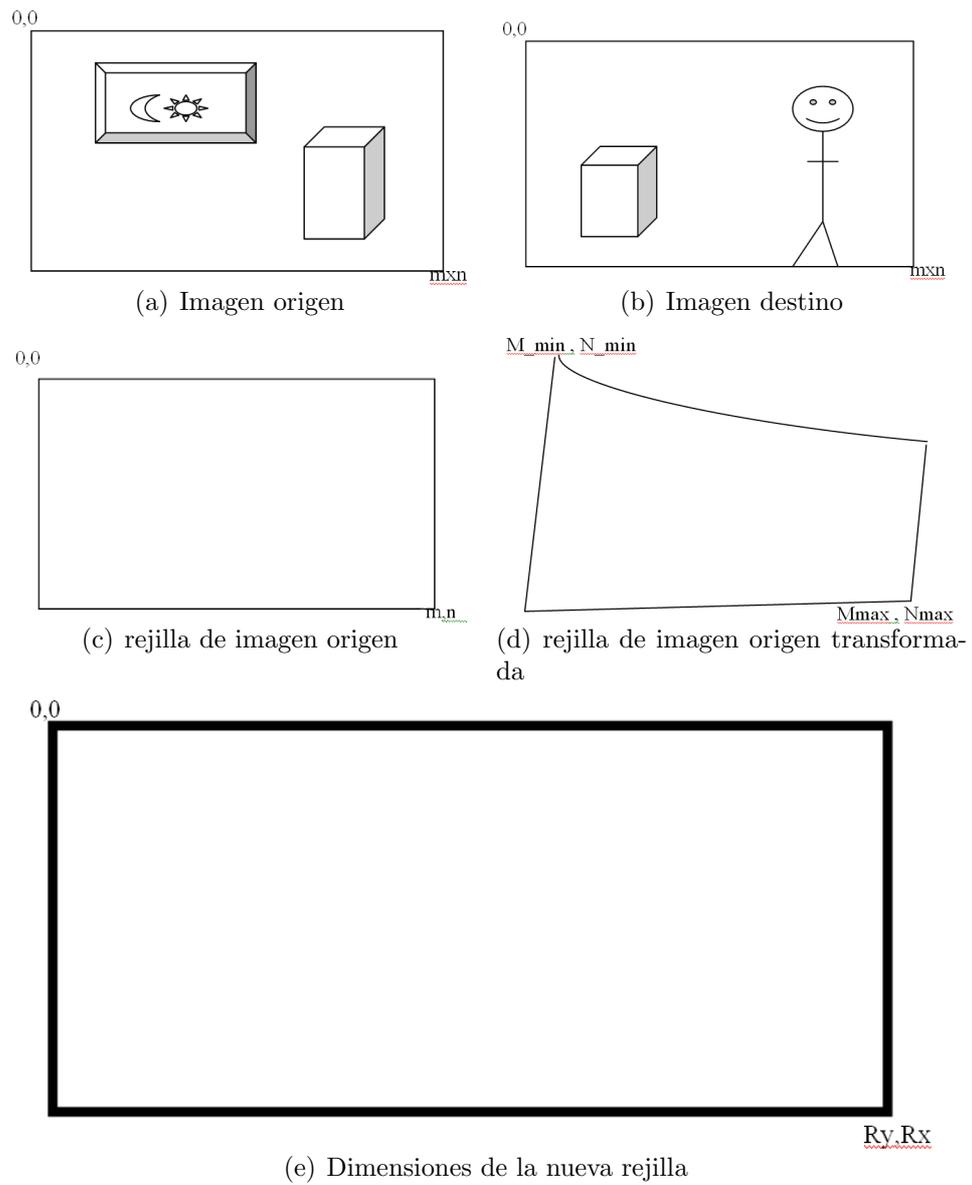


Figura 4.4: (a) Imagen original. (b) Imagen destino. (c) Rejilla con las dimensiones de la *imagen original* pero separada a 0,5 pixeles en ambas direcciones  $x$  y  $y$ . (d) Rejilla transformada sobre las coordenadas de la *imagen destino*. (e) Dimensiones de un marco rectangular que contendrá el mosaico formado por la *imagen original* e *imagen destino*.

entonces las imágenes quedan dentro del marco de la imagen  $I$ , como se muestra en la figura 4.5 (d).

En la capa 3 de  $I$  se hace la mezcla de las imágenes siguiendo la estimación de una transformación de parámetros óptimos, el valor del pixel en la *imagen destino* e *imagen origen-transformada* en una misma rejilla espacial rectangular.

La *imagen destino* que utilizamos está definida como una rejilla rectangular con valores positivos y pixeles separados por una unidad en ambas direcciones  $x$  y  $y$ . Sin embargo, siguiendo la transformación de los pixeles en la imagen distorsionada, ésta es transformada sobre una imagen correcta en la cual se obtienen valores no enteros en referencia a la rejilla de la *imagen destino*.

Por esto es conveniente interpolar los valores de la *imagen origen transformada* sobre los valores discretos de una rejilla rectangular usada por la imagen destino. Este es un problema conocido en computación y un rango de técnicas estándar para interpolación de imágenes son discutidas en Heckbert [1989]. Un enfoque de bajo costo computacional utilizado en este trabajo de tesis es el método de interpolación bajo el criterio de vecino más cercano, el cual trabaja como sigue.

La más simple de las interpolaciones es la del *vecino más cercano*. La idea es que cada punto tome el valor del punto anterior (vecino más cercano). El error de interpolación es por mucho de medio pixel. Este error es perceptible con líneas rectas que pueden aparecer como escaleras tras la interpolación. La figura 4.6 muestra gráficamente la técnica de interpolación del vecino más cercano.

Hasta aquí se tiene ya un mosaico compuesto por dos imágenes. En la figura 4.7 se muestra el proceso descrito de la creación del mosaico utilizando el par de imágenes de fondo de ojo de la figura 3.13 (a) y (b), el conjunto de puntos correspondientes se toma del proceso LMedS que se muestra en la figura 3.13 (e) . La figura 4.7 (a) muestra la imagen origen, la cual será alineada (transformada) con respecto a la imagen destino (figura 4.7 (b) ). En la figura 4.7 (c) se muestra la transformación de la imagen origen con respecto al marco coordenado de la imagen destino. Finalmente en la figura 4.7 (d) se muestra el mosaico resultante de mezclar la imagen origen-transformada con la imagen destino. Se puede notar que la transformación que sufre la imagen origen es prácticamente una simple traslación.

## 4.4. Mosaico de tres o más Imágenes de Fondo de Ojo

Ahora bien, para tener un mosaico con más de dos imágenes se tienen varias opciones. La primera (que es la utilizada en este trabajo de tesis) consiste en repetir los pasos anteriores para cada imagen tomando como imagen destino el mosaico generado

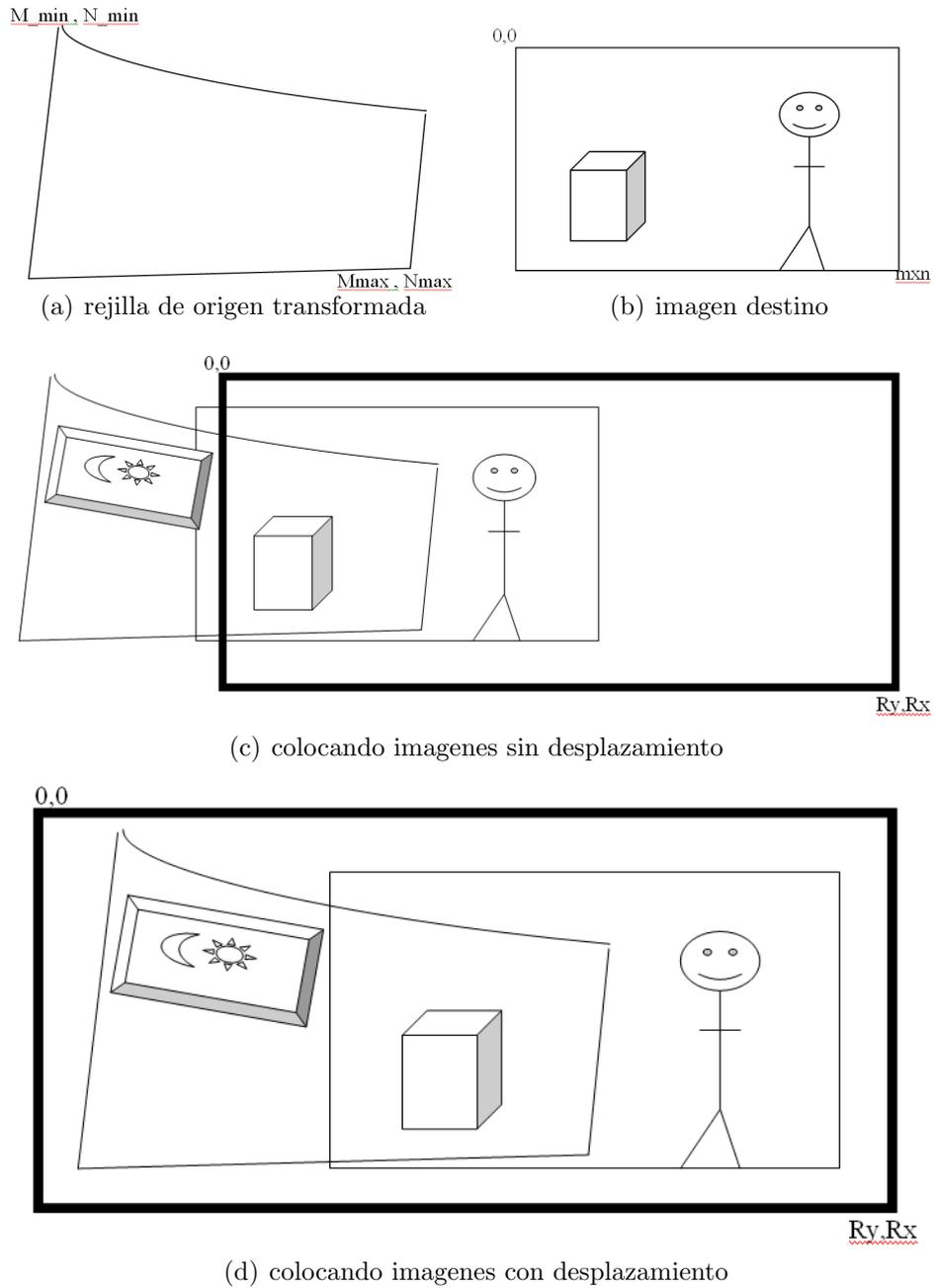


Figura 4.5: Mezcla. (a) Rejilla de origen transformada. (b) Imagen destino. (c) Ejemplo de colocar las imágenes dentro de  $I$  sin utilizar un desplazamiento. (d) Ejemplo de colocar las imágenes dentro de  $I$  utilizando un desplazamiento.

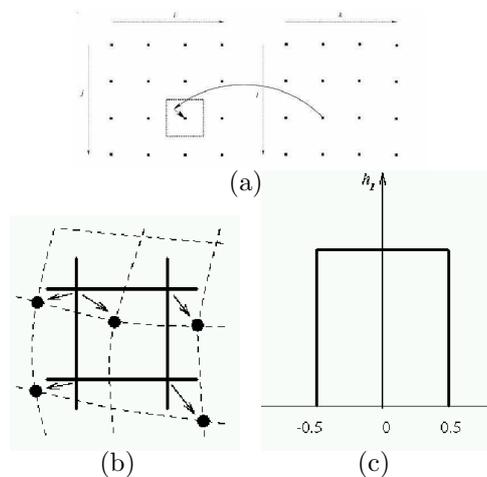


Figura 4.6: Interpolación del vecino más cercano. (a) Si los puntos no coinciden toma el valor de intensidad del vecino más próximo. (b) y (c) se describe esquemáticamente la interpolación del vecino más cercano. Fuente tomada de Researchers [2006].

en el paso anterior.

Otra opción que se tiene si la rotación de la cámara no es mucha o si la imagen de referencia tiene puntos en común con todas las otras imágenes, es la de calcular todos los *puntos de transformación* con respecto a dicha imagen destino, calcular la transformación de todas las esquinas para obtener las dimensiones de la nueva imagen para luego mezclarlas en una sola imagen  $I$ , si el punto cae dentro de varias de las imágenes se promedia su valor.

En la figura 4.8, se muestra el procedimiento de adicionar una imagen más a la figura 4.7 para crear un mosaico de 3 imágenes. Se toma el mosaico generado en la figura 4.7 como *imagen destino* y la *imagen origen* es la imagen que se agrega para formar un mosaico de 3 imágenes.

Para tener un mosaico de 4 imágenes se sigue el mismo procedimiento. En la figura 4.9, se muestra el procedimiento de adicionar una imagen más a la figura 4.8 para crear un mosaico de 4 imágenes.

Finalmente, la tabla 4.10 (b) y figura 4.10 (a) muestran en forma esquemática, la manera en que fue creado el mosaico de 4 imágenes. Las correspondencias de entrada en cada una de los pares fueron tomadas de la etapa de LMedS. (sección 3.3.3).

## 4.5. Experimentos

Se hicieron varios experimentos probando el algoritmo implementado para la creación del mosaico. Se observan algunas imágenes más oscuras que otras debido al cambio de intensidad de la luz al momento de tomar las fotos. Para estas pruebas se utilizó como origen del sistema coordenado, la esquina superior izquierda de las imágenes, tomando el eje  $x$  positivo de izquierda a derecha y el eje  $y$  positivo de arriba a abajo. Un parámetro que influye en gran manera en el resultado es la elección de las correspondencias ya que una sola *correspondencia errónea* puede generar muy malos resultados. Las correspondencias no se deben elegir solamente en una pequeña región de la imagen sino de una manera más distribuida. Según las pruebas realizadas un conjunto entre 8 y 15 correspondencias son suficientes para obtener buenos resultados. Se describen a continuación los experimentos realizados:

- **Experimento 1:**

El objetivo del primer experimento es mostrar cómo afectan las correspondencias erróneas en el proceso de creación del mosaico.

Se toma un par de imágenes del experimento1 de la sección 3.5 (el par número 2) las cuales son mostradas en la figura 4.11 (a) y (b), y se crea un mosaico con estas imágenes tomando como entrada el conjunto de puntos correspondientes obtenidos por la etapa de correlación ZNCC; se puede apreciar en la figura 4.11 (c) que existen *correspondencias erróneas* (tienen una dirección diferente al resto de los vectores). El resultado de construir el mosaico se muestra en la figura 4.11, las imágenes no son alineadas correctamente y por lo tanto no pudo crearse el mosaico.

- **Experimento 2:**

En segundo experimento se toman el mismo par de imágenes del experimento1 de la sección 3.5 (el par número 2), mostradas en la figura 4.12 (a) y (b), pero tomando como entrada el conjunto de puntos correspondientes obtenidos por la etapa de correlación LMedS, se puede apreciar en la figura 4.12 (c) que se tiene un conjunto de *correspondencias correctas* sin errores. El resultado de construir el mosaico se muestra en la figura 4.12, las imágenes son alineadas correctamente.

- **Experimento 3:**

En este experimento, el objetivo es tomar aleatoriamente cuatro imágenes para formar un mosaico (ver figura 4.13 (a)-(d)). En la tabla 4.1, se muestra la forma en que fueron escogidas aleatoriamente las imágenes. En la figura 4.13 (e) se muestra el mosaico obtenido.

Selección aleatoria	Imagen origen	Imagen destino
1		1
2	2	1
3	3	2
4	4	3

Cuadro 4.1: Experimento 3. La *1er. columna*, muestra la forma aleatoria en que fueron seleccionados las imágenes de la figura 4.14 (a)-(e). La *2da. columna*, muestra la imagen origen que será transformada en las coordenadas de la imagen destino. La *3er columna*, muestra la imagen destino.

▪ **Experimento 4:**

Se toman aleatoriamente cinco imágenes (con una secuencia diferente al experimento 3) para formar un mosaico (ver figura 4.14 (a)-(e)). En la tabla 4.2 se muestra la forma en que fueron escogidas aleatoriamente las imágenes. En la figura 4.14 (e) se muestra el mosaico obtenido y en esta misma imagen se resaltan las áreas que no tuvieron una buena alineación con un rectángulo.

Se puede concluir en este experimento que el método escogido para crear mosaicos de 3 o más imágenes tiene desventaja al agregar una imagen que no tiene puntos de transformación en toda el área de solapamiento. Como se muestra en el este experimento donde la imagen 5 (figura 4.14 (e)) es agregada al mosaico teniendo solo puntos de transformación con la imagen 4 (figura 4.14 (d)) (ver tabla 4.2); lo que origina una mala alineación en las áreas que no hay puntos de transformación. Esto se puede evitar si antes de crear el mosaico seleccionamos las imágenes de manera que al ser agregadas tengan el mayor número de puntos de transformación en el área que solapan con el resto de las imágenes.

Selección aleatoria	Imagen origen	Imagen destino
1		1
2	2	1
3	3	2
4	4	3
5	5	4

Cuadro 4.2: Experimento 4. La *1er. columna*, muestra la forma aleatoria en que fueron seleccionados las imágenes de la figura 4.14 (a)-(e). La *2da. columna*, muestra la imagen origen que será transformada en las coordenadas de la imagen destino. La *3er columna*, muestra la imagen destino.

Finalmente concluimos que los experimentos 1 y 2 nos muestran la importancia de tener un conjunto de correspondencias libres de errores (los cuales se obtienen en la sección 3.3.3) y en los experimentos 3 y 4 se muestra que no interesa el orden en que son incorporadas las imágenes para formar el mosaico, siempre y cuando la nueva imagen empate con por lo menos una imagen que pertenezca al mosaico y ésta cubra el mayor número de puntos de transformación en todo el área.

## 4.6. Introducción a la Reconstrucción de un Mosaico de Imágenes en 3D.

Martínez-Pérez y Espinosa-Romero [2004] hacen una reconstrucción en 3D del esqueleto de los vasos sanguíneos utilizando una secuencia de dos vistas (ver análisis bifocal en la sección 2.2).

De un conjunto de puntos correspondientes entre las dos imágenes ( $x_i \leftrightarrow x'_i$ ), se asume que éstas dan un conjunto de puntos 3D  $X_i$  (ver sección 2.2), los cuales son desconocidos. La fórmula para encontrar los puntos 3D, se describe como:

$$x_i = PX_i, \quad x'_i = P'X_i \quad \text{para todo } i \quad (4.5)$$

donde  $P$  y  $P'$  son matrices de proyección. Una matriz de proyección permite recuperar la estructura de la escena y el movimiento de la cámara a partir de la relación obtenida entre los puntos correspondientes de las imágenes. El procedimiento para encontrar la matriz de proyección es descrito en Pollefeys [2000], y de manera general consiste en lo siguiente: primero se seleccionan dos imágenes y se determina un marco inicial de reconstrucción entre ellas, luego la posición de la cámara para las otras vistas se calcula con referencia a este marco, y cada vez que se agrega una nueva imagen la vista se refina.

Una vez que se han definido las matrices de proyección para cada imagen, se hace una reconstrucción proyectiva; para este fin se utiliza un método de triangulación lineal descrito por Hartley y Zisserman [2004]. El método de triangulación lineal está basado en la combinación de las ecuaciones proyectivas descritas en la fórmula 4.5, para tener un sistema de la forma  $AX = 0$ . La matriz  $A$  está dada por:

$$A = \begin{bmatrix} xp^{3\top} - p^{1\top} \\ yp^{3\top} - p^{2\top} \\ x'p^{3\top} - p'^{1\top} \\ y'p^{3\top} - p'^{2\top} \end{bmatrix} \quad (4.6)$$

donde  $p^{iT}$  representa la  $i$ -ésima fila de la matriz de proyección  $P$  y  $x = (x, y, 1)$ ,  $x' = (x', y', 1)$  son las coordenadas homogéneas de puntos correspondientes. Ellos pro-

ponen resolver el sistema por medio de mínimos-cuadrados. El concepto de triangulación puede observarse en la figura 4.15.

Ahora ya que se cuenta con un conjunto de puntos tridimensionales que representan la escena, el siguiente paso que se realiza es el modelado de una superficie cilíndrica para proyectar los puntos en el espacio.

El proceso seguido por Martínez-Pérez y Espinosa-Romero [2004] para la reconstrucción de los vasos sanguíneos es el siguiente:

- Se extrae el árbol de vasos sanguíneos y los puntos de *bifurcación y cruce* utilizando RISA (ver sección 3.1.1) de un par de imágenes. La figura 4.16 (c) y (d) muestra un ejemplo.
- Se detectan los puntos correspondientes entre las dos imágenes, esto lo hace manualmente. En la figura 4.16 (e) se muestran los esqueletos de (a) y (b), los cuales empatan.
- Por último, se reconstruye en 3D el esqueleto de los vasos sanguíneos utilizando el procedimiento descrito anteriormente. La figura 4.16 (f) muestra un ejemplo del resultado obtenido.

En este trabajo de tesis presentado puede ser extendido a *reconstrucción 3D* de un mosaico; partiendo de que en el trabajo presentado por Martínez-Pérez y Espinosa-Romero [2004] utilizan un enfoque bifocal para crear el mosaico de dos imágenes.

Este mismo enfoque puede ser utilizado para crear un mosaico de más de dos imágenes; la hipótesis es la siguiente: en este trabajo de tesis presentado también utilizamos un enfoque bifocal para crear un mosaico con más de dos imágenes (ver sección 4.4) en 2D, es decir, se toma como imagen destino el mosaico generado en un paso previo; tomando ésto como base podemos seguir el proceso que hacen Martínez-Pérez y Espinosa-Romero [2004] para la reconstrucción 3D de los vasos sanguíneos y como resultado obtendríamos un mosaico con más de 3 imágenes en 3D.

Vamos a mencionar las ventajas que existirían sobre el trabajo de Martínez-Pérez y Espinosa-Romero [2004]: el primero es que la etapa de correlación que ellos hacen manualmente sería automático, y el segundo es que se tendría un mosaico en 3D formado por más de dos imágenes.

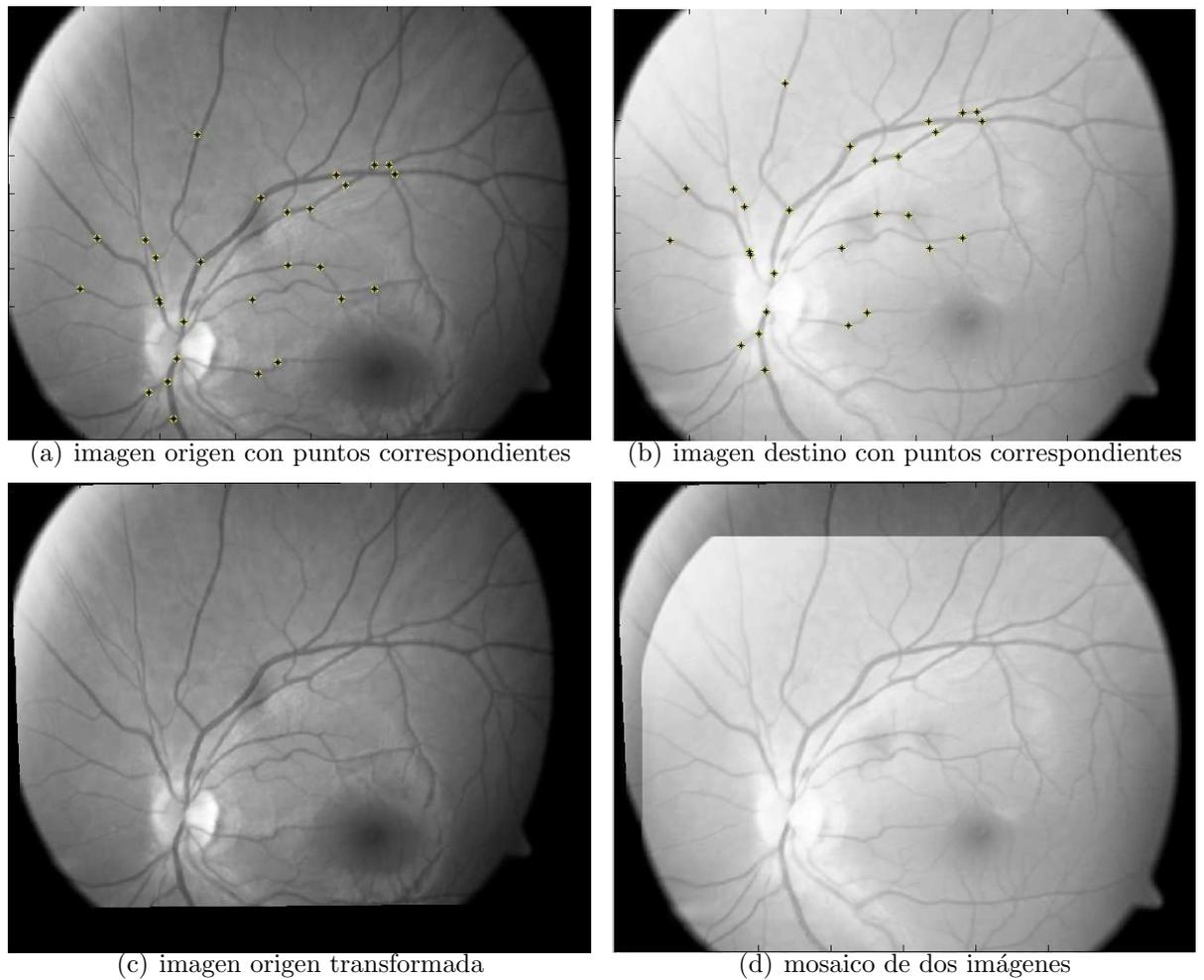
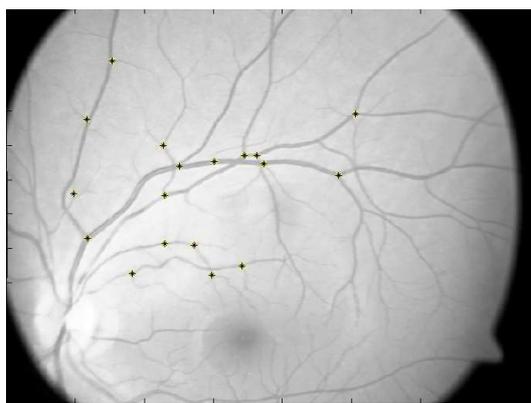
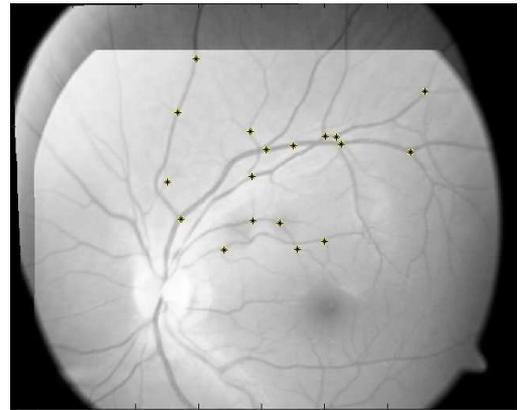


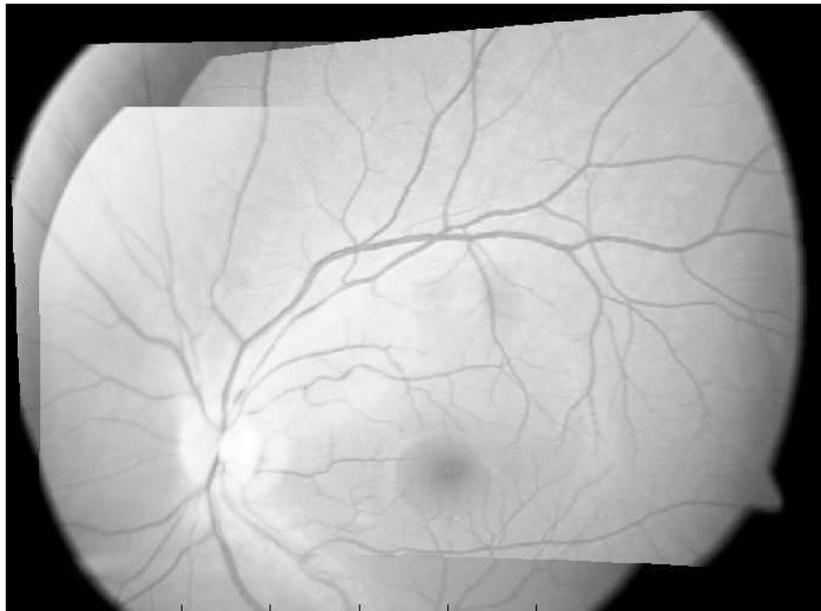
Figura 4.7: Mosaico formado por dos imágenes de fondo de ojo. (a) y (b) Muestra la imagen origen y destino respectivamente, con los puntos correspondientes entre ellas, arrojados por el método LMedS (sección 3.3.3) y mostrados en la figura 3.13. (c) Muestra la imagen origen-transformada. (d) Mezcla la imagen origen-transformada con la imagen destino para crear el mosaico.



(a) imagen origen con puntos correspondientes



(b) imagen origen con puntos correspondientes



(c) mosaico formado por tres imágenes

Figura 4.8: Mosaico formado por tres imágenes de fondo de ojo. (a) y (b) Muestra la imagen origen y destino respectivamente, con los puntos correspondientes. (c) Mezcla la imagen origen-transformada con la imagen destino para crear el mosaico.

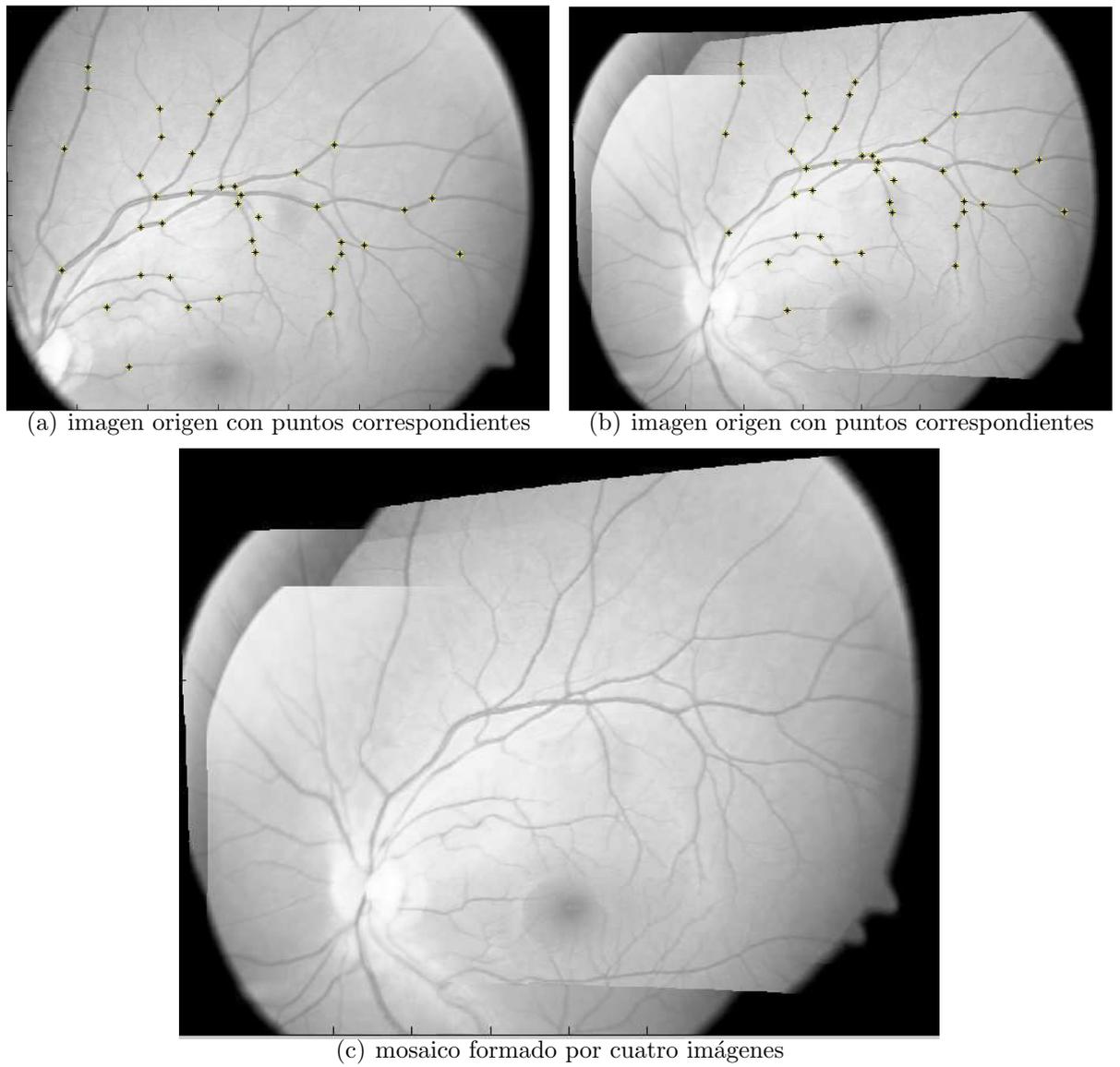
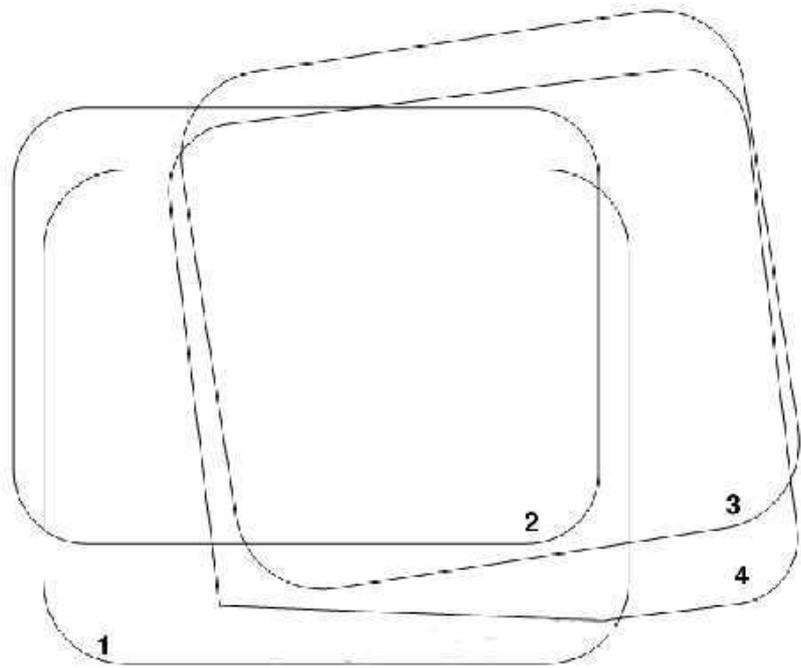


Figura 4.9: (a) y (b) muestran las imágenes origen y destino respectivamente. c) muestra el mosaico construido por cuatro imágenes.



(a) mosaico esquemático

<i>selección automática</i>	<i>imagen origen</i>	<i>imagen destino</i>
1		1
2	2	1
3	3	2
4	4	3

(b) tabla del mosaico 4.8b)

Figura 4.10: (a) Muestra el mosaico esquemático formado por 4 imágenes. (b) Muestra la forma en que fueron seleccionadas e incorporadas al mosaico.

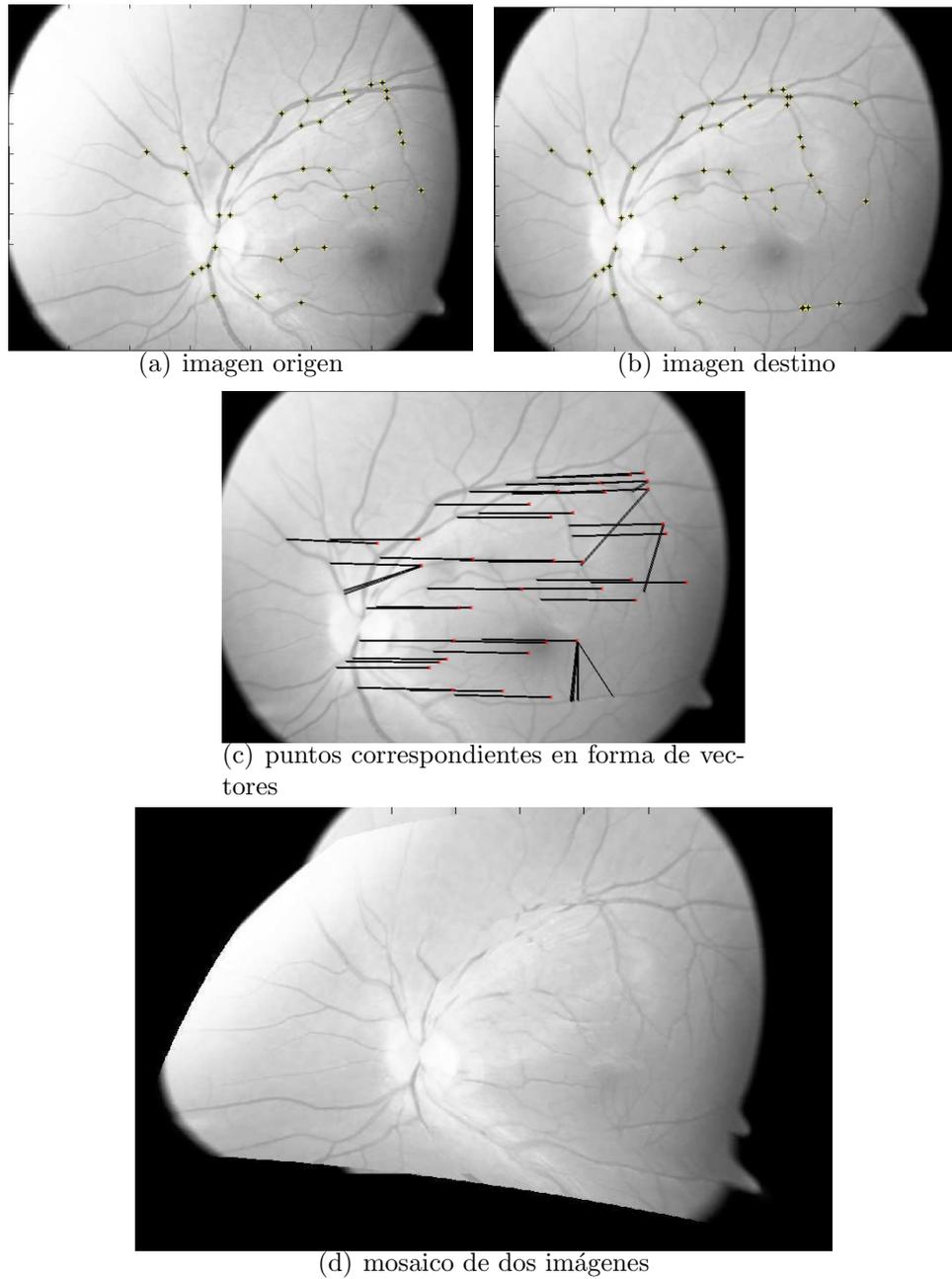


Figura 4.11: Mosaico formado por dos imágenes de fondo de ojo. Tomando los puntos de correspondencias de la etapa de correlación (ZNCC) 3.5. (a) y (b) Muestran el conjunto de puntos correspondientes. (c) Conjunto de puntos correspondientes en forma de vectores, donde se pueden apreciar los puntos erróneos a simple vista. (d) Mosaico mal alineado.

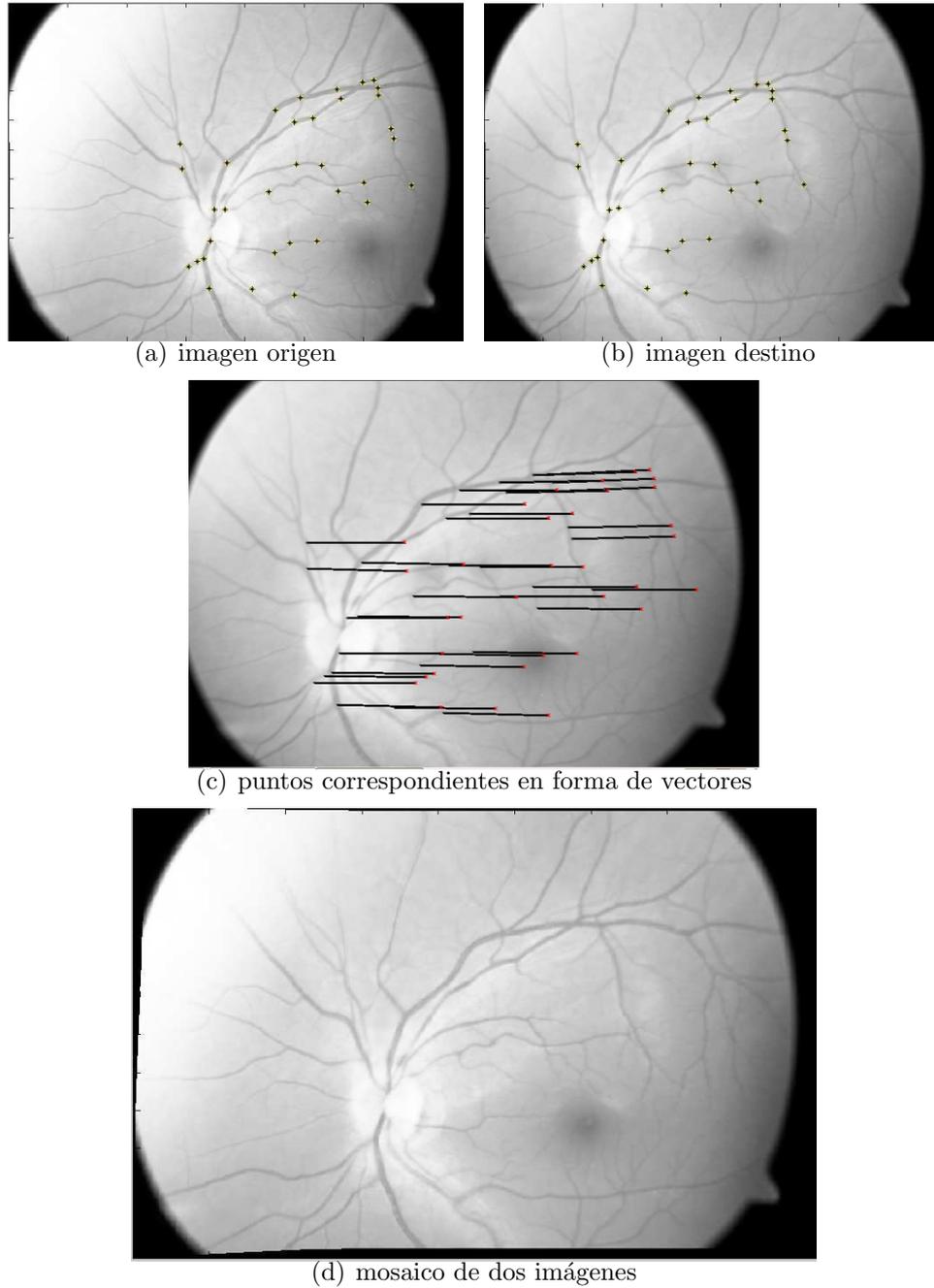


Figura 4.12: (a) y (b) Imagen origen y destino respectivamente con el conjunto de puntos correspondientes entre ambas imágenes. (c) muestra el conjunto de puntos correspondientes de las imágenes en forma de vectores. (d) muestra el mosaico bien alineado.

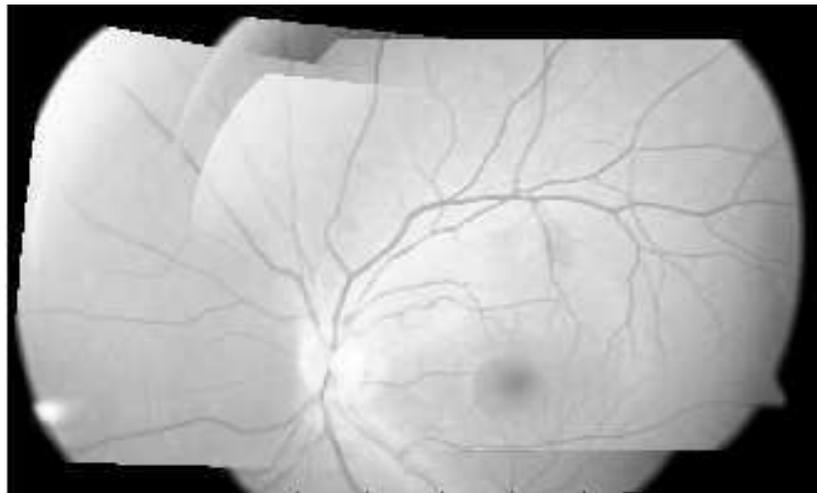
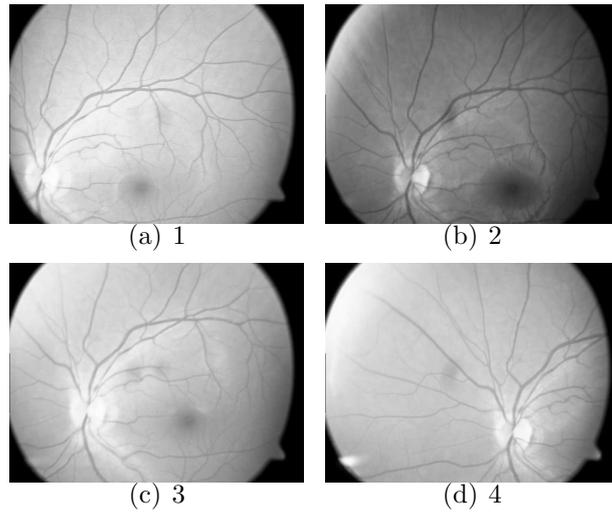


Figura 4.13: Mosaico formado por 4 imágenes de fondo de ojo. (a)-(d) Muestra la secuencia de imágenes en el orden en que fueron escogidas. (e) Muestra el mosaico construido por la secuencia de imágenes de la (a)-(b).

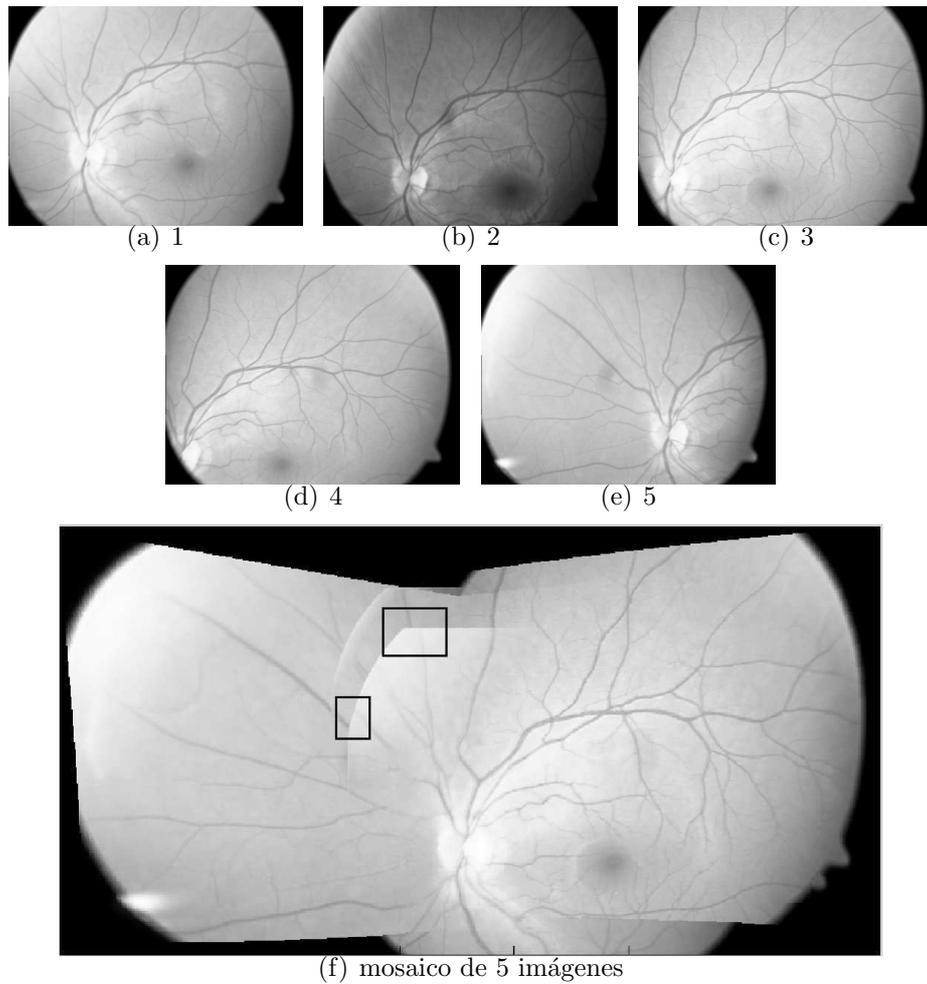


Figura 4.14: Mosaico formado por una secuencia de 5 imágenes de fondo de ojo. (a) - (e) Muestra la secuencia de imágenes en la forma en que fueron escogidas aleatoriamente. f) Muestra el mosaico construido por las imágenes de la (a)-(e) y las áreas mal alineadas son resaltadas por un rectángulo.

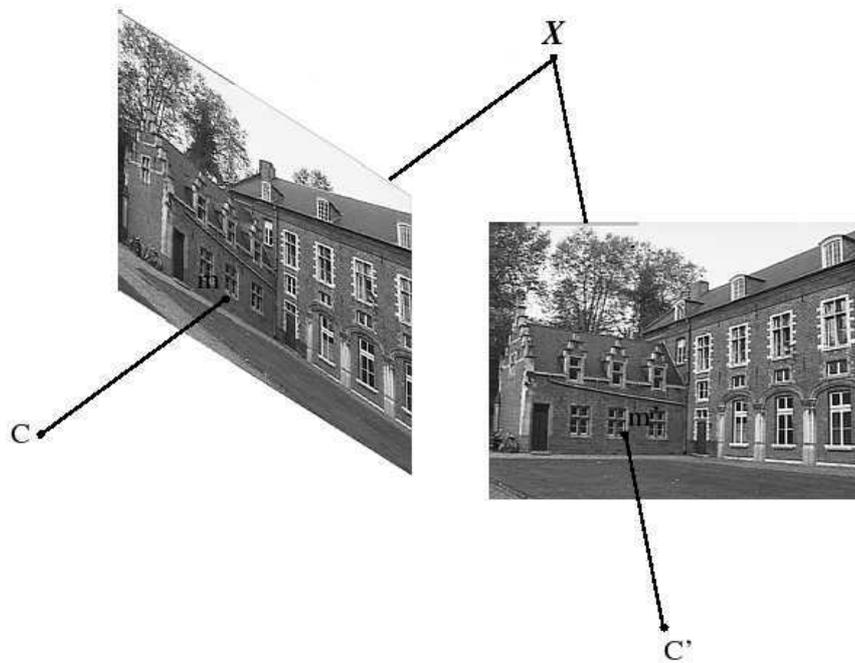


Figura 4.15: Triangulación de puntos correspondientes.  $C$  y  $C'$  son los centros de las cámaras y  $X$  es un punto tridimensional. Fuente tomada de Pollefeys [2000].

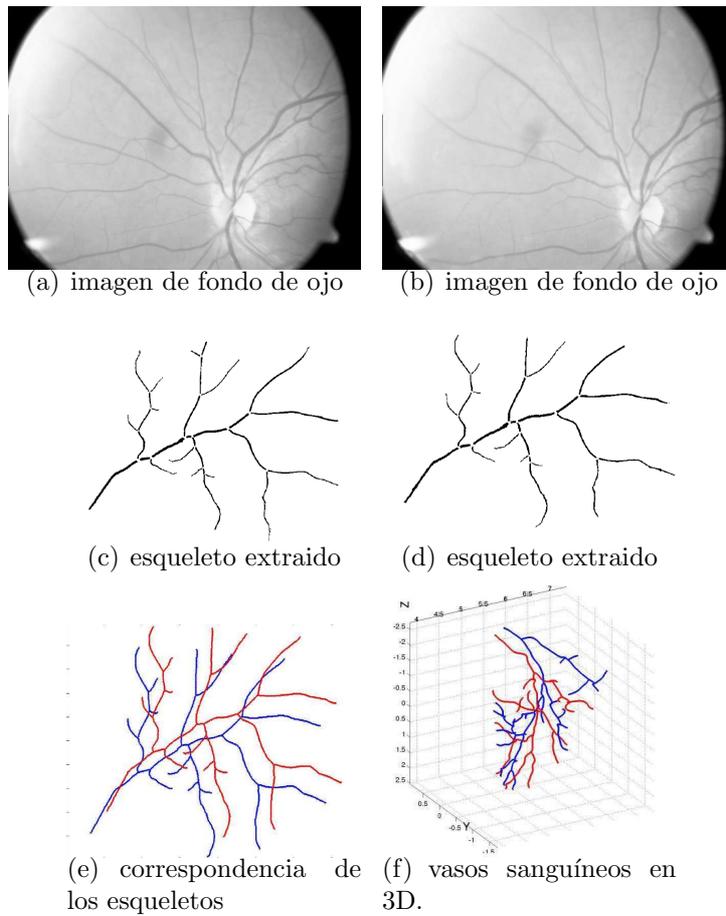


Figura 4.16: (a) y (b) muestran dos imágenes de fondo de ojo. (c) y (d) muestran los esqueletos de los vasos sanguíneos extraídos por RISA. (e) Muestra el resultado del proceso de correlación hecho manualmente. (f) Muestra los vasos sanguíneos de (a) y (b) en 3D.

# Capítulo 5

## Conclusiones y Trabajo Futuro

### 5.1. Conclusiones

El trabajo presentado en esta tesis trata de la creación de un mosaico bidimensional obtenido a partir de una secuencia de imágenes médicas de fondo de ojo. Tomando esto como el objetivo central un enfoque modular al problema de *registro de imágenes* fue basado principalmente en el artículo escrito por Zhang *et al.* [1994]. Para la etapa de la construcción del mosaico nos basamos principalmente en Capel [2004]. Guiándonos por dichos artículos y revisando otros métodos disponibles en la literatura de visión computacional para poder comparar nuestra implementación y hacer notar que los métodos elegidos pueden ser sustituidos sin alterar el proceso, aunque quizá puedan dar resultados ligeramente diferentes.

Todo el proceso de construcción de mosaicos se realiza de manera semi-automática; ésto es debido a que algunos parámetros iniciales como lo son el tamaño de la ventana de búsqueda y la ventana de correlación en la sección 3.2.1 se asignaron de manera independiente y empírica para cada par de imágenes. La reconstrucción se llevó a cabo siguiendo las fases: primero los puntos característicos (bifurcaciones y cruces) fueron obtenidos de las imágenes y se extrajo un conjunto de correspondencias tentativas entre ellos, las cuales encapsulan la información de la geometría epipolar; este conjunto de correspondencias tentativas es empleado para definir y encontrar *correspondencias correctas*, libres de errores. Como paso final se alinearon las imágenes encontrando una función de transformación polinomial de segundo grado, que eliminara la distorsión provocada por la superficie esférica del fondo de ojo. Cabe resaltar que todos los métodos descritos en este trabajo fueron implementados desde cero, y sólo se utilizaron implementaciones ya existentes como el caso del programa de Kovesi [2004] el cual fue usado para hacer una comparación de los resultados obtenidos (en la sección 3.3.3) con nuestra implementación.

La programación de los algoritmos fue realizada en dos etapas y en dos lenguajes de programación. La etapa de *relacionar las imágenes* (capítulo 3) fue programada

en el lenguaje C y la etapa de *creación de mosaicos* (capítulo 4) fue programada en Matlab. Se utilizó una máquina IBM thinkPad T30 con procesador Pentium 4 a 1.80 MHz con 512 MB en RAM. Los tiempos de procesamiento en las dos etapas son los siguientes: en la etapa de *relacionar las imágenes* fue de 0.4 seg para un par imágenes con aproximadamente 59 puntos en cada una; el número de puntos (bifurcaciones y cruces) que contenían las imágenes utilizadas para la creación del mosaico van de 43 a 66 puntos. Otras imágenes de prueba que se utilizaron las cuales contenían 99 y 88 puntos (ver cuadro 3.7) tuvieron un tiempo de procesamiento en la etapa de *relacionar imágenes* de 0.6 seg. En la etapa de *creación de mosaicos* realizado en Matlab el tiempo de procesamiento en la creación de un mosaico con 2 imágenes fue de 5 seg; para el mosaico de 4 imágenes fue de 15 seg y para el mosaico de 5 imágenes fue de 21 seg.

Existen varios puntos que deben ser comentados, ya que deben ser tomados en cuenta a la hora de implementar los métodos descritos y realizar experimentos propios para obtener buenos resultados. Se debe elegir un método para encontrar puntos característicos que nos asegure un buen grado de *repetibilidad*, es decir, el número de puntos repetidos entre dos imágenes con respecto al total de puntos detectados [Patrón Pérez, 2006]. En el caso de nuestra implementación con imágenes de fondo de ojo, obtenemos buenos resultados utilizando RISA. Un buen grado de *repetibilidad*, nos facilita la detección de correspondencias correctas. También es importante una buena selección en las imágenes para estimar el marco inicial de la construcción del mosaico. Una mala selección del marco inicial nos lleva a una pésima reconstrucción. Otro aspecto a considerar es que en nuestra implementación los buenos resultados obtenidos en gran medida se debe a que RISA arroja un conjunto de puntos filtrados, es decir, todos o la mayoría son bifurcaciones y cruces de vasos sanguíneos los cuales nos facilitan las siguientes etapa que nos permiten construir el mosaico.

En la literatura encontramos varios autores que realizan construcción de un mosaico bidimensional con una secuencia de imágenes médicas de fondo de ojo (ver experimentos en el capítulo cuatro, sección 1.1.5), todos estos autores (incluyendo nosotros) utilizan una función polinomial de segundo orden con 6 términos como función de transformación para la creación del mosaico. La mayor diferencia en nuestra implementación con el resto de los autores es en la etapa de selección de puntos característicos extraídos por RISA (sección 3.1.1). Y en la etapa de registro de imágenes; nosotros utilizamos algoritmos clásicos que no habían sido utilizados en la obtención de *correspondencia correctas* (sin errores). Esta etapa es crucial para la construcción del mosaico, ya que una sola correspondencia con un error grande puede darnos resultados no deseables.

Finalmente, el proceso de construcción de mosaico en 2D puede ser útil para lograr la reconstrucción de un mosaico en 3D de vasos sanguíneos (sección 4.6).

## 5.2. Trabajo Futuro

Concluimos indicando algunas de las áreas más importantes de este trabajo donde creemos se pueden realizar modificaciones que ayudarán a mejorar los resultados y la eficiencia del proceso.

- Primero, una investigación a fondo en cuanto a la selección de valores óptimos asignados a los parámetros utilizados por los métodos descritos en este trabajo, tales como las dimensiones de las ventanas de búsqueda y correlación. Éstos pueden ser automatizados midiendo la distancia que existe entre los nervios ópticos de un par de imágenes (desde el centro del nervio óptico). De esta forma se podría saber aproximadamente la distancia que existe entre ellas. Esta hipótesis se basa en que las imágenes de fondo de ojo no tienen una rotación grande, por lo que una distancia de traslación sería una buena aproximación.
- Segundo, realizar de manera automática todo procesamiento de las imágenes y la construcción del mosaico de manera que pueda ser utilizada en tiempo real. Es decir, desde que se toman las imágenes de fondo de ojo al sujeto, poder crear el mosaico.
- Tercero, para evitar el problema de alineación (el mostrado en el experimento 4, sección 4.5) se puede probar obtener los puntos de transformación de una imagen contra todas las imágenes con quien sobrelapa en el mosaico.

# Bibliografía

- Becker, D. E., Can, A., Turner, N. J., Tanenbaum, H. L., y Roysan, B. (1998). Image processing algorithms for retinal montage synthesis, mapping, and real-time location determination. *IEEE Transactions on Biomedical Engineering*, **45**, 105–118.
- Burden, R. y Faires, J. D. (1985). *Análisis Numérico*. Grupo Editorial Iberoamérica.
- Burt, P., Yen, P., y Xu, X. (1982). Local correlation measures for motion analysis: A comparative study. *IEEE Proceedings of Pattern recognition and Information Processing*, pages 269–274.
- Can, A., Stewart, C., Roysam, B., y Tanenbaum, H. (2002). A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: mosaicing the curved human retina. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **24**, 412–419.
- Capel, D. (2004). *Image Mosaicing and Super-resolution*. Springer-Verlag, London.
- Deguchi, K., Noami, J., y Hontani, H. (2000). 3d fundus pattern reconstruction and display from multiple fundus images. *Pattern Recognition, 2000 processing. 15th International Conference on*, **4**, 94–97.
- Gonzalez, R. y Woods, R. (1993). *Digital Image Processing*. Addison-Wesley, 2nd edition.
- Hartley, R. y Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK.
- Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intel Ligence*, **19**, 580–593.
- Heckbert, P. A. (1989). *Fundamentals of Texture Mapping and Image Warping*. Master's thesis, University of California, Berkeley.
- Kovesi, P. (2004). Matlab and octave functions for computer vision and image processing. <http://www.csse.uwa.edu.au/pk/Research/MatlabFns/>.
- Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, **293**, 133–135.

- Martínez-Pérez, M. E. (2000). *Computer Analysis of the Geometry of the Retinal Vasculature*. Ph.D. thesis, Imperial Collage. University of London.
- Martínez-Pérez, M. E. y Espinosa-Romero, A. (2004). Retinal blood vessel 3d reconstruction from two views. In *4th Indian Conference on Computer Vision and Graphics*, pages 258–263.
- Martínez-Pérez, M. E., Hughes, A. D., Stanton, A., Thom, S. A., Bharath, A. A., y Parker, K. H. (2002). Retinal vascular tree morphology: A semi-automatic quantification. *IEEE Transaction on Biomedical Engineering*, **49**, 912–916.
- Martínez-Pérez, M. E., Hughes, A. D., Stanton, A., Thom, S. A., Bharath, A. A., y Parker, K. H. (2007). Segmentation of blood vessels from red-free and fluorescein retinal images. *Medical image analysis*, **11**, 47–61.
- Milgram, D. (1975). Computer methods for creating photomosaics. *IEEE Trans. Computer*, **23**, 1113–1119.
- Patrón Pérez, A. A. (2006). *Reconstrucción tridimensional a partir de una secuencia de imágenes*. Master's thesis, Facultad de Matemáticas, Universidad Autónoma de Yucatán.
- Pollefeys, M. (2000). *Tutorial on 3D Modeling from Images*. Lecture Note, <http://www.cs.unc.edu/marc/tutorial/tutorial02.html>.
- Researchers (2006). Computer vision group. *Universitat, Jaume*. <http://www.vision.uji.es>.
- Rousseeuw, P. J. y Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, New York.
- Ryan, N., Heneghan, C., y DeChazal, P. (2004). Registration of digital retinal images using landmark correspondence by expectation maximization. *Image and Vision Computing*, **22**, 883–898.
- Shen, H., Stewart, C. V., Roysam, B., y Lin, G. (2003). Frame-rate spatial referencing based on invariant indexing and alignment with application to online retinal image registration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **25**, 379–384.
- Tsai, R. y Huang, T. (1984). Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intel Ligence*, **29**, 1147–1152.
- Zhang, Z., Deriche, R., y Faugeras, O. (1994). A robust technique for matching two uncalibrated images through the recovery of the unknow epipolar geometry. Rapport de recherche no. 2273, INRIA Sophia Antipolis.