



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Reconstrucción Tridimensional de Objetos Mediante Técnicas Evolutivas

Tesis que presenta:

Israel Vite Silva

para obtener el grado de:

Doctor en Ciencias

En la especialidad de:

Ingeniería Eléctrica

Opción:

Computación

Director de la Tesis

Dr. Luis Gerardo de la Fraga

Resumen

Uno de los problemas clásicos del área de Visión por Computadora es la reconstrucción tridimensional métrica a partir de imágenes adquiridas por una cámara sin calibración previa. Este problema involucra, específicamente, los siguientes pasos: (1) la triangulación, (2) la calibración de la cámara y (3) la estimación de los parámetros de orientación y posición de la cámara.

El paso (1), la triangulación, tiene por objetivo encontrar la posición de un punto en el espacio a partir de la intersección de dos líneas que parten de la correspondencia de puntos en dos imágenes. Con la presencia de ruido, es muy probable que esta intersección no ocurra. En tal caso es necesario encontrar la mejor aproximación. Este problema es crítico en las reconstrucciones afín y proyectiva, donde no hay definida una métrica. En esta tesis se planteó la triangulación como un problema de optimización multi-objetivo, el cual se resolvió mediante un algoritmo evolutivo llamado NSGA-II. La transformación a un problema multi-objetivo permitió obtener mejores resultados en comparación con los mejores métodos reportados.

Los pasos (2) y (3) consisten en encontrar los parámetros de la cámara (parámetros de calibración, orientación y posición) a partir de un conjunto de correspondencias de puntos extraídas de dos o más imágenes. Sin embargo, existen superficies y configuraciones de la cámara conocidas como críticas donde se ha demostrado que es imposible extraer los parámetros de la cámara al usar, solamente, correspondencias de puntos. En esta tesis se proponen tres soluciones que plantean a la calibración de la cámara, a la estimación de los parámetros de orientación y posición, y al cálculo de los puntos tridimensionales (triangulación), como un problema de optimización, con base en el modelo de la cámara oscura, y emplean un algoritmo evolutivo llamado evolución diferencial para su resolución. Al emplear, directamente, el modelo de la cámara oscura es posible encontrar los parámetros de la cámara y obtener la reconstrucción tridimensional, aún en la configuración crítica de translación pura y en superficies críticas (puntos que yacen sobre un mismo plano).

Otro problema importante dentro del área de Visión por Computadora, consiste en obtener la elipse que mejor se ajuste a un conjunto de puntos dado. Los algoritmos que resuelven este problema sólo minimizan el cuadrado de la distancia euclidiana entre la cónica y los puntos mediante métodos de gradiente, aunque no siempre convergen a la solución óptima. En esta tesis se planteó el ajuste de puntos a una elipse como un problema de optimización, el cual fue resuelto mediante un algoritmo genético y la evolución diferencial. Esta solución tiene la flexibilidad de minimizar la distancia euclidiana o su cuadrado. Al emplear la distancia euclidiana se les da mayor peso a los puntos más cercanos, a diferencia de su cuadrado, en el cual se da mayor peso a los puntos más lejanos como se observa en los resultados.

Abstract

A classic problem for Computer Vision is the three-dimensional metric reconstruction from two uncalibrated views. This problem entails the following steps: (1) triangulation, (2) camera calibration and (3) estimation of orientation and position parameters of each view.

Step (1), triangulation, consists of finding the position of a point in real space as the intersection of two rays given by their corresponding images. When noise is present, the intersection of two rays may not be detected, in which case it is necessary to find the best approximation to the meet point. This problem is critical in affine and projective reconstruction, since no metric is present. In this thesis, we have proposed a solution that transforms the triangulation problem into multi-objective optimization problem, which was solved using an evolutionary algorithm called NSGA-II. The treatment as a multi-objective problem has been able to obtain better results compared with the best reported methods.

Steps (2) and (3) consist on estimating the camera parameters (calibration, orientation and position parameters) using a set of corresponding points extracted from two or more images. However, there are critical surfaces and critical camera configurations where it is not possible to recover these parameters using just the set of corresponding points. In this thesis we have proposed three solutions dealing with the camera calibration, the estimation of the orientation and position parameters of each view, and the calculation of the three-dimensional points (triangulation), all of them were merged into a general optimization problem based on the pinhole camera model and solved with an evolutionary algorithm, called differential evolution. This makes possible to estimate the camera parameters and obtaining thus the three-dimensional isometric reconstruction, even on critical camera configuration (pure translation) and critical surfaces (points lie on the same plane).

We also applied successfully evolutionary algorithms within another Computer Vision problem: ellipse fitting. The problem consists of finding the best ellipse fitting a set of given points. The current algorithms minimize the square of Euclidean distance between the conic and the set of given points through gradient methods. However, such methods do not always converge to the optimal solution. In this thesis our approach based on genetic algorithm and differential evolution were successful, minimizing the Euclidean distance or its square. With the Euclidean distance, the nearest points get greater weight, instead the squared Euclidean distance gives greater weights to distant points.

Dedicada con mucho cariño a mis padres

Paz y Rita y a mi hermana Kenia.

Los quiero mucho.

Agradecimientos

Quiero agradecer de todo corazón a mis padres y hermana por su amor, cariño, apoyo y comprensión que me ofrecieron durante los años que estuve realizando mi tesis, así como la fuerza y el coraje que me inculcaron para terminarla. Es una alegría inmensa ser parte de esta familia.

Agradezco profundamente a mi asesor, Dr. Luis Gerardo de la Fraga, por la oportunidad que me dio de trabajar a su lado y haberme guiado a la culminación de esta tesis. Gracias Doctor por confiar en mí, por tenerme paciencia, así como ayudarme y apoyarme en los momentos más difíciles. Siempre le estaré agradecido.

Agradezco al Dr. Carlos Artemio Coello Coello y al Dr. Guillermo Morales Luna por todo el tiempo dedicado a la corrección de esta tesis, así como sus invaluable comentarios que tuve de su parte para mejorarla.

También agradezco al Dr. Gregorio Toscano por su contribución y por el tiempo que dedicó a ayudarme en este trabajo de tesis. Gracias camarada no lo olvidaré.

Gracias a todos mis amigos, Alfredo, Amílcar, Anahí, Carlos, Edna, Isaí, Jaime, Lucio, Luis G., Nidia, Paco, Rosario y los que me faltan, por brindarme su confianza, amistad y haber hecho mi estadía en el CINVESTAV más amena. En especial quiero agradecer a aquellas personas que me conocen aún más, Carolina, Daniel, Jorge, Luis S., Mario, Mireya, Rolando, Ricardo, Oscar, Noel, con las que he compartido gratas

experiencias, me han hecho crecer como persona y han estado conmigo en las buenas y en las malas. Gracias mejores amigos (estás incluido Goyo).

Además, deseo agradecer a una persona que hizo mi vida especial, siempre me brinda su apoyo y comprensión incondicional, me llena con su amor y me hace pasar agradables momentos. Gracias Ara por ser parte de mi vida.

De igual forma agradezco al CONACYT por la beca otorgada durante los años en los que realicé este trabajo, la extensión que me concedió para culminar mi tesis y los apoyos recibidos mediante el proyecto 45306 del CONACYT.

Índice general

Resumen	I
Abstract	III
Agradecimientos	VII
1. Introducción	1
1.1. Motivación	3
1.2. Objetivo general y objetivos particulares	4
1.3. Contribuciones	5
1.4. Estructura general del documento	5
2. Estado del arte	7
2.1. Reconstrucción 3D por medio de imágenes	7
2.1.1. Reconstrucción 3D partir de cuatro o más imágenes	7
2.1.2. Reconstrucción 3D a partir de tres imágenes	8
2.1.3. Reconstrucción 3D a partir de dos imágenes	8
2.2. Triangulación	11
3. Marco teórico	13
3.1. Fundamentos de Visión por Computadora	13
3.1.1. Geometría proyectiva	13
3.1.2. Transformaciones	15
3.1.3. Distancia euclidiana y su cuadrado	16
3.1.4. Modelo de la cámara oscura	18
3.1.5. Geometría epipolar	23
3.2. Reconstrucción tridimensional	26
3.2.1. Ambigüedad de la reconstrucción	26
3.2.2. Tipos de reconstrucciones	27
3.2.3. Reconstrucción afín	28
3.2.4. Reconstrucción métrica	29
3.2.5. Calibración de la cámara	30
3.2.6. Estimar los parámetros de orientación y posición	31
3.2.7. Triangulación	32
3.2.8. Escenas planas y homografía	33
3.3. Fundamentos de optimización	35
3.3.1. Análisis numérico	37

3.3.2. Computación evolutiva	39
3.3.3. Algoritmos evolutivos empleados para la optimización con pa- rámetros reales	39
3.3.4. Algoritmos genéticos	39
3.3.5. Algoritmos evolutivos de optimización multi-objetivo	49
4. Metodología	53
4.1. Triangulación	53
4.1.1. Definición del problema	54
4.1.2. Solución propuesta mediante algoritmos evolutivos	55
4.2. Reconstrucción tridimensional a partir de dos imágenes	58
4.2.1. Definición del problema	58
4.2.2. Solución propuesta mediante algoritmos evolutivos	60
4.3. Reconstrucción tridimensional a partir de tres o más imágenes	66
4.3.1. Definición del problema	66
4.3.2. Solución propuesta mediante algoritmos evolutivos	67
5. Resultados	69
5.1. Triangulación	69
5.1.1. Solución propuesta aplicada al problema de triangulación mono- objetivo	69
5.1.2. Solución propuesta aplicada al problema de triangulación multi- objetivo	71
5.2. Reconstrucción tridimensional a partir de dos imágenes	73
5.2.1. SPI en configuraciones no críticas	74
5.2.2. SPII en configuraciones no críticas	77
5.2.3. SPII en configuraciones críticas	80
5.3. Reconstrucción tridimensional a partir de tres o más imágenes	85
6. Discusión	91
6.1. Ruido en los puntos bidimensionales	91
6.2. Triangulación	92
6.3. Reconstrucción tridimensional a partir de dos imágenes	94
6.4. Reconstrucción tridimensional a partir de tres o más imágenes	100
7. Conclusiones	101
7.1. Contribuciones	103
7.2. Publicaciones	104
7.3. Trabajo futuro	105
Apéndices	105
A. Ajuste de la mejor elipse a un conjunto de puntos	107
A.1. Descripción del problema	107
A.1.1. Cálculo de los puntos sobre la elipse ortogonales a los puntos dados	109
A.2. Resultados al emplear algoritmos evolutivos	110
A.2.1. Ajuste de la mejor elipse sin restricciones	110

A.2.2. Ajuste de la mejor elipse con restricciones	112
A.3. Algoritmo híbrido	114
A.4. Resultados al emplear el algoritmo híbrido	115
B. Experimentos con la SPII en configuraciones no críticas	117
B.1. Reconstrucción tridimensional al usar puntos de imágenes sintéticas .	117
B.2. Reconstrucción tridimensional al usar puntos de imágenes reales . . .	127
C. Experimentos con la SPII en configuraciones críticas	131
C.1. Reconstrucción tridimensional al usar puntos de imágenes sintéticas .	131
C.2. Reconstrucción tridimensional al usar puntos de imágenes reales . . .	146
D. Comparación de los algoritmos evolutivos	149
D.1. Análisis comparativo de los AE para la optimización con parámetros reales	149
D.1.1. Funciones uni-modales con 10 variables	150
D.1.2. Funciones multi-modales resueltas con 10 variables	150
D.1.3. Funciones uni-modales con 30 variables	151
D.1.4. Funciones multi-modales resueltas con 30 variables	151
D.1.5. Complejidad de los algoritmos	151
D.1.6. Discusión del análisis comparativo	152
D.2. Comparación entre el algoritmo G-CMA-ES y la ED	152
D.2.1. Discusión de resultados	154

Índice de tablas

3.1. Versiones de la evolución diferencial.	48
5.1. Estadísticas del error utilizando la ecuación (5.1) a partir de 30 ejecuciones de la SPI sobre imágenes sintéticas.	77
5.2. Tiempo promedio de las 30 ejecuciones al emplear la SPI.	77
5.3. Estadísticas del error utilizando la ecuación (5.1) a partir de 30 ejecuciones la SPII sobre imágenes sintéticas.	78
5.4. Tiempo promedio de las 30 ejecuciones al emplear la SPII.	79
5.5. Estadísticas del error a partir de 40 ejecuciones de la SPII sobre imágenes reales.	80
5.6. Estadísticas de la longitud foco a partir de 40 ejecuciones de la SPII sobre imágenes reales.	80
5.7. Parámetros de la cámara para la primera configuración crítica.	82
5.8. Parámetros de la cámara para la segunda configuración crítica.	82
5.9. Estadísticas del error a partir de 40 ejecuciones de la SPII para la primera configuración crítica.	82
5.10. Estadísticas del error a partir de 40 ejecuciones de la SPII para la segunda configuración crítica.	82
5.11. Estadísticas del error a partir de 40 ejecuciones de la SPII sobre imágenes reales en la primera configuración crítica.	85
5.12. Parámetros de la cámara estimados por la SPII a partir de imágenes reales en la primera configuración crítica.	85
5.13. Parámetros de la cámara para obtener las cuatro imágenes del anillo.	86
5.14. Parámetros de la cámara para obtener las cuatro imágenes de la figura formada por cubos.	87
5.15. Estadísticas del error de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento del anillo.	89
5.16. Estadísticas del error de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento de la figura formada por cubos.	89
5.17. Estadísticas del error acumulado de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento del anillo.	90
5.18. Estadísticas del error acumulado de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento de la figura formada por cubos.	90

A.1. Resultados del ajuste de la mejor elipse sin restricciones con base en la distancia euclidiana.	112
A.2. Resultados del ajuste de la mejor elipse sin restricciones con base en el cuadrado de la distancia euclidiana.	112
A.3. Resultados del ajuste de la mejor elipse sin restricciones utilizando como función la distancia euclidiana.	113
A.4. Resultados del ajuste de la mejor elipse sin restricciones utilizando como función el cuadrado de la distancia euclidiana.	113
B.1. Parámetros de la cámara utilizados para el tercer experimento.	118
B.2. Estadísticas del error a partir de 40 ejecuciones de la SPII.	118
B.3. Estadísticas del parámetro intrínseco (longitud del foco).	118
B.4. Porcentaje del error relativo al comparar la longitud del foco real y su valor estimado con diferentes niveles de ruido.	118
B.5. Parámetros de la cámara utilizados para el cuarto experimento.	119
B.6. Parámetros de la cámara en su forma canónica.	119
B.7. Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido y correspondencias de puntos.	121
B.8. Parámetros de la cámara utilizados para generar las dos imágenes de cada una de las 6 figuras sobre configuraciones no críticas.	123
B.9. Parámetros de la cámara utilizados para el sexto experimento.	125
B.10. Estadísticas del error a partir de 40 ejecuciones de la SPII para el sexto experimento.	127
B.11. Estadísticas de la longitud del foco a partir de 40 ejecuciones de la SPII para el sexto experimento.	127
B.12. Estadísticas del error de 30 ejecuciones de la SPII sobre imágenes reales de la casa de campo.	129
B.13. Parámetros de la cámara estimados por la SPII.	129
B.14. Estadísticas del error de 40 ejecuciones de la SPII sobre imágenes del juguete.	130
C.1. Parámetros de la cámara para el tercer experimento (configuración crítica I).	132
C.2. Parámetros de la cámara para el tercer experimento (configuración crítica II).	132
C.3. Estadísticas del error a partir de 40 ejecuciones de la SPII para el tercer experimento, sobre la primera configuración crítica.	133
C.4. Estadísticas del error a partir de 40 ejecuciones de la SPII para el tercer experimento, sobre la segunda configuración crítica.	133
C.5. Parámetros de la cámara en forma canónica de la primera configuración crítica para el tercer experimento.	133
C.6. Parámetros de la cámara en forma canónica de la segunda configuración crítica para el tercer experimento.	133
C.7. Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.	133
C.8. Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.	134

C.9. Estadísticas del porcentaje de error relativo entre las magnitudes de los puntos tridimensionales reales y los puntos estimados a partir de 40 ejecuciones de la SPII.	137
C.10. Parámetros de la cámara utilizados para generar las dos imágenes por cada una de las 6 figuras, sobre la primera configuración crítica. . . .	137
C.11. Parámetros de la cámara utilizados para generar las dos imágenes por cada una de las 6 figuras, sobre la segunda configuración crítica. . . .	137
C.12. Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.	142
C.13. Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.	143
C.14. Parámetros de la cámara utilizados en el quinto experimento sobre la primera configuración crítica.	144
C.15. Parámetros de la cámara utilizados en el quinto experimento sobre la segunda configuración crítica.	145
C.16. Estadísticas del error a partir de 40 ejecuciones de la SPII para el quinto experimento, sobre la primera configuración crítica.	145
C.17. Estadísticas del error a partir de 40 ejecuciones de la SPII para el quinto experimento, sobre la segunda configuración crítica.	146
C.18. Comparación entre los parámetros de la cámara reales y los estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.	146
C.19. Comparación entre los parámetros de la cámara reales y los estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.	146
C.20. Estadísticas del error a partir de 40 ejecuciones de la SPII, al utilizar puntos de imágenes reales.	148
C.21. Estadísticas de la longitud foco a partir de 40 ejecuciones de la SPII, al utilizar puntos de imágenes reales.	148
D.1. Algoritmos presentados en la sesión del CEC'2005.	149
D.2. Complejidad de los algoritmos (medida en segundos).	152
D.3. Parámetros de la cámara para generar los puntos en las dos imágenes.	153
D.4. Estadísticas del error del primer experimento.	153
D.5. Estadísticas del error del segundo experimento.	153
D.6. Estadísticas de la longitud del foco del primer experimento.	154
D.7. Estadísticas de la longitud del foco del segundo experimento.	154
D.8. Complejidad de los algoritmos.	154

Índice de figuras

1.1. Proceso de la reconstrucción tridimensional.	2
3.1. Forma de un cuadrado tras aplicar las cuatro clases de transformaciones. (a) isometría, (b) transformación de similitud, (c) transformación afín y (d) transformación proyectiva.	17
3.2. Forma de un cubo tras aplicar las cuatro clases de transformaciones. (a) isometría, (b) transformación de similitud, (c) transformación afín y (d) transformación proyectiva.	17
3.3. Representación del modelo de la cámara oscura.	18
3.4. Plano de la imagen frente al centro de proyección.	19
3.5. Parámetros de la cámara adicionales.	21
3.6. Transformación entre el sistema de coordenadas del mundo y el sistema de coordenadas de la cámara.	22
3.7. Modelo de la geometría epipolar.	24
3.8. Reconstrucción proyectiva.	28
3.9. Reconstrucción afín.	29
3.10. Reconstrucción métrica.	30
3.11. Matriz de calibración: (a) bajo la suposición de la matriz expresada en (3.33) con cierto valor de oblicuidad, (b) bajo la suposición de la matriz expresada en (3.34).	31
3.12. Las cuatro soluciones a partir de la matriz esencial E . Entre los lados izquierdo y derecho se encuentra una línea base que refleja las soluciones (a) y (c) con (b) y (d). Entre las soluciones de arriba y abajo, la cámara B rota 180 grados alrededor de la línea base.	33
3.13. Triangulación.	33
3.14. Método de Newton.	38
3.15. Método de Levenberg-Marquadt.	38
4.1. Triangulación.	54
4.2. Valor seleccionado del frente de Pareto.	58
4.3. Configuración crítica I: ejes ópticos paralelos.	59
4.4. Gráfica de la matriz fundamental no degenerada.	60
4.5. Gráfica de la matriz fundamental en la primera degeneración.	60
4.6. Configuración crítica II: centros ópticos equidistantes al punto donde se intersecan los ejes ópticos.	60
4.7. Gráfica de la matriz fundamental en la segunda degeneración.	61

4.8. Minimización de la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados.	61
5.1. Comparativa de los métodos LLS, Poly-Abs, algoritmo genético (GA) y algoritmo de optimización por cúmulos de partículas (PSO) para el problema de triangulación bajo una reconstrucción proyectiva.	70
5.2. Comparación de los métodos NSGA-II, LLS y Poly-Abs bajo una reconstrucción proyectiva. El número de correspondencias de puntos para calcular la matriz fundamental es de: (a) 8, (b) 12, (c) 16 y (d) 20.	72
5.3. Comparación de los métodos NSGA-II, LLS y Poly-Abs bajo una reconstrucción afín. El número de correspondencias de puntos para calcular la matriz fundamental es de: (a) 8, (b) 12, (c) 16 y (d) 20.	73
5.4. Esfera sintética.	74
5.5. Gráfica de convergencia de una ejecución al emplear la SPI.	75
5.6. Reconstrucción tridimensional de la esfera: (a) generación 10,000, (b) generación 100,000, (c) generación 250,000 y (d) generación 500,000.	76
5.7. Gráfica de convergencia de una ejecución al emplear la SPII.	78
5.8. Reconstrucción tridimensional métrica. (a) esfera real, (b) esfera reconstruida sin ruido en los puntos, (c,d,e) esfera reconstruida cuando hay 1, 2 y 3 píxeles de ruido gaussiano RMS, respectivamente. El subíndice uno representa la ejecución con el menor error y el subíndice dos la ejecución con el mayor error.	79
5.9. Imágenes reales de una cuadrícula. Los puntos que se emplearon están marcados en ambas imágenes.	80
5.10. Reconstrucción tridimensional de la cuadrícula al emplear la SPII.	81
5.11. Figura regular tridimensional.	81
5.12. Gráficas de convergencia a partir de correspondencias de puntos sin ruido en: (a) la primera configuración crítica y (b) la segunda configuración crítica.	83
5.13. Reconstrucción de la figura regular. (a,b,c,d) indica el nivel de ruido y el subíndice (1,2) indica el tipo de configuración crítica.	83
5.14. Diferente vista de la reconstrucción tridimensional métrica al utilizar la SPII, a partir de correspondencias de puntos sin ruido sobre la segunda configuración crítica.	84
5.15. Imágenes reales de un juguete sobre la primera configuración crítica. Los puntos que se emplearon están marcados en ambas imágenes.	85
5.16. Tres vistas de la reconstrucción tridimensional del juguete.	86
5.17. Objetos tridimensionales: (a) parte de un anillo y (b) figura formada por cubos.	86
5.18. Imágenes de una parte de un anillo: (a) imagen 1, (b) imagen 2, (c) imagen 3 y (d) imagen 4.	87
5.19. Imágenes de la figura formada por cubos: (a) imagen 1, (b) imagen 2, (c) imagen 3 y (d) imagen 4.	88
5.20. Reconstrucción tridimensional métrica de una parte de un anillo a partir de: (a) la primera y la segunda imagen, (b) añadir la tercera imagen y (c) añadir la cuarta imagen (con correspondencias de puntos sin ruido).	89

5.21. Reconstrucción tridimensional métrica de la figura formada por cubos a partir de: (a) la primera y la segunda imagen, (b) añadir la tercera imagen y (c) añadir la cuarta imagen (con correspondencias de puntos sin ruido).	89
6.1. Reconstrucción tridimensional métrica. (a) esfera real, (b,c) esferas reconstruidas cuando hay 2 y 3 píxeles de ruido gaussiano a partir de las peores ejecuciones.	95
6.2. Cubo reconstruido mediante el método convencional.	96
A.1. Ajuste de la mejor elipse a un conjunto de puntos	108
A.2. Ejemplos de la curva con base en la ecuación (A.3), para una elipse con $a = 4$ y $b = 2$. En la figura se muestran los valores del punto dado (x, y) para ambas curvas.	110
A.3. Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en la distancia euclidiana (excepto el LSODF).	111
A.4. Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en el cuadrado de la distancia euclidiana.	111
A.5. Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del AG.	112
A.6. Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del ED.	112
A.7. Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en la distancia euclidiana (excepto el LSODF).	114
A.8. Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en el cuadrado de la distancia euclidiana.	114
A.9. Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del algoritmo AG.	114
A.10. Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones de la ED.	114
A.11. Esquema del algoritmo híbrido	115
B.1. Imágenes de la esfera: (a) imagen 1, (b) imagen 2.	117
B.2. Gráficas del porcentaje de error relativo de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado para: 8 (a_1, a_2), 12 (b_1, b_2) y 16 (c_1, c_2) correspondencias de puntos.	120
B.3. Gráficas del porcentaje de error relativo de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado para: 20 (a_1, a_2) y 24 (b_1, b_2) correspondencias de puntos.	121
B.4. Objetos empleados en los experimentos: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.	122
B.5. Media esfera: (a_1) primera imagen y (a_2) segunda imagen.	123
B.6. Cubo: (a_1) primera imagen y (a_2) segunda imagen.	123

B.7. Figura regular: (a_1) primera imagen y (a_2) segunda imagen.	123
B.8. Barco: (a_1) primera imagen y (a_2) segunda imagen.	124
B.9. Parte de un anillo: (a_1) primera imagen y (a_2) segunda imagen.	124
B.10. Figura hecha de cubos: (a_1) primera imagen y (a_2) segunda imagen.	124
B.11. Objetos tridimensionales reconstruidos por la SPII: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.	125
B.12. Objetos reconstruidos por el método convencional: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.	126
B.13. Cuadrícula. (a) imagen 1 y (b) imagen 2.	126
B.14. Reconstrucción tridimensional métrica de la cuadrícula cuando: (a) no hay ruido, (b) un píxel, (c) dos píxeles y (d) tres píxeles de ruido gaussiano.	127
B.15. Imágenes reales de una casa de campo. Los puntos que se emplearon están marcados en ambas imágenes.	128
B.16. Imágenes reales de un juguete. Los puntos que se emplearon están marcados en ambas imágenes.	128
B.17. Tres vistas de la casa reconstruida empleando la SPII.	129
B.18. Reconstrucción del juguete empleando la SPII.	130
C.1. Media esfera.	131
C.2. Imágenes de la media esfera con los 74 puntos sin ruido sobre la primera configuración crítica. (a) imagen 1 y (b) imagen 2.	132
C.3. Imágenes de la media esfera con los 74 puntos sin ruido sobre la segunda configuración crítica. (a) imagen 1 y (b) imagen 2.	132
C.4. Gráficas del porcentaje de error de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado al emplear 24 correspondencias de puntos, con diferentes niveles de ruido en la pri- mera configuración crítica. (a_1, a_2) mejor ejecución, (b_1, b_2) ejecución más cercana a la media y (c_1, c_2) peor ejecución.	135
C.5. Gráficas del porcentaje de error de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado al emplear 24 correspondencias de puntos, con diferentes niveles de ruido en la se- gunda configuración crítica. (a_1, a_2) mejor ejecución, (b_1, b_2) ejecución más cercana a la media y (c_1, c_2) peor ejecución.	136
C.6. Imágenes sintéticas de la media esfera. (a_1, a_2) primera y segunda ima- gen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, res- pectivamente.	138
C.7. Imágenes sintéticas del cubo. (a_1, a_2) primera y segunda imagen, respec- tivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segun- da configuración crítica.	138

C.8. Imágenes sintéticas de la figura regular. (a_1, a_2) primera y segunda imagen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, respectivamente.	139
C.9. Imágenes sintéticas del barco. (a_1, a_2) primera y segunda imagen, respectivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segunda configuración crítica.	139
C.10. Imágenes sintéticas del anillo. (a_1, a_2) primera y segunda imagen, respectivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segunda configuración crítica.	140
C.11. Imágenes sintéticas del objeto formado por cubos. (a_1, a_2) primera y segunda imagen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, respectivamente.	140
C.12. Representación de la matriz fundamental como una cónica degenerada para: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos sobre la primera configuración crítica.	141
C.13. Representación de la matriz fundamental como una cónica degenerada para: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos sobre la segunda configuración crítica.	142
C.14. Objetos reconstruidos por la SPII sobre la primera configuración crítica: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.	143
C.15. Objetos reconstruidos por la SPII sobre la segunda configuración crítica: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.	144
C.16. Imágenes de la cuadrícula generada sobre la primera configuración crítica: (a) imagen 1 y (b) imagen 2.	145
C.17. Imágenes de la cuadrícula generada sobre la segunda configuración crítica: (a) imagen 1 y (b) imagen 2.	145
C.18. Reconstrucción tridimensional métrica de la cuadrícula cuando hay: (a) dos píxeles y (b) tres píxeles de ruido gaussiano sobre la primera configuración crítica.	147
C.19. Reconstrucción tridimensional métrica de la cuadrícula cuando hay: (a) dos píxeles y (b) tres píxeles de ruido gaussiano sobre la segunda configuración crítica.	147
C.20. Imágenes reales de una cuadrícula sobre la primera configuración crítica. Los puntos que se emplearon están marcados en ambas imágenes.	147
C.21. Reconstrucción tridimensional de la cuadrícula empleando la SPII.	148

Índice de algoritmos

1.	- Algoritmo genético simple	40
2.	- Estrategias evolutivas	42
3.	- Programación evolutiva	43
4.	- Optimización mediante cúmulos de partículas	44
5.	- Algoritmo de estimación de distribución	45
6.	- Evolución diferencial	46
7.	- Evolución diferencial <i>rand/1/bin</i>	48
8.	- NSGA-II	52
9.	- SPI, evaluación de un individuo	65
10.	- SPII, evaluación de un individuo	66
11.	- SPIII	67
12.	- Vistas extras	68
13.	- Evaluación del individuo para vistas extras	68

Capítulo 1

Introducción

La extracción de información a partir de imágenes es una tarea muy fácil de realizar para un ser humano, el cual desde muy temprana edad tiene la capacidad de ver y reconocer objetos en el ambiente que le rodea [1].

El sentido de la vista es una de las formas dominantes para percibir el mundo. Imitar su funcionamiento en una computadora, con base en la interpretación de las imágenes, es una tarea extremadamente difícil. Se podría explicar tal proceso a nivel biológico, ya que se sabe como la luz es transformada en la retina y estimula ciertas neuronas perceptivas. Sin embargo, aún no es claro cómo esta transformación causa que la mente reconozca los objetos. Contrario al nivel bajo que se maneja en el área biológica neuronal, el área psicológica intenta dar interpretaciones a un alto nivel de cómo funciona el sistema visual. Sin embargo, los modelos que se proponen en el área psicológica [2] no proporcionan instrucciones de cómo construir un sistema artificial comparable con el sentido visual que tiene el ser humano. En síntesis, se puede decir que vivimos con un sistema visual único, del cual aún no se conoce a detalle cómo funciona. Sin embargo, la comunidad de Visión por Computadora ha realizado avances significativos, a tal grado que se ha podido guiar robots mediante captores visuales (cámaras) [3].

Es claro que el ser humano cuenta con un sistema visual muy complejo. No obstante, las computadoras que operan como sistemas visuales artificiales tienen sus ventajas. Por ejemplo, pueden medir de forma precisa los objetos que se encuentran en un entorno en lugar de interpretarlo. La extracción de estas medidas a partir de una o varias imágenes es el objetivo principal del área de Fotogrametría. Tal objetivo ha sido perseguido desde que se inventó la fotografía. Aunque es posible utilizar sólo una imagen para extraer la información métrica de los objetos, usualmente, en el área de Fotogrametría se utilizan dos o más imágenes. Al utilizar múltiples imágenes es posible emplear técnicas de estereoscopia, con las cuales se puede recuperar información tridimensional real de la escena.

A principios del siglo XX se inventaron los primeros dispositivos para medición estereoscópica. Después se extendieron para actuar como graficadores analógicos. En 1957 se patentó un graficador analítico, el cual tenía la capacidad de establecer la geometría entre imágenes análogas. Cerca del año 1990, fue introducida la primera estación de trabajo fotogramétrica digital, la cual almacenaba imágenes digitales y realizaba el proceso de extracción de información métrica de forma automática mediante una computadora. En esa misma época se desarrollaron diversos algoritmos para extraer información de las imágenes a fin de generar reconstrucciones tridimensionales de forma automatizada.

La extracción de información métrica de los objetos a partir de imágenes y la generación de reconstrucciones tridimensionales de forma automática a partir de esta información, son procesos complementarios. Sin embargo, pertenecen a diferentes áreas de estudio. En los últimos años, tales procesos se han traslapado y ambas áreas comparten, en gran medida, la misma línea de investigación; no obstante, los objetivos de cada área son muy específicos. En el área de la Fotogrametría el objetivo es obtener la información métrica de los objetos que están en el entorno y, mediante esta información, generar modelos tridimensionales. En el área de Visión por Computadora, los objetivos están enfocados a generar algoritmos que realicen la reconstrucción de objetos en forma automática y sistemas que sean capaces de extraer información relevante de las imágenes [4, 5]. Esta última área ha tenido un crecimiento muy importante debido a la explotación que se le ha dado en diversas áreas, a la recreación por computadora de escenas reales y a la automatización de procesos de visión para interactuar con el ambiente.

La recreación por computadora tiene como finalidad generar ambientes virtuales tridimensionales a partir de imágenes de ambientes reales, para ello emplea métodos que pertenecen principalmente al área de Visión por Computadora, pero también se apoya en técnicas que pertenecen a otras áreas tales como: Procesamiento Digital de Imágenes y Graficación. De tal forma que todo el proceso de recreación por computadora, mejor conocido como reconstrucción tridimensional, puede dividirse en tres pasos: 1) procesamiento digital de imágenes, 2) el paso propio de la Visión por Computadora, y 3) la visualización de la reconstrucción. En la figura 1.1 se detallan los pasos del proceso de la reconstrucción tridimensional.

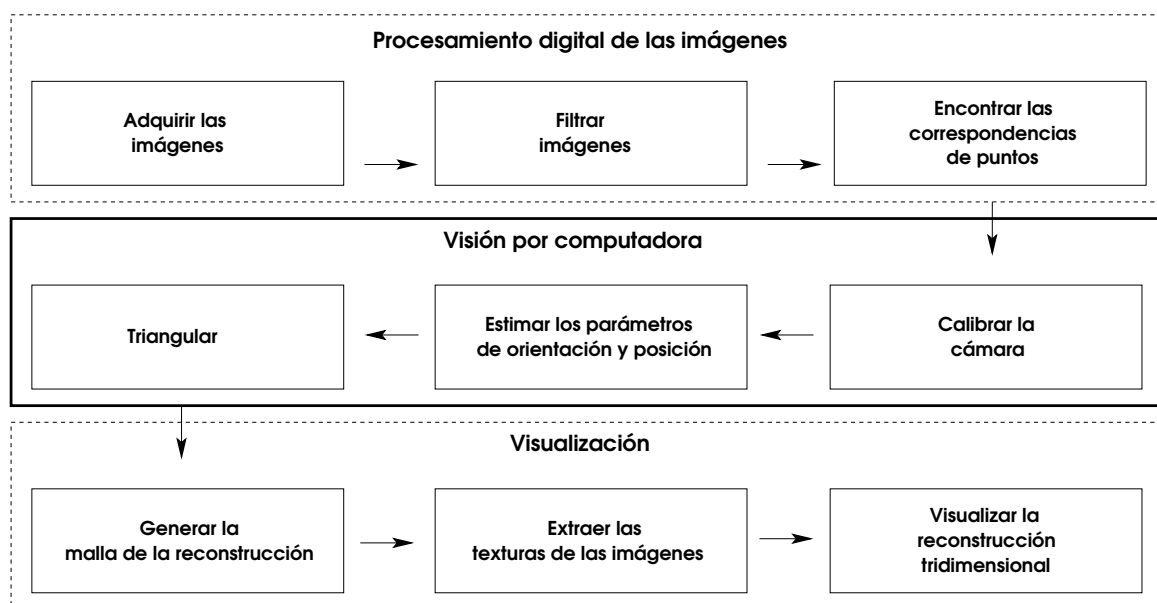


Figura 1.1: Proceso de la reconstrucción tridimensional.

El primer paso involucra la adquisición de las imágenes, el filtrado de ellas para mejorar sus condiciones, y la búsqueda de correspondencia de puntos en las imágenes. El segundo paso consiste en la *calibración de la cámara*, la *estimación de los parámetros de orientación y posición* para la imagen correspondiente, y la *triangula-*

ción (obtener los puntos tridimensionales). El último paso involucra la generación de la malla (generalmente triangular) que da la forma al objeto u objetos reconstruidos, la extracción de las texturas a partir de las imágenes, el mapeo de las texturas sobre los polígonos de la malla y la visualización del objeto.

En esta tesis se propusieron soluciones para el segundo paso a partir de dos o más imágenes mediante la aplicación de algoritmos evolutivos, especialmente, el algoritmo de evolución diferencial.

1.1. Motivación

La reconstrucción de escenas tridimensionales a partir de imágenes ha sido aplicada en diversas áreas. Una de sus principales áreas de aplicación es la industria del entretenimiento [6, 7, 8, 9] donde es utilizada para la generación de personas, animales y objetos. Por ejemplo, hay animaciones que requieren un gran número de animales de cierto tipo desplazándose de un lugar a otro. Este tipo de animaciones se logran mediante la dinámica de desplazamiento de un solo animal. Para obtener esta dinámica, es necesario tener un modelo inicial, el cual por simplicidad, es generado por medio de algún método de reconstrucción tridimensional [10].

La reconstrucción tridimensional también se ha aplicado en el área médica. Se ha usado para generar modelos de todo tipo de órganos, así como del cerebro y los dientes. Otras aplicaciones incluyen modelar cuerpos en movimiento, teleconferencias, navegación de robots, reconocimiento de objetos, simulación de fenómenos físicos, simulación de poblaciones [10], e incluso la reconstrucción de hallazgos arqueológicos [11].

Sin la ayuda de la reconstrucción tridimensional, los diseñadores que generan modelos tridimensionales, desperdiciarían muchas horas en aplicaciones CAD para modelar y dibujar objetos que pertenecen al mundo real. Además, el modelo generado podría distar mucho del objeto real.

Otras técnicas de reconstrucción se basan en marcas de luz que se adhieren en los objetos a reconstruir (comúnmente son utilizados sobre personas). Estas marcas son rastreadas por sensores visuales y, posteriormente, recrean el modelo tridimensional. Aunque pueden ser muy precisas, el costo es alto comparado con el costo de una cámara digital. Además, sólo obtienen la reconstrucción del objeto sin su textura, por lo cual, hay que aplicar otros algoritmos para extraer la textura y empatarla con el modelo reconstruido.

Una alternativa muy atractiva es la reconstrucción tridimensional a partir de imágenes debido al bajo costo y a los buenos resultados que se han obtenido con esta técnica [4, 5]. Dentro del área de Visión por Computadora, el problema de la reconstrucción tridimensional a partir de imágenes, se ha clasificado de acuerdo a [4, 5]: las características de los objetos que se van reconstruir, la información *a priori* de las escenas y de la cámara, y el número de imágenes que se emplean.

Cuando los objetos dentro de las imágenes presentan características con propiedades ortogonales o cuando existe información acerca de las cámaras que se usan para adquirir las imágenes, se cuenta con algoritmos muy robustos que generan reconstrucciones tridimensionales de gran calidad [12, 13, 14, 15, 16, 17, 18]. Sin embargo, tales algoritmos están sujetos, en primer lugar, a que existan objetos con propiedades ortogonales; y en segundo lugar, a que se tenga información como la posición, el des-

plazamiento, la orientación y los parámetros de la cámara que adquiere las imágenes.

Por tal motivo, las investigaciones se han enfocado a imágenes donde no se tengan definidas las características de los objetos o no se tenga información *a priori* de las cámaras (cámaras sin calibrar). De estas investigaciones [19] se ha concluido que, para obtener una reconstrucción tridimensional, se deben tener dos o más imágenes. Para el caso de más de dos imágenes existen algoritmos robustos [5] que generan reconstrucciones tridimensionales realistas. Sin embargo, los inconvenientes que existen con estos algoritmos son: encontrar la correspondencia de puntos en todas las imágenes, resolver los problemas de oclusión que se presenten y manejar un mayor número de puntos que se traduce en más operaciones. Para disminuir estos problemas es preferible tener sólo dos imágenes. Existen diversos algoritmos para obtener una reconstrucción tridimensional a partir de dos imágenes utilizando cámaras sin calibración previa [20, 19, 21, 22, 23]. Estos algoritmos hacen uso de la geometría epipolar y la matriz fundamental para calibrar la cámara, estimar los parámetros de orientación y posición, y obtener la reconstrucción tridimensional. La calibración de la cámara es la parte más importante, ya que permite generar una reconstrucción tridimensional dentro de un espacio euclidiano.

Se ha demostrado que, al aplicar la geometría epipolar, existen configuraciones de la cámara donde no es posible realizar la calibración de la misma [24, 25]. A estas configuraciones se les conoce como degeneraciones o configuraciones críticas. Las investigaciones más recientes [23, 24, 25, 26] han podido obtener reconstrucciones tridimensionales cerca de las configuraciones críticas, pero no en ellas. Asimismo, existen superficies conocidas como críticas donde los algoritmos que emplean la geometría epipolar, tampoco pueden obtener reconstrucciones tridimensionales.

Debido a esto se propusieron soluciones capaces de calibrar la cámara, de estimar los parámetros de orientación y posición, y de obtener la reconstrucción tridimensional en cualquier tipo de superficie (incluso en la superficie crítica) y en cualquier configuración de la cámara (incluso en la traslación pura, la cual es una configuración crítica). Las soluciones propuestas no emplean la geometría epipolar para relacionar las imágenes; en lugar de esto, emplean directamente el modelo de la cámara oscura, una heurística evolutiva (la evolución diferencial [27]) y un método de triangulación inscrito en esta heurística para calcular los puntos estimados y minimizar la distancia euclidiana entre estos puntos y los puntos de la imagen.

1.2. Objetivo general y objetivos particulares

El objetivo de esta tesis es generar soluciones a problemas de Visión por Computadora mediante algoritmos evolutivos, específicamente, a los problemas que presenta el proceso de la reconstrucción tridimensional métrica. Para lograr el objetivo principal, se han establecido los siguientes objetivos particulares:

1. Generar una solución para resolver un problema del área de Visión por Computadora conocido como ajuste de la mejor elipse a un conjunto de puntos mediante algoritmos evolutivos.
2. Generar una solución que resuelva el problema de la triangulación.

3. Generar una solución capaz de reconstruir objetos tridimensionales a partir de correspondencias de puntos de dos imágenes. La reconstrucción debe lograrse en cualquier configuración de la cámara y superficie, incluso dentro de las configuraciones y superficies críticas.
4. Generar una solución capaz de reconstruir objetos tridimensionales a partir de correspondencias de puntos de tres o más imágenes.

1.3. Contribuciones

Las principales contribuciones de esta tesis son las siguientes:

1. Se propuso una solución efectiva que ataca al problema de encontrar la mejor elipse que se ajuste a un conjunto de puntos dados, sin restricciones y con restricciones, mediante un algoritmo genético y la evolución diferencial. La solución propuesta está basada en [28], pero tiene la flexibilidad de minimizar la distancia euclidiana o su cuadrado. Esta solución supera los resultados de los mejores algoritmos reportados en la literatura.
2. Se resolvió, efectivamente, el problema de triangulación. Este problema fue resuelto como un problema de optimización multi-objetivo mediante el algoritmo NSGA-II [29]. Esta solución presenta mejores resultados cuando el nivel de ruido es alto, bajo reconstrucciones proyectivas y afines, en comparación con los mejores algoritmos reportados en la literatura.
3. Se propusieron dos soluciones (SPI y SPII) que permiten calibrar correcta y eficientemente la cámara, estimar los parámetros de orientación y posición, y calcular los puntos tridimensionales *a partir de dos imágenes* adquiridas por una cámara sin calibración previa mediante el uso de la evolución diferencial. Esta contribución es la más importante del trabajo de tesis presentado, debido a que esta solución obtiene reconstrucciones tridimensionales métricas aún cuando sólo exista traslación pura¹ al adquirir las imágenes o cuando se presenta una superficie crítica (puntos que yacen en un mismo plano).
4. Se propuso la SPIII que permite también calibrar la cámara, estimar los parámetros de orientación y posición, y calcular los puntos tridimensionales en el proceso de la reconstrucción tridimensional, pero esta vez *a partir de tres o más imágenes* adquiridas por una cámara sin calibración previa, mediante el uso de la evolución diferencial.

1.4. Estructura general del documento

En esta tesis se presentan soluciones aplicadas a problemas del área de Visión por Computadora, las cuales utilizan algoritmos evolutivos para su resolución. Los principales problemas que aquí se presentan, parten del proceso de la reconstrucción

¹Configuración crítica de la cámara donde los algoritmos que emplean la geometría epipolar, no han podido obtener una reconstrucción tridimensional métrica.

tridimensional. De acuerdo a la literatura, estos problemas han sido atacados con éxito por diversas técnicas y algoritmos muy robustos como se describe en el capítulo 2.

En el capítulo 3 se presentan los fundamentos básicos del área de Visión por Computadora y Optimización. De la primera área se exhibe: la geometría proyectiva (necesaria para comprender el proceso de reconstrucción), la representación de entidades geométricas (puntos y líneas), las transformaciones básicas entre las entidades geométricas, los parámetros de orientación y posición, la calibración de la cámara y el proceso de la reconstrucción tridimensional desde el punto de vista del área de Visión por Computadora. De la segunda área se describe la optimización y algunos métodos empleados por los algoritmos más robustos dentro del área de Visión por Computadora (ventajas y desventajas). Asimismo, se presentan los algoritmos evolutivos para optimización de parámetros reales y sus bondades. Al final de este capítulo se muestran los algoritmos para la optimización multi-objetivo.

En el capítulo 4 se presentan los problemas de la calibración de la cámara, la estimación de los parámetros de orientación y posición, y la triangulación, como problemas de optimización y las soluciones que se propusieron para cada uno de los casos.

El capítulo 5 muestra parte de los experimentos realizados de cada uno de los problemas al emplear las soluciones propuestas. Estos experimentos muestran la robustez de las soluciones y la capacidad que tienen para obtener reconstrucciones tridimensionales en las superficies y configuraciones críticas, para el problema de la calibración de la cámara y la estimación de los parámetros de orientación y posición.

En los capítulos 6 y 7 se exhiben las discusiones de los resultados obtenidos, las conclusiones de la tesis, el trabajo futuro que se desprende de las soluciones que se propusieron y las posibles mejoras de éstas.

En el apéndice A, se muestra otro problema del área de Visión por Computadora llamado ajuste de la mejor elipse a un conjunto de puntos dado, el cual se planteó como un problema de optimización. Asimismo, se presenta la solución propuesta con algoritmos evolutivos y los resultados obtenidos.

En los apéndices B y C, se exhibe el resto de los experimentos resueltos por la SPII en ambientes simulados y reales, en configuraciones críticas y no críticas, y en cualquier tipo de superficies.

Finalmente, en el apéndice D se presenta un análisis comparativo de los algoritmos evolutivos para optimización de parámetros reales, así como un par de experimentos que muestran el desempeño de la heurística empleada en esta tesis.

Capítulo 2

Estado del arte

Los pasos de la reconstrucción tridimensional dentro del área de Visión por Computadora, se resumen a encontrar la estructura de los objetos tridimensionales a partir de entidades geométricas, tales como puntos o líneas presentes en las imágenes.

Las contribuciones de cientos de investigadores han generado una amplia gama de algoritmos que permiten realizar la reconstrucción tridimensional de entidades geométricas en las imágenes. En este capítulo se presentan algunas contribuciones, las cuales se clasifican por el número de imágenes empleadas, la información adicional en las imágenes y la información adicional de la cámara.

2.1. Reconstrucción 3D por medio de imágenes

Para obtener una reconstrucción tridimensional a partir de imágenes, es necesario conocer la relación entre las entidades geométricas del mundo y las entidades geométricas de las imágenes. Esta relación geométrica está dada por los parámetros de la cámara con base en el modelo de la cámara oscura.

Actualmente, existen diversos algoritmos para calcular los parámetros de la cámara. Hartley y Zisserman [5] dividen estos algoritmos de acuerdo al número de imágenes que utilizan, es decir, algoritmos para cuatro o más imágenes, para tres imágenes, y para dos imágenes.

2.1.1. Reconstrucción 3D partir de cuatro o más imágenes

Cuando se tienen cuatro imágenes, uno de los mejores métodos de reconstrucción tridimensional es el *tensor cuadrifocal*, propuesto por Triggs [30]; se basa en la geometría epipolar extendida a cuatro imágenes. Sin embargo, hay ciertas posiciones de la cámara que se exhiben en [31, 32] donde no es posible obtener una reconstrucción tridimensional con este método.

Otro método que ha demostrado tener buen desempeño para la reconstrucción tridimensional a partir de cuatro o más imágenes es la *potencia de factorización* propuesto por Huynh [33]. Con este método iterativo es posible estimar los parámetros de la cámara por cada imagen, alternando con la triangulación en cada iteración para recuperar los puntos tridimensionales. Este método está basado en el *ajuste de bloques* que se generan a partir de los parámetros de la cámara en relación con los puntos del mundo y los puntos de la imagen. El ajuste de bloques, fue propuesto en principio por Tomasi-Kanade [34] para múltiples imágenes (cuatro o más).

Knight y Reid en [35] presentan un método que emplea un robot con dos cámaras que están en una *configuración estereoscópica*. Este método necesita 5 o más pares de imágenes para obtener una reconstrucción tridimensional dentro de una superficie crítica (puntos que yacen sobre un mismo plano). Al tomar las imágenes con las dos cámaras, se producen dos vistas de los puntos, las cuales se incorporan en dos planos distintos para lograr la reconstrucción.

Otro método, propuesto por Malis y Cipolla en [36], emplea estructuras planas dentro de las imágenes para *estimar las alineaciones* entre ellas y obtener los parámetros de la cámara. Este método necesita al menos seis imágenes de la escena para obtener resultados aceptables [36].

2.1.2. Reconstrucción 3D a partir de tres imágenes

Cuando se tienen tres imágenes para realizar la reconstrucción tridimensional, un método muy efectivo es aquel propuesto por Hartley en [37], el cual utiliza el *tensor trifocal* (tensor que tiene como base la *geometría epipolar extendida a tres imágenes*). De esta técnica se derivan diversos métodos para calcular el tensor trifocal [38, 39, 40] cada uno con ciertas ventajas que dependen de los objetos que se desean reconstruir y las posiciones de las cámaras.

Existe otro método propuesto por Hartley en [21], donde se plantea realizar una *reconstrucción métrica directa*. Este método emplea una función que relaciona los puntos en el mundo con los puntos en las imágenes con base en los parámetros de la cámara y, con la ayuda de un método de gradiente iterativo, se estiman los valores de cada parámetro. Hartley señala que al menos necesita tres imágenes para obtener una reconstrucción métrica. También expone que si parte de dos imágenes, es necesaria una estimación inicial de los parámetros de la cámara muy cerca de la solución para que el método obtenga resultados aceptables.

El método propuesto por Ma y Liu en [41] emplea tres imágenes para realizar la reconstrucción. Este método utiliza *formulaciones derivadas de las ecuaciones de Kruppa* para su inicialización. Posteriormente, el método emplea *homografías* sobre los puntos de las imágenes para recuperar los parámetros de la cámara y obtener la reconstrucción tridimensional.

Se debe tener en cuenta que, cuando se tienen tres o más imágenes, es más difícil encontrar correspondencias de puntos debido a que los puntos pueden estar o no presentes; esto complica el paso de hallar las correspondencias. Además, se debe contar con un mecanismo para eliminar puntos que no sean visibles en todas las imágenes y emplear métodos iterativos para refinar los resultados. Pese a estos inconvenientes, cuando se cumplen ciertas restricciones, como la posición de la cámara, las reconstrucciones tridimensionales son muy realistas.

2.1.3. Reconstrucción 3D a partir de dos imágenes

El problema de encontrar correspondencias de puntos que pertenezcan al mismo punto tridimensional tiende a reducirse cuando se emplean dos imágenes para la reconstrucción. Sin embargo, al emplear dos imágenes, es más probable que la precisión de los puntos tridimensionales reconstruidos se vea disminuida.

Para realizar la reconstrucción tridimensional a partir de dos imágenes existen

diversos algoritmos; Soatto y Ma [4] los clasifican en cuatro grupos: *estratificación*, *conocimiento del movimiento de la cámara*, *conocimiento de la escena* y *a partir de correspondencias de puntos en las imágenes*.

Reconstrucción 3D por estratificación

En el primer grupo se encuentran los algoritmos que realizan la reconstrucción tridimensional de forma *estratificada*, es decir, parten de los puntos de la imagen para obtener una reconstrucción en un espacio proyectivo, posteriormente, encuentran una transformación que permite generar una reconstrucción en un *espacio afín* y, finalmente, estiman otra transformación que permite obtener una reconstrucción en el *espacio euclidiano* [4, 5] bajo una cierta escala. El primer método de estratificación fue formulado por Koenderink y van Doorn [42] en 1991, comenzando desde una reconstrucción afín. Basados en esta idea se derivaron algoritmos más robustos [43, 44, 31], los cuales, parten de una reconstrucción en el *espacio proyectivo* y al final obtienen una reconstrucción en el espacio euclidiano. Sin embargo, no siempre es posible obtener una reconstrucción en el espacio euclidiano mediante esta técnica, ya que existen ambigüedades intrínsecas asociadas al movimiento y posición de la cámara, las cuales se muestran en [45, 46, 47]. Además, para encontrar cada una de las matrices de transformación es necesario tener cierta información, por ejemplo, conocer qué objetos se están reconstruyendo, cómo se desplaza la cámara entre las imágenes, o tener calibrada la cámara.

Reconstrucción 3D en base al conocimiento del movimiento de la cámara

El segundo grupo se compone de algoritmos que extraen los parámetros de orientación y posición de la cámara a partir del conocimiento previo del *movimiento de la cámara* al tomar cada una de las imágenes. Uno de los algoritmos importantes en este grupo es aquel desarrollado por Beardsley [48] donde, a partir del desplazamiento la cámara, se obtienen los parámetros de la cámara. Otros algoritmos se basan en ciertos movimientos muy específicos de la cámara, como por ejemplo, *rotación de la cámara en un solo eje cartesiano*, *rotación en los tres ejes cartesianos*, *movimiento sobre un plano* y *movimiento focal* [49, 50, 51, 52]. El principal inconveniente de este grupo de algoritmos es la necesidad de conocer cómo se ha desplazado la cámara para poder aplicar un algoritmo en particular. Por ejemplo, si sólo se realizó una rotación de la cámara entre las dos imágenes, se debe usar un algoritmo específico que permita extraer los parámetros de la cámara y los puntos tridimensionales a partir de la rotación.

Reconstrucción 3D en base al conocimiento de la escena

En el tercer grupo, se encuentran los algoritmos que utilizan el conocimiento de la escena para obtener la reconstrucción tridimensional. Por ejemplo en [53, 54, 13, 17, 18], se exploran las imágenes para encontrar objetos que presenten *ortogonalidad y paralelismo* y, a partir de ellos, se calculan los puntos al infinito los cuales permiten estimar los parámetros de la cámara.

Por otro lado, Zhang en [16] emplea *planos*, de los cuales conoce su posición y, a partir de ellos, obtiene una estimación inicial de los parámetros de la cámara. Con

esta estimación emplea un método de gradiente para refinar los parámetros con base en la función que presenta Hartley [21] en la reconstrucción métrica directa.

En [55] Yang y Cao estiman los parámetros de la cámara mediante un algoritmo lineal que emplea seis puntos. Este algoritmo utiliza un *modelo tridimensional del objeto* (conocido *a priori*), el cual se desea reconstruir. El algoritmo puede obtener los parámetros de la cámara a partir de una imagen.

El inconveniente de estos métodos es que las imágenes deben contener objetos que presenten características ortogonales, como los planos, de tal forma que no pueden ser aplicados a cualquier objeto.

Reconstrucción 3D a partir de correspondencias de puntos

El último grupo no presenta los inconvenientes de los tres anteriores, ya que parte solamente del conocimiento de las correspondencias de puntos para obtener los parámetros de la cámara, con base en las *ecuaciones de Kruppa* y sus derivaciones. Maybank y Faugeras en 1992 [56] desarrollaron un algoritmo que emplea las ecuaciones de Kruppa para realizar la reconstrucción tridimensional. A partir de este algoritmo y con base en estas ecuaciones, diversos métodos fueron desarrollados [57, 58, 59, 60, 61, 24], los cuales se emplearon para generar algoritmos más robustos y precisos que se aplican para encontrar correspondencias de puntos [62, 38, 63]. Sin embargo, la estabilidad numérica y las *configuraciones críticas de la cámara* limitan el uso de las ecuaciones de Kruppa y sus derivaciones [64, 50]. De todas las configuraciones críticas que se describen en [64, 50], éstas se han reducido a sólo dos, en donde se ha demostrado que no es posible extraer de manera correcta los parámetros de la cámara [22, 65, 66, 25, 5, 26, 23, 67].

En [68] Ueshiba y Tomita proponen un método que obtiene los parámetros de la cámara mediante *formulaciones derivadas* de las ecuaciones Kruppa y las emplean dependiendo del desplazamiento que realicen. Los autores muestran que su método no obtiene soluciones dentro de las configuraciones críticas.

Ha y Kang en [69] emplean las formulaciones derivadas de las ecuaciones de Kruppa y los puntos tridimensionales en un espacio proyectivo, los cuales se estiman de las matrices de proyección (matrices de proyección calculadas a partir de la matriz fundamental), para inicializar un *método de minimización lineal*. Esta inicialización permite obtener soluciones más estables aún con ruido en las imágenes.

En [26] Sturm y Cheng presentan un método para obtener una reconstrucción tridimensional a partir de dos imágenes. Ellos analizan los casos cuando se cae en una configuración crítica y cuando se está cerca de ella. En este último caso es cuando pueden obtener una reconstrucción tridimensional métrica mediante las formulaciones derivadas de las ecuaciones de Kruppa.

Kanatani y Nakatsuji desarrollaron un método [23], con base en el trabajo de Hartley y Silpa-Anan [66], el cual es capaz de generar una reconstrucción tridimensional cerca de una de las configuraciones críticas. El algoritmo da resultados aceptables cuando está cerca de esta configuración, sin embargo, tanto el algoritmo de Kanatani y Nakatsuji y el algoritmo de Sturm y Cheng no pueden obtener una reconstrucción tridimensional en las configuraciones críticas.

En general, el problema de este cuarto grupo son las configuraciones críticas de la cámara, donde no es posible estimar los parámetros de la cámara a partir de las

ecuaciones de Kruppa o sus formulaciones derivadas. Por ello no es posible realizar la reconstrucción tridimensional métrica.

2.2. Triangulación

Otro punto en el proceso de reconstrucción tridimensional es la *triangulación*, el cual tiene por objetivo recuperar los puntos tridimensionales a partir de las matrices de proyección y la correspondencia de puntos.

Los mejores métodos, así como los más representativos que realizan la triangulación a partir de correspondencia de puntos de dos imágenes, se describen a continuación. Cabe señalar que la mayoría de estos métodos se han extendido a correspondencias de puntos de tres o más imágenes.

El método más simple para realizar la triangulación obtiene el punto medio sobre la perpendicular común entre ambas rectas proyectadas a partir de los puntos bidimensionales. A este método se le conoce como *método del punto medio* y fue desarrollado por Beardsley [70, 48]. Este método es el más sencillo de implementar y debido a su tiempo de ejecución puede aplicarse en sistemas de tiempo real. Sin embargo, sólo opera bajo reconstrucciones métricas, ya que al usarlo en reconstrucciones del tipo afín o proyectiva, la triangulación que obtiene no es correcta.

El segundo método [71], encuentra una aproximación del punto tridimensional mediante un sistema de ecuaciones lineales [5]. Para resolver este sistema hay dos alternativas, en la primera se manipulan las ecuaciones para obtener un sistema de ecuaciones lineales homogéneas. Este sistema se resuelve por medio del método de mínimos cuadrados, el cual puede ser resuelto mediante la descomposición de valores singulares (DVS). El problema de este método es que su costo computacional es alto. La segunda alternativa es acomodar las ecuaciones de tal forma que se genere un sistema ecuaciones lineales no homogéneo. Tal sistema puede resolverse con distintos algoritmos, de entre ellos, uno de los más estables es el algoritmo de las ecuaciones normales [5], el cual es sencillo de implementar y su tiempo de ejecución es muy similar al primer método. Además, se puede aplicar tanto a la reconstrucción afín como a la reconstrucción métrica, pero no es invariante en la reconstrucción proyectiva.

El último método, conocido como el óptimo bajo ciertas condiciones, fue propuesto por Hartley [71] y se le conoce como método polinomial-absoluto. Una de las propiedades que lo hace robusto es que tiene la capacidad para aplicarse en los tres tipos de reconstrucciones que existen (métrica, afín y proyectiva). Asimismo, si las matrices de proyección con las que opera son calculadas con gran precisión, este método es el óptimo para encontrar el punto tridimensional. Sin embargo, si los puntos en las imágenes presentan ruido y éstos son empleados para estimar las matrices de proyección, los resultados que arroja este algoritmo son deficientes, como se muestra en el capítulo de resultados.

Capítulo 3

Marco teórico

Si se tiene un conjunto de imágenes, con diferentes vistas de un mismo entorno, es muy probable que exista una relación entre ellas. Esta relación está dada por una geometría implícita en el conjunto de imágenes. Gracias a esta geometría es posible obtener la reconstrucción tridimensional del entorno captado por el conjunto de imágenes.

Este capítulo está destinado a definir los conceptos básicos del área de Visión por Computadora. En principio se habla de la geometría proyectiva enfocada a la reconstrucción. Asimismo, se expone la descripción matemática de puntos y líneas en dos y tres dimensiones, a las cuales se les conoce como entidades geométricas básicas. Estas entidades son representadas como subespacios proyectivos, las cuales, a su vez, se pueden expresar como vectores homogéneos. Estos vectores pueden ser subdivididos en otros vectores homogéneos con una parte en el espacio euclidiano, como se propone en [72], donde se describe esta distinción sin el contexto explícito de la geometría proyectiva. Adicional a la descripción de las entidades geométricas, se presentan las transformaciones geométricas también conocidas como homografías y las transformaciones bajo el modelo de la cámara oscura.

Para hacer autocontenida esta tesis, es imprescindible presentar ahora, los conceptos del área de optimización. Con su ayuda es posible plantear el proceso de la reconstrucción tridimensional como un problema de optimización y, posteriormente, emplear algún método para resolverlo.

3.1. Fundamentos de Visión por Computadora

3.1.1. Geometría proyectiva

Cuando se trata con imágenes del mundo real, comúnmente, se observan elementos al infinito, por ejemplo, el horizonte, el cual es una imagen de los puntos en el espacio lejanos más allá de cierto umbral. Desafortunadamente, la geometría euclidiana no es suficiente para representar elementos al infinito, por lo que es muy difícil tratar con estos elementos dentro del marco euclidiano. Por tal motivo, es necesario emplear una geometría más general donde sea posible trabajar con puntos al infinito. Esta geometría es conocida como *Geometría Proyectiva* [3].

Espacio proyectivo y coordenadas homogéneas

El espacio proyectivo \mathbb{P}^n consta de los subespacios vectoriales unidimensionales del espacio vectorial \mathbb{R}^{n+1} . Cada tal subespacio representa un punto proyectivo $\mathbf{x} \in \mathbb{P}^n$.

Si se supone $n = 1$, entonces el espacio correspondiente es la *línea proyectiva* \mathbb{P}^1 y cada $\mathbf{x} \in \mathbb{P}^1$ se identifica con una línea que pasa a través del origen de \mathbb{R}^2 , es decir, es un subespacio vectorial de \mathbb{R}^2 [3].

Un punto proyectivo $\mathbf{x} \in \mathbb{P}^n$ es un elemento del cociente del espacio $\mathbb{R}^{n+1} \setminus \mathbf{0}$ bajo la relación de equivalencia \cong , tal que:

$$\mathbf{x} \cong \mathbf{y} \iff \exists \lambda \neq 0 : \mathbf{x} = \lambda \mathbf{y}. \quad (3.1)$$

En otras palabras, todos los vectores de la forma $\lambda \mathbf{x}^h = [\lambda x_1, \dots, \lambda x_n, \lambda]^T \in \mathbb{R}^{n+1}$, con $\lambda \neq 0$, son equivalentes respecto a la relación \cong y definen un mismo punto $\mathbf{x} \in \mathbb{P}^n$. La relación hace que la longitud de los vectores en \mathbb{R}^{n+1} sea irrelevante, sólo importan sus direcciones. Es posible identificar cada $\mathbf{x} \in \mathbb{P}^n$ con su representación como vector de coordenadas $\mathbf{x}^h \in \mathbb{R}^{n+1}$, o sea $\mathbf{x} = \mathbf{x}^h$, también llamado *vector homogéneo* [3]. Optaremos pues en no hacer distinción entre los puntos y sus vectores coordenados [5]. Con esta notación los puntos al infinito corresponden a los vectores homogéneos cuya última componente es cero.

El espacio euclidiano \mathbb{R}^n se incluye como un subespacio del espacio proyectivo \mathbb{P}^n mediante la transformación $\mathbb{R}^n \mapsto \mathbb{P}^n$

$$\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n \mapsto \mathbf{x}_e = [x_1, \dots, x_n; 1]^T \in \mathbb{P}^n, \quad (3.2)$$

El índice e en el vector homogéneo \mathbf{x}_e denota que la información euclidiana puede ser extraída directamente del vector homogéneo.

En el sentido opuesto cada punto $\mathbf{x} \in \mathbb{P}^n$ (salvo si es uno al infinito), determina un punto euclidiano, es decir, un $\mathbf{x}_e \in \mathbb{R}^n$, mediante

$$\mathbf{x} = [x_1, \dots, x_n; x_{n+1}]^T \in \mathbb{P}^n \mapsto \mathbf{x}_e = \frac{1}{x_{n+1}} [x_1, \dots, x_n]^T \in \mathbb{R}^n. \quad (3.3)$$

Representación de entidades geométricas

Representación de líneas en \mathbb{R}^2 en coordenadas homogéneas. Cada línea en el plano se representa por una ecuación $ax + by + c = 0$, con $a, b, c \in \mathbb{R}$. Así, una línea se representa como un 3-vector,

$$[a, b, c]^T. \quad (3.4)$$

La correspondencia entre líneas y 3-vectores no es inyectiva, ya que las líneas $ax + by + c = 0$ y $(ka)x + (kb)y + (kc) = 0$ con $k \neq 0$, coinciden. De hecho, dos 3-vectores relacionados por un factor global son considerados como equivalentes.

Representación de puntos en \mathbb{R}^2 en coordenadas homogéneas. Un punto $\mathbf{x} = (x, y)$ yace en una línea $\mathbf{l} = [a, b, c]^T$ si y sólo si $ax + by + c = 0$. Esto puede ser escrito en términos del producto “punto” de vectores; al expresar a $(x, y) \in \mathbb{R}^2$ en coordenadas homogéneas,

$$[x, y, 1]^T, \quad (3.5)$$

entonces la relación

$$0 = \mathbf{x}^T \cdot \mathbf{l} = [x, y, 1]^T \cdot [a, b, c] \quad (3.6)$$

es equivalente a $ax + by + c = 0$. De esta forma, al igual que con las líneas, los puntos son representados por vectores en coordenadas homogéneas. Cualquier vector representativo de un punto en coordenadas homogéneas, es de la forma

$$[x_1, x_2, x_3]^T \quad \text{con } x_1, x_2, x_3 \in \mathbb{R}, x_3 \neq 0, \quad (3.7)$$

y representa al punto

$$(x_1/x_3, x_2/x_3) = (x, y) \in \mathbb{R}^2. \quad (3.8)$$

Puntos y líneas al infinito

De la ecuación (3.6) se ve que un punto $[x_1, x_2, x_3]^T \in \mathbb{R}^2$ está en una línea $[a, b, c]^T$ si las tripletas que los representan son ortogonales en \mathbb{R}^3 . Así dadas dos líneas $\mathbf{l} = [a, b, c]^T$ y $\mathbf{l}' = [a', b', c']^T$ su punto de intersección “debe ser ortogonal” a ambas, es decir, es

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' = \begin{vmatrix} i & j & k \\ a & b & c \\ a' & b' & c' \end{vmatrix} = (bc' - b'c)i - (ac' - a'c)j + (ab' - a'b)k.$$

En particular, si \mathbf{l} y \mathbf{l}' son paralelas se tendrá $[a', b', c']^T = [a, b, c']^T$, es decir, \mathbf{l} y \mathbf{l}' sólo difieren en la tercera coordenada. El punto de intersección es pues

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' = \begin{vmatrix} i & j & k \\ a & b & c \\ a & b & c' \end{vmatrix} = (c' - c)[b, -a, 0]^T,$$

lo cual representa al punto $(b/0, -a/0)$, correspondiente a un punto al infinito, esta observación concuerda con la idea de que líneas paralelas se intersecan en el infinito.

En general, puntos en coordenadas homogéneas $[x, y, 0]^T$ no corresponden a puntos en \mathbb{R}^2 . En cambio, como hemos visto, los vectores homogéneos $\mathbf{x} = [x_1, x_2, x_3]^T$ con $x_3 \neq 0$ corresponden a puntos en \mathbb{R}^2 . Los vectores homogéneos $\mathbf{x} = [x_1, x_2, 0]^T$ representan a *puntos ideales* o *puntos al infinito*.

Claramente, las tripletas $[x_1, x_2, 0]^T$ son ortogonales a $[0, 0, c]^T$ por lo que esta última representa a la *línea al infinito*, la cual consiste de todos los puntos al infinito.

3.1.2. Transformaciones

Una *transformación lineal* $\varphi : \mathbb{P}^n \rightarrow \mathbb{P}^m$ queda representada por una matriz H de orden $(m + 1) \times (n + 1)$, tal que $\forall \mathbf{x} \in \mathbb{P}^n$ se tiene:

$$\mathbf{x}' = \varphi(\mathbf{x}) \iff \mathbf{x}' = H\mathbf{x}. \quad (3.9)$$

Si $n = m$ y la matriz H es no singular, se dice que H es una *homografía* $\mathbb{P}^n \rightarrow \mathbb{P}^n$. Las homografías o transformaciones proyectivas forman un grupo llamado *proyectivo lineal*, el cual se divide en cuatro clases:

- **Clase I: Isometrías.** Estas transformaciones preservan la distancia euclidiana. Las *transformaciones euclidianas* son las isometrías más importantes en la práctica y se representan por matrices

$$H = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (3.10)$$

donde $R \in \mathbb{R}^{n \times n}$ es una matriz *unitaria*, i.e., su inversa es su transpuesta, $R^{-1} = R^T$ y $\mathbf{t} \in \mathbb{R}^n$ es un vector de *traslación*.

- **Clase II: Transformaciones de similitud.** Son composiciones de isometrías con escalamientos isotrópicos (aplican un mismo cambio de escala en todas las dimensiones). Esta clase tiene la propiedad de preservar la métrica, salvo un factor de escala. Las transformaciones euclidianas compuestas con escalamientos son similitudes y se representan por matrices:

$$H = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & s \end{bmatrix}, \quad (3.11)$$

donde R y \mathbf{t} son como antes y s es un factor de escala.

- **Clase III: Transformaciones afines.** Estas transformaciones se representan por matrices de la forma (3.10), en donde $R \in \mathbb{R}^{n \times n}$ es no singular, i.e., $\det R \neq 0$. Esta clase tiene la propiedad de preservar el paralelismo.
- **Clase IV: Transformaciones proyectivas.** Estas son transformaciones lineales generales no singulares en términos de coordenadas homogéneas. Generalizan a las transformaciones afines. La forma matricial es:

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}, \quad (3.12)$$

donde A es una matriz no singular de orden $(n \times n)$, $\mathbf{v}, \mathbf{t} \in \mathbb{R}^n$ son vectores arbitrarios y v es un elemento dependiente de los demás, de tal forma que $\det A \neq 0$, en otras palabras la transformación ha de quedar especificada por $(n + 1)^2 - 1$ parámetros.

En la figura 3.1 se muestran las transformaciones aplicadas a un cuadrado y en la figura 3.2 se muestran las transformaciones aplicadas a un cubo.

3.1.3. Distancia euclidiana y su cuadrado

Para dos puntos $\mathbf{x}_a, \mathbf{x}_b \in \mathbb{R}^n$ su *distancia euclidiana* es:

$$d(\mathbf{x}_a, \mathbf{x}_b) = \sqrt{(x_{a,1} - x_{b,1})^2 + (x_{a,2} - x_{b,2})^2 + \dots + (x_{a,n} - x_{b,n})^2}.$$

La raíz cuadrada real $x \mapsto \sqrt{x}$ tiene una singularidad en 0 [73], por lo cual no es analítica. Para evitar este problema se considera el *cuadrado de la distancia euclidiana*

$$(\mathbf{x}_a, \mathbf{x}_b) \mapsto d(\mathbf{x}_a, \mathbf{x}_b)^2 = (x_{a,1} - x_{b,1})^2 + (x_{a,2} - x_{b,2})^2 + \dots + (x_{a,n} - x_{b,n})^2,$$

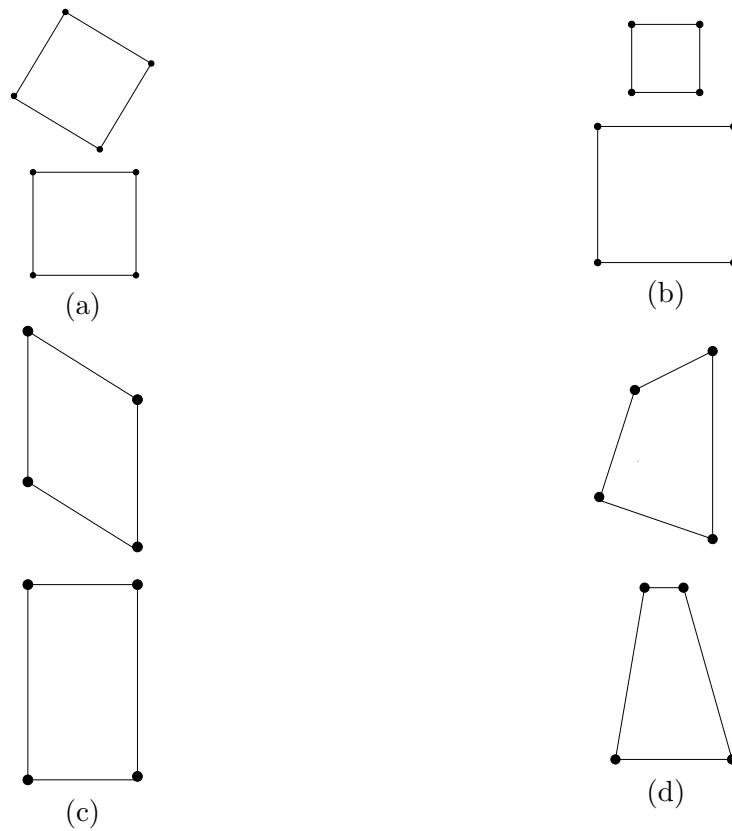


Figura 3.1: Forma de un cuadrado tras aplicar las cuatro clases de transformaciones. (a) isometría, (b) transformación de similitud, (c) transformación afín y (d) transformación proyectiva.

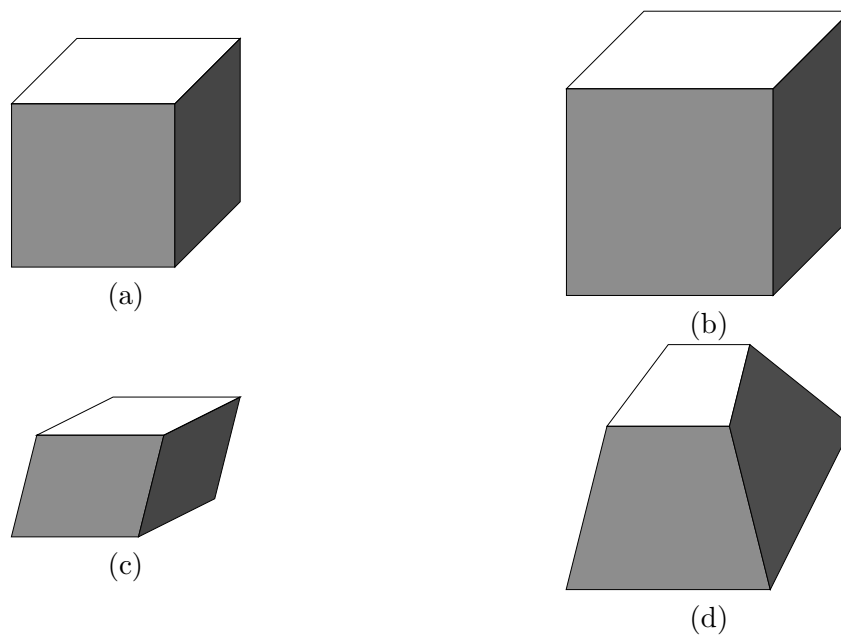


Figura 3.2: Forma de un cubo tras aplicar las cuatro clases de transformaciones. (a) isometría, (b) transformación de similitud, (c) transformación afín y (d) transformación proyectiva.

la cual sí es analítica. Con fines de optimización puede ser tratada con diversos métodos numéricos. Observamos que al ser la función “cuadrado” $y \mapsto y^2$ creciente en \mathbb{R}^+ , cualquier mínimo considerando el cuadrado de la distancia, lo será también al emplear la distancia sola, con la diferencia de que puntos “alejados” incrementarán su lejanía en el cuadrado de la distancia.

En el apéndice A, se exponen las diferencias que se presentan al emplear la distancia euclidiana y su cuadrado cuando se aplica a un problema geométrico.

3.1.4. Modelo de la cámara oscura

Al modelo de la cámara que nos permite transformar un punto tridimensional en un punto bidimensional sobre un plano llamado *imagen*, se le conoce como el *modelo de la cámara oscura*, el cual está diseñado principalmente para cámaras que tienen dispositivos de cargas eléctricas interconectadas (DCEI) como sensores.

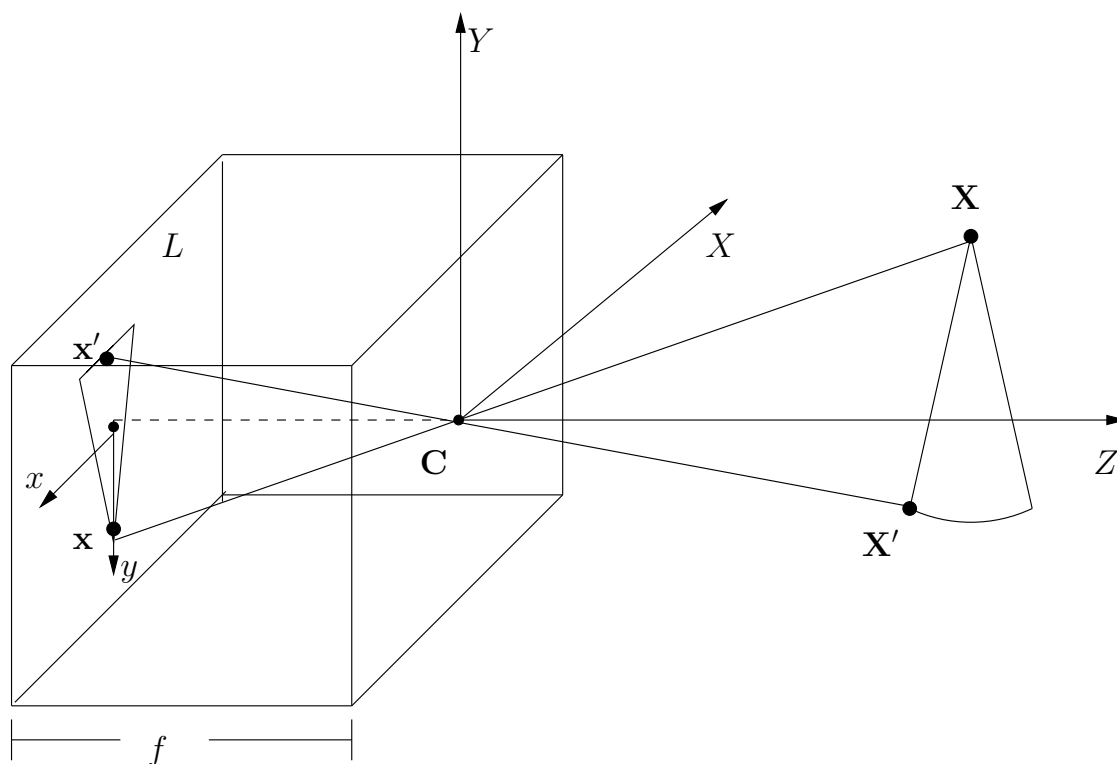


Figura 3.3: Representación del modelo de la cámara oscura.

La base del modelo de la cámara oscura

Una cámara oscura posee un *obturador* a través del cual se proyecta el mundo real sobre un *plano de la imagen*, así se considera la proyección de los puntos que yacen en el espacio sobre un plano.

Nos referiremos a la figura 3.3. Se ha de convenir en representar a los puntos en el espacio real con letras mayúsculas y se utilizará letras minúsculas para representar a sus proyecciones en el plano.

3. MARCO TEÓRICO

Se denota por \mathbf{C} al punto del obturador, también llamado *centro de proyección*, y por L al plano de la imagen. El origen del sistema de coordenadas euclidianas se fija en el punto \mathbf{C} . En este modelo, un punto con coordenadas $\mathbf{X} = (X, Y, Z)$ en el espacio es transformado a un punto sobre el plano de la imagen, donde la línea que conecta \mathbf{X} con \mathbf{C} se extiende hasta el plano de la imagen (ver fig. 3.3). Con base a la semejanza de triángulos, la relación entre el punto tridimensional con su proyección bidimensional en la imagen está dada por,

$$x = -f \frac{X}{Z}, \quad y = -f \frac{Y}{Z}. \quad (3.13)$$

donde f es la *longitud del foco*, o sea la distancia entre el centro de proyección y el plano de la imagen. Nótese que hay un signo negativo en la expresión (3.13) debido a que el punto en el espacio y su proyección están en direcciones opuestas respecto al centro de proyección.

Esto determina una transformación del espacio euclidiano tridimensional \mathbb{R}^3 al espacio euclidiano bidimensional \mathbb{R}^2 , y hace que la imagen de un objeto aparezca invertida en el plano de la imagen. Para eliminar este efecto, es posible voltear la imagen: $(x, y) \mapsto (-x, -y)$. Esto corresponde a posicionar el plano de la imagen (con ecuación $z = -f$) enfrente del centro de proyección (ver fig. 3.4), por lo tanto, la expresión (3.13) queda como

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}. \quad (3.14)$$

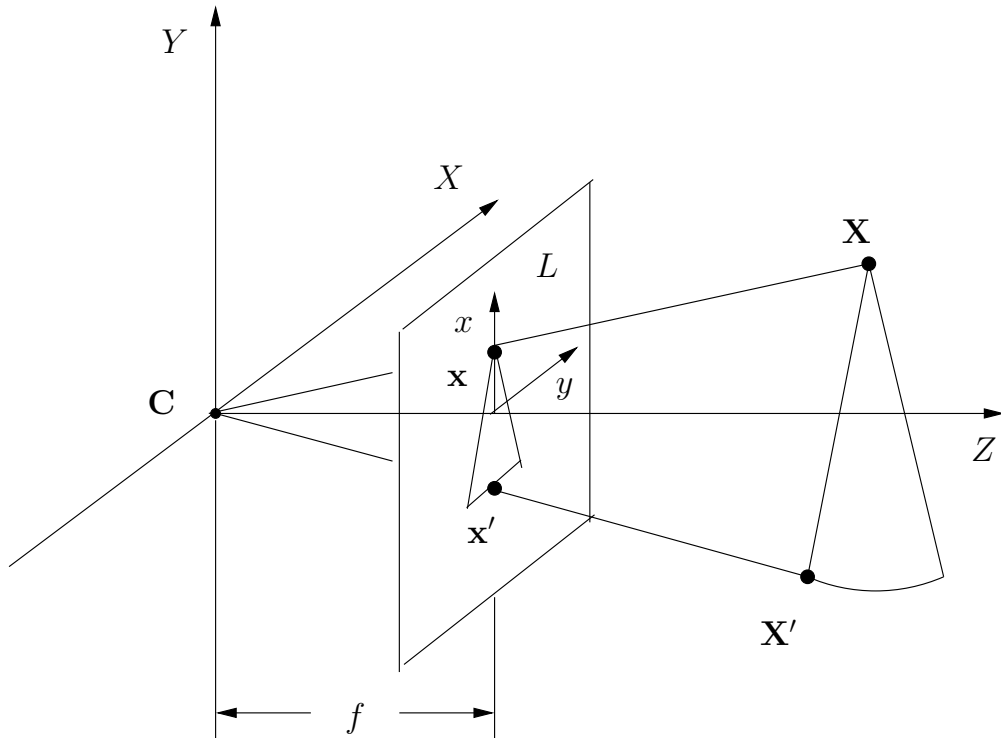


Figura 3.4: Plano de la imagen frente al centro de proyección.

Al centro de proyección también se le conoce como el *centro de la cámara* o *centro*

óptico. A la línea que pasa por el centro de la cámara y es perpendicular al plano de la imagen se le llama *eje principal de la cámara*, y al punto donde el eje principal se une con el plano de la imagen, se le conoce como *punto principal*.

Modelo de la cámara oscura empleando coordenadas homogéneas

Si los puntos en el mundo y los puntos en la imagen están representados por vectores homogéneos, entonces la proyección se puede expresar como una transformación lineal entre sus coordenadas homogéneas. En particular la expresión (3.14) puede ser descrita en términos de una matriz de multiplicación como

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.15)$$

La matriz de la expresión anterior se puede escribir como $diag(f, f, 1)[I \mid \mathbf{0}]$ donde $diag(f, f, 1)$ es la matriz diagonal y $[I \mid \mathbf{0}]$ es la matriz formada por dos bloques: la matriz identidad de orden 3×3 y el vector columna de dimensión 3, que en este caso es el vector cero. En esta parte se introduce la notación para expresar a un punto en el mundo \mathbf{X} como vector homogéneo de dimensión 4, un punto en la imagen \mathbf{x} como vector homogéneo de dimensión 3 y una matriz M para representar la matriz de proyección de la cámara en su forma homogénea. Gracias a esta notación, se puede escribir de forma compacta la expresión (3.15) como

$$\mathbf{x} = M\mathbf{X},$$

la cual define la matriz de la cámara para el modelo de la cámara oscura como

$$M = diag(f, f, 1)[I \mid \mathbf{0}].$$

Posición del punto principal

La expresión (3.14) presupone que el origen del sistema de coordenadas de la imagen es el punto principal. Sin embargo, en la práctica, esto puede no ser cierto, de modo que en general existe una transformación que se expresa como

$$x = f \frac{X}{Z} + o_x, \quad y = f \frac{Y}{Z} + o_y,$$

donde $[o_x, o_y]^T$ son las coordenadas del punto principal en la imagen (ver fig. 3.5). La expresión anterior se puede escribir en coordenadas homogéneas como

$$\begin{bmatrix} fX + o_x \\ fY + o_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.16)$$

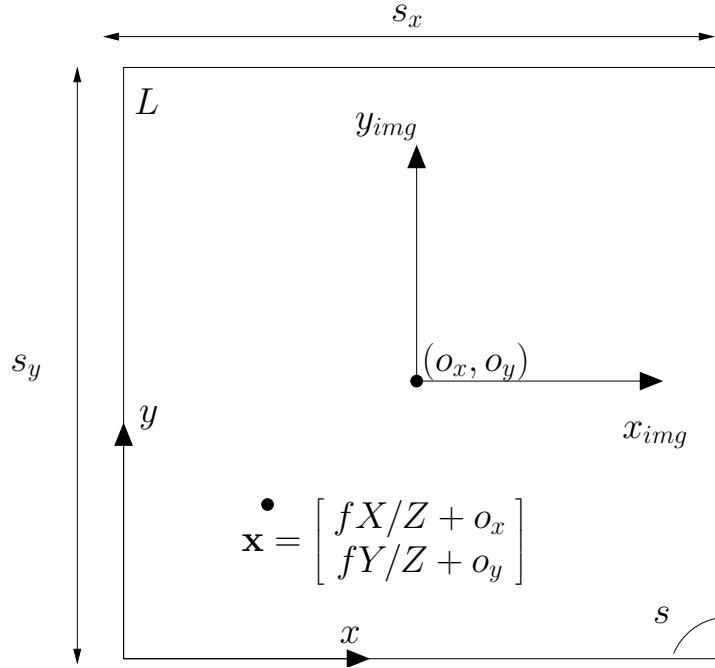


Figura 3.5: Parámetros de la cámara adicionales.

A partir de la ecuación anterior es posible definir la siguiente matriz,

$$K = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.17)$$

a la cual se le conoce como *matriz de calibración*.

El modelo de la cámara oscura que se ha presentado hasta ahora, presupone que las coordenadas de la imagen son coordenadas euclidianas con una misma escala entre los ejes coordenados, no obstante, es posible que los píxeles no sean cuadrados. Si las coordenadas de la imagen son medidas en píxeles, entonces los factores de escala entre los ejes coordenados de la imagen son diferentes. En particular, si el número de píxeles por unidad de distancia en las coordenadas de la imagen son s_x y s_y en las direcciones x y y , respectivamente, entonces la transformación de las coordenadas del mundo a las coordenadas en píxeles es obtenida multiplicando por la izquierda a la matriz expresada en (3.17) con la matriz dada por $diag(s_x, s_y, 1)$. De esta manera, la forma general de la matriz de calibración es:

$$K = \begin{bmatrix} fs_x & 0 & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.18)$$

Matriz de calibración general

Para tener un modelo más general de la cámara oscura, es necesario añadir un parámetro interno extra, el cual se identifica como el parámetro de oblicuidad s . Este parámetro es proporcional a la cotangente del ángulo entre los ejes de coordenadas

de la imagen. Típicamente, este ángulo es muy cercano a 90 grados y, por lo tanto, s es muy cercano a 0. Este parámetro se expresa dentro de la matriz de calibración de la siguiente forma:

$$K = \begin{bmatrix} fs_x & s & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.19)$$

Al final, la transformación de un punto tridimensional \mathbf{X}_c a un punto de la imagen \mathbf{x} , está dada por

$$\mathbf{x} = K[I \mid \mathbf{0}]\mathbf{X}_c. \quad (3.20)$$

En esta expresión, se ha escrito el punto tridimensional $[X, Y, Z, 1]^T$ como \mathbf{X}_c para enfatizar que la cámara está situada en el origen del sistema de coordenadas euclidiano, con el eje principal de la cámara puesto sobre el eje Z , por lo tanto, el punto \mathbf{X}_c está expresado en este sistema de coordenadas, el cual es llamado *sistema de coordenadas de la cámara*.

Parámetros de orientación y posición de la cámara

En general, los puntos en el espacio son expresados en términos de un sistema de coordenadas euclidiano diferente, conocido como *sistema de coordenadas del mundo*. Este sistema de coordenadas está relacionado con el sistema de coordenadas de la cámara, mediante una transformación dada por una matriz de rotación y un vector de traslación (ver fig. 3.6).

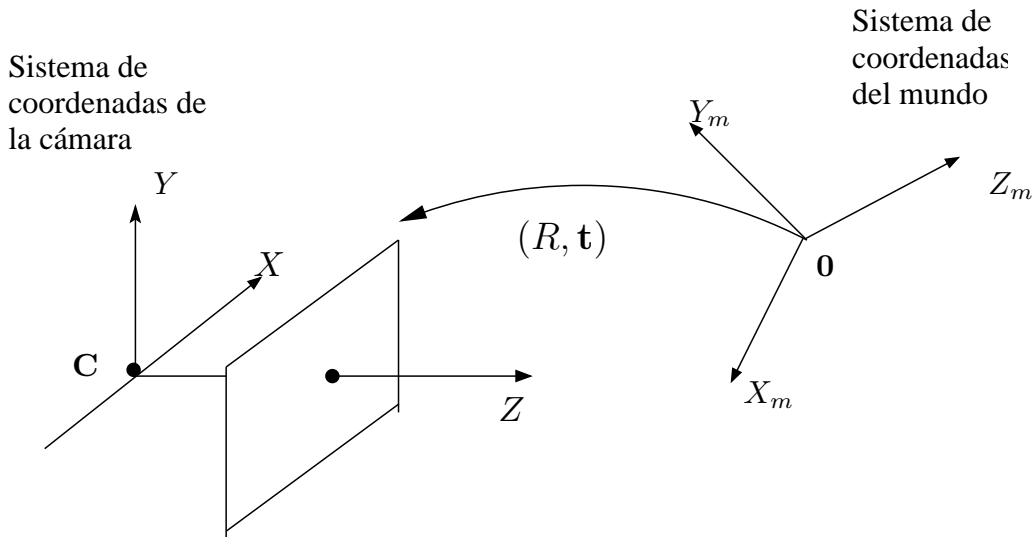


Figura 3.6: Transformación entre el sistema de coordenadas del mundo y el sistema de coordenadas de la cámara.

Sea $\tilde{\mathbf{X}}$ el vector de coordenadas euclidianas de un punto en el mundo, respecto al sistema de coordenadas de este último, y sea $\tilde{\mathbf{X}}_c$ el vector de coordenadas euclidianas del mismo punto, respecto al sistema de coordenadas de la cámara. Entonces la relación entre ambos es

$$\tilde{\mathbf{X}}_c = R[\tilde{\mathbf{X}} - \tilde{\mathbf{C}}]. \quad (3.21)$$

donde $\tilde{\mathbf{C}}$ es el 3-vector que representa las coordenadas euclidianas del centro óptico

en el sistema de coordenadas del mundo, y R es la matriz de rotación de orden 3×3 que representa la orientación del sistema de coordenadas de la cámara. Por tanto, la ecuación (3.21) se representa en coordenadas homogéneas como

$$\mathbf{X}_c = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}. \quad (3.22)$$

Al sustituir el punto \mathbf{X}_c en la ecuación (3.20) se obtiene la siguiente ecuación:

$$\mathbf{x} = KR[I \mid -\tilde{\mathbf{C}}]\mathbf{X}, \quad (3.23)$$

donde \mathbf{X} es el punto en el sistema de coordenadas del mundo. Esta es la transformación general dada por el modelo de la cámara oscura. Los parámetros contenidos en K son conocidos como *parámetros internos de la cámara*. A los parámetros R y $\tilde{\mathbf{C}}$ que relacionan la orientación y posición de la cámara con el sistema de coordenadas del mundo se les conoce como: *parámetros de orientación y posición de la cámara* o *parámetros externos de la cámara*. Los parámetros internos, los parámetros de orientación y los parámetros de posición de la cámara son conocidos como los parámetros de la cámara.

Por conveniencia, el centro de la cámara no se expresa explícitamente. En su lugar se emplea un vector $\mathbf{t} = -R\tilde{\mathbf{C}}$ conocido como vector de traslación. De tal forma que la matriz de la cámara o matriz de proyección se expresa como

$$M = K[R \mid \mathbf{t}]. \quad (3.24)$$

Entonces la transformación del punto \mathbf{X} en el sistema de coordenadas del mundo al punto \mathbf{x} en el sistema de coordenadas de la imagen está dada por

$$\mathbf{x} = M\mathbf{X}. \quad (3.25)$$

3.1.5. Geometría epipolar

A la geometría proyectiva intrínseca entre dos imágenes, la cual es independiente de la estructura de la escena y sólo depende de los parámetros internos de la cámara y su pose relativa, se le conoce como *geometría epipolar*.

Esencialmente, la geometría epipolar entre dos imágenes es la geometría dada por la intersección del plano de cada imagen con una familia de planos, donde todos los planos de esta familia contienen la línea base (línea que une los centros ópticos \mathbf{C} y \mathbf{C}').

Para ver la importancia de esta familia de planos es necesario enfocarse a uno solo. Si existe un punto tridimensional \mathbf{X} y éste es proyectado a dos imágenes distintas, entonces los puntos bidimensionales de cada imagen con el punto tridimensional forman un plano Π , el cual también contiene a la línea base.

En el caso de que sólo se conozca el punto bidimensional \mathbf{x} de la primera imagen y se desea encontrar el punto bidimensional correspondiente \mathbf{x}' en la segunda imagen, es necesario hacer uso del plano Π , donde ahora, sólo está determinado por la línea

base y la línea definida por \mathbf{x} . Sin embargo, se conoce que el punto \mathbf{x}' también yace en el plano Π , por lo tanto, el punto \mathbf{x}' yace en una línea l'_x , que es la intersección del plano Π con el plano de la segunda imagen L' (ver fig. 3.7). Esta línea l' es la proyección en la segunda imagen de la línea que une al punto \mathbf{X} con el punto \mathbf{x} y se le conoce como *línea epipolar* correspondiente al punto \mathbf{x} , y al plano Π se le conoce como *plano epipolar*.

De forma general, todas las líneas epipolares, correspondientes a cada uno de los puntos de la primera imagen, pasan a través de un punto común, conocido como *epipolo* \mathbf{e}' , el cual es la intersección de la línea base con el plano L' . Este epipolo es la proyección del centro óptico de la primera vista en el plano L' (segunda imagen).

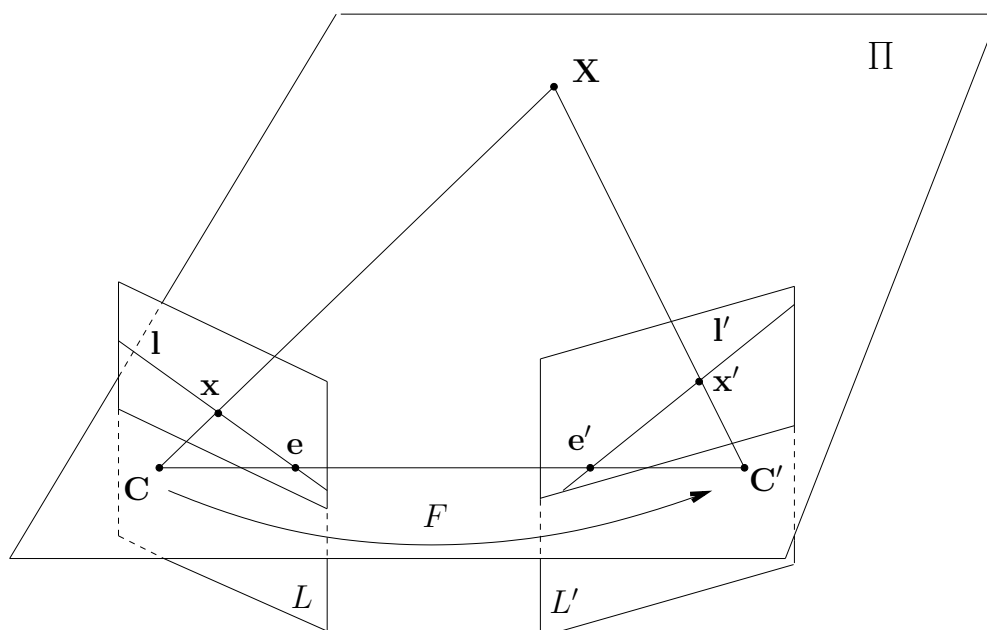


Figura 3.7: Modelo de la geometría epipolar.

Asimismo, el punto \mathbf{x}' en el plano L' presenta su línea epipolar $l_{x'}$ en el plano L de la primera imagen y en esta línea epipolar yace el punto \mathbf{x} . El epipolo \mathbf{e} que yace en el plano L , es la proyección del centro óptico de la segunda imagen C' , y en él se intersecan todas las líneas epipolares que pertenecen al plano L .

Matriz fundamental

Cuando se tiene un par de imágenes de una misma escena, se ha descrito que por cada punto \mathbf{x} de la primera imagen existe su correspondiente línea epipolar l' en la segunda imagen. De tal forma que cualquier punto \mathbf{x}' en la segunda imagen que corresponde al punto \mathbf{x} debe yacer en la línea epipolar l' . La línea epipolar es la proyección, en la segunda imagen, de la línea que parte del punto \mathbf{x} hacia el centro óptico de la primera cámara. Entonces, existe una transformación proyectiva de un punto en la primera imagen a su línea epipolar correspondiente en la segunda imagen, es decir, transformación de puntos a líneas. Esta transformación está representada por una matriz F de orden 3×3 y rango 2 conocida como *la matriz fundamental* [5].

La forma de la matriz fundamental se puede derivar algebraicamente [74], en términos de las dos matrices de proyección M y M' . Esta derivación se presenta a

continuación. El rayo proyectado a partir de \mathbf{x} mediante M se obtiene al resolver $M\mathbf{X} = \mathbf{x}$. La familia de soluciones es de la forma:

$$\mathbf{X}(\lambda) = M^+\mathbf{x} + \lambda\mathbf{C}, \quad (3.26)$$

donde M^+ es la pseudo-inversa de M , es decir, $MM^+ = I$ (matriz identidad de orden 3×3), \mathbf{C} es el centro óptico de la primera imagen, y λ es el valor escalar para parametrizar al rayo. Dos puntos en particular del rayo son: $M^+\mathbf{x}$ si $\lambda = 0$, y el primer centro óptico \mathbf{C} si $\lambda = \infty$. Estos dos puntos son proyectados en la segunda imagen, por la segunda matriz de proyección M' como $M'M^+\mathbf{x}$ y $M'\mathbf{C}$, respectivamente. La línea epipolar es la línea que une a este par de puntos proyectados y se expresa como $l' = (M'\mathbf{C}) \times (M'M^+\mathbf{x})$. El punto $M'\mathbf{C}$ es el epipolo \mathbf{e}' en la segunda imagen, entonces la línea epipolar queda expresada como $l' = [\mathbf{e}']_{\times}(M'M^+)\mathbf{x} = F\mathbf{x}$, donde $[\mathbf{e}']_{\times}$ es la matriz antisimétrica del epipolo $\mathbf{e}' = [e'_1, e'_2, e'_3]^T$ expresada como,

$$[\mathbf{e}']_{\times} = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix}, \quad (3.27)$$

de tal forma que la expresión que representa a la matriz fundamental es

$$F = [\mathbf{e}']_{\times}(M'M^+). \quad (3.28)$$

Al emplear la forma canónica en las matrices de proyección, es decir, la primera matriz de proyección M está en el origen del sistema de coordenadas del mundo $M = K[I \mid \mathbf{0}]$ y la segunda matriz de proyección se expresa como $M' = K'[R \mid \mathbf{t}]$ con respecto a la primera matriz de proyección, la pseudo-inversa de M y el centro de proyección quedan expresados como,

$$M^+ = \begin{bmatrix} K^{-1} \\ \mathbf{0}^T \end{bmatrix} \quad \mathbf{C} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}.$$

y los epipolos se definen como,

$$\mathbf{e} = M \begin{pmatrix} -R^T\mathbf{t} \\ 1 \end{pmatrix} = KR^T\mathbf{t} \quad \mathbf{e}' = M' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = K'\mathbf{t}.$$

Al sustituir estos valores en la expresión (3.28) se tiene que,

$$\begin{aligned} F &= [\mathbf{e}']_{\times}K'RK^{-1} = [K'\mathbf{t}]_{\times}K'RK^{-1} = K'^{-T}[\mathbf{t}]_{\times}RK^{-1} \\ &= K'^{-T}R[R^T\mathbf{t}]_{\times}K^{-1} = K'^{-T}RK^T[KR^T\mathbf{t}]_{\times} = K'^{-T}RK^T[\mathbf{e}]_{\times}. \end{aligned} \quad (3.29)$$

Condición de correspondencia de la matriz fundamental

Hasta ahora sólo se ha considerado la transformación de un punto a una línea mediante F . Sin embargo, esta transformación se puede extender a una de las propiedades más importantes de la matriz fundamental [5]. Esta propiedad, conocida como condición de correspondencia, establece que para cualquier correspondencia de puntos $\mathbf{x} \leftrightarrow \mathbf{x}'$ en dos imágenes se satisface la siguiente condición:

$$\mathbf{x}'^T F \mathbf{x} = 0. \quad (3.30)$$

Esta condición se cumple dado que, si \mathbf{x} y \mathbf{x}' son correspondientes, \mathbf{x}' yace sobre la línea epipolar $l' = F\mathbf{x}$ que corresponde al punto \mathbf{x} , de tal forma que $0 = \mathbf{x}'^T l' = \mathbf{x}'^T F\mathbf{x}$. De forma inversa, si los puntos en la imagen satisfacen la relación $\mathbf{x}'^T F \mathbf{x} = 0$, entonces las líneas definidas por estos puntos son coplanares. Esta es una condición necesaria para determinar que los puntos son correspondientes.

La importancia de esta propiedad radica en la forma de caracterizar a la matriz fundamental sin emplear las matrices de proyección; es decir, esta matriz puede ser caracterizada solamente con la correspondencia de puntos.

3.2. Reconstrucción tridimensional

Para realizar la reconstrucción tridimensional a partir de dos imágenes, es necesario conocer la correspondencia de puntos $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, entre ambas vistas o imágenes, donde \mathbf{x}_i pertenece al conjunto de puntos de la primera imagen y \mathbf{x}'_i pertenece al conjunto de puntos de la segunda imagen. Se presupone que la correspondencia de puntos proviene de un conjunto de puntos tridimensionales desconocidos \mathbf{X}_i . Del mismo modo, la posición, la orientación y la matriz de calibración de la cámara son desconocidas. Por ello, la parte principal de la reconstrucción tridimensional es encontrar estos parámetros desconocidos.

Cuando se parte de las correspondencias de puntos de dos imágenes, comúnmente se emplea la matriz fundamental para obtener los parámetros de la cámara y la reconstrucción de los puntos tridimensionales. Es posible estimar la matriz fundamental a partir de siete o más correspondencias de puntos [5]. Al usar esta matriz se pueden obtener las matrices de proyección y realizar la triangulación para obtener una reconstrucción tridimensional de la escena en un espacio proyectivo. Sin embargo, esta reconstrucción (ver fig. 3.8) dista de la estructura real de la escena.

3.2.1. Ambigüedad de la reconstrucción

Si no se conoce la relación entre los puntos tridimensionales de la escena y los puntos de las imágenes, generalmente, no es posible obtener la posición ni la orientación absoluta de la escena. Esto también sucede cuando se tienen tres o más vistas. De hecho, si se tiene el conocimiento de la matriz de calibración, tampoco es posible obtener su posición ni orientación absoluta (latitud, longitud y altitud) en el mundo. De tal forma que la mejor reconstrucción posible de la escena es sobre un espacio euclidiano con respecto a un sistema de coordenadas tridimensional propio.

Se debe tener en cuenta que, aún teniendo un sistema de coordenadas local en la escena, la reconstrucción que se obtiene mediante del método convencional de reconstrucción, se encuentra bajo un factor de escala arbitrario. Por ello, existe una transformación del sistema de coordenadas local con el sistema de coordenadas de la reconstrucción. Esta transformación pertenece a la clase II conocida como similitud. Al aplicar esta transformación H_s a los puntos tridimensionales y a las matrices de

proyección se tiene que,

$$MX_i = (MH_s^{-1})(H_s\mathbf{X}_i) \quad M'\mathbf{X}'_i = (M'H_s^{-1})(H_s\mathbf{X}'_i),$$

donde se observa que los puntos en ambas imágenes no cambian, pero sí las matrices de proyección y los puntos tridimensionales. Además, si la primera matriz de proyección se descompone como $M = K[R_m | \mathbf{t}_m]$, entonces se puede calcular a M con la transformación de similitud H_s^{-1} como,

$$MH_s^{-1} = K[R_m R^{-1} | \mathbf{t}'],$$

para un cierto vector de traslación \mathbf{t}' . Este resultado muestra que multiplicar por la transformación de similitud H_s^{-1} no produce un cambio en la matriz de calibración K . Por consiguiente, esta ambigüedad de reconstrucción existe aún para las matrices de proyección generadas a partir de cámaras calibradas (aquellas cámaras en las que se conozcan, previamente, las matrices de calibración [5]). Tal ambigüedad fue demostrada por Longuet-Higgins en [20], quienes concluyen que la reconstrucción tridimensional es posible hasta una transformación de similitud si se conocen las matrices de calibración.

Ambigüedad proyectiva

Si no se tiene conocimiento de las matrices de calibración, ni de la posición y orientación de la cámara, entonces la ambigüedad de la reconstrucción queda representada por una transformación proyectiva. En particular, si H es cualquier matriz invertible de orden 4×4 , la cual representa una transformación proyectiva de \mathbb{P}^3 , entonces al reemplazar los puntos \mathbf{X}_i por $H\mathbf{X}_i$ y las matrices de proyección M y M' por MH^{-1} y $M'H^{-1}$ los puntos en la imagen no cambian. Esto permite mostrar que los puntos tridimensionales \mathbf{X}_i y las matrices de proyección pueden ser determinadas bajo una transformación proyectiva, donde esta transformación es la ambigüedad que se tiene cuando sólo se cuenta con las correspondencia de puntos o, de forma más específica, cuando se emplea la matriz fundamental [5]. De tal forma que, la reconstrucción a partir de cámaras sin calibración es posible, solamente, bajo una transformación proyectiva.

3.2.2. Tipos de reconstrucciones

En cualquier problema de reconstrucción existe una escena real o verdadera que consiste de puntos reales $\bar{\mathbf{X}}_i$ y matrices de proyección de las cámaras reales \bar{M} y \bar{M}' que generan la correspondencia de puntos en las imágenes. La reconstrucción de los puntos tridimensionales y la estimación de las matrices de proyección a partir de las correspondencias difiere de la escena real por una transformación que pertenece a la clase II (similitud), III (afín) o IV (proyectiva). El tipo de transformación que está involucrada en las matrices de proyección indica el tipo de reconstrucción que se obtuvo (reconstrucción proyectiva, reconstrucción afín o reconstrucción similar). Cabe señalar que el término reconstrucción métrica es normalmente usado en lugar de reconstrucción similar, pero el significado de ambas es el mismo.

Reconstrucción proyectiva

Si las matrices de proyección están definidas por alguna transformación proyectiva, entonces, al obtener los puntos tridimensionales a partir de las correspondencias de puntos, se genera una reconstrucción que yace en un espacio proyectivo. En este espacio no existe métrica alguna y el concepto de paralelismo no está definido. En la figura 3.8 se observa un ejemplo de una reconstrucción proyectiva de una casa de campo.

Si una de las matrices de proyección se define a partir de la matriz fundamental, entonces, sólo podemos aspirar a una reconstrucción proyectiva, dado que la matriz fundamental no contiene información explícita de la matriz de calibración. Esto genera una ambigüedad en las matrices de proyección dada por la clase de transformación proyectiva. De manera más formal, esto se expresa mediante el siguiente teorema:

Teorema 1 (Reconstrucción proyectiva) *Supongamos que $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ es un conjunto de correspondencia de puntos de dos imágenes y que la matriz fundamental F es única, determinada por la condición $\mathbf{x}_i^T F \mathbf{x}_i = 0 \quad \forall i$. Sean $(M_1, M'_1, \mathbf{X}_{1i})$ y $(M_2, M'_2, \mathbf{X}_{2i})$ dos reconstrucciones de la correspondencia de puntos $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. Entonces existe una matriz no singular H tal que $M_2 = M_1 H^{-1}$, $M'_2 = M'_1 H^{-1}$ y $\mathbf{X}_{2i} = H \mathbf{X}_{1i}$ para toda i **excepto** para aquellas i tal que $F \mathbf{x}_i = \mathbf{x}_i^T F = \mathbf{0}$ (ver demostración en [5]).*

Es claro que la matriz fundamental es determinada, únicamente, por un par de matrices de proyección M y M' y, por el contrario, n diferentes pares de matrices de proyección M_i y $M'_i \quad \forall i = 1, 2, \dots, n$ son determinadas por la matriz fundamental, bajo una ambigüedad proyectiva tridimensional. De tal forma que la matriz fundamental encapsula completamente la geometría proyectiva del par de cámaras, y no sufre cambios al realizar una transformación proyectiva tridimensional.

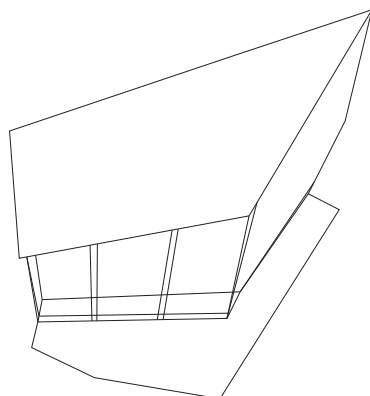


Figura 3.8: Reconstrucción proyectiva.

3.2.3. Reconstrucción afín

Con la ayuda de ciertas características de la escena, es posible pasar de una reconstrucción proyectiva a una reconstrucción métrica. Una forma de lograr esta transformación de espacios tridimensionales, es obtener una reconstrucción intermedia.

Si se parte de una reconstrucción proyectiva de la escena, es decir, se tienen las matrices de proyección para la reconstrucción proyectiva M_p y M'_p , los puntos tridimensionales en un espacio proyectivo \mathbf{X}_i , y además, se conoce un plano π identificado como el plano al infinito en las coordenadas del espacio proyectivo (en el espacio euclidiano las coordenadas de π son $[0, 0, 0, 1]^T$), entonces existe una matriz de transformación H que convierte al plano al infinito, el cual está en el espacio proyectivo, a otro espacio ($H^{-T}\pi = [0, 0, 0, 1]^T$). La matriz de transformación H está dada por,

$$H = \begin{bmatrix} I & | & \mathbf{0} \\ \hline & & \pi^T \end{bmatrix}, \quad (3.31)$$

donde I es la matriz de identidad de orden 3×3 y $\mathbf{0}$ es el vector nulo de dimensión 3. Esta matriz de transformación se aplica a todos los puntos tridimensionales \mathbf{X}_i y a las dos matrices de proyección M_p y M'_p . No obstante, la reconstrucción obtenida con la aplicación de esta matriz de transformación H no es una reconstrucción euclidiana. Esta reconstrucción difiere de la reconstrucción euclidiana por una transformación que fija el plano al infinito llamada *transformación afín*. Por lo tanto, la reconstrucción que difiera por una transformación afín de la reconstrucción euclidiana es una *reconstrucción afín*. En la figura 3.9 se observa un ejemplo de una reconstrucción afín de una casa de campo.

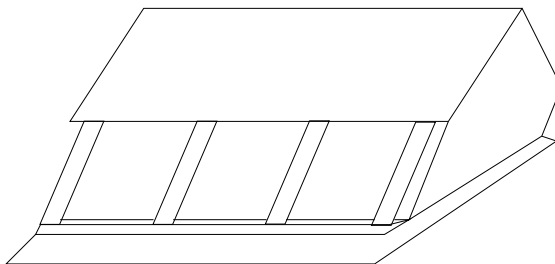


Figura 3.9: Reconstrucción afín.

3.2.4. Reconstrucción métrica

Si se definen las matrices de proyección (M, M') de la siguiente forma,

$$M = K[R | \mathbf{t}] \quad M' = K'[R' | \mathbf{t}'],$$

y se estiman los puntos tridimensionales \mathbf{X}_i a partir de puntos correspondientes $\mathbf{x} \leftrightarrow \mathbf{x}'$, entonces se obtiene una reconstrucción métrica. Nótese que una reconstrucción métrica es una reconstrucción euclidiana bajo una transformación de similitud en R^3 , por lo tanto, yace en el espacio euclidiano. De hecho el término reconstrucción euclidiana es frecuentemente usado en la literatura [49, 21, 61, 64] para hacer referencia a una reconstrucción similar. En la figura 3.10 se observa un ejemplo de una reconstrucción métrica de una casa de campo.

Para obtener esta reconstrucción es necesario conocer la matriz de calibración de la cámara y aplicarla a la matriz fundamental. A continuación se muestran los pasos del método convencional para realizar la reconstrucción tridimensional métrica a partir de las correspondencias de puntos [5].

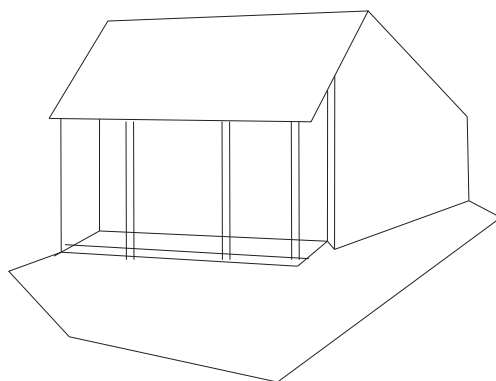


Figura 3.10: Reconstrucción métrica.

1. Estimar la matriz fundamental a partir de las correspondencias de puntos (un método que se emplea comúnmente para estimar esta matriz, es el algoritmo de los ocho puntos [75, 5]).
2. Calibrar la cámara a partir de la matriz fundamental mediante las ecuaciones de Kruppa o sus derivaciones [5, 23, 26].
3. Estimar los parámetros de orientación y posición.
4. Estimar los puntos tridimensionales (triangular).

El segundo paso de este método es el más importante ya que permite eliminar la ambigüedad proyectiva de la reconstrucción y permiten obtener una reconstrucción métrica de la escena.

3.2.5. Calibración de la cámara

Para calcular los puntos tridimensionales a partir de dos imágenes es necesario conocer las matrices de proyección [5]. Al partir de la matriz fundamental, las matrices de proyección quedan expresadas en su forma canónica de la siguiente manera:

$$M = [I \mid \mathbf{0}] \quad \text{y} \quad M' = [[\mathbf{e}']_{\times} F \mid \mathbf{e}']. \quad (3.32)$$

Al tener las matrices de proyección, el paso final es calcular los puntos tridimensionales. Sin embargo, la reconstrucción que se obtiene es una reconstrucción proyectiva, como se mencionó anteriormente. Para eliminar la ambigüedad proyectiva en la reconstrucción es necesario encontrar las matrices de calibración K y K' de ambas cámaras [4]. A este proceso se le conoce como *calibración de la cámara*.

En esta tesis se presupone que las imágenes fueron adquiridas por la misma cámara. Esto implica que ambas matrices de calibración son iguales ($K = K'$). Por otro lado, dada la tecnología que emplean las cámaras actuales, la mayoría de los autores de los métodos de calibración de la cámara [5] presuponen que los píxeles son cuadrados, es decir, que los factores de escala por píxel son iguales ($s_x = s_y = 1$) y el parámetro de oblicuidad es $s = 0$.

En la figura 3.11(a) se muestra cómo se deforma un círculo cuando el parámetro de oblicuidad es diferente de cero y los factores de escala son diferentes. En la

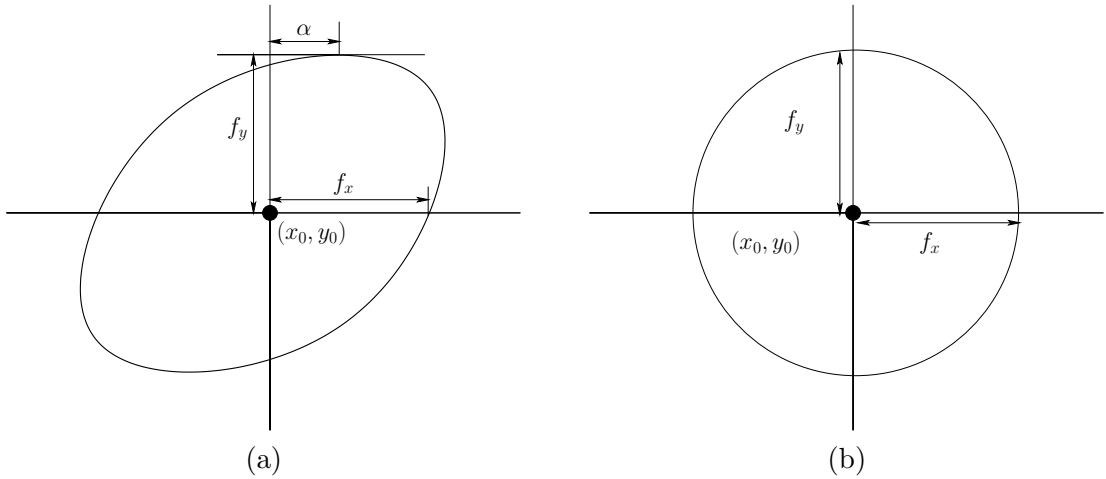


Figura 3.11: Matriz de calibración: (a) bajo la suposición de la matriz expresada en (3.33) con cierto valor de oblicuidad, (b) bajo la suposición de la matriz expresada en (3.34).

figura 3.11(b) se muestra que el círculo no se deforma debido a que el parámetro de oblicuidad es cero y las factores de escala son iguales.

Por otro lado, estos autores también presuponen que las coordenadas del punto principal (o_x, o_y) están a la mitad de la imagen, es decir, $o_x = w/2$ y $o_y = h/2$, donde w es el ancho de la imagen y h es el alto de la misma. Por consiguiente, la matriz de calibración que era de la forma,

$$K = \begin{bmatrix} f s_x & s & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.33)$$

queda expresada como,

$$K = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.34)$$

de tal forma que el único parámetro de la cámara a estimar es la longitud del foco f . Cuando se parte solamente de las correspondencias de puntos y por ende de la matriz fundamental, la longitud del foco se calcula con las ecuaciones de Kruppa¹ o formulaciones derivadas de estas ecuaciones.

3.2.6. Estimar los parámetros de orientación y posición

El siguiente paso del método convencional de la reconstrucción tridimensional es la estimación de los parámetros de orientación y posición de la cámara. La estimación de tales parámetros se obtiene por medio de una matriz E , la cual se calcula con la matriz fundamental y las matrices de calibración de la siguiente manera,

$$E = K^T F K, \quad (3.35)$$

¹En [5, 23, 26] se muestra cómo se derivan y emplean las ecuaciones de Kruppa, y las formulaciones derivadas de ellas para estimar la longitud del foco.

donde la matriz E es conocida como la *matriz esencial*. Al aplicar las matrices de calibración a la matriz fundamental se pasa de la relación de puntos entre cámaras sin calibrar a la relación de puntos entre cámaras calibradas. Esta transformación elimina la ambigüedad proyectiva, quedando simplemente la ambigüedad de similitud.

Nótese que la matriz fundamental se puede expresar en términos de los parámetros de la cámara, es decir, $F = K'^{-T}[\mathbf{t}]_{\times} R K^{-1}$ como se mostró anteriormente. Al aplicar las matrices de calibración K y K' a la matriz fundamental F se obtiene que,

$$E = K'^T F K = K'^T K'^{-T} [\mathbf{t}]_{\times} R K^{-1} K = [\mathbf{t}]_{\times} R, \quad (3.36)$$

de tal forma que la matriz esencial contiene los parámetros de orientación R y posición \mathbf{t} . Para extraer estos parámetros de la matriz esencial, es necesario emplear la descomposición de valores singulares (DVS), es decir,

$$DVS(E) = U_e D_e V_e^T,$$

donde U_e y V_e^T son las matrices ortogonales de orden 3×3 y D_e es la matriz diagonal. A partir de esta descomposición los posibles valores para los parámetros de orientación y posición son:

$$\begin{aligned} R_1 &= U_e W V_e^T & R_2 &= U_e W^T V_e^T \\ \mathbf{t}_1 &= \mathbf{u}_3 & \mathbf{t}_2 &= -\mathbf{u}_3, \end{aligned}$$

donde W_e se define como,

$$W_e = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

y el vector \mathbf{u}_3 como,

$$\mathbf{u}_3 = U[0, 0, 1]^T.$$

La primer matriz de proyección está en su forma canónica $M = K[I \mid \mathbf{0}]$ y la combinación de los parámetros de orientación y posición obtenidos producen cuatro posibles valores para la segunda matriz de proyección,

$$\begin{aligned} M' &= K[R_1 \mid t_1], M' = K[R_1 \mid t_2], \\ M' &= K[R_2 \mid t_1] \text{ y } M' = K[R_2 \mid t_2]. \end{aligned}$$

Al estimar el punto tridimensional con estas matrices, se obtienen cuatro soluciones las cuales se muestran en la figura 3.12. Se puede observar que el punto reconstruido \mathbf{X} puede quedar enfrente de las cámaras 3.12(a), detrás de las cámaras 3.12(b), enfrente de la cámara B' y detrás de la cámara A 3.12(c) o enfrente de la cámara A y detrás de la cámara B' 3.12(d). Para determinar cuál solución se debe elegir, basta con probar que el punto \mathbf{X} reconstruido esté enfrente de ambas cámaras. Esto es suficiente para decir que es una reconstrucción tridimensional métrica correcta.

3.2.7. Triangulación

El último paso del método convencional para la reconstrucción es la triangulación. En este paso se extrae cada uno de los puntos tridimensionales de la intersección de las líneas proyectadas a partir de las correspondencias de puntos y las matrices

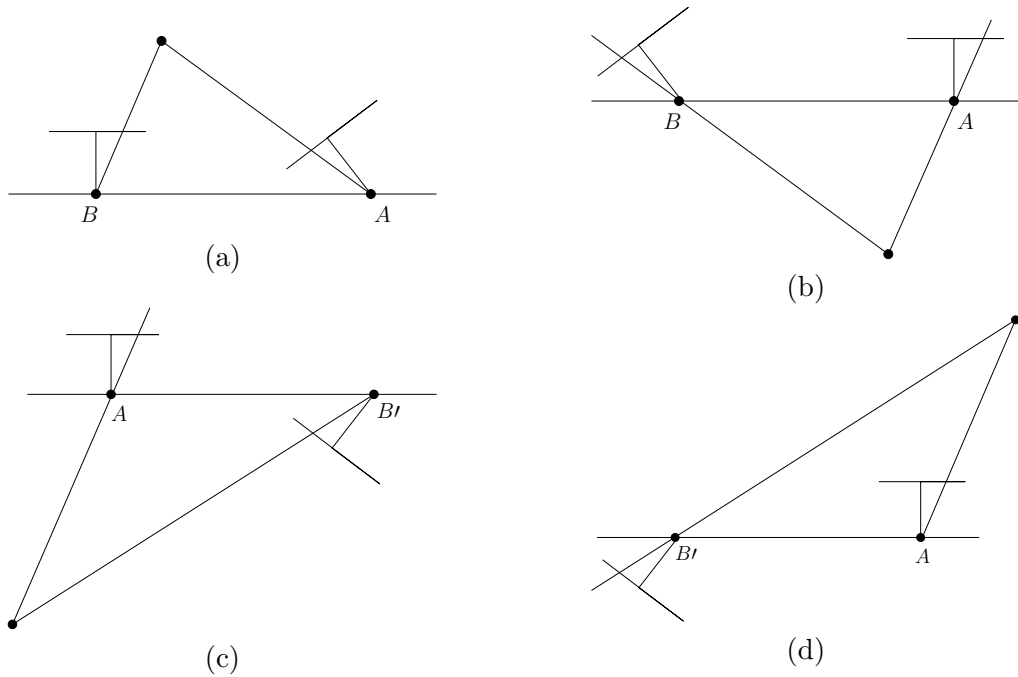


Figura 3.12: Las cuatro soluciones a partir de la matriz esencial E . Entre los lados izquierdo y derecho se encuentra una línea base que refleja las soluciones (a) y (c) con (b) y (d). Entre las soluciones de arriba y abajo, la cámara B rota 180 grados alrededor de la línea base.

de proyección [5, 71]. La triangulación de un punto tridimensional a partir de la correspondencia de puntos y las matrices de proyección se muestra en la figura 3.13.

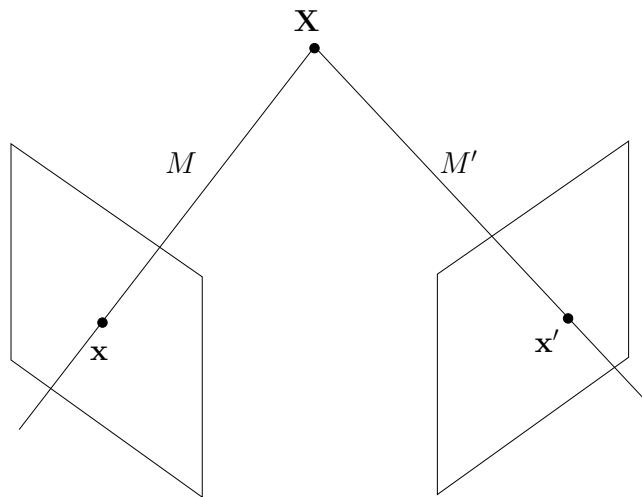


Figura 3.13: Triangulación.

En el siguiente capítulo se detalla este paso ya que es parte de uno de los problemas que se resolvió en esta tesis.

3.2.8. Escenas planas y homografía

Para que los algoritmos que calculan la matriz fundamental obtengan una solución única, es indispensable que todos los puntos tridimensionales a estimar no estén so-

bre ciertas superficies conocidas como superficies críticas². De lo contrario, la matriz fundamental no será única y, por lo tanto, la reconstrucción tridimensional será incorrecta [4]. La mayoría de estas superficies críticas se presentan en raras ocasiones y su importancia es limitada. Sin embargo, existe una superficie crítica llamada plano que se presenta frecuentemente en los ambientes reales, por ejemplo, en ambientes construidos por el hombre (edificios, casas, muros, entre otros). Al aplicar, por ejemplo, el algoritmo de los ocho puntos [75] sobre puntos que yacen sobre un mismo plano, el algoritmo fallará en obtener una solución única.

Homografía entre el plano en el mundo y la imagen

Sin pérdida de generalidad, se presupone que el plano en el sistema de coordenadas del mundo está sobre $Z = 0$. Si se denota la i -ésima columna de la matriz de rotación R como \mathbf{r}_i , entonces tenemos que, con base en el modelo de la cámara oscura,

$$\begin{aligned} \mathbf{x} = M \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} &= K[R \mid \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \\ &= K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ &= K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \tilde{\mathbf{X}}, \end{aligned} \tag{3.37}$$

donde $\tilde{\mathbf{X}} = [X, Y, 1]^T$ representa el punto tridimensional en coordenadas homogéneas sobre el plano sin la coordenada Z , debido a que su valor es 0, y $\mathbf{x} = [x, y, 1]^T$ es el punto en la imagen bajo una escala arbitraria. De tal forma que el punto $\tilde{\mathbf{X}}$ y el punto \mathbf{x} están relacionados por una homografía H , tal que,

$$\mathbf{x} = H\tilde{\mathbf{X}} \quad \text{con} \quad H = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}], \tag{3.38}$$

donde H es una matriz de 3×3 definida bajo un factor de escala arbitrario. Si se denota la homografía como $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$, y se sustituye en la ecuación (3.38), se tiene que

$$[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = \lambda K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}], \tag{3.39}$$

donde λ es una escala arbitraria. Por otro lado, se conoce que los vectores \mathbf{r}_1 y \mathbf{r}_2 son ortonormales, por lo que se pueden obtener dos restricciones,

$$\mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_2 = 0 \tag{3.40}$$

$$\mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_1 = \mathbf{h}_2^T K^{-T} K^{-1} \mathbf{h}_2 \tag{3.41}$$

$$\tag{3.42}$$

²En general, estas superficies pueden ser descritas por ecuaciones cuadráticas en las coordenadas (X, Y, Z) del punto y, por lo tanto, se les denomina superficies cuadráticas.

Estas son dos restricciones básicas sobre la matriz de calibración, dada una homografía, ya que la homografía tiene 8 grados de libertad y se tienen 3 parámetros de orientación y 3 parámetros de posición, sólo se pueden tener dos restricciones para los parámetros intrínsecos. Se debe observar que, en ambas restricciones aparece la expresión $K^{-T}K^{-1}$, la cual describe la imagen de la cónica absoluta [5].

Si se presupone que la matriz de calibración está dada por la expresión (3.34), entonces es posible calibrar la cámara a partir de una imagen mediante las restricciones mostradas anteriormente. No obstante, es necesario conocer a priori que la escena tiene un plano donde yacen los puntos y, a su vez, es necesario conocer los valores de estos puntos en el mundo.

3.3. Fundamentos de optimización

La optimización es la acción de obtener el mejor resultado bajo ciertas circunstancias [76]. El objetivo es decidir cuál es la mejor solución, tal que se minimice el esfuerzo requerido o se maximicen los beneficios deseados. En situaciones prácticas, el esfuerzo requerido o los beneficios deseados pueden ser expresados como una función o funciones que actúan sobre ciertas variables de decisión. Por lo tanto, la optimización puede ser definida como el proceso de encontrar las condiciones que proporcionen el valor máximo o mínimo de una función [76]. Otra definición de la optimización es el intento de maximizar las propiedades deseadas de un sistema, mientras simultáneamente se minimizan las características no deseadas [27].

Las funciones de un problema de optimización son llamadas *funciones objetivo*, ya que sus valores más extremos representan la meta de la optimización. Cuando su valor mínimo es buscado, la función objetivo es frecuentemente referida como *función de costo*. En el caso especial donde el valor mínimo a buscar es cero, la función objetivo es conocida como *función de error* [27].

Los problemas de optimización de una sola función objetivo, también conocidos como *mono-objetivo*, se definen como [77]:

Definición 1 (Problema general de optimización mono-objetivo): *Un problema general de optimización mono-objetivo se define como la minimización o maximización de la función $g(\mathbf{a})$ sujeta a $l_i(\mathbf{a}) \leq 0$, $i = \{1, \dots, m\}$, y $h_j(\mathbf{a}) = 0$, $j = \{1, \dots, p\}$ $\mathbf{a} \in \Omega$. Una solución minimiza (o maximiza) el valor escalar de $g(\mathbf{a})$ donde \mathbf{a} es un vector de dimensión d con las variables de decisión $\mathbf{a} = [a_1, \dots, a_d]^T$ de algún universo Ω .*

Nótese que $l_i(\mathbf{a}) \leq 0$ y $h_j(\mathbf{a}) = 0$ representan las restricciones que deben de cumplirse mientras se optimiza a $g(\mathbf{a})$. Ω contiene todos los posibles vectores \mathbf{a} que pueden ser empleados para cumplir con una evaluación de $g(\mathbf{a})$ y sus restricciones.

El método para encontrar el óptimo global (no necesariamente único) de cualquier función es conocido como *optimización global* [77]. En general, el mínimo global de una función mono-objetivo se presenta en la definición (2).

Definición 2 (Optimización mono-objetivo del mínimo global): *Dada una función $g : \Omega \subseteq \mathbb{R}^d \mapsto \mathbb{R}$, $\Omega \neq \emptyset$, para $\mathbf{a} \in \Omega$ el valor $g^* \triangleq g(\mathbf{a}^*) > -\infty$ es llamado un **mínimo global** si y sólo si,*

$$\forall \mathbf{a} \in \Omega : g(\mathbf{a}^*) \leq g(\mathbf{a}). \quad (3.43)$$

\mathbf{a}^* es por definición la mínima solución global, g es la función objetivo, y el conjunto Ω es la región factible de \mathbf{a} . El objetivo de determinar la solución o soluciones mínimas globales es conocido como **problema de optimización global** para un problema mono-objetivo.

Los problemas de optimización de dos o más funciones objetivo (normalmente estas funciones se encuentran en conflicto entre sí), también conocidos como *multi-objetivo*, se definen como [78, 79, 80, 81]:

Definición 3 Problema general de optimización multi-objetivo: *Un problema general de optimización multi-objetivo se define como la minimización o maximización de las funciones $G(\mathbf{a}) = (g_1(\mathbf{a}), \dots, g_k(\mathbf{a}))$ sujetas a $l_i(\mathbf{a}) \leq 0$, $i = \{1, \dots, m\}$, y $h_j(\mathbf{a}) = 0$, $j = \{1, \dots, p\}$ $\mathbf{a} \in \Omega$. Una solución de un problema de optimización multi-objetivo minimiza (o maximiza) los componentes de un vector $G(\mathbf{a})$ donde \mathbf{a} es un vector de dimensión d con las variables de decisión $\mathbf{a} = [a_1, \dots, a_d]^T$ de algún universo Ω .*

Al igual que en la optimización mono-objetivo, $l_i(\mathbf{a}) \leq 0$ y $h_j(\mathbf{a}) = 0$ representan las restricciones que deben de cumplirse mientras se optimiza a $G(\mathbf{a})$. Ω contiene todos los posibles vectores \mathbf{a} que pueden ser empleados para satisfacer una evaluación de $G(\mathbf{a})$ y sus restricciones.

En los problemas de optimización multi-objetivo, la noción de *óptimo* es diferente, dado que el objetivo es encontrar buenos compromisos en lugar de encontrar un solución única como en la optimización global [77]. La notación de *óptimo*, comúnmente adoptada, es presentada en la definición (4), la cual fue generalizada por Vilfredo Pareto [79] y se le conoce como *óptimo de Pareto*.

Definición 4 (Optimalidad de Pareto): *Una solución $\mathbf{a} \in \Omega$ se dice que es un óptimo de Pareto con respecto a Ω si y sólo si, no existe un vector $\mathbf{a}' \in \Omega$ tal que $\mathbf{v} = G(\mathbf{a}') = (g_1(\mathbf{a}'), \dots, g_k(\mathbf{a}'))$ domina a $\mathbf{u} = G(\mathbf{a}) = (g_1(\mathbf{a}), \dots, g_k(\mathbf{a}))$. La frase **optimalidad de Pareto** está definida con respecto al espacio de las variables de decisión salvo que se especifique lo contrario.*

Asimismo, existen otras tres definiciones presentadas a continuación que también son empleadas en la optimización multi-objetivo [79, 77].

Definición 5 (Dominancia de Pareto): *Un vector $\mathbf{u} = [u_1, \dots, u_k]^T$ se dice que es **dominado** por otro vector $\mathbf{v} = [v_1, \dots, v_k]^T$ (denotado por $\mathbf{u} \preceq \mathbf{v}$) si y sólo si, \mathbf{u} es parcialmente menor que \mathbf{v} , es decir, $\forall i \in \{1, \dots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$.*

Definición 6 (Conjunto óptimo de Pareto): *Para un problema de optimización multi-objetivo dado, $G(\mathbf{a})$, el conjunto óptimo de Pareto, \mathcal{P}^* , está definido como:*

$$\mathcal{P}^* := \{\mathbf{a} \in \Omega \mid \neg \exists \mathbf{a}' \in \Omega \ G(\mathbf{a}') \preceq G(\mathbf{a})\}. \quad (3.44)$$

Definición 7 (Frente de Pareto): Para un problema de optimización multi-objetivo dado, $G(\mathbf{a})$, y un conjunto óptimo de Pareto \mathcal{P}^* , el frente de Pareto \mathcal{PF}^* , está definido como:

$$\mathcal{PF}^* := \{\mathbf{u} = G(\mathbf{a}) \mid \mathbf{a} \in \mathcal{P}^*\}. \quad (3.45)$$

A los componentes del conjunto de óptimos de Pareto se les denomina soluciones no-inferiores, soluciones admisibles o eficientes, con respecto al conjunto representado por \mathcal{P}^* . Cuando se grafican en el espacio de las funciones objetivo a los vectores no dominados, el resultado es conocido como *frente de Pareto*.

Dado que no existe un método capaz de resolver todos los problemas de optimización de manera eficiente [76], se ha desarrollado una amplia gama de métodos que resuelven cierto tipo o clase de problemas.

3.3.1. Análisis numérico

El análisis numérico es una disciplina que se ocupa en describir, analizar y crear métodos numéricos que permitan resolver problemas matemáticos [82].

Los métodos numéricos se pueden clasificar en dos grupos (de acuerdo al tipo de funciones a resolver):

- Aplicados a funciones lineales.
- Aplicados a funciones no lineales.

Los métodos numéricos que son aplicados a este tipo de funciones, realizan una búsqueda para

$$\text{optimizar } g(\mathbf{a}); \quad (3.46)$$

donde g es una función matemática dada y \mathbf{a} el vector solución. Este tipo de métodos numéricos emplean iteraciones con base en ciertos mecanismos (por ejemplo, el gradiente) que producen una sucesión de números, los cuales, se desea que converjan a una solución óptima global que minimice la función g .

El primer paso de estos métodos y el más importante es la solución inicial. A partir de la solución inicial se genera la siguiente solución posible. Sucesivamente, la solución actual se calcula con base en la anterior. Si la solución inicial está lejos del óptimo, existen altas probabilidades de que el método no converja. Este tipo de métodos son efectivos cuando se conocen de antemano las características de la función. Por ejemplo, si se sabe que la función es uni-modal y dos veces diferenciable, los métodos de gradiente son los más adecuados [27].

A continuación se muestran dos métodos de optimización que se emplean comúnmente en diversos problemas del área de Visión por Computadora [83, 4, 5].

Método de Newton

El método de Newton para optimización, es un algoritmo muy eficiente para minimizar el valor de una función g (donde g debe ser una función derivable). Se basa en la evaluación de la primera y la segunda derivada de la función g , las cuales, dan el paso y la dirección a seguir en la búsqueda de la solución óptima [84]. Este algoritmo se usa para el ajuste de puntos a una cónica por el mejor método reportado en la literatura [28]. En la figura 3.14 se muestra el mecanismo de convergencia.

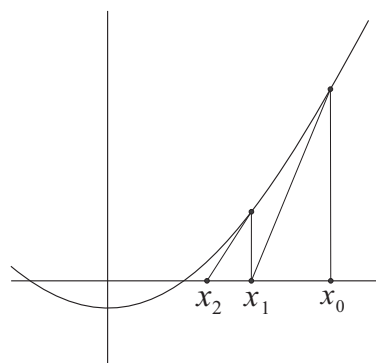


Figura 3.14: Método de Newton.

Este método es muy sencillo de programar, sin embargo, presenta diversos problemas de convergencia. Uno de los problemas más fuertes de este método es cuando el espacio de búsqueda tiene más de un óptimo local (en caso de ser sólo un óptimo local se le nombra el óptimo global). De tal forma que si no tenemos una solución inicial cerca de óptimo global, el método no converge.

Método de Levenberg-Marquadt

El método de Levenberg-Marquadt es una técnica numérica basada en el método de Newton y el método de gradiente descendente. Este método da mejores resultados que el método de Newton, porque lleva un control sobre el paso en cada iteración [85]. Esto le da una gran robustez, que le permite en muchos casos, encontrar el óptimo global, aún cuando la solución inicial está lejos de éste. No obstante, su convergencia es más lenta que el método de Newton y también es susceptible a quedar atrapado en un óptimo local. En la figura 3.15 se observa su proceso de convergencia.

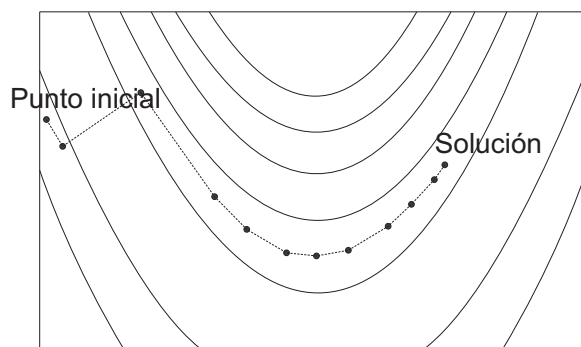


Figura 3.15: Método de Levenberg-Marquadt.

Este método es muy utilizado por diversos algoritmos del área de Visión por Computadora considerados como los mejores [4, 5] para minimizar las funciones como el cuadrado de la distancia euclidiana. Sin embargo, es necesario proporcionar una solución cercana al óptimo global para que el método converja, y en diversos problemas de esta área no es posible [49, 5]. Además, no se debe olvidar que la mayoría de estas funciones son no lineales, multimodales y sus derivadas no son continuas [4, 5, 86, 69]. Es en este tipo de funciones donde el uso de los algoritmos evolutivos resulta particularmente útil y su desempeño muestra una alta competitividad [27].

3.3.2. Computación evolutiva

La computación evolutiva (CE) engloba diversos algoritmos inspirados biológicamente con base en la teoría Neo-Darwiniana³ de la evolución natural [89]. La evolución natural es vista como un proceso de optimización, en el cual los individuos de una población gradualmente se adaptan a su ambiente.

En los algoritmos pertenecientes a la Computación Evolutiva, un individuo es una solución potencial del problema y el ambiente al que pertenece es la función o funciones objetivo y sus restricciones. Este ambiente determinará la capacidad de supervivencia del individuo.

Al igual que en los ambientes naturales, los algoritmos evolutivos trabajan con una población (varias posibles soluciones a la vez) a diferencia de los métodos iterativos, la cual proporciona a los algoritmos evolutivos la capacidad de escapar de un óptimo local más fácilmente [90].

Mientras los algoritmos evolutivos están en ejecución, ciertos operadores son aplicados a la población para obtener nuevos individuos (soluciones posibles del problema), los cuales son preservados o descartados por medio de un mecanismo de selección. El algoritmo mantiene su ejecución hasta un cierto número de iteraciones conocidas como generaciones.

Los operadores más comunes que presenta un algoritmo evolutivo son: la selección, la cruce y la mutación. La selección es un mecanismo (probabilista o determinista) que permite elegir a los individuos de la nueva generación. La cruce consiste en combinar dos o más individuos (padres) para generar uno o más nuevos individuos (hijos). La mutación permite la alteración de los nuevos individuos [90].

A continuación, se presentan los principales paradigmas de la computación evolutiva que comúnmente se usan en optimización con parámetros reales [91], dado que el problema principal que se ataca en esta tesis (reconstrucción tridimensional), es un problema de optimización de este tipo.

3.3.3. Algoritmos evolutivos empleados para la optimización con parámetros reales

Las principales heurísticas empleadas para la optimización con parámetros reales [91] son: los algoritmos genéticos, las estrategias evolutivas, la programación evolutiva, los algoritmos de optimización mediante cúmulos de partículas, los algoritmos de estimación de distribución y la evolución diferencial. Enseguida se presenta una breve descripción de cada una de ellas y sus características principales, además, se exhiben sus ventajas y desventajas al ser empleadas para la optimización con parámetros reales. En el caso de la evolución diferencial su descripción es más detallada, dado que es la heurística que se emplea en esta tesis.

3.3.4. Algoritmos genéticos

El primer algoritmo genético (originalmente llamado plan reproductivo) fue desarrollado por John Holland en 1962, como un medio para estudiar el comportamiento

³El Neo-Darwinismo es el nombre dado a la fusión de la teoría de la selección natural, con las leyes de la herencia de Mendel [87] y el seleccionismo de Weismann [88].

de adaptación y aplicarlo al aprendizaje de máquina [92]. Sin embargo, este tipo de algoritmos ha sido ampliamente utilizado como método de optimización [89, 93].

El algoritmo genético, considerado como clásico, emplea representación binaria, selección con base en la aptitud, baja probabilidad de mutación y una cruce inspirada genéticamente para producir nuevos individuos [93]. El principal operador es la cruce y el operador secundario es la mutación. Este algoritmo es conocido también como *algoritmo genético simple* (AGS) y se muestra en el algoritmo 1.

Algoritmo 1 - Algoritmo genético simple

Entrada: Información para evolucionar y los parámetros del AGS.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar a los individuos de G .
 - 3: **Repetir**
 - 4: Seleccionar padres de G .
 - 5: Aplicar la *cruza* entre padres para crear hijos.
 - 6: Aplicar la *mutación* a cada hijo.
 - 7: Evaluar a los hijos.
 - 8: La población G es sustituida por los nuevos hijos.
 - 9: **Hasta** que la condición de paro se satisfaga.
-

No obstante, los algoritmos genéticos empleados para la optimización con parámetros reales difieren del AGS. Para la optimización con parámetros reales, comúnmente se emplea representación de punto flotante (representación real) y los operadores de mutación y cruce son especiales para esta representación [94, 95]. En los algoritmos genéticos con representación real, una solución es representada directamente como un vector de parámetros reales que contiene a las variables de decisión. El algoritmo parte de una población (usualmente creada de forma aleatoria) y un conjunto de operadores genéticos (selección, cruce y mutación) y luego crea nuevas poblaciones de manera iterativa.

A pesar de que la mayoría de los algoritmos genéticos con parámetros reales difieren unos con otros, en términos de sus operadores de cruce y mutación, la mayoría sigue el modelo del algoritmo genético simple [96].

Las características para la optimización con parámetros reales son:

1. Cuando operan con representación de punto flotante, se pueden ver como una estrategia evolutiva [97, 27].
2. Para obtener mejores resultados en problemas con parámetros reales, el operador de mutación se vuelve una parte crucial del algoritmo genético [98].
3. En la mayoría de los algoritmos genéticos, la auto-adaptación se ve reflejada en el operador de cruce [99].

Las ventajas para la optimización con parámetros reales son:

1. Responden bien en funciones donde la aptitud de las soluciones es mayor a la varianza de la aptitud causada por el ruido [98].

2. Presentan un buen desempeño en problemas con funciones multi-modales [100].

Las desventajas para la optimización con parámetros reales son:

1. La velocidad de convergencia del algoritmo es lenta en comparación con las demás heurísticas [101, 99, 102].
2. Aunque existen varios estudios que sugieren ciertos parámetros y tipos de operadores para cierto tipo de problemas, el tiempo para poner a punto tales parámetros, encontrar los operadores adecuados e implementarlos, tiende a ser mucho mayor que las demás heurísticas [103, 100]. Además, es necesario tener un amplio conocimiento de los algoritmos genéticos.
3. El desarrollo de nuevos operadores para realizar la mutación o la cruce, tienden a aumentar aún más la complejidad de este tipo de algoritmos [100, 104, 102].
4. Aún con la variedad de operadores, es probable que en diversas clases de problemas no obtengan buenos resultados [105, 102].

Estrategias evolutivas

Las estrategias evolutivas (EE) fueron creadas en el año 1964 por Ingo Rechenberg y Hans-Paul Schwefel, mientras trabajaban en una aplicación concerniente a la optimización [106]. La versión original $(1 + 1) - EE$ sólo utiliza un padre para generar un hijo, al cual se le aplica la mutación. Si la aptitud del hijo es mejor a la del padre, el hijo reemplaza al padre. Posteriormente, aparecieron otras versiones como $(\mu + \lambda) - EE$ y $(\mu, \lambda) - EE$ propuestas por Hans-Paul Schwefel en [107]. En estas propuestas, μ representa el número de padres empleados para realizar la cruce y crear λ hijos. En $(\mu + \lambda) - EE$ (con el signo más), se seleccionan a los mejores μ individuos a partir de los μ padres y λ hijos, mientras que en $(\mu, \lambda) - EE$ (con el signo coma) los mejores μ individuos se seleccionan sólo de los λ hijos. El algoritmo 2 muestra cómo operan las estrategias evolutivas.

En esta técnica, no sólo los individuos evolucionan, sino también sus parámetros, como los valores de la mutación. Por tal motivo, se dice que estos algoritmos son auto-adaptables [108].

Las características de las estrategias evolutivas son:

1. Frecuentemente son empleadas para la optimización con parámetros reales [96].
2. Existe un gran énfasis en la mutación para la creación de hijos [109].
3. Comúnmente, la implementación de la mutación se realiza mediante la adición aleatoria de algún tipo de perturbación con base en la distribución gaussiana.
4. Los parámetros de la mutación están en constante cambio durante la ejecución del algoritmo y se encargan de la exploración del espacio de búsqueda [109].

Las ventajas de las estrategias evolutivas son:

Algoritmo 2 - Estrategias evolutivas

Entrada: Información para evolucionar y los parámetros de las EE.**Salida:** Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar a los individuos de G .
 - 3: **Repetir**
 - 4: Seleccionar a μ padres de G .
 - 5: Aplicar la *cruza* entre los μ padres para crear λ hijos.
 - 6: Aplicar la *mutación* a los λ hijos.
 - 7: Evaluar a los λ hijos.
 - 8: **Si** la estrategia es con signo más **entonces**
 - 9: La población G es sustituida por los μ mejores individuos seleccionados entre los μ padres y los λ hijos.
 - 10: **De lo contrario, sí** la estrategia es con signo coma **entonces**
 - 11: La población G es sustituida por los μ mejores individuos seleccionados entre los λ hijos.
 - 12: **Fin**
 - 13: **Hasta** que la condición de paro se satisfaga.
-

1. El tiempo que se emplea para poner a punto a las estrategias evolutivas es relativamente bajo, ya que la mayoría de los parámetros que emplea son auto-adaptables [103].
2. Se ha comprobado con diversos problemas con parámetros reales, que la auto-adaptación que emplea en su operador de mutación, permite tener una mejor exploración y explotación del espacio de búsqueda [99].

Las desventajas de las estrategias evolutivas son:

1. Se tienen indicios que en problemas del mundo real, el algoritmo presenta convergencia prematura [110].
2. Las nuevas versiones de las estrategias evolutivas incorporan más parámetros, los cuales incrementan la complejidad del algoritmo [108, 109, 111, 110].

Programación evolutiva

La primera versión de la programación evolutiva (PE) fue desarrollada por Lawrence Fogel en 1960 [112], donde utiliza una máquina de estados finitos como la estructura a ser evolucionada. Originalmente se empleó para simular la evolución como un proceso de aprendizaje, con el objetivo de generar inteligencia artificial [113, 112]; donde la inteligencia es vista como la capacidad de un sistema para adaptar su comportamiento y alcanzar ciertas metas.

Es difícil definir una versión estándar para la programación evolutiva [93]. No obstante, en el algoritmo 3 se muestra una versión básica de esta heurística.

Las características esenciales de la programación evolutiva son [113]:

1. Al igual que las estrategias evolutivas, existe un gran énfasis en la mutación para la creación de hijos.

3. MARCO TEÓRICO

Algoritmo 3 - Programación evolutiva

Entrada: Información para evolucionar y los parámetros de la PE.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar a los individuos de G .
 - 3: **Repetir**
 - 4: Aplicar la *mutación* a todos los individuos de G para crear a los hijos.
 - 5: Evaluar a cada uno de los hijos.
 - 6: Seleccionar (usualmente se emplea torneo estocástico) a los individuos para sustituir a los individuos de G .
 - 7: **Hasta** que la condición de paro se satisfaga.
-

2. Simulan la evolución a nivel de las especies, por tal motivo no utiliza el operador de cruza.
3. El mecanismo de selección es probabilista (usualmente emplea torneo estocástico), de tal forma que tiende a converger lentamente [93].

Las ventajas de la programación evolutiva son:

1. Sólo se debe lidiar con la mutación para poner a punto al algoritmo [112].
2. El tiempo de ejecución es relativamente bajo [114], en comparación con las demás heurísticas.
3. La auto-adaptación del operador de mutación permite tener una mejor exploración y explotación del espacio de búsqueda [99, 115].

Las desventajas de la programación evolutiva son:

1. La combinación entre el mecanismo de selección que emplea y que no realiza la cruza, parece tener un impacto negativo en su desempeño [116].
2. La convergencia del algoritmo representativo de esta heurística puede ser lenta debido a la selección suave que emplea [114].
3. Al igual que las demás heurísticas, las nuevas versiones aumentan el número de parámetros y su complejidad [114].

Optimización mediante cúmulos de partículas

La optimización mediante cúmulos de partículas (PSO por sus siglas en inglés) es una técnica de optimización poblacional, la cual fue desarrollada por James Kennedy y Russell Eberhart [117] en 1995. Forma parte de un grupo de sistemas de optimización evolutiva llamado *Inteligencia Colectiva*, inspirados en el comportamiento social.

La motivación para el desarrollo de esta heurística es la simulación simplificada del vuelo de las aves. Este algoritmo emplea una población de partículas que evoluciona con el tiempo mediante el vuelo sobre el espacio de búsqueda. Cada partícula imita a sus partículas vecinas de mayor éxito y modifica su componente de velocidad

para seguir la dirección de éstas. Como entidades sociales, cada partícula tiene una memoria limitada que mantiene la mejor posición de su propia experiencia de vuelo (el componente cognitivo) y la mejor posición alcanzada hasta ese momento, dada por las partículas vecinas (el componente social) [118]. La versión básica se presenta en el algoritmo 4.

Algoritmo 4 - Optimización mediante cúmulos de partículas

Entrada: Información para evolucionar y los parámetros del PSO.

Salida: Una población de partículas evolucionadas.

- 1: Iniciar aleatoriamente cada partícula de la población G y su velocidad.
 - 2: Evaluar todas las partículas de G
 - 3: **Repetir**
 - 4: Calcular la nueva velocidad de cada partícula, con base en su velocidad anterior, mejor posición y mejor partícula.
 - 5: Generar nuevas partículas a partir de las partículas en G y las velocidades.
 - 6: Evaluar las nuevas partículas.
 - 7: Mantener la mejor posición de cada partícula.
 - 8: Obtener la mejor partícula.
 - 9: Sustituir las partículas en G por las nuevas partículas.
 - 10: **Hasta** que la condición de paro se satisfaga.
-

Las características de la optimización mediante cúmulos de partículas son [119]:

1. No emplea operadores de cruce o mutación.
2. La búsqueda emplea operadores que alteran la velocidad de cada partícula, con base en su experiencia de vuelo y la experiencia de vuelo de las otras partículas.
3. Cada partícula contiene un vector con una posible solución (posición de la partícula en el espacio de búsqueda) y, a su vez, tiene asociada una velocidad por cada variable del vector [117].

Las ventajas de los algoritmos de optimización mediante cúmulos de partículas son:

1. El tiempo para poner a punto al algoritmo es relativamente bajo, debido a que sólo emplea tres parámetros [117].
2. El tiempo de convergencia es mucho menor, comparado con las demás heurísticas, y su desempeño es muy bueno en problemas donde el espacio de búsqueda es relativamente simple y con baja dimensionalidad [122].

Las desventajas de los algoritmos de optimización mediante cúmulos de partículas son:

1. Existen indicios de que tiene un desempeño bajo, al ser comparado con la mayoría de las heurísticas, en problemas del mundo real [101].
2. En problemas de optimización con parámetros reales, donde las funciones son de tipo uni-modal, multi-modal y, además, presentan ruido gaussiano, el desempeño de este algoritmo decrece considerablemente [98].
3. El desempeño depende de los valores iniciales de las partículas y es muy sensible al valor de sus parámetros [122, 123, 124].

Algoritmos de estimación de distribución

Los algoritmos de estimación de distribución (EDAs por sus siglas en inglés) fueron propuestos en 1996 por Heinz Mühlenbein y Gerhard Paaß [125, 126]. Estos algoritmos se destacan por no presentar los operadores de cruce y mutación. En lugar de estos operadores, utilizan el aprendizaje y la toma de muestras de la distribución de probabilidad de los mejores individuos, para llevar a cabo la evolución de la población en cada generación. Al operar de esta forma, las interrelaciones entre las variables involucradas en el dominio del problema son capturadas y explotadas explícita y eficazmente [125].

Frecuentemente, estos algoritmos inician con una población aleatoria. Posteriormente, se realizan tres operaciones sobre la población de forma iterativa. La primera operación consiste en la generación de un subconjunto de los mejores individuos de la población. En la segunda operación se realiza un proceso de aprendizaje a partir de un modelo de distribución de probabilidad de los individuos seleccionados. En la tercera operación se generan nuevos individuos mediante la simulación del modelo de distribución obtenido. El algoritmo se detiene cuando alcanza un cierto número de generaciones o cuando el rendimiento de la población deja de mejorar significativamente [127]. Esta técnica se describe en el algoritmo 5.

Algoritmo 5 - Algoritmo de estimación de distribución

Entrada: Información para evolucionar y los parámetros del EDA.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar a los individuos de G .
 - 3: **Repetir**
 - 4: Seleccionar los mejores individuos de G .
 - 5: Generar nuevos individuos por medio del modelo de distribución de probabilidad a partir de los individuos seleccionados.
 - 6: Reemplazar parte o todos los individuos de G por los nuevos individuos.
 - 7: **Hasta** que la condición de paro se satisfaga.
-

La versión original de este tipo de heurísticas no se recomienda para la optimización con parámetros reales. Es por ello que se han desarrollado algoritmos con modelos de distribución de probabilidad adecuados para la optimización con parámetros reales, como *redes gaussianas* [128].

Las características de los algoritmos de estimación de distribución son:

1. El funcionamiento de estos algoritmos está basado en un modelo probabilista de la población de individuos [125].
2. El modelo probabilista de la población representa las dependencias entre las distintas variables (atributos) del problema [125].
3. Se enfocan más en tratar de captar la estructura de las variables del problema, en lugar de encontrar la solución del problema [129].

Las ventajas de los algoritmos de estimación de distribución son:

1. Se tienen indicios de que estos algoritmos presentan un buen desempeño en funciones uni-modales y multi-modales [129, 91], aunque no superan a las estrategias evolutivas ni a la evolución diferencial.
2. Tienden a converger más rápido que un algoritmo genético, debido al modelo probabilista que emplean [125].

Las desventajas de los algoritmos de estimación de distribución son:

1. El principal problema al que se enfrentan estos algoritmos es la estimación de la distribución de probabilidad [125].
2. La complejidad de estos algoritmos, enfocados a problemas de optimización con parámetros reales, se incrementa aún más debido a los modelos probabilísticos que se utilizan [129, 127].
3. Han sido empleados muy poco en problemas del mundo real en relación a las demás heurísticas [131].

Evolución diferencial

La evolución diferencial (ED) es una heurística de búsqueda directa, propuesta por Rainer Storn y Kenneth Price en 1995 [132]. Esta heurística fue diseñada para la optimización con parámetros reales. La principal idea de la evolución diferencial es emplear la diferencia entre los valores de cada individuo para generar nuevos individuos. Una versión general de la evolución diferencial se presenta en el algoritmo 6.

Algoritmo 6 - Evolución diferencial

Entrada: Información para evolucionar y los parámetros de la ED.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar a los individuos de G .
 - 3: **Repetir**
 - 4: **Para** cada individuo \mathbf{i} de G **hacer**
 - 5: Seleccionar individuos de G de acuerdo a la variante seleccionada.
 - 6: Aplicar la *mutación* con base en la diferencia de los individuos seleccionados y obtener un individuo mutado.
 - 7: Aplicar la *cruza* entre el individuo \mathbf{i} y el individuo mutado para generar un nuevo individuo.
 - 8: Evaluar el nuevo individuo.
 - 9: Si la aptitud del nuevo individuo es mejor que la aptitud del individuo \mathbf{i} , éste es remplazado en G por el nuevo individuo.
 - 10: **Fin**
 - 11: **Hasta** que la condición de paro se satisfaga.
-

La evolución diferencial maneja un conjunto de individuos, los cuales contienen una posible solución del problema. Este conjunto de individuos, conocido como *población* G , se representa por

$$\mathbf{i}_c^r \in G^r \quad \forall c = 1, 2, \dots, o, \quad (3.47)$$

3. MARCO TEÓRICO

donde r es la generación actual y o es el número de individuos de la población G . Cada individuo contiene d parámetros reales (d variables del problema), es decir,

$$\mathbf{i}_c^r = (i_{c,1}^r, i_{c,2}^r, \dots, i_{c,d}^r) = (\mathbf{i}_{c,b}^r) \quad \forall c = 1, 2, \dots, o, b = 1, 2, \dots, d. \quad (3.48)$$

En la evolución diferencial, comúnmente, la inicialización de cada individuo se realiza de forma aleatoria en la generación inicial (generación 0), bajo los límites de cada variable, esto es,

$$i_{c,b}^0 = rand()_{c,b} \cdot (ub_b - lb_b) + lb_b \quad \forall c = 1, 2, \dots, o, b = 1, 2, \dots, d, \quad (3.49)$$

donde $rand()$ es una función que genera valores aleatorios con una distribución uniforme en el intervalo de $[0, 1]$, \mathbf{ub} y \mathbf{lb} son vectores que representan los límites superiores e inferiores de cada variable, respectivamente.

Posteriormente, la población se actualiza por medio de un esquema de reproducción en cada generación. Este esquema toma de la población, un conjunto de individuos extraídos aleatoriamente $\phi = \xi_1, \xi_2, \dots, \xi_t$ por cada individuo \mathbf{i}_c^r , necesarios para efectuar la diferencia. El resultado de esta diferencia es un nuevo individuo ρ_c^r . Si la aptitud del nuevo individuo es mejor a la aptitud del individuo \mathbf{i}_c^r , entonces ρ_c^r se agrega en la población de la siguiente generación ($r + 1$). En caso contrario, \mathbf{i}_c^r se agrega a la población de la siguiente generación.

En el algoritmo 7 se exhibe la versión *rand/1/bin* de la evolución diferencial, donde f_d es la constante de la diferencia, C_r la constante de recombinación, TG el número total de generaciones y $randint(ib, sb)$ es una función que genera números enteros aleatorios con una distribución uniforme en el intervalo de $[ib, sb]$ (ib es el entero inferior y sb el entero superior).

En este algoritmo hay tres parámetros de control (constante de la diferencia, constante de recombinación y tamaño de población). Para la constante de diferencia se recomienda utilizar valores entre $[0.3, 0.9]$, mientras que para la constante de recombinación se sugiere tomar valores extremos $[0.0, 0.3]$ o $[0.8, 0.9]$ [133]. Asimismo, el incremento en el tamaño de la población aumenta la diversidad de los posibles movimientos de la misma, promoviendo la exploración del espacio de búsqueda. Sin embargo, la probabilidad de encontrar la dirección correcta de la búsqueda se decrementa considerablemente. Por tal motivo, debe existir un balance en el tamaño de la población para encontrar la solución óptima [134].

Las diferentes estrategias usadas en la evolución diferencial son producto de la variación de los individuos extraídos, las diferencias y el individuo base. La versión mostrada en el algoritmo 7 con la nomenclatura *rand/1/bin* es la versión más popular, la cual ha mostrado mayor robustez en las funciones de prueba [132, 133] y en problemas del mundo real [101, 27]. Esta versión es empleada en este trabajo de tesis.

La nomenclatura de cada versión (*na/nb/nc*) se divide en tres partes: la primera es cómo se elige el individuo base, la segunda es el número de padres elegidos para realizar la diferencia y la tercera es el tipo de cruce.

La tabla 3.1 muestra las versiones que se tienen de la evolución diferencial, donde *rand* es una elección aleatoria, *best* es el mejor individuo, *bin* es cruce binomial, *exp* es cruce exponencial, *dir* es el operador de dirección y *ce* es una constante que va de $[0.3, 0.8]$ para la versión *curr-to-rand*.

Algoritmo 7 - Evolución diferencial *rand/1/bin*
Entrada: Información para evolucionar y los parámetros de la ED.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente a cada individuo $\mathbf{i}_c^0 \forall c = 1, 2, \dots, o$
- 2: Evaluar cada individuo $g(\mathbf{i}_c^r) \forall c = 1, 2, \dots, o$.
- 3: **Para** $1 \leq r \leq TG$ **hacer**
- 4: **Para** $1 \leq c \leq o$ **hacer**
- 5: Seleccionar tres individuos aleatoriamente: $\xi_1 \neq \xi_2 \neq \xi_3$
- 6: Obtener una posición aleatoria, $b_{\text{rand}} = \text{randint}(1, d)$
- 7: **Para** $1 \leq b \leq d$ **hacer**
- 8: **Si** $\text{rand}(0, 1) < C_r$ or $b = b_{\text{rand}}$ **entonces**
- 9: $\rho_{c,b}^r = \xi_{3,b} + f_d(\xi_{1,b} - \xi_{2,b})$
- 10: **De lo contrario,**
- 11: $\rho_{c,b}^r = i_{c,b}^r$
- 12: **Fin**
- 13: **Fin**
- 14: **Si** $g(\rho_c^r) < g(\mathbf{i}_c^r)$ **entonces**
- 15: $\mathbf{i}_c^{r+1} = \rho_c^r$
- 16: **De lo contrario,**
- 17: $\mathbf{i}_c^{r+1} = \mathbf{i}_c^r$
- 18: **Fin**
- 19: **Fin**
- 20: **Fin**

Nomenclatura	Variantes
<i>rand/1/bin</i>	$\rho_{c,b} = \begin{cases} \xi_{3,b} + f_d \cdot (\xi_{1,b} - \xi_{2,b}) & \text{si } (\text{rand}(0, 1) < C_r \text{ o } b = b_{\text{rand}}) \\ i_{c,b} & \text{de otra forma (d.o.f.)} \end{cases}$
<i>best/1/bin</i>	$\rho_{c,b} = \begin{cases} \xi_{\text{best},b} + f_d \cdot (\xi_{1,b} - \xi_{2,b}) & \text{si } (\text{rand}(0, 1) < C_r \text{ o } b = b_{\text{rand}}) \\ i_{c,b} & \text{d.o.f.} \end{cases}$
<i>rand/1/exp</i>	$\rho_{c,b} = \begin{cases} \xi_{3,b} + f_d \cdot (\xi_{1,b} - \xi_{2,b}) & \text{mientras } (\text{rand}(0, 1) < C_r \text{ o } b = b_{\text{rand}}) \\ i_{c,b} & \text{d.o.f.} \end{cases}$
<i>rand/1/dir</i>	$\rho_{c,b} = \begin{cases} \xi_{3,b} + f_d/2 \cdot (v_{c+,b} - v_{c-,b} + v_s) & \text{si } (\text{rand}(0, 1) < C_r \text{ o } b = b_{\text{rand}}) \\ i_{c,b} & \text{d.o.f.} \end{cases}$
<i>curr to rand/1</i>	$\rho_{c,b} = \begin{cases} i_{c,b} + ce \cdot (\xi_{3,b} - i_{i,b}) + f_d \cdot (\xi_{1,b} - \xi_{2,b}) & \text{si } (\text{rand}(0, 1) < C_r \text{ o } b = b_{\text{rand}}) \\ i_{c,b} & \text{d.o.f.} \end{cases}$

Tabla 3.1: Versiones de la evolución diferencial.

En general, las características de la evolución diferencial son:

1. La representación que emplea es de punto flotante y el operador de mutación está basado en la diferencia de los individuos [132], de tal forma que, éste es guiado por los individuos de la población [135]. Por ello se va auto-adaptando tras cada generación.
2. Al calcular la diferencia entre los individuos de la población, los autores de esta técnica dicen estar estimando el gradiente de esa zona [27].

Las ventajas de la evolución diferencial son:

1. Es sencilla de implementar y su complejidad es baja en relación a los demás algoritmos evolutivos [140, 101].

2. Ha sido empleada exitosamente en diversos problemas de optimización con parámetros reales [27, 132, 136, 137, 138, 98, 139, 140] y hay indicios de que es una muy buena alternativa en problemas del mundo real [101].
3. En diversas ocasiones se ha seleccionado como la heurística con mejor desempeño (en comparación con los algoritmos genéticos con representación real, estrategias evolutivas y algoritmos de optimización mediante cúmulos de partículas) en problemas del mundo real [141, 142, 122, 143, 140, 101, 27, 144], e incluso, se sugiere como primera opción cuando se tiene un nuevo problema de optimización con parámetros reales [122].
4. El número de parámetros que emplea es bajo, por lo cual, la puesta a punto es sencilla. Asimismo, existen diversos estudios en la literatura que sugieren ciertos valores iniciales para estos parámetros, los cuales se mantienen constantes en diversos problemas [122, 98, 27, 133].

Las desventajas de la evolución diferencial son:

1. Existen indicios de que su desempeño se degrada más rápido, en relación a las demás heurísticas, conforme se incrementa el ruido [98, 144].
2. Debido al tipo de selección que utiliza (selección determinista), es posible que este algoritmo presente convergencia prematura.

En el apéndice D se presenta un análisis comparativo entre estas heurísticas, con base en los resultados expuestos en la sesión especial del Congreso de Computación Evolutiva (CEC'2005) sobre optimización con parámetros reales [130], y con base en los experimentos sobre el problema principal de esta tesis.

3.3.5. Algoritmos evolutivos de optimización multi-objetivo

Los algoritmos evolutivos que se han desarrollado para resolver problemas de optimización multi-objetivo se pueden clasificar en dos grupos [145] (los algoritmos que se exhiben a continuación son los más conocidos y representativos del área):

1. Algoritmos no basados en la optimalidad de Pareto: algoritmos de muestreo independiente, algoritmos de selección agregativas y VEGA.
2. Algoritmos basados en la optimalidad de Pareto: MOGA, NPGA, NPGA2, SPEA, SPEA2, PAES, PESA, PESA-II, Micro-AG, NSGA y NSGA-II.

Enseguida se presenta una breve descripción de cada uno de estos algoritmos. El algoritmo NSGA-II se detalla más, debido a que es empleado para optimizar el problema de la triangulación (estimación del punto tridimensional).

Algoritmos no basados en la optimalidad de Pareto

Algoritmos de muestreo independiente. En estos algoritmos se usa una combinación de pesos para la función de aptitud; tales pesos se modifican de manera uniforme en varias ejecuciones del algoritmo evolutivo [146, 147, 148]. Con estos algoritmos se logran encontrar diversas soluciones no dominadas después de varias ejecuciones. Sin embargo, su principal desventaja es su elevado costo computacional.

Algoritmos de selección agregativas. En estos algoritmos se aplica un criterio de selección de acuerdo a la combinación ponderada de los objetivos, ya sea usando una función agregativa lineal o no lineal [149, 150]. Los pesos varían entre generaciones, en cada función de evaluación o en ambos. Su ventaja principal es que en una sola ejecución se pueden alcanzar varios segmentos del frente de Pareto; su principal desventaja es que en ciertos casos (cuando se usan funciones agregativas lineales), existen puntos del frente de Pareto que no pueden generarse.

VEGA (*Vector Evaluated Genetic Algorithm*). Algoritmo genético propuesto por Schaffer [151] en el que la población se divide en un número de subpoblaciones igual a la cantidad de objetivos que se desean optimizar. La idea es hacer que cada subpoblación se seleccione con base en un solo objetivo y, posteriormente, las subpoblaciones se mezclan en una sola a la que se le aplican los operadores genéticos convencionales (cruza y mutación). Este algoritmo es sencillo, pero no garantiza la generación de soluciones no dominadas.

Algoritmos basados en la optimalidad de Pareto

MOGA (*Multi Objective Genetic Algorithm*). Algoritmo genético propuesto por Fonseca y Fleming [152], el cual es una variante del algoritmo de Goldberg. En este algoritmo, el proceso de jerarquía está basado en la dominancia de Pareto, es decir, los individuos no dominados tienen la aptitud más elevada de la población y reciben el mismo valor de aptitud, mientras que los dominados son penalizados de acuerdo a la densidad de la población. Este algoritmo emplea un método de formación de nichos para distribuir a la población sobre la región óptima de Pareto.

NPGA (*Niched Pareto Genetic Algorithm*). Algoritmo genético propuesto por Horn y Nafpliotis [153], donde se aplica la dominancia de Pareto con base en la selección por torneo de toda la población. En esta técnica se seleccionan dos posibles padres y se comparan contra un subconjunto (seleccionado aleatoriamente) de la población; aquél que resulte no dominado con respecto al subconjunto es el ganador. Si existe un empate (ambos son dominados o no dominados), entonces el resultado del torneo se decide por medio de compartir la aptitud (el individuo con menos vecinos gana). El desempeño de esta propuesta puede ser negativamente afectada si el valor del tamaño del torneo es seleccionado incorrectamente, por ejemplo, si se emplea un valor muy pequeño. Una versión mejorada de este algoritmo llamada NPGA2, fue propuesta por Erickson en [154]. En esta nueva versión se aplica ordenamiento usando el concepto de Pareto y se sigue realizando selección mediante torneo, aunque se usa un esquema diferente para compartir la aptitud.

SPEA (*Strength Pareto Evolutionary Algorithm*). Algoritmo evolutivo propuesto por Zitzler y Thiele [155], en el cual se introdujo el uso de un archivo externo que se actualiza en cada generación y contiene las soluciones no dominadas encontradas previamente, para mantener un registro histórico. En este algoritmo se

calcula la *fuerza* de cada uno de sus individuos de manera similar a la jerarquización de Pareto que hace MOGA, pues depende de la cantidad de individuos a los que domina. La aptitud de los individuos de la generación actual se calcula de acuerdo a la *fuerza* de las soluciones del archivo externo que lo dominan. Para lograr una buena distribución de soluciones no dominadas, SPEA usa un método de *clustering* conocido como los *enlaces promedio*. La versión mejorada de este algoritmo se llama SPEA2, y fue propuesta en [156]. SPEA2 tiene tres diferencias principales con respecto a su versión SPEA: se asigna la aptitud con base en la cantidad de individuos que lo dominan y en los que domina, se usa una estimación de densidad de los vecinos para eficientar la búsqueda, y se usa un método de truncamiento del archivo externo para asegurar que no se pierdan las soluciones de los extremos del frente de Pareto.

PAES (*Pareto Archived Evolution Strategy*). Es una estrategia evolutiva (1+1) (significa que se tiene una población de un padre que genera un solo hijo) propuesta por Knowles y Corne [157]. Esta estrategia usa un archivo externo para retener las soluciones no dominadas obtenidas a lo largo del proceso evolutivo. Para mantener la diversidad se usa un mecanismo de *multitud* que se basa en una división recursiva del espacio de las funciones objetivo, de manera que los individuos quedan localizados en una retícula que facilita determinar y controlar la distribución de las soluciones obtenidas.

PESA (*Pareto Envelope-Based Algorithm*). Algoritmo propuesto por Corne [158], en el que se usa una población interna pequeña y una externa (o secundaria) grande. Al igual que PAES, también usa una retícula adaptativa aplicada en el espacio de las funciones objetivo para mantener la diversidad, con la diferencia de que la selección, en este caso, está basada en la medida de *multitud*. La versión mejorada de este algoritmo (PESA-II [159]), posee un mecanismo de selección basado en regiones y no a nivel de individuos, es decir, primero se selecciona una región y después se selecciona un individuo de ésta. Gracias a esto, el algoritmo reduce el costo computacional asociado con la jerarquización de Pareto.

Micro algoritmo genético (micro-AG). Algoritmo genético propuesto por Coello y Toscano [160, 161], donde la población es muy pequeña (sólo 4 individuos) y existe un proceso de re-inicialización. Este algoritmo posee dos memorias, una fija y otra reemplazable. También usa un archivo externo o histórico (similar al que se emplea en PAES) donde se retienen las mejores soluciones encontradas. El micro-AG usa tres formas de elitismo. La primera retiene las soluciones no dominadas encontradas en el ciclo interno del micro-AG. La segunda emplea una memoria reemplazable que se refresca de manera parcial a ciertos intervalos de tiempo. La tercera sustituye a los individuos de la memoria reemplazable del micro-AG con soluciones no dominadas producidas por el micro-AG.

NSGA (*Nondominated Sorting Genetic Algorithm*). Algoritmo genético propuesto por Srinivas y Deb [162], el cual es una modificación del proceso de jerarquización, propuesto originalmente por Goldberg. Este algoritmo emplea capas de clasificaciones de los individuos. Antes de realizarse la selección, la población se jerarquiza con base en la no dominancia (todos los individuos no dominados son clasificados en una categoría con un valor de aptitud *falso*). La selección de tipo proporcional restante estocástica es utilizada por este algoritmo. El NSGA fue un algoritmo relativamente exitoso durante muchos años [163, 164, 165], sin embargo, es altamente ineficiente debido a la forma de clasificar a los individuos.

Posteriormente, Deb propuso una versión mejorada de este algoritmo llamada NSGA-II [166, 29], la cual es una versión genérica de un algoritmo evolutivo multi-objetivo con bloques constructores no explícitos con base en el diseño del NSGA. Esta versión construye una población de individuos competitivos, jerarquizados y ordenados de acuerdo al nivel de no dominancia. A esta población se le aplican operadores evolutivos para crear un conjunto de hijos y estos se combinan con los padres para, posteriormente, dividir los nuevos conjuntos combinados en los frentes. Después, se lleva a cabo la generación de nichos mediante la adición de la distancia de la multitud para cada miembro. El NSGA-II emplea la distancia de la multitud en su operador de selección para mantener la diversidad del frente, esto le permite asegurar que se mantenga una distancia de multitud entre cada miembro. El uso de esta distancia es la clave de este algoritmo, ya que le permite mantener la diversidad de la población y ayuda al algoritmo a explorar el espacio de búsqueda. El NSGA-II se presenta en el algoritmo 8. Actualmente, el NSGA-II es el algoritmo que se emplea en la mayoría de las comparaciones de los algoritmos evolutivos multi-objetivo, debido a su buen desempeño y robustez [77].

Algoritmo 8 - NSGA-II

Entrada: Información para evolucionar y los parámetros del NSGA-II.

Salida: Una población evolucionada.

- 1: Iniciar aleatoriamente cada individuo de la población G .
 - 2: Evaluar las funciones objetivo con los individuos de G .
 - 3: Asignar un nivel de jerarquización con base en la dominancia de Pareto.
 - 4: Generar una población de hijos.
 - 5: Seleccionar mediante torneo binario.
 - 6: Aplicar cruza y mutación.
 - 7: **Repetir**
 - 8: **Para** cada padre e hijo en G **hacer**
 - 9: Asignar un nivel de jerarquización con base en la dominancia de Pareto.
 - 10: Generar conjuntos de vectores no dominados a lo largo del frente de Pareto conocido hasta la generación actual.
 - 11: Ciclar (internamente) mediante la adición de soluciones a la siguiente generación; se comienza con el primer frente y se termina hasta que todos los individuos encontrados determinen la distancia de multitud entre los puntos sobre cada frente.
 - 12: **Fin**
 - 13: Seleccionar puntos (elitismo) sobre el frente con el nivel de jerarquización más bajo y que estén fuera de una distancia de multitud.
 - 14: Crear la siguiente generación.
 - 15: Seleccionar mediante torneo binario.
 - 16: Aplicar cruza y mutación.
 - 17: **Hasta** que la condición de paro se satisfaga.
-

Capítulo 4

Metodología

Dentro del área de Visión por Computadora, el proceso de la reconstrucción tridimensional a partir de imágenes involucra tres pasos: la calibración de la cámara, la estimación de los parámetros de orientación y posición, y la triangulación. Los algoritmos que resuelven cada uno de estos pasos deben ser muy robustos y capaces de integrarse con los demás algoritmos de todo el proceso de la reconstrucción.

No obstante, en ciertos escenarios, estos algoritmos producen resultados no deseados. Por tal motivo, se estudiaron tales escenarios y se plantearon soluciones, las cuales se muestran en este capítulo

En principio, se presenta la triangulación que, aunque es un paso posterior a la estimación de los parámetros de la cámara, es necesaria definirla, así como conocer sus propiedades, ya que en nuestro caso, fue empleada dentro de las soluciones propuestas para realizar la reconstrucción tridimensional.

Después de definir la triangulación, se muestran las desventajas de los mejores algoritmos reportados en la literatura [5, 71] que calculan la triangulación, y la forma en que se atacó este problema para obtener mejores resultados.

Posteriormente, se exhibe el problema de la reconstrucción tridimensional, es decir, la estimación de la matriz de calibración, los parámetros de orientación y posición, y el cálculo de los puntos tridimensionales (triangulación). Asimismo, se muestran las configuraciones de las cámaras donde no es posible estimar los parámetros de la cámara con los algoritmos actuales. Finalmente, se presentan las soluciones propuestas para resolver este problema al emplear dos o más imágenes.

4.1. Triangulación

Si se tiene una correspondencia de puntos $\mathbf{x} \leftrightarrow \mathbf{x}'$ de dos imágenes y se conocen sus matrices de proyección (M, M') , es posible proyectar una línea por cada punto $(\mathbf{x}_i, \mathbf{x}'_i)$ hacia el espacio tridimensional, como se muestra en la figura 4.1. En el mejor de los casos, estas dos líneas deben intersectarse en un punto, al cual se le denomina *punto tridimensional* \mathbf{X} . A este proceso se le conoce como *triangulación* [5, 71].

Sin embargo, las imágenes adquiridas por una cámara presentan ruido, el cual se define como la diferencia o variación entre los puntos de los objetos en la imagen $(\mathbf{x}, \mathbf{x}')$ y los puntos verdaderos $(\bar{\mathbf{x}}, \bar{\mathbf{x}}')$ que se obtienen cuando no hay ruido en las imágenes [5]. Por tal motivo, es necesario emplear métodos que encuentren el mejor punto tridimensional a partir de las matrices de proyección y las correspondencias de puntos extraídas de las imágenes, y que se puedan aplicar en los tres tipos de reconstrucciones (proyectiva, afín y métrica).

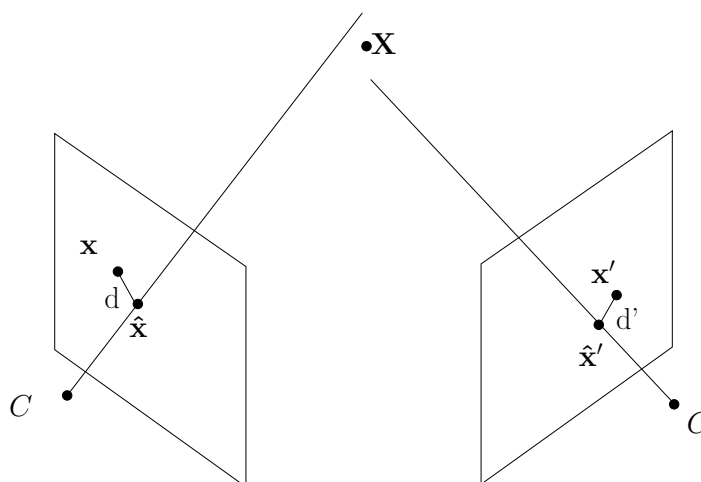


Figura 4.1: Triangulación.

4.1.1. Definición del problema

Si se tiene dos matrices de proyección M y M' , y una correspondencia de puntos $\mathbf{x} \leftrightarrow \mathbf{x}'$ con ruido, entonces, se pueden proyectar dos líneas a partir de estos puntos, pero es muy posible, que estas líneas no se unan en el espacio tridimensional. Por lo cual, no existe un punto \mathbf{X} que satisfaga exactamente la siguiente relación $\mathbf{x} = M\mathbf{X}$ y $\mathbf{x}' = M'\mathbf{X}$. Por este motivo, es necesario tener un método que estime el punto tridimensional más cercano a ambas líneas proyectadas y, además, debe ser invariante bajo los tres tipos de reconstrucciones.

El método de triangulación se expresa como una función τ , la cual calcula el punto tridimensional a partir de la correspondencia de puntos y un par de matrices de proyección mediante la siguiente expresión,

$$\mathbf{X} = \tau(\mathbf{x}, \mathbf{x}', M, M'), \quad (4.1)$$

esta función se dice ser invariante bajo cualquier transformación H si,

$$\tau(\mathbf{x}, \mathbf{x}', M, M') = H^{-1}\tau(\mathbf{x}, \mathbf{x}', MH^{-1}, M'H^{-1}). \quad (4.2)$$

Por ejemplo, al emplear el *método del punto medio*, la función no cumple que sea invariante en una reconstrucción proyectiva [71]. Debido a que los conceptos de distancia y perpendicularidad no son válidos en el contexto del espacio proyectivo y, por lo tanto, no se cumple la igualdad de la ecuación (4.2).

Para cumplir con la igualdad de la ecuación (4.2), es decir, que el método de triangulación sea invariante, se debe trabajar en un espacio donde el concepto de distancia exista y sea válido. El único espacio que cumple con esta propiedad y está implícito en la triangulación es el espacio euclidiano bidimensional, en el cual se encuentran las correspondencias de puntos de ambas imágenes. El método de triangulación que opera sobre este espacio, actúa sobre los puntos bidimensionales, los cuales no dependen del espacio al que pertenecen los puntos tridimensionales (proyectivo, afín o métrico).

La idea principal para operar en el espacio bidimensional es calcular la correspondencia de puntos bidimensionales estimados $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}'}$ para cada una de las dos imágenes, mediante la relación geométrica entre el punto tridimensional y las matri-

ces de proyección, es decir,

$$\hat{\mathbf{x}} = M\mathbf{X} \quad \hat{\mathbf{x}}' = M'\mathbf{X}.$$

Posteriormente, se minimiza la distancia entre la correspondencia de puntos de la imagen y la correspondencia de puntos estimados.

4.1.2. Solución propuesta mediante algoritmos evolutivos

Con base en la definición del problema, se requiere de una solución capaz de realizar el proceso de la triangulación a partir de puntos en la imagen y con la propiedad de que sea invariante en cualquiera de los tres tipos de reconstrucción, es decir, que se cumpla la igualdad de la ecuación (4.2) en la reconstrucción proyectiva, afín y métrica. Asimismo, tal solución debe ser capaz de dar buenos resultados aún cuando la matriz fundamental o las matrices de proyección sean calculadas a partir de correspondencias de puntos con ruido.

Los métodos que resuelven el problema de la triangulación y son invariantes ante cualquier tipo de reconstrucción son: el método polinomial-absoluto y los métodos iterativos (en particular el método de Levenberg-Marquadt [5]). El método polinomial absoluto encuentra la solución óptima cuando las matrices de proyección son generadas a partir de correspondencias de puntos sin ruido, de lo contrario, los resultados que se obtienen con este método son deplorables, como se muestra en el capítulo de resultados. Los métodos iterativos definen la función a minimizar por medio del cuadrado de la distancia euclidiana entre la correspondencia de puntos de la imagen y la correspondencia de puntos estimados, es decir,

$$g_{ce}(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2 = d(\mathbf{x}, M\mathbf{X})^2 + d(\mathbf{x}', M'\mathbf{X})^2 \quad (4.3)$$

donde $d(*, *)$ es la distancia euclidiana. Sin embargo, esta distancia no puede ser minimizada analíticamente, por lo cual, no puede ser empleada en los métodos iterativos de gradiente. En su lugar se emplea el cuadrado de la distancia euclidiana $d(*, *)^2$. Al aplicar los métodos iterativos sobre la función g_{ce} , se ha demostrado en [71], que éstos no siempre convergen a soluciones óptimas o cercanas a éstas.

Una alternativa para minimizar esta función es el uso de algoritmos evolutivos. En general, los algoritmos evolutivos son técnicas de optimización robustas, capaces de encontrar buenas soluciones, aún en espacios de búsqueda grandes, accidentados y con ruido, además están entre las menos susceptibles a quedar atrapadas en óptimos locales, debido a su naturaleza estocástica y al esquema poblacional que adoptan.

Solución propuesta para resolver el problema de la triangulación (mono-objetivo) con algoritmos evolutivos

La función de la triangulación que se deriva del cuadrado de la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados es, por naturaleza, una función mono-objetivo. En principio se propuso utilizar un algoritmo genético [93] y un algoritmo de optimización por cúmulos de partículas (llamado PSO por sus siglas en inglés) [167] para minimizar esta función. Estos algoritmos tienen parámetros que se manipulan para aumentar o disminuir: la exploración en el espacio

de búsqueda, la explotación para encontrar soluciones mínimas sobre una zona del espacio de búsqueda y la velocidad de convergencia.

Una de las bondades de estas heurísticas es que se pueden emplear para minimizar cualquier tipo de funciones, aunque éstas no sean analíticamente tratables. Por este motivo, el problema mono-objetivo de la triangulación se definió como: encontrar el punto tridimensional \mathbf{X} , tal que minimice la función g definida como,

$$g(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}'), \quad (4.4)$$

donde $d(*, *)$ es la distancia euclidiana, \mathbf{x} y \mathbf{x}' son la correspondencia de los puntos de cada imagen, $\hat{\mathbf{x}}$ y $\hat{\mathbf{x}}'$ son la estimación de los puntos bidimensionales.

Las variables a buscar por los algoritmos evolutivos (AG y PSO) de la función g , es el punto tridimensional \mathbf{X} . El punto tridimensional \mathbf{X} tiene cuatro variables (X, Y, Z, W) en coordenadas homogéneas. Sin embargo, es posible normalizar este punto para que la última entrada sea igual a uno, es decir, $[X, Y, Z, W]^T = \lambda[X/W, Y/W, Z/W, 1]^T = \lambda[X', Y', Z', 1]^T$, donde λ es un factor de escala igual a $1/W$. De tal forma que sólo se buscan tres variables (X', Y', Z') por cada punto.

Al emplear heurísticas evolutivas es necesario definir los límites del espacio de búsqueda por cada variable. Para este problema se definió que cada variable esté limitada de -1 a 1 con una precisión de 10 decimales. Los límites de las variables (X', Y', Z') se definieron de esta forma, debido a que no existe inconveniente en que el punto tridimensional esté normalizado bajo un cierto factor de escala dado por λ .

La primera heurística empleada es el algoritmo genético. Para utilizar este algoritmo, se necesita codificar cada una de las variables dentro de un cromosoma (se empleó representación binaria). El tamaño del cromosoma en bits depende del número de variables, los límites de cada variable y la precisión que se maneje. Este tamaño es calculado mediante la siguiente fórmula,

$$nb_i = \log_2((ls_i - li_i) * 10^{pr} + 1) + 1, \quad (4.5)$$

donde nb_i es el número de bits de la i -ésima variable, ls_i y li_i es el límite superior y el límite inferior de la i -ésima variable y pr es la precisión. Se le suma uno, ya que se debe agregar el 0 y al final se le suma de nuevo para que, al truncar el valor a un número entero, tome un bit más para cumplir con la precisión y el intervalo dado.

El siguiente paso es definir la función de aptitud del problema de la triangulación. Esta función se define mediante la ecuación (4.4), la cual representa la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados.

La siguiente heurística a utilizar es el algoritmo de optimización por cúmulos de partículas. En este algoritmo se definen las variables (X', Y', Z') en cada partícula directamente sin tener que pasar de una representación real a una representación binaria como en el caso del algoritmo genético. Al igual que en cualquier heurística evolutiva, es necesario definir los límites del espacio de búsqueda por cada variable. Para este algoritmo se utilizaron los mismos límites que en la heurística anterior. Asimismo, la función de aptitud se definió de igual forma que la función de aptitud empleada en el algoritmo genético.

En el siguiente capítulo se muestran las gráficas de los resultados de estas dos heurísticas evolutivas. Estas gráficas muestran el desempeño de ambas heurísticas, así como los resultados de los métodos de triangulación ya existentes. Al realizar la

comparación se observa claramente que los resultados obtenidos con las heurísticas no son satisfactorios. Tal comportamiento puede ser causado porque el espacio de búsqueda se compone de un gran número de óptimos locales y, aunque las heurísticas empleadas son muy robustas, en este caso, no fue posible encontrar el óptimo global o valores en su vecindad. Se dice entonces que, el espacio de búsqueda tiene un gran número de óptimos locales porque se observó que una modificación pequeña del estado actual en el proceso del algoritmo genético o del algoritmo de optimización por cúmulos de partículas no producen una mejor solución.

Solución propuesta para resolver el problema de la triangulación (multi-objetivo) con algoritmos evolutivos

Para remover los mínimos locales del espacio de búsqueda, Knowles [168] sugiere replantear el problema mono-objetivo como un problema multi-objetivo. Esta transformación es llamada multi-objetivización. Jensen muestra en [169], que en una aplicación donde se tiene un problema muy complejo con una función mono-objetivo, tras aplicar esta transformación, se mejoraron considerablemente los resultados.

Para realizar la transformación, es necesario separar la función objetivo original del problema por un conjunto de funciones objetivo o agregar nuevas funciones objetivo a la función original. En este caso se separó la función principal del problema de la triangulación y se planteó como dos funciones objetivo. De tal forma que el problema multi-objetivo de la triangulación se definió como: encontrar el punto tridimensional \mathbf{X} , tal que minimice las siguientes funciones:

$$g_1(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) \quad \text{y} \quad g_2(\mathbf{X}) = d(\mathbf{x}', \hat{\mathbf{x}}'), \quad (4.6)$$

donde cada una de estas funciones representa un objetivo del problema.

Dentro de la optimización evolutiva multi-objetivo (EMOO por sus siglas en inglés) se han desarrollado un gran número de algoritmos en los últimos años [159, 170, 171]. Sin embargo, uno de los algoritmos más representativos y robustos del estado del arte en esta área es el NSGA-II, el cual ha demostrado ser muy competitivo en diversos tipos de problemas multi-objetivo [29]. Por tal motivo, se empleó este algoritmo para resolver el problema de la triangulación en su planteamiento multi-objetivo. El NSGA-II utiliza representación binaria y real. Diversos autores sugieren el uso de la representación real cuando se ataca un problema con parámetros reales [90, 172, 173], por ello se empleó esta representación. La aptitud de los individuos es igual al valor de las funciones objetivo g_1 y g_2 . Los límites de cada variable son los mismos que se emplearon en los algoritmos evolutivos mono-objetivo.

Cuando se tiene un problema multi-objetivo, las posibles soluciones óptimas se encuentran en el conjunto óptimo de Pareto [29]. Estas soluciones coinciden con el óptimo del problema mono-objetivo, de tal forma que la solución al problema original se debe tomar de este conjunto de soluciones. Para elegir la solución más equitativa se emplea el frente de Pareto. El frente de Pareto se obtiene al evaluar las soluciones del conjunto óptimo de Pareto en las funciones g_1 y g_2 . La solución más equitativa es la solución con el menor valor absoluto de la diferencia de los dos objetivos en el frente de Pareto (ver figura 4.2). Al aplicar esta elección, se toman en cuenta por igual o de forma similar, el valor de los puntos de la primera y la segunda imagen.

En el capítulo siguiente se muestran los resultados obtenidos con esta solución

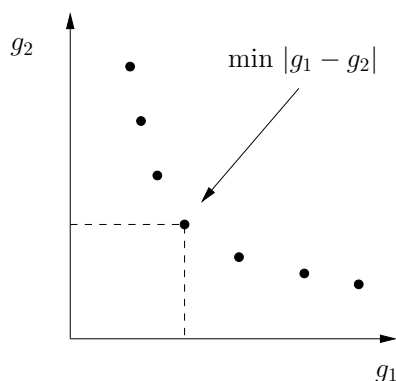


Figura 4.2: Valor seleccionado del frente de Pareto.

propuesta. En ellos se puede ver que, efectivamente, al transformar el problema mono-objetivo a multi-objetivo, se obtuvieron mejores resultados, e incluso mejor que los mejores métodos que resuelven el problema de la triangulación.

4.2. Reconstrucción tridimensional a partir de dos imágenes

En esta sección se define el problema de la reconstrucción a partir de dos imágenes, así como las configuraciones críticas donde no es posible obtener una reconstrucción tridimensional métrica por medio de la geometría epipolar y, finalmente, la solución que propusimos para evitar estas configuraciones.

4.2.1. Definición del problema

Para obtener una reconstrucción en el espacio métrico a partir de dos imágenes adquiridas por una cámara sin calibración previa, es necesario calcular las matrices de proyección. Con base en el modelo de la cámara oscura, las matrices de proyección se calculan a partir de la matriz de calibración y los parámetros de orientación y posición, es decir,

$$M = K[R | \mathbf{t}] \quad M' = K[R' | \mathbf{t}'], \quad (4.7)$$

de tal manera que el problema principal en la reconstrucción tridimensional a partir de puntos en dos imágenes es calibrar la cámara y estimar los parámetros de orientación y posición de la misma.

El punto de partida de los métodos más robustos que estiman los parámetros de la cámara es, el cálculo de la matriz fundamental, con base en la geometría epipolar, a partir de la correspondencia de puntos.

Con la matriz fundamental es posible calibrar la cámara, es decir, encontrar la matriz K ; posteriormente se calcula los parámetros de orientación y posición de la cámara. Al tener todos los parámetros de la cámara se generan las matrices de proyección y, finalmente se reconstruyen cada uno de los puntos tridimensional (triangulación) en un espacio métrico.

El proceso de calibración, como se mencionó en los capítulos anteriores, se puede realizar por distintas técnicas. La técnica más importante es el uso de las ecuaciones de

Kruppa o las formulaciones que se derivan de ellas [22, 65, 66, 25, 5, 26, 23, 67], debido a que es la única que sólo necesita conocer la correspondencia de puntos de ambas imágenes para poder operar. Estas ecuaciones y sus derivaciones actúan directamente sobre la matriz fundamental.

Configuraciones críticas

El problema principal al emplear las ecuaciones de Kruppa (o sus formulaciones derivadas), es que existen configuraciones (posiciones de la cámara) críticas, en las cuales no hay suficientes restricciones para calibrar la cámara y estimar los parámetros de orientación y posición. En los últimos años estas configuraciones críticas (también conocidas como degeneraciones) se han reducido a dos [26].

La primera configuración crítica ocurre cuando el eje óptico de la primera cámara es paralelo al eje óptico de la segunda cámara [26]. Esta configuración se da cuando, tras haber tomado la primera imagen, la cámara se desplaza a través de una línea recta y se toma la segunda imagen; es decir, sólo hay una traslación entre los puntos de la primera imagen y los puntos de la segunda imagen. En este caso, el epipolo e , el cual es la proyección del centro óptico C de la primera imagen en la segunda imagen, está en el infinito. De igual forma sucede con el epipolo e' en la primera imagen. Esta degeneración se muestra en la figura 4.3.

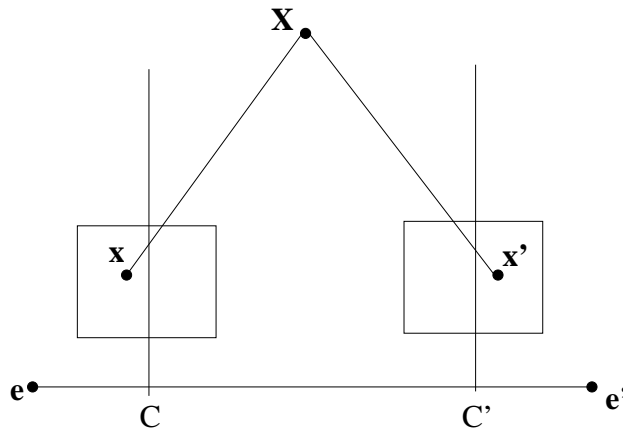


Figura 4.3: Configuración crítica I: ejes ópticos paralelos.

Cuando se calcula la matriz fundamental en una configuración no crítica, ésta se puede graficar y visualizar como una cónica no degenerada, es decir, un círculo, una elipse, una parábola o una hipérbola, como se muestra en la figura 4.4.

Sin embargo, cuando se está en la primera degeneración y se calcula la matriz fundamental, ésta se visualiza como una cónica degenerada. Esta cónica se muestra en la figura 4.5.

La segunda configuración crítica o degeneración ocurre cuando los ejes ópticos se intersecan en un punto tridimensional finito y los centros ópticos son equidistantes a este punto tridimensional [26]. En otras palabras, la distancia del centro óptico de la primera cámara C al punto donde se intersecan los ejes ópticos, es igual a la distancia del centro óptico de la segunda cámara C' al punto donde se intersecan. Esta configuración crítica se puede ver como una rotación pura, donde no existe una traslación. En la figura 4.6 se muestra la segunda configuración crítica.

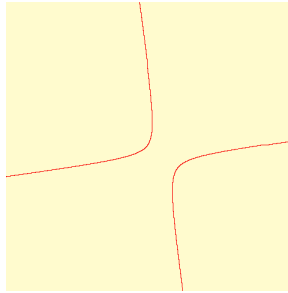


Figura 4.4: Gráfica de la matriz fundamental no degenerada.

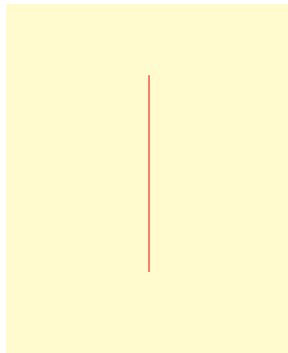


Figura 4.5: Gráfica de la matriz fundamental en la primera degeneración.

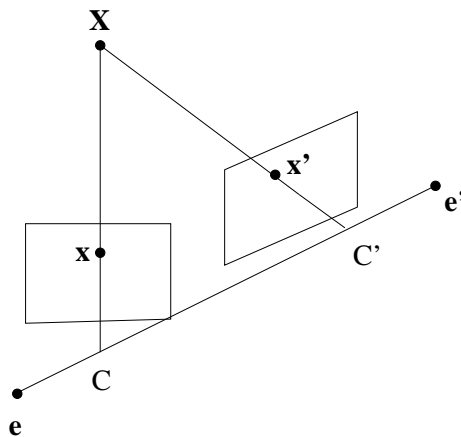


Figura 4.6: Configuración crítica II: centros ópticos equidistantes al punto donde se intersecan los ejes ópticos.

Al calcular la matriz fundamental en esta degeneración, ésta se visualiza como una cónica degenerada, pero diferente a la cónica de la primera degeneración. Esta cónica se puede ver en la figura 4.7.

4.2.2. Solución propuesta mediante algoritmos evolutivos

Para evitar caer en las configuraciones críticas se propuso estimar directamente la matriz de calibración y los parámetros de orientación y posición de la cámara para cada una de las vistas mediante el modelo de la cámara oscura. Con este modelo es posible estimar puntos en ambas imágenes conocidos como puntos bidimensionales

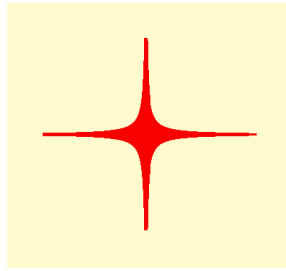


Figura 4.7: Gráfica de la matriz fundamental en la segunda degeneración.

estimados. Al variar los parámetros de la cámara y los puntos tridimensionales se pretende encontrar aquellos valores que generen los puntos bidimensionales estimados más cercanos a los puntos de la imagen. Este proceso se muestra en la figura 4.8.

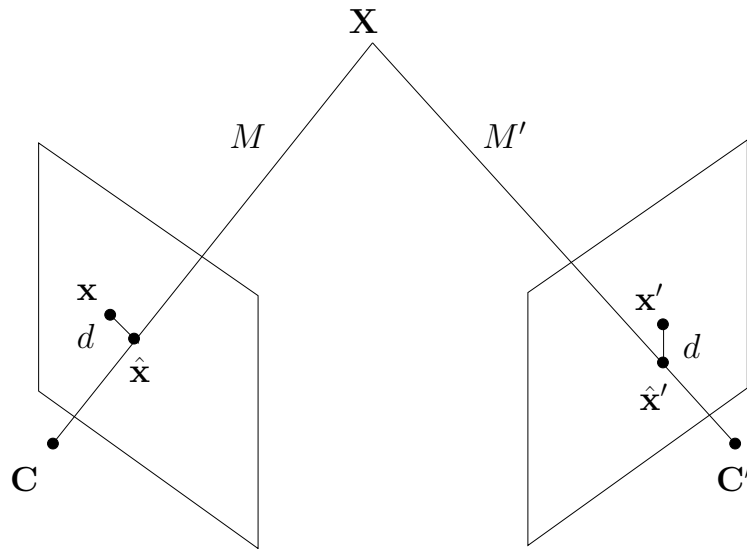


Figura 4.8: Minimización de la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados.

De tal forma que la estimación de los parámetros de la cámara se logra a partir de minimizar la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados [5], es decir, minimizar la siguiente función,

$$g(M, M', \mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}') = d(\mathbf{x}, M\mathbf{X}) + d(\mathbf{x}', M'\mathbf{X}) \quad (4.8)$$

donde $d(*, *)$ es la distancia euclidiana, \mathbf{x} y \mathbf{x}' son la correspondencia de puntos en ambas imágenes, $\hat{\mathbf{x}}$ y $\hat{\mathbf{x}}'$ son la correspondencia de puntos bidimensionales estimados, M y M' son las matrices de proyección de la primera y segunda imagen, respectivamente, y \mathbf{X} es el punto tridimensional.

Los parámetros que se desean encontrar para generar las dos matrices de proyección son: tres parámetros de orientación y tres parámetros de posición por cada matriz de proyección, y cinco parámetros de la matriz de calibración.

Asimismo, con las cámaras fotográficas o las cámaras de video que se tienen en la actualidad es posible asumir que los píxeles son cuadrados (como se mencionó en el capítulo anterior). Por tal motivo, los factores de escala tiene el mismo valor y el

parámetro de oblicuidad es igual con cero. Además, es posible suponer que el punto principal está en el centro de la imagen [66, 26, 23]. De esta manera, el problema se reduce a encontrar una sola variable de la matriz de calibración (longitud del foco).

Al final, el número total de variables a estimar del problema es de: una variable de la matriz de calibración (f), tres variables de orientación de la primera matriz de proyección (α , β y γ), tres variables de orientación de la segunda matriz de proyección (α' , β' y γ'), tres variables de posición de la primera matriz de proyección (t_x , t_y y t_z), tres variables de posición de la segunda matriz de proyección (t'_x , t'_y y t'_z), y $4 \times n$ variables de los n puntos tridimensionales, donde cada punto tridimensional se expresa como $\mathbf{X} = [X, Y, Z, W]^T$.

La función g a minimizar, definida en la expresión (4.8), se evalúa en términos de las variables a estimar de la siguiente forma:

$$\begin{aligned} g(M, M', \mathbf{X}) &= d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}') \\ &= \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} + \sqrt{(x' - \hat{x}')^2 + (y' - \hat{y}')^2} \end{aligned}$$

donde (x, y) y (x', y') son los puntos de la primera y segunda imagen, respectivamente, (\hat{x}, \hat{y}) y (\hat{x}', \hat{y}') son los puntos bidimensionales estimados para la primera y segunda imagen, respectivamente. Cada punto estimado de la primera imagen se obtiene mediante la expresión $\hat{\mathbf{x}} = M\mathbf{X}^1$, es decir,

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = \frac{1}{\hat{w}} \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix}, \quad \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix},$$

de tal forma que el valor de \hat{x} y \hat{y} queda expresado como,

$$\hat{x} = \frac{Xm_{11} + Ym_{12} + Zm_{13} + Wm_{14}}{Xm_{31} + Ym_{32} + Zm_{33} + Wm_{34}}, \quad (4.9)$$

y,

$$\hat{y} = \frac{Xm_{21} + Ym_{22} + Zm_{23} + Wm_{24}}{Xm_{31} + Ym_{32} + Zm_{33} + Wm_{34}}, \quad (4.10)$$

donde m_{ij} es el valor del i -ésimo renglón y la j -ésima columna de la matriz M . Al sustituir los parámetros de la cámara en la matriz M se obtiene la siguiente expresión,

$$\begin{aligned} M &= \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \\ &= \begin{bmatrix} fr_{11} + o_x r_{31} & fr_{12} + o_x r_{32} & fr_{13} + o_x r_{33} & ft_x + o_x t_z \\ fr_{21} + o_y r_{31} & fr_{22} + o_y r_{32} & fr_{23} + o_y r_{33} & ft_y + o_y t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}. \end{aligned}$$

¹Es claro que el cálculo de cada punto estimado de la segunda imagen emplea la misma expresión pero con su propia matriz de proyección M' .

4. METODOLOGÍA

Si se sustituyen estos valores en las ecuaciones (4.9) y (4.10) se generan las siguientes ecuaciones,

$$\begin{aligned}\hat{x} &= \frac{X(fr_{11} + o_x r_{31}) + Y(fr_{12} + o_x r_{32}) + Z(fr_{13} + o_x r_{33}) + W(ft_x + o_x t_z)}{Xr_{31} + Yr_{32} + Zr_{33} + Wt_z}, \\ \hat{y} &= \frac{X(fr_{21} + o_y r_{31}) + Y(fr_{22} + o_y r_{32}) + Z(fr_{23} + o_y r_{33}) + W(ft_y + o_y t_z)}{Xr_{31} + Yr_{32} + Zr_{33} + Wt_z},\end{aligned}$$

donde r_{ij} es el valor del i -ésimo renglón y la j -ésima columna de la matriz de rotación. La matriz de rotación R se representa mediante la convención de los ángulos de balanceo, inclinación y orientación [174, 175] de la siguiente forma,

$$\begin{aligned}R &= R_Z(\gamma)R_Y(\beta)R_X(\alpha) \\ &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \cos \beta \sin \alpha & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha \\ -\sin \beta & \sin \gamma \cos \beta & \cos \gamma \cos \beta \end{bmatrix},\end{aligned}$$

la cual representa la rotación de cada eje coordenado, comenzando por el eje X y terminando en el eje Z .

Ya que se ha mostrado como se evalúa la función g mediante las variables a estimar, es necesario definir esta función como la función objetivo del problema. Para lograrlo, se debe contemplar las n correspondencias de puntos de ambas imágenes, de tal forma que la función objetivo se define como:

$$g_{obj}(M, M', \mathbf{X}_i) = \frac{1}{n} \sum_{i=1}^n \left(d(\mathbf{x}_i, \hat{\mathbf{x}}_i) + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i) \right) \quad (4.11)$$

Se ha demostrado que el espacio de búsqueda de esta función, altamente no lineal [176] y multimodal [177], presenta una gran cantidad de óptimos locales [178, 179, 180, 176]. En consecuencia, los métodos iterativos para optimización de funciones no lineales necesitan de un valor inicial cercano a la solución óptima para converger, de lo contrario se quedan atrapados en óptimos locales [19, 178, 5, 179]. Por tal motivo, es necesario emplear un algoritmo de optimización que tenga un mecanismo de exploración tan eficaz que sea muy difícil que quede atrapado en óptimos locales. Los algoritmos que cumplen con lo anterior son los algoritmos evolutivos, además, tienen la capacidad de operar en dimensiones como las que presenta este problema.

En el Congreso de Computación Evolutiva de 2005 hubo una sesión especial donde se pusieron a prueba los diferentes algoritmos evolutivos para la optimización de funciones con parámetros reales sobre un conjunto de 25 funciones [130]. En la sección D.1 del apéndice D se presentan los algoritmos evolutivos que se pusieron a prueba y un análisis comparativo con base a los resultados presentados en esta sesión. De acuerdo a este análisis, los mejores algoritmos para funciones con diez variables son: las estrategias evolutivas (en su versión $G-CMA-ES$ [111]) y la evolución diferencial (en su

versión *rand/1/bin* [140]).

No obstante, el problema de la reconstrucción tridimensional tiene 13 variables que pertenecen a los parámetros de la cámara y $4 \times n$ variables de los n puntos tridimensionales. En la segunda solución propuesta para resolver este problema, la cual se presenta más adelante, el número de variables se reduce a 13 (calcular solamente los parámetros de la cámara). Sin embargo, en algunos experimentos se añadieron las variables o_x y o_y , las cuales pertenecen al punto principal. Es por ello que el número máximo de variables, para el problema de la reconstrucción tridimensional que se presenta en esta tesis, es de 15.

Con base en el número máximo de variables, se generó un experimento donde se pusieron a prueba las estrategias evolutivas y la evolución diferencial. En la sección D.2 del apéndice D se muestra la configuración de este experimento y los resultados obtenidos por ambos algoritmos. Estos resultados muestran que ambos algoritmos tienen un buen desempeño, aunque la evolución diferencial es ligeramente mejor que las estrategias evolutivas. Sin embargo, la elección entre ambos algoritmos fue marcada por su complejidad. Las estrategias evolutivas presentan una complejidad 5.2 veces mayor a la complejidad de la evolución diferencial en el caso de 90,000 evaluaciones de la función objetivo, y ésta se incrementa cuando se realiza un número mayor de evaluaciones. Es por ello que la evolución diferencial es una muy buena opción para aplicarse en este problema. De hecho se sugiere como primera opción cuando se tiene un problema de optimización con parámetros reales [122].

El siguiente paso es definir los límites de cada variable. En principio, la longitud del foco se estableció dentro del intervalo de $[10, 1000]$, debido a que los artículos consultados hasta ese momento [66, 4, 5, 26] encontraban el valor del foco entre $[300, 900]$. Posteriormente, este intervalo se incrementó a $[10, 1500]$, ya que se encontró en [24, 178, 69] que el foco puede alcanzar estos valores. En cuanto a los ángulos de rotación, las cámaras pueden tener una rotación, en su caso extremo, de 180 grados al tomar cada imagen ², por ello el intervalo se estableció de $[0, \pi]$. Para el vector de traslación y las variables de los puntos tridimensionales, es posible normalizar sus valores, de tal forma que queden dentro de un cubo de 2 unidades de tamaño, es decir, dentro del intervalo de $[-1, 1]$.

La primera solución propuesta (SPI) para atacar este problema, estima los trece parámetros de la cámara más las $4 \times n$ variables de los puntos tridimensionales mediante la minimización de la función g_{obj} (expresada por la ecuación (4.11)). Para minimizar esta función se empleó la evolución diferencial. La evaluación de la función objetivo para cada individuo se muestra en el algoritmo 9.

Al ejecutar la SPI, se observó que se necesitan en promedio de 450,000 a 500,000 generaciones para obtener una solución cercana al óptimo o la solución óptima; este comportamiento se muestra en las gráficas de convergencia del capítulo de resultados. Al necesitar de un número tan alto de generaciones, el tiempo de ejecución también es demasiado alto. Para disminuir el número de generaciones, se propuso la solución II (SPII), la cual en lugar de estimar los puntos tridimensionales mediante la evolución diferencial, calcula tales puntos con un método de triangulación. Con la SPII, además de reducir el número de generaciones, como se muestra en los resultados, también se reduce el número de variables a 13, las cuales pertenecen solamente a los parámetros de la cámara de ambas vistas.

²En el caso extremo no sería posible obtener las correspondencias de puntos.

Algoritmo 9 - SPI, evaluación de un individuo

Entrada: Un individuo \mathbf{i}_k^g .

Salida: La evaluación de \mathbf{i}_k^g mediante la función objetivo g_{obj} expresada en (4.11).

- 1: Extraer de \mathbf{i}_k^g el parámetro intrínseco (f) de la cámara.
 - 2: Extraer de \mathbf{i}_k^g los parámetros de orientación (θ, β, γ) y posición (t_x, t_y, t_z) para la primera vista.
 - 3: Extraer de \mathbf{i}_k^g los parámetros de orientación (θ', β', γ') y posición (t'_x, t'_y, t'_z) para la segunda vista.
 - 4: Extraer de \mathbf{i}_k^g las $4 \times n$ variables de los puntos tridimensionales.
 - 5: Generar las dos matrices de proyección $M = K[R \mid \mathbf{t}]$ y $M' = K[R' \mid \mathbf{t}']$.
 - 6: **Para** $i = 1$ hasta n **hacer**
 - 7: Calcular los puntos bidimensionales estimados ($\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$) a partir de las matrices de proyección y el i -ésimo punto tridimensional extraído de \mathbf{i}_k^g .
 - 8: Evaluar la función g mediante la ecuación (4.8).
 - 9: **Fin**
 - 10: Regresar la media de las n evaluaciones de la función g , es decir, la evaluación de la función objetivo g_{obj} .
-

Existen diversos métodos para realizar el proceso de la triangulación [71, 181] a partir de las correspondencias de puntos (como se muestra en el segundo capítulo). Sin embargo, los métodos que dan buenos resultados y, además, presentan un tiempo de ejecución bajo, son los métodos lineales [71, 5].

Estos métodos se derivan de las ecuaciones que relacionan el punto tridimensional en el mundo con el punto bidimensional en la imagen, con base en el modelo de la cámara oscura, es decir, $\hat{\mathbf{x}}_i = M\mathbf{X}_i$ y $\hat{\mathbf{x}}'_i = M'\mathbf{X}_i$.

Por cada vista se generan dos ecuaciones linealmente independientes. Estas ecuaciones se combinan para formar un sistema de ecuaciones lineales homogéneo $A\mathbf{X} = \mathbf{0}$, donde la matriz A está representada como,

$$A = \begin{pmatrix} \hat{x}\mathbf{m}^{3T} - \mathbf{m}^{1T} \\ \hat{y}\mathbf{m}^{3T} - \mathbf{m}^{2T} \\ \hat{x}'\mathbf{m}'^{3T} - \mathbf{m}'^{1T} \\ \hat{y}'\mathbf{m}'^{3T} - \mathbf{m}'^{2T} \end{pmatrix}, \quad (4.12)$$

donde \mathbf{m}^{iT} and \mathbf{m}'^{iT} son los i -ésimos renglones de las matrices de proyección M y M' , respectivamente. El sistema de ecuaciones lineales homogéneo se puede resolver mediante la descomposición en valores singulares (DVS) de la matriz A [71, 5].

Al utilizar la SPII para estimar los parámetros de la cámara y reconstruir los puntos tridimensionales, el número de generaciones se redujo, en promedio, a 5,000 y 10,000 generaciones. No obstante, aunque se redujo el tiempo de ejecución, éste aún sigue siendo alto debido al cálculo de la DVS. Para reducir aún más el tiempo de ejecución, se puede transformar el sistema lineal homogéneo en un sistema lineal no homogéneo al emplear el punto tridimensional como $\mathbf{X} = [X, Y, Z, 1]^T$. De esta forma, se tienen cuatro ecuaciones con tres incógnitas $A\mathbf{X} = \mathbf{b}$.

Para resolver este sistema de ecuaciones, se usó la relación $A^{-1}A\mathbf{X} = A^{-1}\mathbf{b}$, que es el método de las ecuaciones normales, equivalente a encontrar una estimación por mínimos cuadrados; este sistema de ecuaciones lineales se resolvió explícitamente con

la regla de Cramer [182].

Al utilizar este método se reduce el tiempo de ejecución y se obtienen los mismos resultados con el mismo número de generaciones. La evaluación de la función objetivo sobre cada individuo se muestra en el algoritmo 10.

Algoritmo 10 - SPII, evaluación de un individuo

Entrada: Un individuo \mathbf{i}_k^g .

Salida: La evaluación de \mathbf{i}_k^g mediante la función objetivo g_{obj} expresada en (4.11).

- 1: Extraer de \mathbf{i}_k^g el parámetros intrínseco (f) de la cámara.
 - 2: Extraer de \mathbf{i}_k^g los parámetros de orientación (θ, β, γ) y posición (t_x, t_y, t_z) para la primera vista.
 - 3: Extraer de \mathbf{i}_k^g los parámetros de orientación (θ', β', γ') y posición (t'_x, t'_y, t'_z) para la segunda vista.
 - 4: Generar las dos matrices de proyección $M = K[R | \mathbf{t}]$, $M' = K[R' | \mathbf{t}']$.
 - 5: **Para** $i = 1$ hasta n **hacer**
 - 6: Estimar el punto tridimensional con el método de ecuaciones normales.
 - 7: Calcular los puntos estimados ($\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$) a partir de las matrices de proyección y el i -ésimo punto tridimensional estimado.
 - 8: Evaluar la función g mediante la ecuación (4.8).
 - 9: **Fin**
 - 10: Regresar la media de las n evaluaciones de la función g , es decir, la evaluación de la función objetivo g_{obj} .
-

En el siguiente capítulo se muestran los resultados de la SPII, al ser aplicada tanto en las configuraciones no críticas como en las críticas. Los resultados muestran que la SPII es capaz de encontrar reconstrucciones correctas en estas configuraciones.

4.3. Reconstrucción tridimensional a partir de tres o más imágenes

En ocasiones, es posible contar con tres o más imágenes para obtener la reconstrucción tridimensional de un objeto o de una escena. Por lo cual, es necesario contar con una solución que sea capaz de operar con tres o más imágenes, las cuales deben ser adquiridas por una misma cámara.

4.3.1. Definición del problema

Cuando se cuenta con tres o más imágenes adquiridas por la misma cámara y se requiere realizar una reconstrucción tridimensional, es necesario calibrar la cámara y estimar los parámetros de orientación y posición de la misma por cada imagen, con la finalidad de generar cada una de las matrices de proyección,

$$M_j = K[R_j | \mathbf{t}_j] \quad \forall 3 \leq j \leq m, \quad (4.13)$$

donde m es el número total de imágenes, y calcular los puntos tridimensionales mediante las matrices de proyección y las correspondencias de puntos.

El problema parte de n correspondencias de puntos, las cuales se generan a partir de puntos en las imágenes. Sin embargo, ningún punto es visible en todas las imágenes (si se adquirieron imágenes que cubran todo el objeto), por lo cual se debe conocer a qué imágenes pertenece, para obtener el punto tridimensional.

4.3.2. Solución propuesta mediante algoritmos evolutivos

Para resolver el problema de la reconstrucción tridimensional a partir de tres o más imágenes se propuso la solución III (SPIII), la cual es una extensión de la SPII y emplea también la evolución diferencial. Esta solución se divide en dos fases: la primera se encarga de calibrar la cámara, estimar los parámetros de orientación y posición de las dos vistas, y calcular los puntos tridimensionales correspondientes a tales vistas a partir de las primeras dos imágenes; en la segunda fase se estiman los parámetros de orientación y posición de las vistas restantes, así como los puntos tridimensionales correspondientes a estas vistas. Cabe señalar que todas las correspondencias de puntos cuentan con un identificador para saber a qué vistas pertenecen. En el algoritmo 11 se muestra la SPIII.

Algoritmo 11 - SPIII

Entrada: Las n correspondencias de puntos de las m imágenes.

Salida: Los n puntos tridimensionales, la matriz de calibración y los parámetros de orientación y posición de cada una de las vistas.

- 1: Primera fase: estimar la matriz de calibración, los parámetros de orientación y posición, y los r puntos tridimensionales de las primeras dos vistas mediante la SPII. Las r correspondencias de puntos se toman de las n correspondencias de puntos con la restricción de que pertenezcan a la primera y a la segunda imagen.
- 2: **Para** $j = 3$ hasta m **hacer**
- 3: Segunda fase: estimar los parámetros de orientación y posición de la j -ésima vista y los puntos tridimensionales restantes que pertenecen a la vista anterior ($j - 1$) y a la vista actual (j) los cuales no han sido estimados previamente. Para calcular estos parámetros y los puntos tridimensionales se emplea el algoritmo 12 llamado vistas extras.

4: **Fin**

La primera fase emplea la SPII y sólo actúa sobre r correspondencias de puntos que pertenecen a la primera y a la segunda imagen, las cuales se pueden elegir arbitrariamente, siempre y cuando existan correspondencias de puntos entre ellas.

La segunda fase se repite hasta que se hayan empleado las m vistas. En cada iteración se almacenan los puntos tridimensionales obtenidos y los parámetros de orientación y posición de la vista actual. Posteriormente, estos valores se emplean en la siguiente iteración como entrada del algoritmo, catalogados como valores de la vista anterior. Al menos deben existir tres vistas que compartan cierto número de correspondencias de puntos (en los experimentos fueron necesarios al menos 8 correspondencias de puntos). En el algoritmo 12 se muestra la segunda fase de la SPIII, con el nombre de vistas extras. Este algoritmo también emplea la evolución

diferencial y evalúa a cada individuo mediante la siguiente función,

$$g_{ve}(R_j, \mathbf{t}_j, \mathbf{X}_i^j, s) = \frac{1}{q} \sum_{i=1}^q \left(d(\mathbf{x}_i^{j-1}, \hat{\mathbf{x}}_i^{j-1}) + d(\mathbf{x}_i^j, \hat{\mathbf{x}}_i^j) \right), \quad (4.14)$$

donde s es un factor de escala que también se estima y es necesario para lograr la convergencia del algoritmo en la segunda fase.

Los pasos para realizar esta evaluación se muestran en el algoritmo 13. En el siguiente capítulo en la Sec. 5.3, (en la pág. 85) se muestra el desempeño de la SPIII al reconstruir dos objetos diferentes a partir de cuatro imágenes.

Algoritmo 12 - Vistas extras

Entrada: K , R_{j-1} , \mathbf{t}_{j-1} , las c correspondencias de puntos pertenecientes a la vista anterior ($j - 1$) y a la vista actual (j), y los q puntos tridimensionales calculados previamente que pertenecen a ambas vistas.

Salida: Los parámetros de orientación y posición de la vista actual y los c puntos tridimensionales que pertenecen a ambas vistas.

- 1: Inicializar la población.
 - 2: Evaluar cada uno de los p individuos de la población mediante el algoritmo 13.
 - 3: **Para** $1 \leq g \leq$ número de generaciones **hacer**
 - 4: **Para** $1 \leq k \leq p$ **hacer**
 - 5: Ejecutar los pasos propios de la evolución diferencial y efectuar la evaluación de \mathbf{i}_k^g mediante el algoritmo 13.
 - 6: **Fin**
 - 7: **Fin**
 - 8: Extraer los parámetros de orientación y posición de la vista actual del mejor individuo de la población.
 - 9: Estimar los $c - q$ puntos tridimensionales que pertenecen a ambas vistas, los cuales no han sido estimados previamente.
-

Algoritmo 13 - Evaluación del individuo para vistas extras

Entrada: Un individuo \mathbf{i}_k^g , K , R_{j-1} y \mathbf{t}_{j-1} .

Salida: La evaluación de \mathbf{i}_k^g mediante la función g_{ev} expresada en (4.14).

- 1: Extraer de \mathbf{i}_k^g los parámetros de orientación $(\theta_j, \beta_j, \gamma_j)$ y posición (t_{xj}, t_{yj}, t_{zj}) pertenecientes a la j -ésima vista.
 - 2: Extraer de \mathbf{i}_k^g el factor de escala s .
 - 3: Generar las dos matrices de proyección $M_{j-1} = K[R_{j-1}|\mathbf{t}_{j-1}]$ y $M_j = K[R_j|s\mathbf{t}_j]$.
 - 4: **Para** $i = 1$ hasta q **hacer**
 - 5: Calcular los puntos estimados a partir de las matrices de proyección y el i -ésimo punto tridimensional, el cual es estimado en la fase o iteración anterior.
 - 6: Evaluar la función g mediante la ecuación (4.8).
 - 7: **Fin**
 - 8: Regresar la media de las q evaluaciones de la función g , es decir, la evaluación de la función objetivo g_{ve} .
-

Para demostrar el desempeño de las soluciones propuestas es necesario ponerlas a prueba bajo diversos escenarios. En principio, se muestran los experimentos del problema de la triangulación, donde se emplearon puntos con ruido para observar su desempeño y, posteriormente, se presentan los experimentos del problema de la reconstrucción tridimensional. En estos experimentos se emplearon correspondencias de puntos con cierto nivel de ruido, a partir de imágenes simuladas y reales, en configuraciones y superficies críticas y no críticas.

5.1. Triangulación

El problema de la triangulación se atacó de dos formas distintas: como un problema de optimización mono-objetivo y como un problema de optimización multi-objetivo. En ambas pruebas se tomaron los mismos escenarios. Sin embargo, se observa una gran diferencia en los resultados, inclinándose la balanza hacia la solución del problema mediante optimización multi-objetivo. A continuación se muestran los resultados y la comparación con los mejores métodos de triangulación publicados.

5.1.1. Solución propuesta aplicada al problema de triangulación mono-objetivo

Para realizar las pruebas se utilizó un conjunto de correspondencias de puntos bidimensionales que fueron generadas de forma sintética, a partir de proyectar un poliedro en dos imágenes con sus respectivas matrices de proyección arbitrarias.

El conjunto se compone de 24 correspondencias de puntos, de las cuales se tomaron aleatoriamente 8, 12, 16 y 20 correspondencias de puntos. Posteriormente, se le añadió ruido gaussiano con base en la raíz cuadrada media (RMS por sus siglas en inglés) con media cero y desviación estándar k (en píxeles), para k desde 1 hasta 8. Esto permite generar imágenes sintéticas con diferentes niveles de ruido. Este tipo de ruido comúnmente se emplea para modelar el ruido presente en las imágenes y se usa para observar el desempeño de los algoritmos en el área de visión por computadora [5].

Para obtener las matrices de proyección, se parte del cálculo de la matriz fundamental y el uso de un método de optimización (Levenberg-Marquadt), que minimiza el cuadrado de la distancia euclidiana entre las líneas epipolares y los puntos en ambas imágenes [5]. Este algoritmo tiene como entrada la correspondencia de puntos y a la salida la matriz fundamental, de la cual se desprenden las matrices de proyección. Es-

tas matrices de proyección están bajo una transformación proyectiva. Por tal motivo, los puntos tridimensionales obtenidos están en un espacio proyectivo.

Para resolver este problema de optimización mono-objetivo se emplearon dos algoritmos evolutivos: un algoritmo genético y el algoritmo de optimización por cúmulos de partículas. Para el algoritmo genético se implementaron las siguientes variantes:

- Tipo de selección: torneo binario, ruleta, sobrante estocástico.
- Tipo de cruza: un punto, dos puntos.

Con los siguientes parámetros:

- Población: 50, 100 y 200 individuos.
- Número de generaciones: 10,000, 20,000 y 50,000.
- Porcentaje de mutación: 0.1, $1/L$, $2/L$, donde L es la longitud de la cadena binaria.

Por cada variante y combinación de parámetros se realizaron 30 ejecuciones, de las cuales se calculó la media con base en la función de aptitud.

En el caso del algoritmo de optimización por cúmulos de partículas, también se realizaron 30 ejecuciones y se utilizaron los siguientes parámetros:

- Población: 50, 100 y 200 individuos.
- Número de generaciones: 100,000, 500,000 y 1,000,000.
- $C1 = 1.4962$.
- $C2 = 1.4962$.
- $W = 0.7298$.

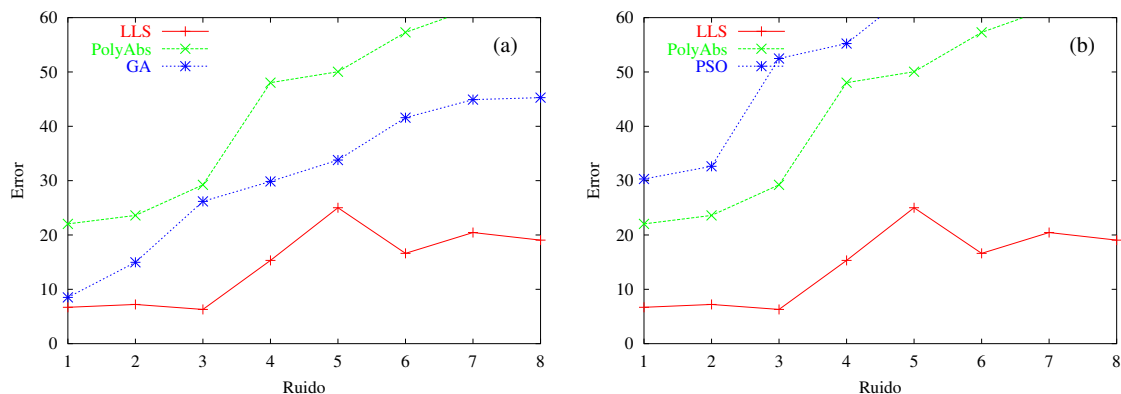


Figura 5.1: Comparativa de los métodos LLS, Poly-Abs, algoritmo genético (GA) y algoritmo de optimización por cúmulos de partículas (PSO) para el problema de triangulación bajo una reconstrucción proyectiva.

Con la finalidad de realizar una comparativa más visible, en la figura 5.1 se muestran las gráficas de los resultados de ambos algoritmos (media de la función objetivo): el algoritmo genético (GA) y el de optimización por cúmulos de partículas (PSO). Asimismo, se presentan los resultados del método lineal de mínimos cuadrados (LLS) y del método polinomial-absoluto (Poly-Abs) conocidos como los mejores métodos de triangulación [5, 71]. En las gráficas de la figura 5.1 se observa que los resultados obtenidos no fueron satisfactorios para ninguno de los dos algoritmos evolutivos. Es muy probable que el espacio de búsqueda esté compuesto por un gran número de óptimos locales, de tal forma que estas dos heurísticas quedan atrapadas en ellos.

5.1.2. Solución propuesta aplicada al problema de triangulación multi-objetivo

En vista de los resultados obtenidos se optó por modificar la función del problema y tratarla como una función multi-objetivo. Esta nueva función se ajustó al algoritmo NSGA-II de la siguiente manera: cada variable del punto tridimensional (X', Y', Z') fueron codificadas usando representación real, la aptitud de los individuos es igual al valor de las funciones objetivo $g_1(\mathbf{X})$ y $g_2(\mathbf{X})$ (Eqs. (4.3)) y los límites de las variables (X', Y', Z') van de $[-1, 1]$.

Para mostrar el desempeño de esta solución se utilizaron 24 correspondencias de puntos generadas sintéticamente a partir de un poliedro regular al igual que el experimento anterior. De éstas se tomaron aleatoriamente 8, 12, 16 y 20 correspondencias. A cada punto se le agregó ruido gaussiano con base en la raíz cuadrada media con media cero y desviación estándar de k píxeles, con k desde 1 hasta 8. A partir de las correspondencias de puntos se calculó la matriz fundamental y se estimaron las matrices de proyección, necesarias para realizar la triangulación.

El algoritmo NSGA-II tiene ciertos parámetros para su operación. Al igual que en los algoritmos evolutivos mono-objetivo, éste fue probado con distintos parámetros que se listan a continuación:

- Número de generaciones: 100, 300, 500 y 1,000
- Porcentaje de cruza: 0.1, 0.3, 0.7 y 0.9
- Porcentaje de mutación: 0.33, 0.66 y 1
- Tamaño de población: 50, 100 y 200

Al calcular la media de las dos funciones a partir de 30 ejecuciones, el conjunto de parámetros que dio mejores resultados es el siguiente: 300 generaciones, 100 individuos en la población, porcentaje de cruza de 0.9 y porcentaje de mutación de 0.33.

Las gráficas que se muestran en la figura 5.2 pertenecen a la media del error de las 30 ejecuciones que se realizaron con el algoritmo NSGA-II y el error obtenido con los métodos de triangulación LLS y Poly-Abs. Se observa claramente que, si sólo se toman 8 o 12 correspondencias de puntos (figs. 5.2a y 5.2b) para estimar la matriz fundamental y por ende, las matrices de proyección, el NSGA-II ofrece mejores resultados que los otros dos métodos. Esto se ve más marcado cuando se presenta un nivel de ruido mayor en los puntos. Sin embargo, cuando se toman 16 o

20 correspondencias de puntos (figs. 5.2c y 5.2d) el desempeño del NSGA-II es similar al desempeño del método LLS.

Debido a su buen desempeño bajo un espacio proyectivo, el siguiente paso fue probarlo bajo un espacio afín. Al igual que en el experimento anterior, se tomaron aleatoriamente 8, 12, 16 y 20 correspondencias de puntos de un conjunto de 24. Con estas correspondencias se estimó la matriz fundamental y se calcularon las matrices de proyección. Con estas matrices de proyección se obtienen puntos tridimensionales en el espacio proyectivo, por lo cual es necesario aplicar una transformación a estas matrices que permita obtener puntos tridimensionales en el espacio afín. Esta transformación fue previamente calculada a partir de las 24 correspondencias de puntos sin ruido.

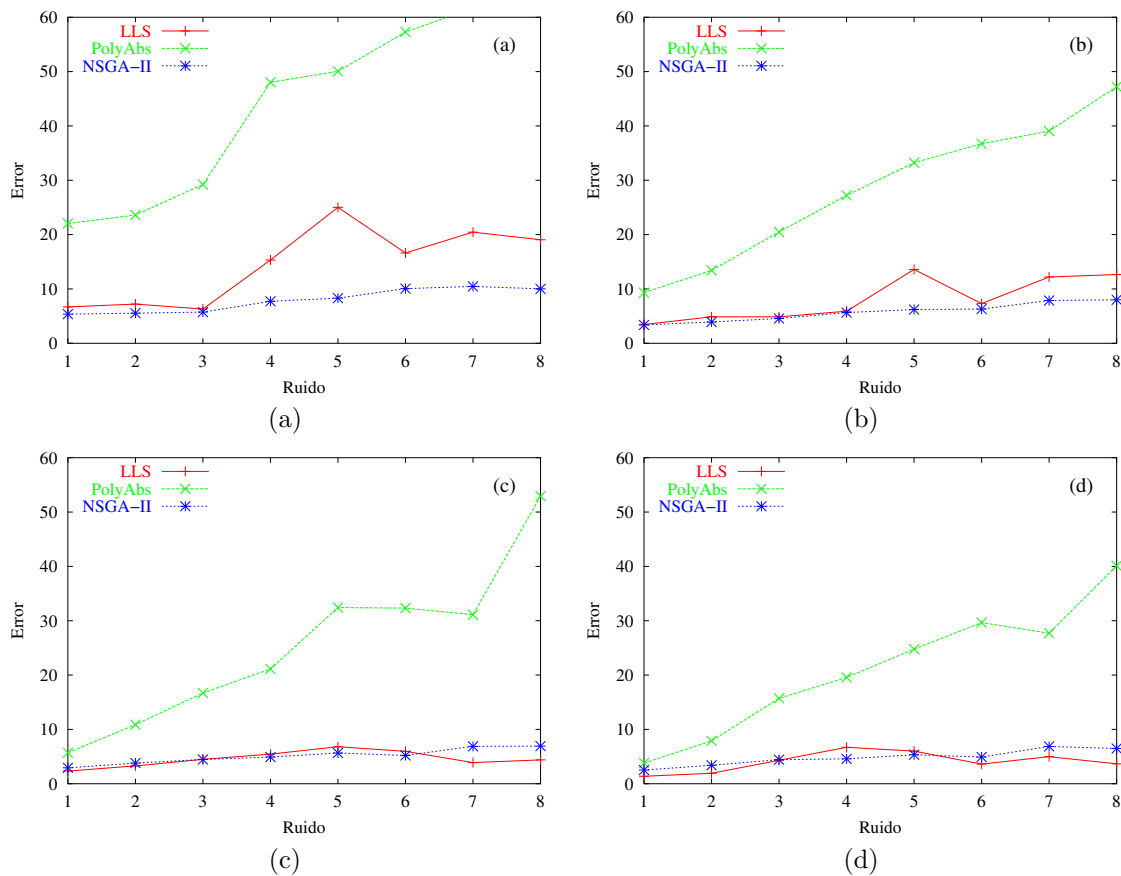


Figura 5.2: Comparación de los métodos NSGA-II, LLS y Poly-Abs bajo una reconstrucción proyectiva. El número de correspondencias de puntos para calcular la matriz fundamental es de: (a) 8, (b) 12, (c) 16 y (d) 20.

Las gráficas que se muestran en la figura 5.3 corresponden a la media del error de 30 ejecuciones del algoritmo NSGA-II con el mismo conjunto de parámetros que en el experimento anterior. Asimismo, se muestran los resultados de los métodos de triangulación LLS y Poly-Abs. En estas gráficas se observa que, bajo el espacio afín, si sólo se toman 8 correspondencias de puntos (ver fig. 5.3a) el NSGA-II obtiene menor error que los otros dos métodos y este desempeño se incrementa cuando el nivel de ruido en los puntos es mayor.

Cuando se tienen 12 correspondencias de puntos y mayor nivel de ruido en los puntos, se observa que la solución propuesta con el algoritmo NSGA-II es ligeramente

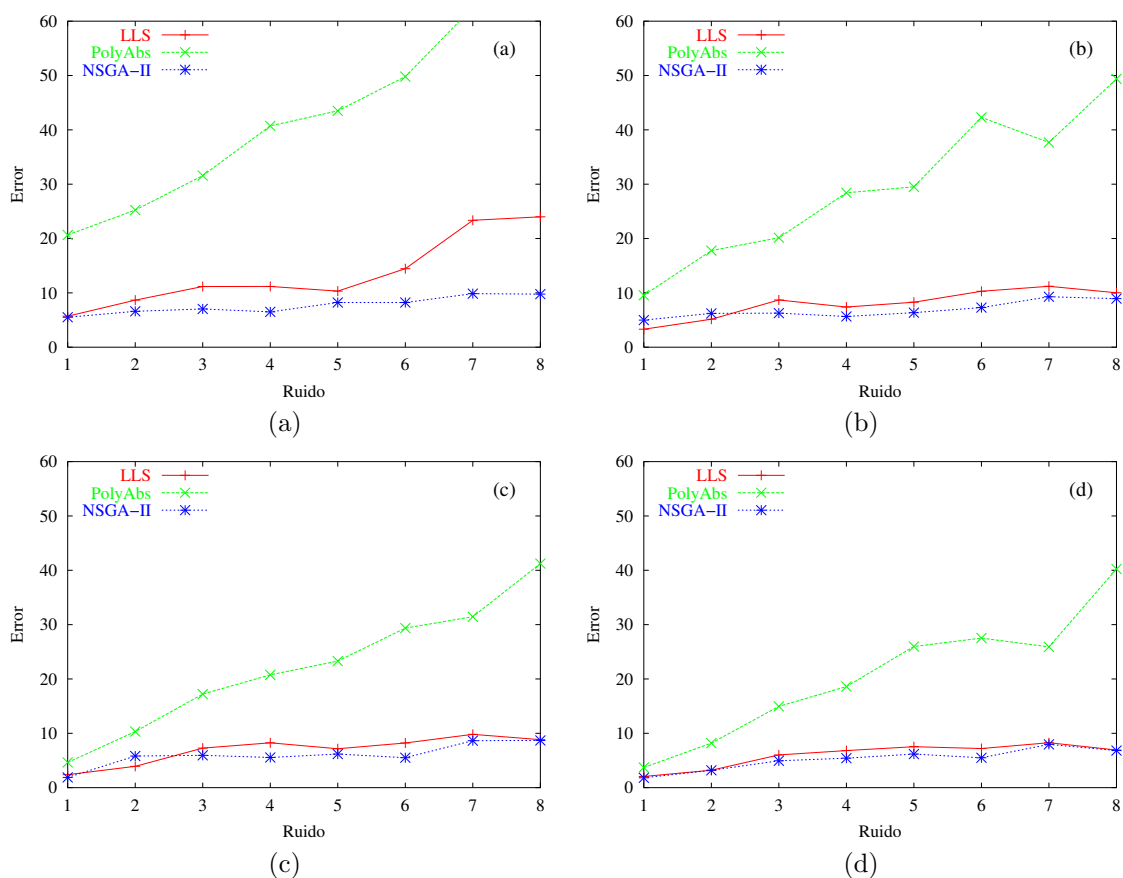


Figura 5.3: Comparación de los métodos NSGA-II, LLS y Poly-Abs bajo una reconstrucción afín. El número de correspondencias de puntos para calcular la matriz fundamental es de: (a) 8, (b) 12, (c) 16 y (d) 20.

mejor que el método LLS (ver fig. 5.3b). No obstante, cuando se toman 12, 16 o 20 correspondencias de puntos se observa en las figuras 5.3b, 5.3c y 5.3d que el desempeño del NSGA-II es similar al desempeño del método LLS.

La base del método Poly-Abs es la matriz fundamental y dado que ésta fue estimada a partir de las correspondencias de puntos con ruido, este método presenta un pésimo desempeño. Cabe señalar que si la matriz fundamental es calculada a partir de correspondencias de puntos sin ruido, este algoritmo obtiene el óptimo global [5, 71].

5.2. Reconstrucción tridimensional a partir de dos imágenes

Cuando se adquieren las imágenes de un cierto escenario del mundo real, éstas puede ser tomadas de distintas posiciones (configuraciones de la cámara). Es muy probable que la persona no preste atención a la posición de la cámara al tomar las imágenes. Si esta persona desea realizar una reconstrucción tridimensional métrica de los objetos a partir de dos imágenes con los algoritmos actuales, sólo podrá hacerlo si no cae dentro de una configuración crítica. Por desgracia es probable que esté dentro de una configuración crítica, ya que la persona puede adquirir las imágenes hacia un

costado caminando en línea recta o girando alrededor del objeto a reconstruir.

En esta sección se presenta parte de un extenso conjunto de experimentos donde se emplean las soluciones propuestas para mostrar su desempeño. El resto de los experimentos se muestran en los apéndices B y C. Estos experimentos incluyen escenarios virtuales con diferentes niveles de ruido y escenarios reales. Para ambos casos, solamente se ocupan dos imágenes que contienen objetos con superficies regulares e irregulares. Estas imágenes se generan o adquieren sobre configuraciones no críticas y, la parte más importante, sobre configuraciones críticas (donde los algoritmos actuales no pueden obtener una reconstrucción).

5.2.1. SPI en configuraciones no críticas

El primer experimento exhibe a la solución propuesta I (SPI) bajo un escenario con diferentes niveles de ruido y en una configuración no crítica. Esta prueba se realizó sobre un par de imágenes sintéticas, las cuales fueron generadas a partir de puntos tridimensionales que conforman a una esfera (ver fig. 5.4).

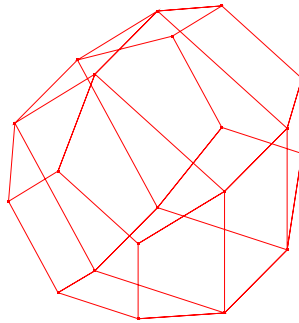


Figura 5.4: Esfera sintética.

Se entiende por imagen sintética al conjunto de puntos bidimensionales que fueron calculados a partir de proyectar los puntos tridimensionales del objeto en cuestión, mediante el modelo de la cámara oscura por cada imagen, es decir, $\mathbf{x} = M\mathbf{X}$ y $\mathbf{x}' = M'\mathbf{X}$. Para esta prueba, la matriz de calibración que se utiliza en ambas cámara, es de la siguiente forma:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

es decir, las coordenadas del centro de proyección están en el origen, los píxeles son cuadrados y el único valor que se asigna es la longitud del foco.

Los valores que se utilizaron para generar la matriz de rotación, el vector de traslación y la matriz de los parámetros intrínsecos de la primera matriz de proyección son:

- $\alpha = 10, \beta = 20, \gamma = 10$.

5. RESULTADOS

- $t_x = 3, t_y = 6, t_z = 12$.
- Longitud del foco: 150.

para la segunda matriz de proyección los valores propuestos son:

- $\alpha' = 30, \beta' = 15, \gamma' = 20$.
- $t'_x = 5, t'_y = 2, t'_z = 13$.
- Longitud del foco: 150.

Los ángulos de ambas matrices de proyección están expresados en grados. Para este experimento, se utilizaron n correspondencias de puntos sin ruido y con k niveles de ruido gaussiano. El ruido gaussiano RMS utilizado va de 1 a 3 píxeles.

El siguiente paso es definir los parámetros de control (constante de diferencia, constante de recombinación, tamaño de población y número de generaciones) del algoritmo de evolución diferencial. De acuerdo al criterio para la elección de los parámetros de la evolución diferencial propuesto por Mezura et al., en [133], éstos se fijaron con los siguientes valores: 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

El número de variables que se emplean en este experimento son $13 + 4 \times n$; seis variables que corresponden a los ángulos que generan las matrices de rotación de cada vista, seis variables que conforman los vectores de traslación de cada vista, una variable que corresponde a la longitud del foco y $4 \times n$ variables que corresponden a las coordenadas homogéneas de los n puntos en el mundo.

El espacio de búsqueda está definido por el intervalo de cada variable. Para las variables que generan la matriz de rotación (α, β y γ), el intervalo es de $[0, \pi]$; para las variables que conforman el vector de traslación, el intervalo es de $[-1, 1]$; para la longitud del foco, el intervalo que se estableció es de $[10, 1000]$, finalmente, para las variables (X, Y, Z, W) el intervalo es de $[-1, 1]$.

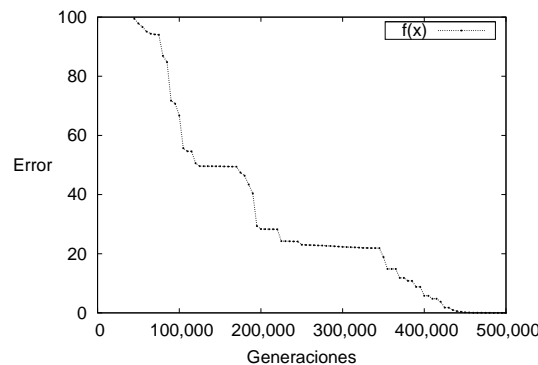


Figura 5.5: Gráfica de convergencia de una ejecución al emplear la SPI.

Se necesitaron 500,000 generaciones para que el algoritmo converja a una solución cercana al óptimo global¹. Se realizaron 30 ejecuciones del algoritmo sobre 8, 12, 16

¹En el siguiente capítulo, se presenta el criterio para decidir, si la solución obtenida está en la vecindad del óptimo global o probablemente es la solución global.

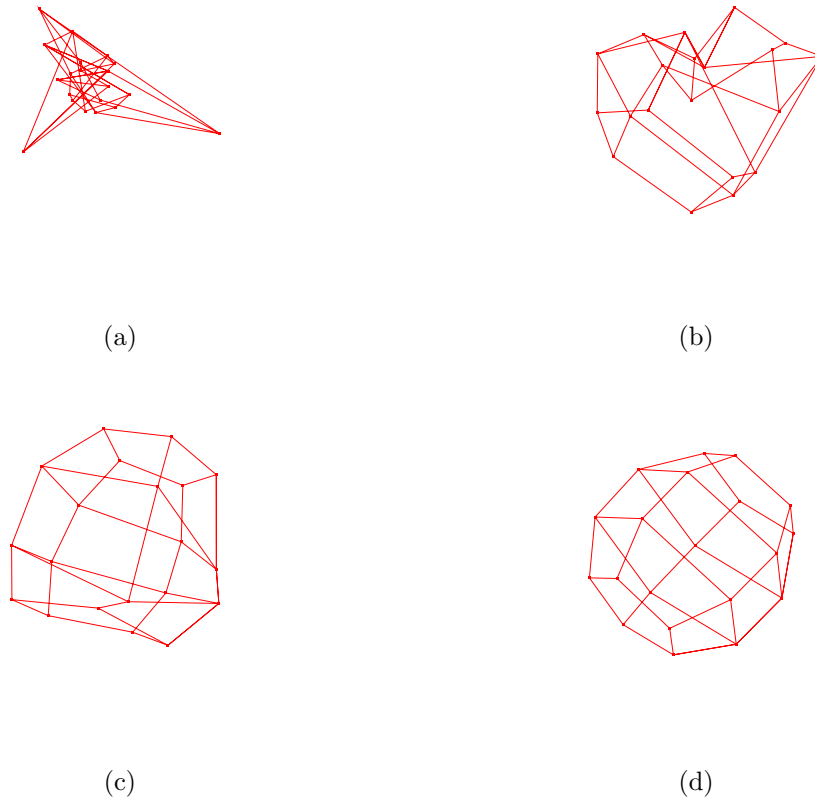


Figura 5.6: Reconstrucción tridimensional de la esfera: (a) generación 10,000, (b) generación 100,000, (c) generación 250,000 y (d) generación 500,000.

y 20 correspondencias de puntos sin ruido y con ruido gaussiano RMS de 1, 2 y 3 píxeles. En la figura 5.5 se muestra la gráfica de convergencia de una ejecución con 16 correspondencias de puntos sin ruido. En ella se observa que el algoritmo necesita de un número alto de generaciones para alcanzar una solución en la vecindad del óptimo global y, por lo tanto, una reconstrucción tridimensional métrica semejante al objeto real. Asimismo, se puede observar en la figura 5.6 cómo se está reconstruyendo el objeto a través de las generaciones y al llegar a la generación 500,000, la figura es casi idéntica en estructura (bajo una transformación de similitud) a la esfera real.

En la tabla 5.2.1 se muestran las estadísticas pertenecientes al error calculado por la siguiente ecuación,

$$e_r = \frac{1}{\sqrt{4n}} \left(\sum_{i=1}^n d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + \sum_{i=1}^n d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right)^{1/2} \quad (5.1)$$

Este error corresponde a la raíz cuadrada media (RMS por sus siglas en inglés).

El desempeño del algoritmo es muy bueno de acuerdo las estadísticas mostradas y a la visualización del objeto reconstruido (ver fig. 5.6d). Sin embargo, el tiempo de ejecución es bastante alto y se incrementa de forma lineal al aumentar el número de puntos, como se muestra en la tabla 5.2.

5. RESULTADOS

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
8	0	0.00	0.00	0.01	0.00	[0.00,0.01]
	1	0.06	0.23	0.45	0.13	[0.16,0.29]
	2	0.51	0.75	0.99	0.15	[0.67,0.82]
	3	1.14	1.28	1.52	0.13	[1.22,1.35]
12	0	0.02	0.04	0.10	0.02	[0.03,0.06]
	1	0.24	0.30	0.51	0.06	[0.28,0.35]
	2	1.35	1.44	1.56	0.06	[1.41,1.47]
	3	1.46	1.57	1.69	0.08	[1.54,1.61]
16	0	0.02	0.08	0.23	0.06	[0.06,0.12]
	1	0.29	0.45	0.99	0.20	[0.37,0.61]
	2	1.37	1.61	1.96	0.16	[1.54,1.75]
	3	2.04	2.23	2.73	0.15	[2.17,2.32]
20	0	0.05	0.11	0.20	0.05	[0.10,0.14]
	1	0.35	0.47	0.95	0.16	[0.41,0.58]
	2	1.54	1.70	1.99	0.16	[1.63,1.79]
	3	2.17	2.73	2.95	0.23	[2.56,2.82]

Tabla 5.1: Estadísticas del error utilizando la ecuación (5.1) a partir de 30 ejecuciones de la SPI sobre imágenes sintéticas.

Puntos	Tiempo promedio (segundos)
8	101.630
12	137.575
16	174.810
20	208.265

Tabla 5.2: Tiempo promedio de las 30 ejecuciones al emplear la SPI.

5.2.2. SPII en configuraciones no críticas

El primer experimento, en el cual se emplea la solución propuesta II (SPII), muestra el desempeño y la convergencia de esta solución bajo un escenario con diferentes niveles de ruido en una configuración no crítica. Esta prueba se realizó sobre un par de imágenes sintéticas, las cuales fueron generadas a partir de puntos tridimensionales que conforman una esfera (ver fig. 5.4).

Los valores empleados para generar las matrices de proyección fueron los mismos que en la prueba anterior. Para este experimento se utilizaron n correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados para la evolución diferencial son los mismos que en el experimento anterior, es decir, 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85. El número de variables que se emplean en este experimento es el mismo que en el experimento anterior, al igual que los intervalos del espacio de búsqueda de cada variable.

El número de generaciones que se utilizaron para que el algoritmo converja a una solución cercana al óptimo global es de 5,000. A diferencia de la SPI, la SPII requiere dos órdenes de magnitud menos para obtener una solución cercana al óptimo global.

La SPII se ejecutó 30 veces por cada 8, 12, 16 y 20 correspondencias de puntos sin ruido y con ruido gaussiano RMS de 1, 2 y 3 píxeles. En la figura 5.7 se muestra la gráfica de convergencia de una ejecución con 16 correspondencias de puntos sin ruido. En ella se observa que el algoritmo necesita pocas generaciones, en comparación con la SPI, para alcanzar una solución cercana al óptimo global. En la mayoría de las ejecuciones se observa que puede emplearse un número menor de generaciones para que el algoritmo converja, por ejemplo 4,000. Sin embargo, hay ejecuciones que necesitan hasta 5,000 generaciones para converger. En la tabla 5.2.2 se muestran las

estadísticas pertenecientes al error calculado con la ecuación (5.1).

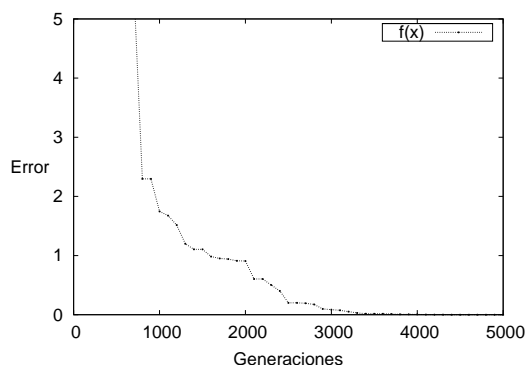


Figura 5.7: Gráfica de convergencia de una ejecución al emplear la SPII.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
8	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.14	0.17	0.18	0.01	[0.16,0.17]
	2	0.48	0.65	0.67	0.04	[0.60,0.66]
	3	0.59	0.64	0.65	0.01	[0.63,0.64]
12	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.38	0.47	0.55	0.05	[0.45,0.49]
	2	1.08	1.21	1.24	0.03	[1.19,1.22]
	3	1.23	1.27	1.34	0.03	[1.26,1.29]
16	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.55	0.59	0.59	0.01	[0.58,0.59]
	2	1.32	1.36	1.37	0.01	[1.35,1.36]
	3	2.91	2.97	2.99	0.01	[2.96,2.98]
20	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.48	0.49	0.51	0.01	[0.49,0.50]
	2	1.58	1.61	1.61	0.01	[1.60,1.61]
	3	3.11	3.14	3.15	0.01	[3.14,3.15]

Tabla 5.3: Estadísticas del error utilizando la ecuación (5.1) a partir de 30 ejecuciones la SPII sobre imágenes sintéticas.

En la figura 5.8a se muestran los puntos que conforman a la esfera tridimensional real. En la figura 5.8b se muestra la reconstrucción tridimensional métrica de la esfera cuando no hay ruido en las imágenes. En las figuras 5.8c, 5.8d y 5.8e se muestran la reconstrucciones de la esfera cuando hay 1, 2 y 3 píxeles de ruido gaussiano RMS, respectivamente. Las figuras con subíndice uno son las reconstrucciones con el menor error y las figuras con subíndice dos son las reconstrucciones con el mayor error.

El desempeño de la SPII es similar a la SPI, en cuanto al error (ver tabla 5.2.2), pero notablemente más rápido en cuanto al tiempo de ejecución, el cual también se incrementa de forma lineal de acuerdo al número de puntos a reconstruir, como se muestra en la tabla 5.4. Debido a estos resultados, el resto de los experimentos se realizan con la SPII.

El segundo experimento lidia con puntos que yacen bajo el mismo plano (superficie crítica) en ambientes reales. Para realizar este experimento se imprimió una cuadrícula similar a un tablero de ajedrez, la cual fue adherida a una pared blanca. Posteriormente, se adquirieron un par de imágenes de esta cuadrícula. El escenario de esta prueba es un conjunto de puntos que se extrajeron manualmente de imágenes

5. RESULTADOS

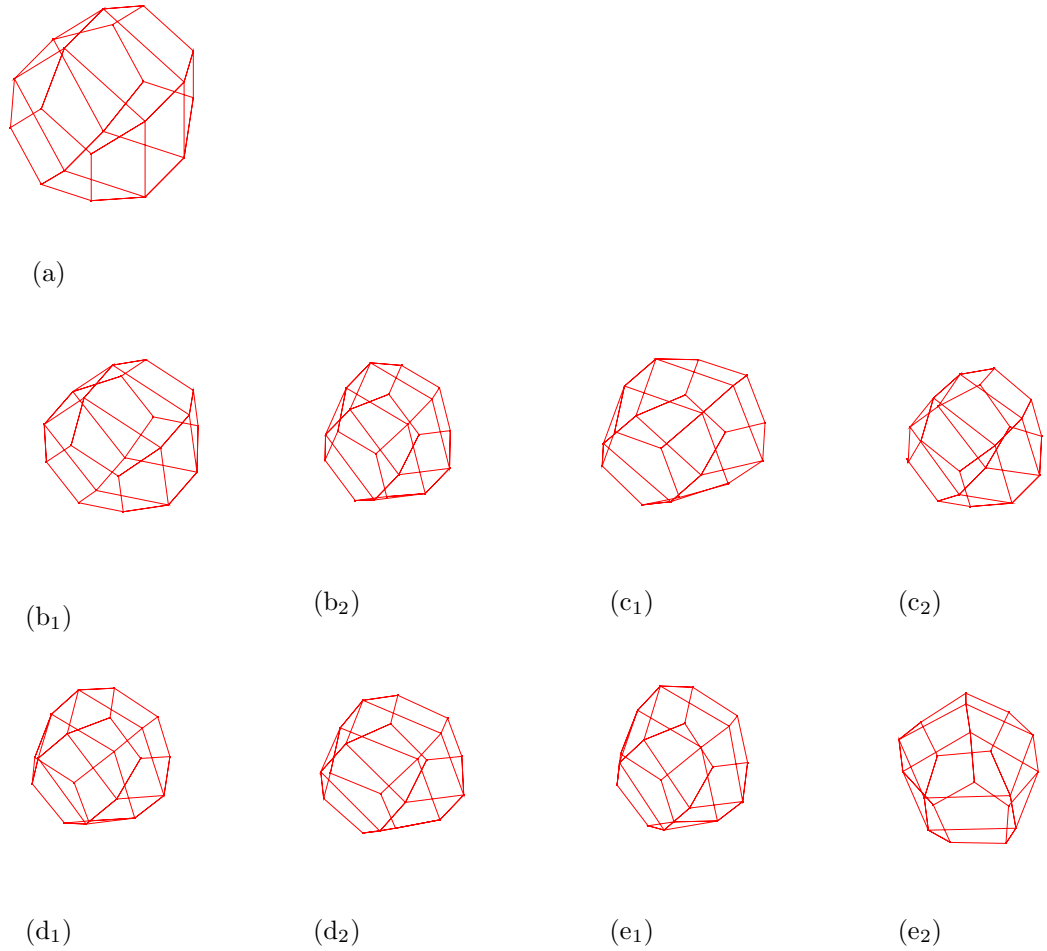


Figura 5.8: Reconstrucción tridimensional métrica. (a) esfera real, (b) esfera reconstruida sin ruido en los puntos, (c,d,e) esfera reconstruida cuando hay 1, 2 y 3 píxeles de ruido gaussiano RMS, respectivamente. El subíndice uno representa la ejecución con el menor error y el subíndice dos la ejecución con el mayor error.

Puntos	Tiempo promedio (segundos)
8	1.272
12	1.575
16	1.990
20	2.372

Tabla 5.4: Tiempo promedio de las 30 ejecuciones al emplear la SPII.

reales, bajo una configuración no crítica y una superficie crítica. En la figura 5.9 se visualizan ambas imágenes con marcas en los puntos elegidos.

Los parámetros empleados para la evolución diferencial son los mismos que los empleados en el experimento anterior, excepto por el número de generaciones, el cual es de 10,000 generaciones. Este es el número de generaciones necesario para que el algoritmo obtenga una solución óptima global o cerca de ella.

El número de variables que se definieron para este experimento es de quince, debido a que se agregó el centro de proyección (o_x, o_y) . El intervalo de cada variable de los parámetros de orientación y posición se mantiene como en los experimentos anteriores. Para el caso de la longitud del foco, el intervalo es de $[10, 1500]$ y, para

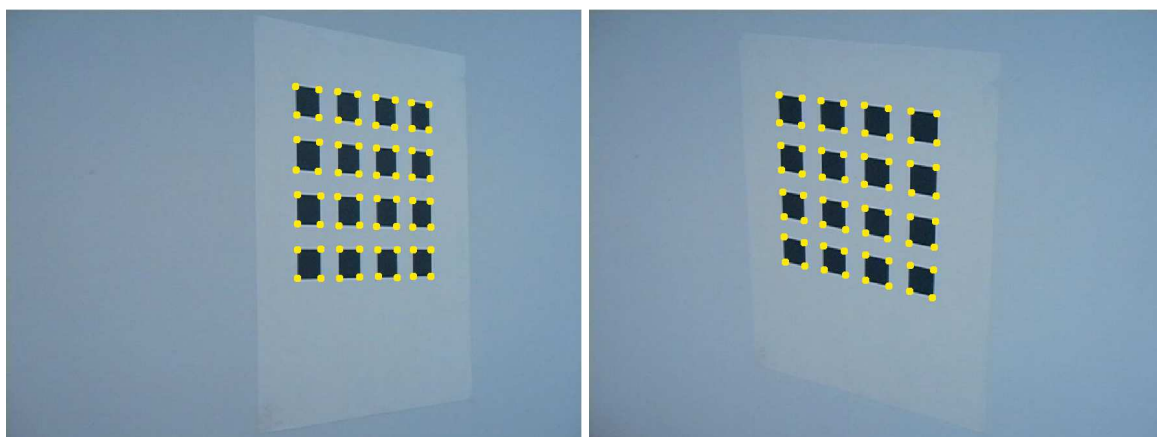


Figura 5.9: Imágenes reales de una cuadrícula. Los puntos que se emplearon están marcados en ambas imágenes.

el centro de proyección, los intervalos son: $o_x = [270, 370]$ y $o_y = [190, 290]$. Estos intervalos fueron tomados con base en el tamaño de la imagen que, para este caso, es de 640×480 .

Las estadísticas del error, al realizar 40 ejecuciones de la SPII, se muestran en la tabla 5.5. Estas estadísticas muestran que la SPII proporciona muy buenos resultados y la reconstrucción que se obtiene (reconstrucción que se tomó de la ejecución más cercana a la media) es casi idéntica a la cuadrícula real (ver fig. 5.10). Asimismo, en la tabla 5.6 se muestra las estadísticas de la longitud del foco, en las cuales se percibe una estabilidad del mismo en todas las ejecuciones.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
64	Real	0.425	0.430	0.486	0.011	[0.427,0.437]

Tabla 5.5: Estadísticas del error a partir de 40 ejecuciones de la SPII sobre imágenes reales.

Puntos	Ruido	Mínimo	Media	Máximo	d.e.	i.c. de la media
64	Real	940.00	949.19	979.88	10.60	[945.89,953.73]

Tabla 5.6: Estadísticas de la longitud foco a partir de 40 ejecuciones de la SPII sobre imágenes reales.

En el apéndice B se presentan más experimentos en las configuraciones no críticas a partir de imágenes sintéticas e imágenes reales.

5.2.3. SPII en configuraciones críticas

El siguiente paso es poner a prueba la SPII sobre las configuraciones críticas. En el capítulo anterior se mencionó acerca de tales configuraciones; en resumen, existen dos configuraciones donde los algoritmos actuales no pueden estimar los parámetros intrínsecos de la cámara: cuando el eje óptico de la primera cámara es paralelo al eje óptico de la segunda cámara (traslación en un solo eje coordenado) y cuando los ejes ópticos se intersecan en un punto tridimensional finito y los centros ópticos son equidistantes a este punto tridimensional (rotación en un solo eje).

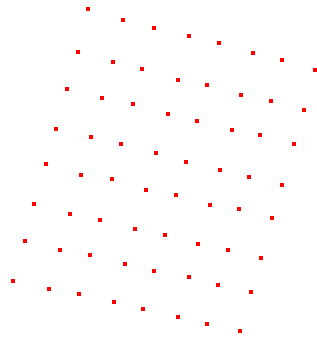


Figura 5.10: Reconstrucción tridimensional de la cuadrícula al emplear la SPII.

El primer experimento muestra el comportamiento de la SPII en las dos configuraciones críticas. El escenario es un conjunto de puntos que pertenecen a una figura regular (ver fig. 5.11), con diferentes niveles de ruido gaussiano RMS en los puntos y bajo configuraciones críticas.

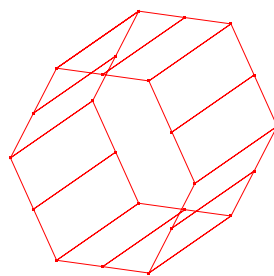


Figura 5.11: Figura regular tridimensional.

Los valores empleados para generar las matrices de proyección de la primera y la segunda configuración crítica se muestran en las tablas 5.7 y 5.8, respectivamente.

Para este experimento se utilizaron 24 correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados en la evolución diferencial son: 5,000 generaciones, 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

5.2 Reconstrucción tridimensional a partir de dos imágenes

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	5	7	20	500
2	0	0	0	25	7	20	500

Tabla 5.7: Parámetros de la cámara para la primera configuración crítica.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	20	500
2	0	40	0	0	0	20	500

Tabla 5.8: Parámetros de la cámara para la segunda configuración crítica.

El número de variables que se emplean en este experimento es trece y el intervalo de cada variable es el mismo que en el primer experimento de la SPII en las configuraciones no críticas. Para obtener las estadísticas del error que se muestran en las tablas 5.2.3 y 5.2.3 se realizaron 40 ejecuciones sobre la primera y segunda configuración crítica, respectivamente.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.75	0.75	0.75	0.00	[0.75,0.75]
	2	1.26	1.26	1.26	0.00	[1.26,1.26]
	3	2.51	2.51	2.51	0.00	[2.51,2.51]

Tabla 5.9: Estadísticas del error a partir de 40 ejecuciones de la SPII para la primera configuración crítica.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.68	0.68	0.68	0.00	[0.68,0.68]
	2	1.27	1.27	1.27	0.00	[1.27,1.27]
	3	2.47	2.47	2.49	0.00	[2.47,2.49]

Tabla 5.10: Estadísticas del error a partir de 40 ejecuciones de la SPII para la segunda configuración crítica.

En las figuras 5.12a y 5.12b se muestran las gráficas de convergencia al emplear correspondencias de puntos sin ruido para la primera y segunda configuración crítica, respectivamente. Estas gráficas se tomaron de las ejecuciones más cercanas a la media. En ellas se observa que el algoritmo necesita de 3,000 generaciones para llegar a una solución óptima global o en su vecindad.

En las figuras 5.13a₁, 5.13b₁, 5.13c₁ y 5.13d₁ se visualiza la reconstrucción tridimensional de la figura regular sin ruido, con 1, 2 y 3 píxeles de ruido gaussiano RMS, respectivamente, para el caso de la primera configuración crítica. En las figuras 5.13a₂, 5.13b₂, 5.13c₂ y 5.13d₂ se muestra la reconstrucción tridimensional de la figura regular sin ruido, con 1, 2 y 3 píxeles de ruido gaussiano RMS, respectivamente, para el caso de la segunda configuración crítica.

Para el caso de la primera configuración crítica, la reconstrucción tridimensional métrica es muy parecida a la figura real; las estadísticas y las gráficas de convergencia exhiben que el algoritmo es muy estable y robusto. Por tales motivos se comprueba el éxito de la SPII en la primera configuración crítica. En el caso de la segunda configuración crítica, las reconstrucciones que se presentan en las figuras 5.13a₁, 5.13b₁, 5.13c₁ y 5.13d₁ son engañosas, ya que a primera vista parecen mostrar una reconstrucción

5. RESULTADOS

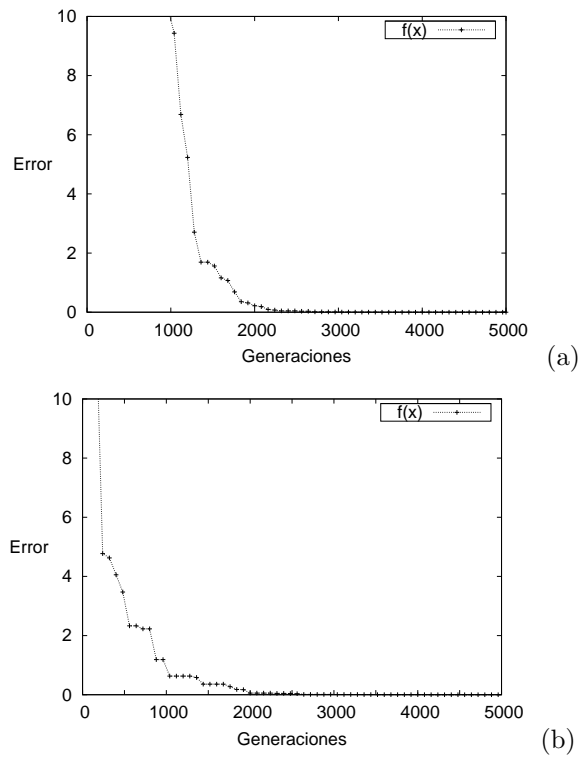


Figura 5.12: Gráficas de convergencia a partir de correspondencias de puntos sin ruido en: (a) la primera configuración crítica y (b) la segunda configuración crítica.

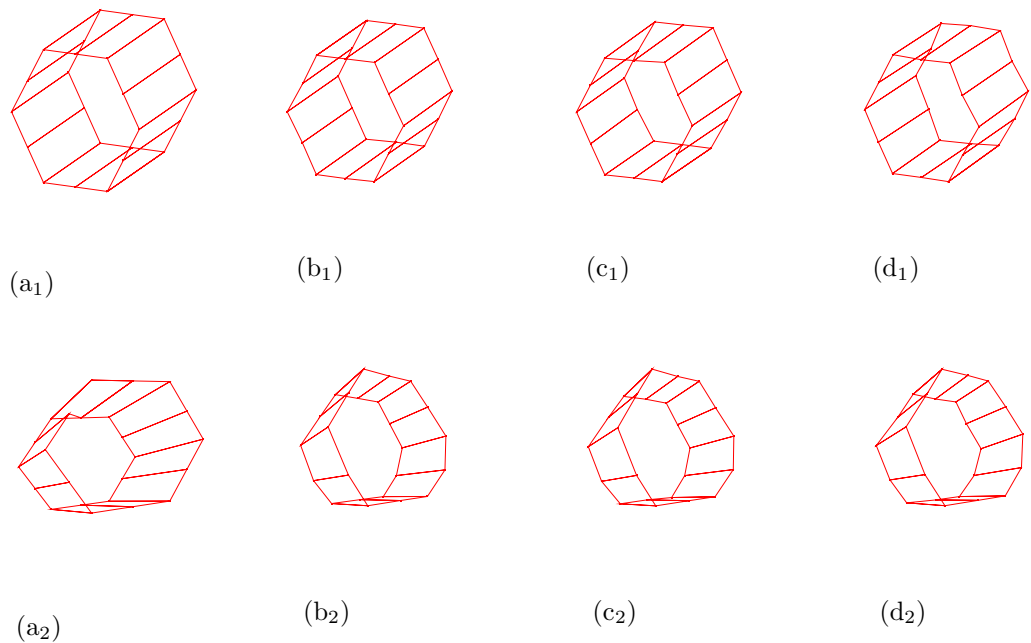


Figura 5.13: Reconstrucción de la figura regular. (a,b,c,d) indica el nivel de ruido y el subíndice (1,2) indica el tipo de configuración crítica.

tridimensional correcta, sin embargo, esto no es cierto. En la figura 5.14 se visualiza otro ángulo de la reconstrucción tridimensional métrica sobre la segunda configuración



Figura 5.14: Diferente vista de la reconstrucción tridimensional métrica al utilizar la SPII, a partir de correspondencias de puntos sin ruido sobre la segunda configuración crítica.

crítica. En ella se observa que todos los puntos reconstruidos tienden a estar sobre un mismo plano, es decir, la profundidad que se calcula con el método de triangulación empleado es muy similar en todos los puntos.

Para analizar este caso se realizó una prueba de triangulación que parte de las correspondencias de puntos sin ruido y las matrices de proyección, las cuales se calcularon directamente de los parámetros de la cámara (ver tabla 5.7). En el proceso de la triangulación, al generar la matriz A de tamaño 4×4 a partir del sistema de ecuaciones lineales, se obtuvo una matriz singular (rango 2), de la cual no es posible obtener una solución. Para que la triangulación sea posible es necesario que la matriz A tenga rango 3. De tal forma que, aunque se utilice cualquier método para resolver el sistema de ecuaciones lineales en la SPII y la función que se minimiza presente un valor muy cercano a cero, la reconstrucción tridimensional será diferente al objeto real.

En *el segundo experimento* se emplean imágenes de ambientes reales sobre la primera configuración crítica. El escenario de este experimento es un conjunto de puntos que se extraen manualmente de imágenes reales pertenecientes a un juguete bajo una configuración no crítica.

El juguete consiste de 47 correspondencias de puntos, las cuales se obtuvieron manualmente de imágenes reales. Estas imágenes se adquirieron dentro del laboratorio de pruebas y se exhiben en la figura 5.15.

Los parámetros empleados para la evolución diferencial son los mismos que los empleados en los experimentos anteriores, excepto por el número de generaciones, el cual es de 7,000. Esto se realizó con la finalidad de que el algoritmo obtenga una solución óptima global o en su vecindad.

El número de variables que se emplean para este caso es de quince, porque se agregó el centro de proyección (o_x, o_y) como parte de los parámetros a estimar. El intervalo de cada variable de los parámetros de orientación y posición se mantiene como en los experimentos anteriores. Para el caso de la longitud del foco se incrementó el intervalo a $[10, 1500]$ y para el centro de proyección, los intervalos son: $o_x = [270, 370]$ y $o_y = [190, 290]$. Estos intervalos fueron tomados con base en el tamaño de la imagen que, para este caso, es de 640×480 . De tal forma que el punto principal se encuentra en el centro de la imagen $[320, 240]$ o cerca de él.



Figura 5.15: Imágenes reales de un juguete sobre la primera configuración crítica. Los puntos que se emplearon están marcados en ambas imágenes.

Las estadísticas del error, al realizar 40 ejecuciones de la SPII, se muestran en la tabla 5.11. Estas estadísticas muestran que el algoritmo proporciona muy buenos resultados bajo la primera configuración crítica. Esto se corrobora con los parámetros obtenidos por el algoritmo (ver tabla 5.12), en los cuales se observa que no hay rotación en ninguno de los ejes coordenados y sólo hay traslación sobre el eje x . Esto es correcto, ya que sólo se trasladó la cámara sobre el eje x . Además, la estructura del juguete reconstruido (ver fig. 5.16) es muy similar a la del juguete real. Esta reconstrucción tridimensional se tomó de la ejecución más cercana a la media del error.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
47	Real	0.391	0.391	0.394	0.000	[0.391,0.394]

Tabla 5.11: Estadísticas del error a partir de 40 ejecuciones de la SPII sobre imágenes reales en la primera configuración crítica.

47 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Mejor	Real	-0.294	0.046	-0.001	0.988	-0.002	-0.009	1499.99	310.06	230.00
Media	Real	-0.294	0.047	-0.001	0.988	-0.002	-0.009	1499.99	318.19	230.36
Peor	Real	-0.294	0.047	-0.001	0.988	-0.002	-0.008	1499.95	310.23	234.26

Tabla 5.12: Parámetros de la cámara estimados por la SPII a partir de imágenes reales en la primera configuración crítica.

En el apéndice C se exponen más experimentos en las configuraciones críticas a partir de imágenes sintéticas e imágenes reales.

5.3. Reconstrucción tridimensional a partir de tres o más imágenes

Esta sección está destinada a presentar el desempeño de la SPIII en dos experimentos diferentes en configuraciones no críticas.

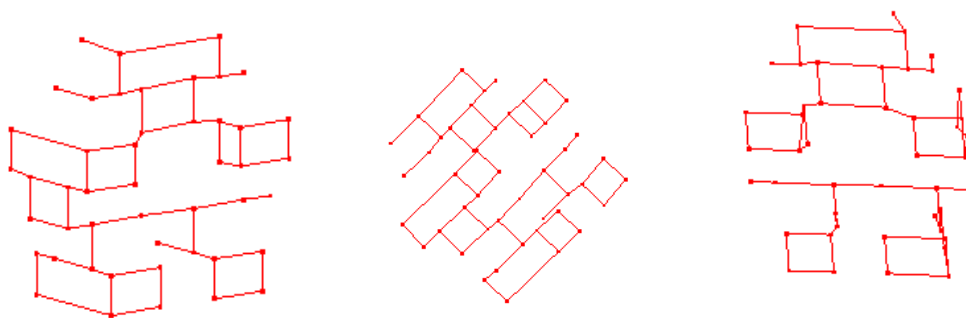


Figura 5.16: Tres vistas de la reconstrucción tridimensional del juguete.

Los dos experimentos muestran el desempeño y la convergencia de la SPIII sobre un escenario con diferentes niveles de ruido y bajo una configuración no crítica. Estas pruebas se realizaron sobre cuatro imágenes sintéticas, las cuales fueron generadas a partir de puntos tridimensionales que conforman parte de un anillo (ver fig. 5.17a) y una figura formada por cubos (ver fig. 5.17b). Sólo se toman los puntos que son visibles en cada imagen. Cabe señalar que las imágenes se generaron con la misma cámara, es decir, los parámetros intrínsecos son iguales en todas las imágenes.



Figura 5.17: Objetos tridimensionales: (a) parte de un anillo y (b) figura formada por cubos.

Los valores empleados para generar las matrices de proyección se muestran en la tabla 5.13 para el primer objeto y en la tabla 5.14 para el segundo objeto. En las figuras 5.18 y 5.19 se muestran las cuatro imágenes de una parte de un anillo y las cuatro imágenes de la figura formada por cubos.

Imagen	α	β	γ	t_x	t_y	t_z	f	f	f
1	270	1	250	2	1	25	700	250	300
2	272	1	240	2	1	25	700	250	300
3	268	2	220	1	3	25	700	250	300
4	268	0	205	1	3	25	700	250	300

Tabla 5.13: Parámetros de la cámara para obtener las cuatro imágenes del anillo.

Este experimento se ejecuta en dos fases, la primera calibra la cámara y estima los parámetros de orientación y posición de las primeras dos vistas y, en la segunda fase estima los parámetros de orientación y posición de las vistas restantes, más un factor

5. RESULTADOS

Imagen	α	β	γ	t_x	t_y	t_z	f	f	f
1	1	20	2	2	1	25	700	250	300
2	1	55	3	3	2	25	700	250	300
3	2	90	1	1	1	25	700	250	300
4	1	140	3	2	3	25	700	250	300

Tabla 5.14: Parámetros de la cámara para obtener las cuatro imágenes de la figura formada por cubos.

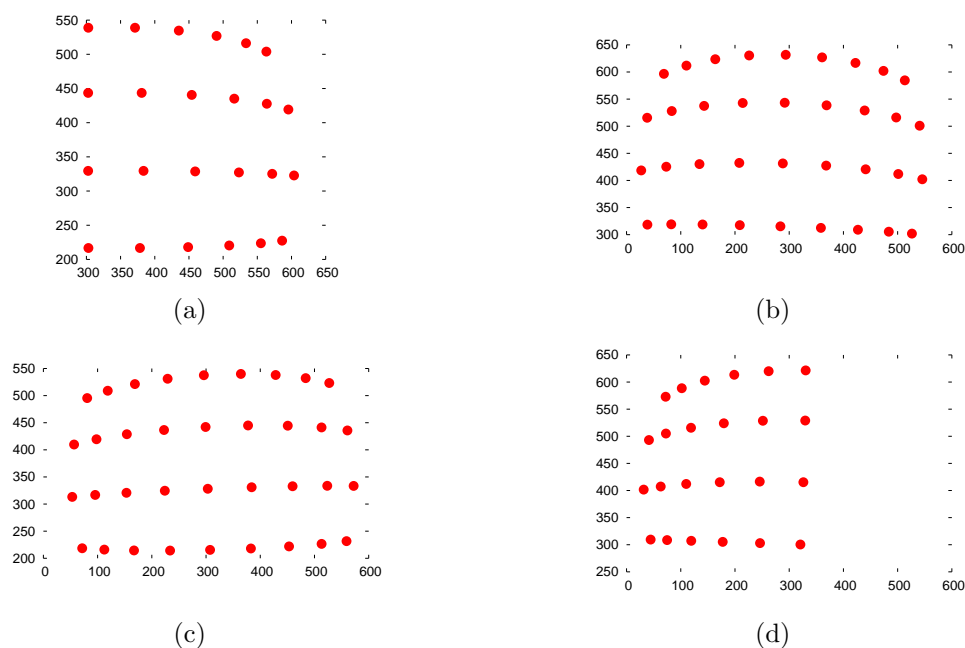


Figura 5.18: Imágenes de una parte de un anillo: (a) imagen 1, (b) imagen 2, (c) imagen 3 y (d) imagen 4.

de escala que ayuda a la convergencia del algoritmo. Para el primer experimento se utilizaron 48 correspondencias de puntos y en el segundo experimento se utilizaron 29 correspondencias de puntos. En ambos experimentos se emplearon las correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que van de 1 a 3 píxeles. Para el primer experimento, en la fase uno se utilizan 24 correspondencias de puntos y en la segunda fase las restantes. Para el segundo experimento, en la fase uno se utilizan 16 correspondencias de puntos y en la segunda fase las restantes (algunos puntos están presentes en tres o en las cuatro imágenes).

Los parámetros empleados para la evolución diferencial en la primera y segunda fase para ambos experimentos son: 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

En la primera fase se emplearon 10,000 generaciones, ya que se observó que los parámetros de la cámara y los puntos tridimensionales se estabilizan más, al aumentar el número de generaciones en ambos experimentos. Esto es necesario para que la fase dos obtenga resultados satisfactorios, de lo contrario, la fase dos tiene más probabilidades de no converger. En la segunda fase los parámetros se estabilizan en menos generaciones, por lo cual se emplean 5,000 generaciones.

Asimismo, para ambos experimentos, el número de variables que se emplean para la primera fase es de: seis variables que conforman las dos matrices de rotación de

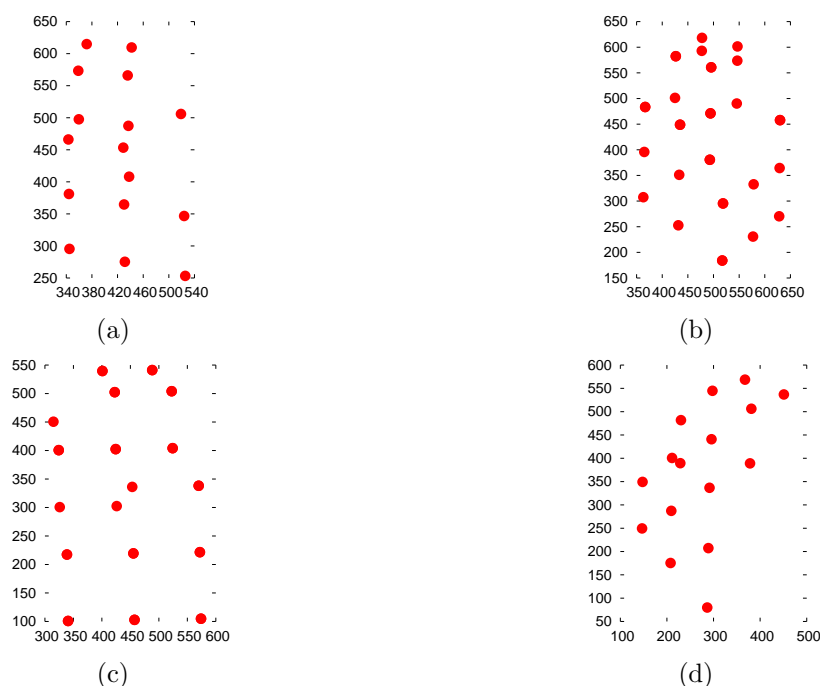


Figura 5.19: Imágenes de la figura formada por cubos: (a) imagen 1, (b) imagen 2, (c) imagen 3 y (d) imagen 4.

cada vista, seis variables que conforman los dos vectores de traslación de cada vista, una variable que corresponde a la longitud del foco y dos variables que pertenecen al centro óptico. El intervalo para las variables que conforman la matriz de rotación (α , β y γ) es de $[0, 2\pi]^2$; para las variables que conforman el vector de traslación es de $[-1, 1]$; para la longitud del foco el intervalo se estableció de $[10, 1000]$; y para el centro de proyección los intervalos son: $o_x = [270, 370]$ y $o_y = [190, 290]$.

El número de variables utilizadas en la segunda fase (por cada imagen que se añade) es de: tres variables que conforman la matriz de rotación, tres variables que conforman al vector de traslación (de la vista actual con el mismo intervalo) y un factor de escala s con un intervalo de $[-1, 1]$.

En la figura 5.20(a) se muestra la reconstrucción tridimensional métrica que se obtiene al finalizar la primera fase de la SPIII para el primer experimento. En las figuras 5.20(b) y 5.20(c) se muestra la reconstrucción tridimensional métrica que se obtiene al añadir la tercera imagen y la cuarta imagen, respectivamente. En la figura 5.21(a) se muestra la reconstrucción tridimensional métrica que se obtiene al finalizar la primera fase de la SPIII para el segundo experimento y en las figuras 5.21(b) y 5.21(c) se muestra la reconstrucción que se obtiene al añadir la tercera imagen y la cuarta imagen, respectivamente. Estas reconstrucciones se extrajeron de las ejecuciones más cercanas a la media del error con correspondencias de puntos sin ruido.

Para mostrar el desempeño de la SPIII se calcularon las estadísticas del error de 40 ejecuciones al término de la primera fase, y posteriormente, al término de cada imagen añadida en la segunda fase. El error en la segunda fase sólo contempla las correspondencias de puntos usadas en la actual iteración y no el total de correspon-

²En este caso se aumentó el intervalo a 2π debido a que la cámara puede rotarse en su totalidad para tomar las imágenes alrededor del objeto.

5. RESULTADOS

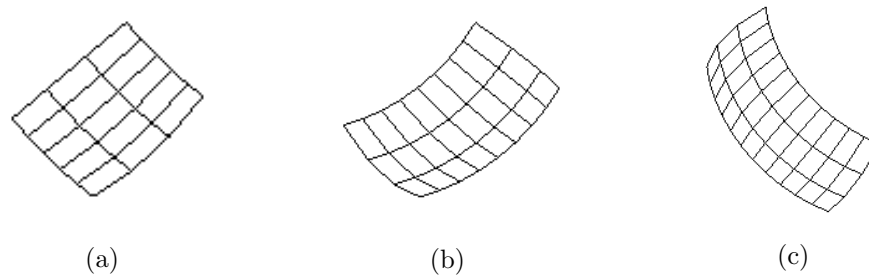


Figura 5.20: Reconstrucción tridimensional métrica de una parte de un anillo a partir de: (a) la primera y la segunda imagen, (b) añadir la tercera imagen y (c) añadir la cuarta imagen (con correspondencias de puntos sin ruido).

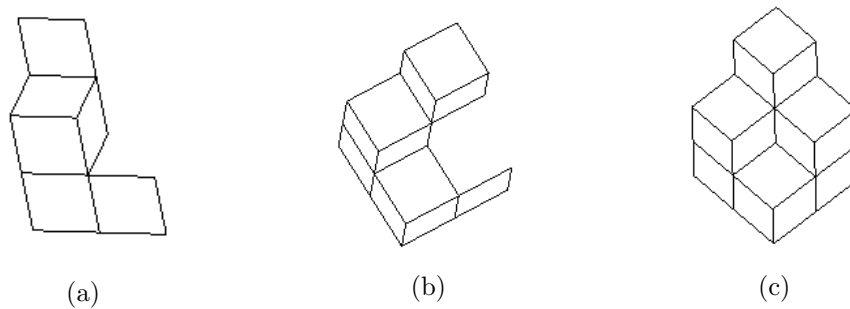


Figura 5.21: Reconstrucción tridimensional métrica de la figura formada por cubos a partir de: (a) la primera y la segunda imagen, (b) añadir la tercera imagen y (c) añadir la cuarta imagen (con correspondencias de puntos sin ruido).

dencias que se han empleado. Las estadísticas del primer y segundo experimento se exhiben en las tablas 5.15 y 5.16, de las cuales se observa que el algoritmo obtiene muy buenos resultados cuando tiene un mayor número de puntos, de lo contrario su desempeño se ve disminuido. No obstante, en la segunda fase se observa un nivel de ruido bajo (menor a 0.19) aún cuando los puntos no presentan ruido. Esto puede ser un factor de impacto en el desempeño del algoritmo si se utilizan más imágenes para reconstruir el objeto.

Fase	Puntos	Sin ruido	1 píxel	2 píxeles	3 píxeles
1 (imagen 1 y 2)	24	0.00	0.76	1.39	2.10
2 (imagen 2 y 3)	12	0.06	0.92	2.20	3.34
2 (imagen 3 y 4)	12	0.05	0.82	1.65	2.73

Tabla 5.15: Estadísticas del error de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento del anillo.

Fase	Puntos	Sin ruido	1 píxel	2 píxeles	3 píxeles
1 (imagen 1 y 2)	16	0.00	0.75	1.26	1.73
2 (imagen 2 y 3)	8	0.11	1.59	2.31	5.29
2 (imagen 3 y 4)	8	0.18	1.91	3.00	6.93

Tabla 5.16: Estadísticas del error de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento de la figura formada por cubos.

Asimismo, se calculó el error acumulado, es decir, se calculó el error con base en

5.3 Reconstrucción tridimensional a partir de tres o más imágenes

todas las correspondencias de puntos utilizadas hasta ese momento. Esto se hizo con la finalidad de mostrar el error total obtenido por la SPIII (ver tablas 5.17 y 5.17). Al igual que en las estadísticas mostradas anteriormente se observa que el error se incrementa al emplear menos puntos.

Fase	Puntos	Sin ruido	1 píxel	2 píxeles	3 píxeles
1 (imagen 1 y 2)	24	0.00	0.76	1.39	2.10
2 (imagen 2 y 3)	36	0.03	1.22	2.49	3.77
2 (imagen 3 y 4)	48	0.03	1.08	2.21	3.42

Tabla 5.17: Estadísticas del error acumulado de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento del anillo.

Fase	Puntos	Sin ruido	1 píxel	2 píxeles	3 píxeles
1 (imagen 1 y 2)	16	0.00	0.75	1.26	1.73
2 (imagen 2 y 3)	24	0.06	1.55	2.61	4.37
2 (imagen 3 y 4)	32	0.10	1.67	2.90	5.23

Tabla 5.18: Estadísticas del error acumulado de 40 ejecuciones por cada fase e imagen añadida para diferentes niveles de ruido en el experimento de la figura formada por cubos.

Capítulo 6

Discusión

El proceso de la reconstrucción tridimensional a partir de imágenes puede dividirse en tres pasos principales: 1) procesamiento de la imagen, 2) el paso propio de visión por computadora, y 3) la visualización de la reconstrucción.

El paso (1) involucra la adquisición de las imágenes, el filtrado de ellas para mejorar sus condiciones, y encontrar la correspondencia de puntos en las imágenes. Este paso se resuelve con técnicas de procesamiento de imagen.

El paso (3) involucra la generación de la malla (generalmente triangular) que da la forma al objeto reconstruido, la extracción de las texturas a partir de las imágenes para mapearlas sobre los polígonos de la malla y la visualización del objeto.

En esta tesis se desarrolló el paso (2) aplicando algoritmos evolutivos, especialmente, el algoritmo de la evolución diferencial, que ofrece muy buenos resultados según los experimentos realizados en este trabajo.

El paso (2) consiste en la *calibración de la cámara y la obtención de los parámetros de orientación y posición* para la vista correspondiente, y la *triangulación*, la cual permite obtener los puntos tridimensionales. Estos dos problemas de visión pueden expresarse como problemas de optimización no lineales (por los senos y cosenos de la orientación de cada vista, los cuadrados y la raíz de la distancia euclidiana entre los puntos de la imagen y los puntos estimados), e involucran hasta 15 variables a estimar. A continuación, se discute sobre estos dos problemas, cómo se plantearon para ser resueltos por algoritmos evolutivos, las ventajas y desventajas de usar tales algoritmos, y el desempeño con base en los resultados obtenidos.

6.1. Ruido en los puntos bidimensionales

Los datos necesarios para realizar una reconstrucción tridimensional son puntos seleccionados en una o más imágenes. Estos puntos pueden ser, por ejemplo, las esquinas de una caja o de un edificio.

Las valores de las coordenadas de los puntos seleccionados no pueden considerarse exactos debido: a la misma digitalización de la imagen (las unidades están en píxeles), al proceso mediante el cual se seleccionan los puntos (el proceso más simple es la selección manual de esos puntos con el ratón), al proceso de adquisición de la imagen por la cámara, y a las medidas de los propios objetos, que no son una representación exacta de un modelo matemático.

En principio se propuso usar simulaciones con modelos matemáticos para probar las soluciones desarrolladas en este trabajo. Puede pensarse en las coordenadas de los vértices de un cubo, por ejemplo. Ahora bien, para modelar la variabilidad en los

puntos seleccionados, se considera que cada punto se ve afectado por ruido; un ruido que se considera aditivo y con una distribución gaussiana. Entonces, un punto con ruido \mathbf{u} (que modela un punto en una imagen real) se calcula como:

$$\mathbf{u} = [u_x, u_y]^T = [x + \eta, y + \eta]^T,$$

donde $[x, y]^T$ son las coordenadas del punto en la imagen, generado a partir de un modelo sintético; η es una variable aleatoria, con una distribución gaussiana de media cero y desviación estándar $k/\sqrt{2}$. El valor de k es entonces, el nivel de ruido en píxeles de los puntos en la imagen. El factor $1/\sqrt{2}$ se agrega para que la magnitud del ruido adicionado sea precisamente k .

Para medir el error de la reconstrucción obtenida, se calculan las diferencias entre el punto bidimensional con ruido \mathbf{u} y el punto estimado $\hat{\mathbf{u}}$. Este error debe ser equiparable al nivel de ruido que se añadió, si esto se cumple, entonces el algoritmo presenta un buen desempeño y se puede decir que, estadísticamente, ha obtenido una solución óptima global o cerca de su vecindad [5]. Para el caso de dos imágenes, el error se calcula como:

$$\frac{1}{2n} \sum_{i=1}^n \left(d(\mathbf{u}, \hat{\mathbf{u}}) + d(\mathbf{u}', \hat{\mathbf{u}}') \right),$$

para n correspondencias de puntos.

6.2. Triangulación

La triangulación es el problema de encontrar el punto tridimensional resultado de la intersección de dos líneas correspondientes a los puntos bidimensionales en ambas imágenes. Cuando los dos puntos no tienen ruido, el problema se resuelve calculando la intersección de esas líneas, pero cuando los puntos presentan ruido, esta condición no se cumple más y debe estimarse la mejor solución.

La solución propuesta parte del modelo de la cámara oscura $\mathbf{x} = M\mathbf{X}$, donde M es la matriz de proyección, \mathbf{x} el punto de la imagen y \mathbf{X} el punto tridimensional. Con este modelo se empleó una función que minimiza la distancia euclidiana entre el punto de la imagen \mathbf{x} y el punto estimado $\hat{\mathbf{x}}$ por cada imagen, es decir,

$$g(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}') = d(\mathbf{x}, M\mathbf{X}) + d(\mathbf{x}', M'\mathbf{X}). \quad (6.1)$$

El propósito de esta función es estimar el punto tridimensional \mathbf{X} tal que minimice a la función g . Al emplear esta función se calcula la distancia dentro de un espacio euclidiano bidimensional (en donde si existe una métrica definida), y no importa en que espacio se encuentre el punto tridimensional (proyectivo, afín o métrico), la función garantiza ser invariante en estos espacios [5].

A partir de la función expresada en la ecuación (6.1), se planteó el problema de la triangulación como un problema multi-objetivo mediante la separación de dicha función en dos funciones objetivo, es decir,

$$g_1(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) \quad (6.2)$$

$$g_2(\mathbf{X}) = d(\mathbf{x}', \hat{\mathbf{x}}'). \quad (6.3)$$

Entonces se estableció el *problema de la triangulación* como: encontrar el punto tridimensional \mathbf{X} tal que minimice las funciones g_1 y g_2 .

Cuando se resuelve un problema multi-objetivo se obtiene un conjunto de soluciones llamado conjunto de óptimos de Pareto. En nuestro caso se eligió la solución que sea equitativa para ambos objetivos, es decir, donde el error obtenido de cada objetivo sea el mismo o la diferencia entre los errores de los objetivos sea la menor. Esta elección tiene como suposición que ambas imágenes tiene el mismo nivel de error.

Para resolver el problema se empleó uno de los algoritmos más competitivo y representativo del estado del arte en el área de optimización multi-objetivo, conocido como NSGA-II. Este algoritmo utiliza representación binaria y real. Diversos autores sugieren el uso de la representación real cuando se ataca un problema con parámetros reales [90, 172, 173], es por ello que se empleó esta representación.

Con base en las pruebas realizadas, los parámetros del NSGA-II que generaron mejores resultados son: 300 generaciones, 100 individuos en la población, porcentaje de cruce de 0.9 y porcentaje de mutación de 0.33. Al comparar estos resultados contra los mejores algoritmos del estado del arte (LLS [71] y Poly-Abs [71]) se observó lo siguiente (ver gráficas 5.2 y 5.3):

1. Si se tienen pocas correspondencias de puntos y el nivel de ruido es alto, entonces se debe emplear el NSGA-II, ya que ofrece mejores resultados que los algoritmos (LLS y Poly-Abs) para ambos tipos de reconstrucciones (proyectivo y afín).
2. Si se tiene un mayor número de correspondencias de puntos no importando el nivel de ruido, se puede emplear tanto el LLS como el NSGA-II. El LLS tiene un tiempo de ejecución mucho menor comparado con el NSGA-II.

Asimismo, se observa que el algoritmo Poly-Abs no obtiene los mejores resultados como se esperaba. Esto es porque emplea la matriz fundamental. En las pruebas realizadas, la matriz fundamental fue calculada a partir de correspondencias de puntos con ruido y esto afectó significativamente el desempeño del algoritmo. Por consiguiente, se sugiere utilizar la solución propuesta o el LLS, dependiendo del número de puntos y el tipo de reconstrucción. Cabe señalar que la solución propuesta sólo debe ser empleada en problemas donde el tiempo de ejecución no sea un factor crítico, ya que el tiempo de ejecución del NSGA-II es mucho mayor que el de los otros dos algoritmos. En comparación con el algoritmo Poly-Abs son cuatro órdenes de magnitud¹.

Por otro lado, en la literatura especializada sólo se han reportado dos problemas en los cuales se aplicó exitosamente la metodología de transformar el problema mono-objetivo a multi-objetivo [181]. La *triangulación* se suma a estos problemas, e incluso es posible sugerir que los problemas donde se involucren transformaciones geométricas pueden conseguir mejores resultados si se tratan como problemas multi-objetivo.

¹Esta comparación se tomó de la media de cien ejecuciones de ambos algoritmos. Por cada ejecución del NSGA-II se pueden realizar en promedio 30360 ejecuciones del algoritmo Poly-Abs.

6.3. Reconstrucción tridimensional a partir de dos imágenes

La calibración de la cámara consiste en encontrar la matriz de los parámetros intrínsecos que rigen a la cámara. Actualmente, la mayoría de las cámaras fotográficas o de video comerciales emplean lentes y sensores DCEI² que permiten simplificar la matriz de los parámetros intrínsecos, es decir,

$$K = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.4)$$

donde f es la longitud del foco, w el ancho de la imagen y h el alto de la imagen. En las cámaras actuales es posible suponer que los píxeles son cuadrados y los ejes son ortogonales, condiciones que anteriormente la tecnología no lo permitía. De tal forma que el único parámetro desconocido al calibrar la cámara es la longitud del foco.

La estimación de los parámetros de orientación y posición consiste en encontrar la rotación (matriz R) y la traslación (vector \mathbf{t}) que sufrió la cámara al tomar cada una de las imágenes.

La solución propuesta I (SPI) establece la calibración de la cámara y la estimación de los parámetros de rotación y traslación como un problema de optimización mono-objetivo. La SPI emplea la función

$$g_{obj}(M, M', \mathbf{X}_i) = \frac{1}{n} \sum_{i=1}^n \left(d(\mathbf{x}_i, \hat{\mathbf{x}}_i) + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i) \right) \quad (6.5)$$

como función objetivo a minimizar. La diferencia con el problema de la triangulación radica, en que este problema emplea n correspondencias de puntos y, además, de calcular los puntos tridimensionales, se calibra la cámara y se estiman los parámetros de orientación y rotación al mismo tiempo.

El análisis estadístico, con base en el error y las reconstrucciones obtenidas del primer experimento bajo una configuración no crítica (experimento descrito en la Sec. 5.2, en la pág. 74), muestra que:

1. Todas las ejecuciones convergen siempre a un resultado muy similar de acuerdo a la desviación estándar, la cual es menor a 0.25 píxeles en todos los casos, y al intervalo de confianza, el cual se aproxima en menos de 0.5 píxeles a las mejores y peores ejecuciones (ver la tabla 5.2.1) .
2. Los resultados obtenidos presuponen ser soluciones óptimas globales o muy cercanas a éstas, debido a que el nivel de ruido gaussiano que se añadió es similar al error.
3. A mayor número de puntos empleados, el error aumenta y se aproxima al nivel del ruido gaussiano añadido, tal como se esperaba.
4. El error tiene un crecimiento lineal con respecto al nivel de ruido gaussiano.

²Dispositivos de cargas eléctricas interconectadas.

6. DISCUSIÓN

5. Las reconstrucciones obtenidas son muy similares al objeto real (no se nota visualmente la diferencia entre la original y la reconstruida).

La desventaja es que se necesitaron 500,000 generaciones en la ED para obtener estas estadísticas. Esto se traduce en un tiempo de ejecución bastante alto, verbigracia, para 20 puntos se tarda en promedio 208 segundos.

Posteriormente, se planteó *la solución II* (SPII) donde se utiliza la misma función objetivo, con la diferencia que los puntos tridimensionales son calculados a partir de un método lineal de triangulación llamado *ecuaciones normales*. Al usar ecuaciones normales en la SPII, se requirieron 5,000 generaciones y el tiempo promedio se redujo dos órdenes de magnitud, es decir, se mejoró su desempeño considerablemente, en cuanto a tiempo de ejecución se refiere.

En relación al análisis estadístico del error, la SPII es similar a la SPI. Sin embargo, la SPII muestra ser ligeramente más estable, ya que la desviación estándar es menor a 0.06 píxeles para todos los casos (ver la tabla 5.2.2 del segundo experimento en la pág. 78), por consiguiente fue empleada para los experimentos restantes. Por otro lado, las reconstrucciones obtenidas son muy similares al objeto real para los diferentes niveles de ruido (ver fig. 5.8), incluso en las ejecuciones que se obtuvo mayor error, como se muestra en la figura 6.1.

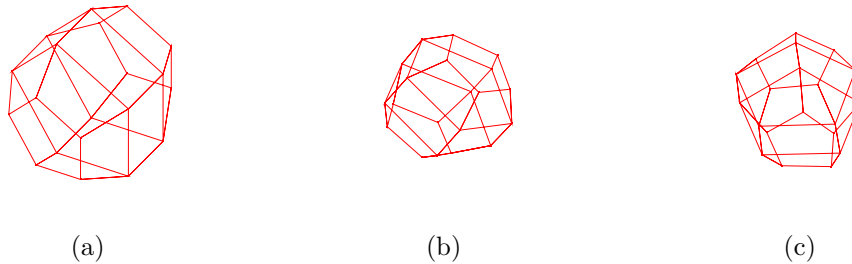


Figura 6.1: Reconstrucción tridimensional métrica. (a) esfera real, (b,c) esferas reconstruidas cuando hay 2 y 3 píxeles de ruido gaussiano a partir de las peores ejecuciones.

En el segundo experimento, la SPII se puso a prueba con puntos de imágenes reales bajo una superficie crítica y en una configuración no crítica. La SPII respondió de manera satisfactoria. En este experimento se utilizaron imágenes de una cuadrícula que está sobre un mismo plano, ambas adquiridas en el laboratorio de pruebas. El análisis estadístico muestra que la SPII obtiene en todas las ejecuciones, soluciones muy cercanas al óptimo global, ya que la desviación estándar es de 0.011 píxeles (ver la tabla 5.5 que se muestra en la pág. 80).

El resto de los experimentos realizados en las configuraciones no críticas, los cuales se discuten a continuación, se muestran en el apéndice B. Los resultados que arroja el tercer experimento, muestran la capacidad que tiene la SPII para calibrar la cámara. Cuando no hay ruido se obtiene la longitud del foco casi idéntico al valor original (0.01 % de error en relación a la media), y cuando hay ruido en los puntos, el porcentaje de error con base en la media sigue siendo bajo, aún en los experimentos con el nivel más alto de ruido (3 píxeles). Según [16], la reconstrucción tridimensional comienza a degradarse cuando el valor del foco difiere en más del 10 % del valor original. Por consiguiente, el algoritmo obtuvo excelentes resultados en todas las eje-

cuciones, ya que la longitud del foco obtenida en todas las ejecuciones está por debajo del 10 % de error.

Posteriormente, se realizó un experimento similar al anterior, pero en éste se muestran tanto los parámetros obtenidos por la calibración de la cámara como los parámetros de orientación y posición. Para realizar esta comparación fue necesario transformar las matrices de proyección a su forma canónica. En la literatura no se encontró el porcentaje de error permitido para que los parámetros de orientación y posición no influyan en la reconstrucción tridimensional. No obstante, los parámetros de orientación y posición estimados varían por debajo del 3 % del error relativo en relación a los parámetros originales y para la longitud del foco en un 7 % en el peor de los casos para los diferentes niveles de ruido y con diferente número de correspondencias de puntos (ver tablas B.1 y B.7 en la pág. 119, que representan los parámetros reales y estimados respectivamente). Por ejemplo, la peor ejecución obtiene una longitud del foco de 140.26 (8 correspondencias de puntos con nivel de ruido de 3 píxeles) y el valor original es de 150, es decir, 6.4 % de error relativo. En el caso de los parámetros de orientación, el peor resultado se dio en el ángulo β (resultado de emplear 8 correspondencias de puntos con nivel de ruido de 3 píxeles), donde el valor estimado es de 38.12 mientras que el valor original es de 36, lo cual representa un error relativo de 1.1 %. Para los parámetros de posición el peor caso se presentó en t_z donde en todas las ejecuciones obtuvo un error relativo cercano al 3 %. También se observa que, aunque son muy similares los valores en todas las ejecuciones, a mayor nivel de ruido, el porcentaje de error se incrementa, y a mayor número de puntos, el porcentaje de error disminuye ligeramente.

En el quinto experimento se realizó una comparación de la SPII con el método convencional, el cual emplea las formulaciones derivadas de las ecuaciones de Kruppa para calibrar la cámara [26], utilizando diferentes objetos (diferentes superficies y números de puntos). La SPII siempre obtuvo reconstrucciones muy similares a los objetos reales. En cuanto al método convencional se esperaban los mismos resultados, sin embargo, esto no sucedió. Este algoritmo obtiene reconstrucciones que distan del objeto real cuando se emplean pocas correspondencias de puntos. Por ejemplo, en el cubo (ver fig. 6.2) se emplearon 8 puntos y en la reconstrucción se perdieron las proporciones de sus dimensiones.

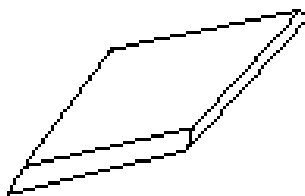


Figura 6.2: Cubo reconstruido mediante el método convencional.

Posteriormente, se probó la SPII sobre puntos que yacen en un mismo plano (ver sexto experimento en la pág. 124). Este es un tipo de superficie crítica. Sin embargo, se han reportado algoritmos muy robustos que emplean homografías, tal es el caso

del algoritmo de Zhang [16], para estimar la matriz de calibración. No obstante, el inconveniente de estos algoritmos es que sólo se pueden aplicar en ese caso. En este experimento, el algoritmo propuesto siempre converge a un valor cercano a la solución óptima global y la reconstrucción es muy parecida al objeto original. Esto muestra la robustez que tiene el algoritmo, ya que no es necesario modificarlo o desarrollar uno nuevo para obtener resultados en esta superficie, a diferencia de los algoritmos del estado del arte. Asimismo, la estimación de la longitud del foco de acuerdo a la media está dentro del intervalo donde no se afecta la reconstrucción, aunque para el valor mínimo y máximo sí se salen de este intervalo.

En el séptimo experimento, al aplicar la SPII sobre imágenes reales bajo configuraciones no críticas, la SPII obtuvo resultados satisfactorios. En este caso se utilizaron imágenes de una casa de campo pertenecientes al estado del arte e imágenes de un juguete adquiridas en el laboratorio de pruebas. El análisis estadístico muestra que la SPII obtiene, en todas las ejecuciones y en los dos objetos, un error muy bajo con una desviación estándar de 0.004 y 0.000871 píxeles, respectivamente (ver las tablas B.12 y B.14 que se muestran en la pág. 129).

No obstante, se tuvo que aumentar a 10,000 generaciones, en el caso de la casa de campo para obtener tales resultados. El error que se obtuvo en todas las ejecuciones para los dos objetos se encuentran dentro del intervalo de error que pueden tener las imágenes, el cual va de 0.2 a 2.0 píxeles [183]. Asimismo, los objetos tridimensionales reconstruidos, al igual que con los objetos sintéticos, presentan una gran similitud con los objetos reales, por ejemplo, se observan ortogonalidad entre las líneas, paralelismo y dimensiones acordes con el objeto real bajo una transformación de similitud.

Finalmente, la SPII se puso a prueba sobre las configuraciones críticas. En el primer experimento realizado bajo estas configuraciones (experimento presentado en la pág. 80), se observa que la información estadística con base en el error de la función, es muy aproximado a los experimentos anteriores para los diferentes niveles de ruido, e incluso, es más estable, ya que la desviación estándar es de 0.0 (ver tablas 5.2.3 y 5.2.3 para ambas configuraciones críticas que se muestran en la pág. 82).

El objeto reconstruido se aprecia visualmente idéntico al objeto real bajo la primera configuración crítica hasta 2 píxeles de ruido; con 3 píxeles se visualiza que las líneas formadas por tres puntos ya no son totalmente rectas, pero no pierde la estructura. En la segunda configuración crítica, a primera vista, la reconstrucción parece ser similar al objeto real, sin embargo, al observarlo desde otro punto de vista, todos los puntos se encuentran sobre un mismo plano. Esto es consecuencia de la configuración crítica, debido a que la matriz, la cual se genera a partir del sistema de ecuaciones lineales en la triangulación (en caso de tener los parámetros originales), es singular.

La SPII emplea ecuaciones normales y la matriz se resuelve por medio de determinantes. Al emplear determinantes, si la matriz, en algún momento fuese singular, el determinante sería cero, y al evaluar la función, ésta generaría un error muy grande. Como la SPII debe minimizar la función, la matriz que se genera no debe ser singular. Para lograr esto, los vectores de traslación \mathbf{t} y \mathbf{t}' deben ser diferentes entre ellos. Por tal motivo, los parámetros de posición no se obtienen correctamente. Tal situación se observa en el tercer experimento (experimento expuesto en el apéndice C), al comparar los valores de los parámetros de la cámara (ver tabla C.8 en la pág. 134) bajo la segunda configuración crítica, es decir, la estimación de los parámetros de orientación y la calibración de la cámara se realiza correctamente, aún en las ejecuciones con

mayor error, sin embargo, no es posible obtener la estimación de los parámetros de la posición. Al observar las gráficas del porcentaje de error (ver fig. C.5 en la pág. 136) se visualiza claramente que los parámetros de la posición presentan un error extremadamente alto y no existe relación con el nivel de ruido añadido. En cuanto a los parámetros de orientación y la longitud del foco, el porcentaje de error es bajo, incluso para las ejecuciones que obtuvieron peores resultados. Por otro lado, el error obtenido a partir de la función objetivo es cercano al nivel de ruido inducido en ambas configuraciones críticas y se mantiene estable en todas las ejecuciones. En el caso de la primera configuración crítica, las gráficas (ver fig. C.4 en la pág. 135) muestran que todos los parámetros de la cámara tienen un porcentaje de error menor al 6%, para todos los niveles de ruido. Además, existe un efecto interesante, en el cual algunos parámetros de orientación y posición disminuyen su porcentaje de error cuando el nivel de ruido se incrementa. Tal efecto supone que los parámetros de orientación y posición no presentan una relación estrecha con la función objetivo, a diferencia del parámetro intrínseco (longitud del foco), donde el porcentaje de error se incrementa linealmente en relación al nivel del ruido.

En el mismo experimento se midió el porcentaje de error entre las magnitudes de los puntos tridimensionales reales y los puntos tridimensionales estimados. Esta medición confirma la robustez del algoritmo, ya que se están comparando los puntos reales (no los puntos con ruido) contra los puntos estimados. Además, es muy probable que los mejores resultados obtenidos sean los óptimos globales, ya que están por debajo del nivel de ruido que se les añadió a los puntos.

En la literatura se reportan diversos algoritmos para calibrar la cámara y estimar los parámetros de orientación y posición mediante la matriz fundamental. Entre los algoritmos que destacan para calibrar la cámara de forma automática están: el algoritmo de Sturm et al. [26] y el algoritmo de Kanatani et al. [23]. Sin embargo, existen configuraciones críticas de la cámara donde estos algoritmos y, en general, los algoritmos que emplean la matriz fundamental no pueden calibrar la cámara ni estimar los parámetros de orientación y posición. Tal situación se visualiza en el cuarto experimento bajo configuraciones críticas (mostrado en el apéndice C, en la pág. 117).

En este experimento sólo se compara la SPII contra el método convencional utilizando diferentes objetos. Como ya se mencionó, el método convencional hace uso de la matriz fundamental y emplea el algoritmo de Sturm et al. [26] para calibrar la cámara. Al emplear este método en las configuraciones críticas, no se obtiene resultados debido a la matriz fundamental calculada, la cual se observa como una cónica degenerada. En la primera configuración crítica, la cónica que se generó se exhibe como una línea vertical recta en todos los objetos; en la segunda configuración crítica, la cónica se visualiza como una hipérbola degenerada, la cual se une formando una cruz. Estas cónicas muestran la deficiencia que tiene la matriz fundamental de forma gráfica. Este problema afecta a todos los algoritmos que emplean la matriz fundamental, ya que imposibilita la extracción de los parámetros de la cámara. La solución propuesta no sufrió de este problema, aún cuando se le añadieron dos variables más a estimar, las cuales corresponden al punto principal. Todos los parámetros estimados de la cámara son muy similares a los parámetros reales bajo la primera configuración crítica y esto se ve reflejado en las reconstrucciones, las cuales visualmente, aparentan ser idénticas a las reales bajo una transformación de similitud. En la segunda configuración crítica, al igual que en resultados anteriores, sólo se obtienen los parámetros de orientación

y los parámetros intrínsecos, los cuales ahora son tres y no parecen afectados por la configuración crítica, como sucede con los parámetros de posición. Las reconstrucciones de todos los objetos en esta configuración mantienen cierta forma de los objetos originales, pero los puntos yacen sobre un mismo plano (tal efecto se observa mejor con la media esfera, el cubo y la figura formada por cubos).

Otro experimento que arrojó resultados esperados, en cuanto a la reconstrucción tridimensional, es el quinto experimento (experimento expuesto en el apéndice C), el cual emplea puntos que yacen sobre un mismo plano. En este caso se esperaba obtener una reconstrucción correcta para ambas configuraciones porque, de acuerdo a los resultados anteriores, específicamente en la segunda configuración crítica, los puntos tridimensionales reconstruidos son coplanares. Al visualizar las reconstrucciones se observa que en efecto, en ambas configuraciones los puntos tridimensionales fueron reconstruidos correctamente y, al igual que en los experimentos previos, los parámetros de traslación no se obtuvieron correctamente en la segunda configuración crítica. Por otro lado, el desempeño se mantiene firme de acuerdo al análisis estadístico del error mostrado en las tablas C.16 y C.17 en la pág. 145, en donde se observa que la desviación estándar es menor a 0.24 píxeles.

El segundo y sexto experimento (experimentos mostrados en el capítulo de resultados y en el apéndice C, respectivamente) se realizaron con puntos de imágenes reales bajo la primera configuración crítica. El primer par de imágenes pertenece a un juguete y el otro pertenece a una cuadrícula (puntos sobre un mismo plano). El análisis estadístico del error muestra el excelente desempeño que tuvo el algoritmo en todas las ejecuciones; siempre se mantuvo dentro del intervalo del error para las imágenes reales, e incluso el error fue menor que en los experimentos de las configuraciones no críticas (la media del error es de 0.625 para el juguete y 0.43 para la cuadrícula en configuraciones no críticas, mientras que en las configuraciones críticas es de 0.3911 para el juguete y 0.256 para la cuadrícula) y ligeramente más estable, ya que la desviación estándar es cercana a cero para el juguete y cero para la cuadrícula (ver las tablas 5.11 y C.20 en las págs. 85 y 148, respectivamente). Es posible que el decremento del error sea consecuencia de agregar o_x y o_y (punto principal) como variables. Asimismo, los parámetros de orientación y posición estimados concuerda con gran exactitud (menos del 3% de error relativo). En cuanto a los parámetros intrínsecos, la longitud del foco se encuentra dentro del intervalo de las cámaras actuales y el punto principal presenta un pequeño desplazamiento del centro de la imagen, el cual probablemente contribuyó en la disminución del error. No obstante, fue necesario incrementar el número de generaciones a 10,000 para obtener estos resultados.

En todos los experimentos realizados, la variante de la evolución diferencial que se usó fue la *rand/1/bin*; los parámetros empleados que generaron los mejores resultados y que permitieron que todas las ejecuciones llegaran a soluciones óptimas globales o muy cercanas a ellas son: tamaño de población de 30, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

6.4. Reconstrucción tridimensional a partir de tres o más imágenes

Cuando se tienen tres o más imágenes adquiridas por la misma cámara, el problema radica en estimar los parámetros de orientación y posición de la misma al tomar cada una de las imágenes. La calibración de la cámara es posible obtenerla con sólo dos imágenes, ya que los parámetros internos de la cámara no se modifican. Para lidiar con este problema se propuso *la solución III* (SPIII). Esta solución se divide en dos fases: la primera emplea la SPII para calibrar la cámara, estimar los parámetros de orientación y posición, y calcular los puntos tridimensionales, a partir de las primeras dos imágenes, con base en la función objetivo g de la ecuación (6.5); la segunda sólo estima los parámetros de orientación y posición de las imágenes restantes y los puntos tridimensionales, para lo cual, emplea la misma función objetivo g con los valores de la matriz de calibración y los parámetros de orientación y posición de la imagen anterior, es decir,

$$g_{ve}(R_j, \mathbf{t}_j, \mathbf{X}_i^j, s) = \frac{1}{c} \sum_{i=1}^c \left(d(\mathbf{x}_i^{j-1}, \hat{\mathbf{x}}_i^{j-1}) + d(\mathbf{x}_i^j, \hat{\mathbf{x}}_i^j) \right) \quad \forall j > 2, \quad (6.6)$$

donde \mathbf{x}_j y \mathbf{x}'_j son los j -ésimos puntos de la imagen y los puntos estimados que pertenecen a la j -ésima matriz de proyección M_j de la j -ésima vista, y s un factor de escala, el cual se aplica al vector de traslación de la j -ésima vista para ayudar a la convergencia del algoritmo. Obviamente, es necesario que existan correspondencias de puntos en al menos tres imágenes.

En los experimentos realizados se emplearon cuatro imágenes sintéticas por cada objeto (parte de un anillo y figura formada por cubos) con diferentes matrices de proyección. En un principio se emplearon 5,000 generaciones para la primera fase de la SPIII. Sin embargo, al ejecutar la segunda fase, con la segunda y tercera imagen, aproximadamente la mitad de ejecuciones generaron un error mucho mayor al nivel de ruido añadido, el cual al propagarse al siguiente par de imágenes (tercera y cuarta imagen) éste se incrementaba de forma cuadrática (aproximadamente).

Posteriormente se emplearon 10,000 generaciones en la primera fase, esto trajo consigo mayor estabilidad en los parámetros de la cámara (ver tablas 5.15 y 5.16 de las estadísticas del error en la pág. 89) y las ejecuciones en las segunda fase generaron una media del error muy cercana al nivel de ruido añadido. Por ejemplo, en el primer experimento se obtuvo una media del error de 0.06, 0.92, 2.20 y 3.34 píxeles para los diferentes niveles de ruido. Sin embargo, cuando se reconstruyó la figura formada por cubos (segundo experimento), en el nivel de ruido de 3 píxeles y bajo la fase dos, el algoritmo obtienen un error de 5.29 y 6.93 píxeles, los cuales son valores similares al doble del nivel de ruido. Es muy probable que si se tiene un mayor número de imágenes y un nivel de ruido alto (3 o más píxeles), las últimas iteraciones de la segunda fase podrían verse afectadas severamente. Este efecto también se observa en las estadísticas del error acumulado. Por otro lado, es posible que las diferencias de los parámetros de orientación entre las imágenes afecte el desempeño del algoritmo aunado al nivel de ruido. Por ejemplo, el mayor intervalo de rotación entre las imágenes del anillo fue de 45° , mientras que, para el objeto formado por cubos fue de 120° .

Capítulo 7

Conclusiones

En esta tesis se propusieron cuatro soluciones: la primera solución realiza el proceso de la triangulación, y las otras tres soluciones obtienen la reconstrucción tridimensional métrica, es decir, *calibran la cámara, estiman los parámetros de orientación y posición de la misma por cada vista y realizan la triangulación*. Estas soluciones hacen uso de los algoritmos evolutivos como método de optimización. Asimismo, se propuso otra solución para resolver el ajuste de la mejor elipse dado un conjunto de puntos, el cual es otro problema del área de Visión por Computadora.

1. La primera propuesta¹ es una solución que plantea al problema de la triangulación como un problema de optimización, donde la función a minimizar es la distancia euclidiana entre los puntos de la imagen y los puntos bidimensionales estimados. En principio, se mostró que, al resolver la triangulación como un problema mono-objetivo mediante un algoritmo genético y un algoritmo de optimización por cúmulos de partículas, se obtienen resultados que distan en gran medida de la solución óptima global. Posteriormente, se aplicó una metodología que transforma el problema mono-objetivo a un problema multi-objetivo, el cual fue resuelto mediante el algoritmo NSGA-II. Esta transformación trajo consigo una mejora dramática en los resultados. Además, se obtuvieron mejores resultados, en comparación con los mejores métodos reportados en la literatura (LLS y Poly-Abs), sobre las reconstrucciones afín y proyectiva cuando se tienen pocas correspondencias de puntos para calcular las matrices de proyección y tales puntos presentan un nivel de ruido alto. El problema de la solución propuesta es el tiempo para obtener los resultados, de tal forma que, no es posible integrarla a un sistema de tiempo real. Cabe señalar que este problema se suma a los dos casos reportados en la literatura [168, 169], donde se aplicó exitosamente la metodología de transformar un problema mono-objetivo a uno multi-objetivo.
2. Las soluciones propuestas (SPI, SPII y SPIII) realizan la calibración de la cámara, la estimación de los parámetros de orientación y posición sobre cada una de las vistas adquiridas por la cámara (sin calibración previa) y, al mismo tiempo, calcula los puntos tridimensionales (triangulación) a partir de dos o más imágenes. Estas soluciones plantean a estos tres problemas como un problema de optimización (bajo el modelo de la cámara oscura), el cual tiene como función objetivo la media de la distancia euclidiana entre los puntos de cada imagen y los puntos bidimensionales estimados. Esta función no emplea la matriz fundamental, de tal forma que las soluciones propuestas tienen la capacidad de

¹Propuesta aceptada y publicada en el congreso de EvoWorkshops (EvoIASP) 2007 [181].

obtener los parámetros de la cámara y reconstruir los objetos sobre cualquier superficie y en cualquier configuración de la cámara, incluso en la primera configuración crítica, donde hasta el momento los algoritmos actuales no han podido obtener una reconstrucción tridimensional métrica. En todas las soluciones, el problema de optimización fue resuelto con la evolución diferencial. La primera solución propuesta (SPI) estima todos los parámetros de la cámara y los puntos tridimensionales mediante la evolución diferencial a partir de dos imágenes. Esto trae consigo que se necesiten alrededor de 500,000 generaciones para que la evolución diferencial presente un excelente desempeño, es decir, converger a una solución óptima global o cercana a ella. Sin embargo, esta cantidad de generaciones se traduce en minutos. La segunda solución propuesta² (SPII) estima todos los parámetros de la cámara a partir de dos imágenes mediante la evolución diferencial y calcula los puntos tridimensionales a través de un sistema de ecuaciones lineales no homogéneo, resuelto con ecuaciones normales. Con esta nueva solución se redujo en dos órdenes de magnitud el número de generaciones (5,000 generaciones), al igual que el tiempo de ejecución, aunque no es suficiente para emplearse en sistemas de tiempo real. El desempeño de la SPII es ligeramente mejor que la SPI y más estable. Asimismo, las reconstrucciones generadas son muy similares de manera visual y al medir las dimensiones de los objetos reales bajo una escala arbitraria. La robustez de la SPII permite obtener un buen desempeño tanto en escenarios virtuales con diferentes niveles de ruido, como en escenarios del mundo real. La tercera solución propuesta³ (SPIII) estima todos los parámetros de la cámara mediante la evolución diferencial y calcula los puntos tridimensionales mediante un sistema de ecuaciones lineales no homogéneo, resuelto con ecuaciones normales, a partir de tres o más imágenes. La SPIII se divide en dos fases: la primera calibra la cámara y estima los parámetros de orientación y posición de las primeras dos vistas mediante la SPII; la segunda fase estima los parámetros de orientación y posición de las vistas restantes mediante la evolución diferencial, y calcula los puntos tridimensionales que pertenecen a esas imágenes restantes, mediante ecuaciones normales. La SPIII permite emplear más de dos imágenes, de tal forma que es posible hacer una reconstrucción tridimensional del objeto en su totalidad. El desempeño de la SPIII es bastante bueno de acuerdo a los resultados obtenidos (donde se emplean cuatro imágenes), sin embargo, parece que su desempeño se ve disminuido cuando se incrementa el número de vistas.

3. La última propuesta⁴ es una solución para resolver el problema de ajuste de puntos a una elipse. La definición del problema y la solución propuesta se presentan en el apéndice A. Esta solución plantea al problema de ajuste geométrico como un problema de optimización, en el cual, la función objetivo a minimizar es la suma de la distancia euclidiana entre los puntos y la elipse. La solución emplea la función paramétrica de la elipse para estimar el punto sobre la cónica, ortogonal al punto a ajustar. Asimismo, se mostró que la evolución diferencial necesita de un número menor de generaciones y tiempo, en comparación con un

²Propuesta aceptada y publicada en el congreso mundial en Computational Intelligence sesión especial Evolutionary Computer Vision (WCCI/CEC) 2008 [184].

³Propuesta aceptada y publicada en el congreso de EvoWorkshops (EvoIASP) 2008 [185].

⁴Propuesta aceptada y publicada en el congreso de EvoWorkshops (EvoIASP) 2007 [186].

algoritmo genético para converger, debido a la representación real (no binaria como el algoritmo genético) y a los operadores que se aplican a la población en la evolución diferencial, los cuales simulan a los métodos de gradiente. Asimismo, se mostró que el tipo de distancia empleada en la función objetivo (euclidiana o su cuadrado) influye en la forma de la cónica resultante. Por otro lado, se encontraron casos (cuando se le incorporan restricciones a la cónica), donde el mejor algoritmo reportado queda atrapado en un óptimo local y la solución propuesta siempre obtiene valores muy cercanos al óptimo global. Finalmente, se desarrolló un algoritmo híbrido que emplea la evolución diferencial para inicializar el algoritmo propuesto por Ahn [28] y obtener la solución óptima global en 10 generaciones con 10 individuos.

Cabe señalar que una de las bondades de estas soluciones es que no requieren una estimación inicial cercana a la solución óptima global para su convergencia. Por otro lado, la evolución diferencial puede lidiar con funciones que no pueden ser tratadas analíticamente, a diferencia de los métodos de gradiente, de tal forma que es posible emplear la distancia euclidiana como función objetivo a minimizar.

7.1. Contribuciones

Las principales contribuciones de esta tesis son las siguientes:

1. Se resolvió, efectivamente, el problema de triangulación. Este problema fue resuelto como un problema de optimización multi-objetivo mediante el algoritmo NSGA-II [29]. Esta solución presenta mejores resultados cuando el nivel de ruido es alto, bajo reconstrucciones proyectivas y afines, en comparación con los mejores algoritmos reportados en la literatura.
2. Se propusieron dos soluciones (SPI y SPII) que permiten calibrar correcta y eficientemente la cámara, estimar los parámetros de orientación y posición, y calcular los puntos tridimensionales *a partir de dos imágenes* adquiridas por una cámara sin calibración previa mediante el uso de la evolución diferencial. Esta contribución es la más importante del trabajo de tesis presentado, debido a que esta solución obtiene reconstrucciones tridimensionales métricas aún cuando sólo exista traslación pura⁵ al adquirir las imágenes o cuando se presenta una superficie crítica (puntos que yacen en el mismo plano).
3. Se propuso la SPIII que permite también calibrar la cámara, estimar los parámetros de orientación y posición, y calcular los puntos tridimensionales en el proceso de la reconstrucción tridimensional, pero esta vez *a partir de tres o más imágenes* adquiridas por una cámara sin calibración previa, mediante el uso de la evolución diferencial.
4. Se propuso una solución efectiva que ataca al problema de encontrar la mejor elipse que se ajuste a un conjunto de puntos dados, sin restricciones y con

⁵Configuración crítica de la cámara donde los algoritmos que emplean la geometría epipolar, no han podido obtener una reconstrucción tridimensional métrica.

restricciones, mediante un algoritmo genético y la evolución diferencial. La solución propuesta está basada en [28], pero tiene la flexibilidad de minimizar la distancia euclidiana o su cuadrado. Esta solución supera los resultados de los mejores algoritmos reportados en la literatura.

7.2. Publicaciones

Enseguida se presentan las publicaciones elaboradas a partir del trabajo realizado en esta tesis. En principio se exhibe el trabajo realizado y, posteriormente, la bibliografía de la publicación.

1. Se elaboró y publicó un artículo donde se muestran los resultados de la solución propuesta que ataca al problema de triangulación como un problema de optimización multi-objetivo. Ver:

Israel Vite Silva, Luis Gerardo de la Fraga, Nareli Cruz Cortés y Gregorio Toscano Pulido. Optimal triangulation in 3D Computer Vision using a multi-objective evolutionary algorithm. In M. Giacobini et al., editor, *EvoWorkshops 2007 (EvoIASP)*, LNCS 4448, pages 330-339, Springer-Verlag, Valencia, España, 2007.

2. Se elaboró y publicó un artículo donde se muestran los resultados de la SPII, la cual permite la reconstrucción tridimensional métrica a partir de dos imágenes en configuraciones y superficies críticas y no críticas. Ver:

Luis Gerardo de la Fraga e Israel Vite Silva. Direct 3D metric reconstruction from two views using differential evolution. In V. Ciesielski et al., editor, *WCCI '08: Proceedings of the IEEE World Congress on Computational Intelligence and Evolutionary Computer Vision*, pages 3265-3272, Springer-Verlag, Hong Kong, China, 2008.

3. Se elaboró y publicó un artículo donde se muestran los resultados de la SPIII, la cual permite obtener la reconstrucción tridimensional métrica a partir de tres o más imágenes. Ver:

Luis Gerardo de la Fraga y Israel Vite Silva. Direct 3D metric reconstruction from multiple views using differential evolution. In M. Giacobini et al., editor, *EvoWorkshops 2008 (EvoIASP)*, LNCS 4974, pages 341-346, Springer-Verlag, Napoli, Italy 2008.

4. Se elaboró y publicó un artículo donde se muestran los resultados obtenidos de la solución propuesta que lidia con el problema de ajuste de puntos a un elipse mediante un algoritmo genético. Ver:

Luis Gerardo de la Fraga, Israel Vite Silva y Nareli Cruz Cortés. Euclidean distance fit of ellipses with a genetic algorithm. In M. Giacobini et al., editor, *EvoWorkshops 2007 (EvoIASP)*, LNCS 4448, pages 359-366, Springer-Verlag, Heidelberg, Berlin, 2007.

5. Se elaboró y fue aceptado un capítulo con los resultados obtenidos de la solución propuesta que ataca el problema de ajuste de puntos a una cónica mediante la evolución diferencial. Ver:

Euclidean distance fit of conics using differential evolution, del libro, *Evolutionary Image Analysis and Signal Processing in their Studies in Computational Intelligence*, Springer-Verlag, 2008. Autores: Luis Gerardo de la Fraga, Israel Vite Silva y Nareli Cruz Cortés.

7.3. Trabajo futuro

De las soluciones propuestas en esta tesis se desprenden posibles trabajos futuros con base en los problemas que se atacaron.

En el problema de la triangulación, sería interesante emplear la evolución diferencial para atacar el problema mono-objetivo o aplicar una versión multi-objetivo de la evolución diferencial para comparar contra el NSGA-II con base en el desempeño y el tiempo de ejecución. Además, se podría analizar la relación entre el ruido de las imágenes y el frente de Pareto cuando la triangulación se plantea como un problema multi-objetivo. Esta relación puede ser utilizada para conocer qué imagen presenta un nivel de ruido mayor o eliminar correspondencias que son incorrectas. También, las soluciones del frente de Pareto pueden ser elegidas si se conoce el ruido que presentan las imágenes. Finalmente, la solución puede extenderse para realizar la triangulación a partir de tres o más puntos bidimensionales que pertenezcan al mismo punto tridimensional. Para ello basta con agregar a la función objetivo, la distancia euclidiana entre los puntos de la siguiente imagen y los puntos bidimensionales estimados.

En el problema de la reconstrucción tridimensional, se podría emplear algún método clásico de optimización para formar un algoritmo híbrido que necesite menos tiempo para converger. Por otro lado, para calcular la matriz de rotación se podrían emplear los cuaterniones o coordenadas exponenciales, esto podría disminuir el tiempo de ejecución y hacer al algoritmo más estable. También sería importante generar un conjunto de experimentos con parámetros bien definidos, que contemplen las configuraciones críticas y las superficies críticas, así como la reconstrucción tridimensional y el error obtenido para tenerlos como referencia de comparación. De igual forma, es necesario profundizar y analizar el tema de la triangulación cuando se está en la segunda configuración crítica, de tal forma que sea posible obtener una reconstrucción tridimensional correcta. Por otra parte, se pueden extender las soluciones propuestas (SPII y SPIII) para lidiar con correspondencias de puntos que no pertenezcan al mismo punto tridimensional o que presenten mucho ruido. Asimismo, sería conveniente que se probara, cuál es el número máximo de imágenes que puede manejar la SPIII sin que el error se incremente a tal grado que la reconstrucción obtenida sea incorrecta, y así realizar las comparaciones pertinentes con algoritmos del estado del arte.

Apéndice A

Ajuste de la mejor elipse a un conjunto de puntos

Otro problema de visión por computadora que fue resuelto en esta tesis es el ajuste de la mejor elipse a un conjunto de puntos dado. Este es un problema que puede verse como un problema simplificado de los algoritmos sobre reconstrucción tridimensional, tratados en esta tesis; conserva las mismas características de ellos, esto es, es un problema geométrico con una función no lineal que puede establecerse como un problema de optimización.

La idea en mente al resolver este problema fue visualizar, de una forma más simple, cómo se comporta un algoritmo evolutivo al resolver un problema geométrico. La hipótesis de trabajo que se tuvo, fue que la metodología usada para resolver este problema del ajuste de la mejor elipse se puede extender a problemas más difíciles.

El ajuste de la mejor elipse a un conjunto puntos fue resuelto, en principio, mediante un algoritmo genético (AG) [186]. Posteriormente, se utilizó la evolución diferencial, la cual emplea una representación real (en punto flotante) de las variables a diferencia de la representación binaria del AG, lo cual constituye un motor de búsqueda más eficiente [132]. La metodología usada puede extenderse a todas las cónicas (hipérbola, parábola y círculo), como se hizo en [186].

El ajuste de la mejor elipse es un problema importante de visión por computadora, ya que la proyección de un círculo es una elipse y se puede usar para recuperar las dimensiones de las llantas de un auto, estimar las deformaciones de los ojos, ajustar la forma de una cara en reconocimiento de rostros, conocer las órbitas de los planetas, entre otras aplicaciones. Los métodos alternativos para la detección y análisis de características geométricas son la transformada de Hough y el método de momentos [187]. En [188] se presenta un algoritmo interesante para detectar elipses usando la transformada de Hough y la evolución diferencial.

A.1. Descripción del problema

El problema de ajuste de la mejor elipse tiene dos posibles soluciones: *el ajuste algebraico* y *el ajuste geométrico* [28]. Una elipse puede ser representada implícitamente por la ecuación general de las cónicas $ax^2 + bxy + cy^2 + dx + ey + f = 0$, si esta ecuación cumple con $b^2 - 4ac < 0$. Cuando se sustituye el valor de un punto (x, y) en la ecuación de la elipse, al valor obtenido de la ecuación implícita se le conoce como *distancia algebraica*. Esta formulación permite generar algoritmos muy eficientes, debido a que

el problema de ajuste es lineal y puede ser resuelto de forma determinista con un método que resuelva el problema generalizado de los vectores propios [189, 190].

La principal desventaja al emplear el ajuste algebraico es que la elipse estará distorsionada si los valores de los puntos no se conocen con suficiente precisión [28]. El mejor ajuste se obtiene al emplear un algoritmo con ajuste geométrico, ya que este ajuste toma en cuenta la distancia euclidiana entre el punto y la elipse a ajustar. Al emplear esta distancia se genera un problema no lineal, el cual puede resolverse mediante la expansión en series de Taylor de la ecuación de la distancia euclidiana, y si se tiene un valor inicial, cercano a la solución real. En este caso, es necesario iterar hasta encontrar la solución óptima global¹. Sin embargo, la distancia euclidiana utiliza la raíz cuadrada y su derivada no es continua en todo el espacio de búsqueda. Al elevar al cuadrado la distancia euclidiana, su derivada se torna continua en todo el espacio de búsqueda. A esta distancia se le conoce como el cuadrado de la distancia euclidiana y puede ser minimizada mediante métodos iterativos de gradiente [29].

La principal desventaja al emplear el cuadrado de la distancia euclidiana es que los puntos más lejanos tienen mayor peso que los puntos más cercanos a la curva al momento de minimizar la distancia. Otra importante desventaja es que los métodos de gradiente necesitan de un valor inicial, cercano a la solución óptima global, para evitar que quede atrapado en un óptimo local.

Por conveniencia y sólo en este apéndice, los puntos en el sistema de coordenadas global se representa con letras mayúsculas, mientras que los puntos en el sistema de coordenadas rotado se representa con minúsculas.

La elipse se caracteriza geoméricamente por cinco parámetros: X_c, Y_c, a, b , y α . $\mathbf{X}_c = (X_c, Y_c)$ es el centro de la elipse, a y b son los semiejes mayor y menor, respectivamente, y α es el ángulo de pose definido entre el semieje mayor y el eje X , como se observa en el diagrama de la figura A.1.

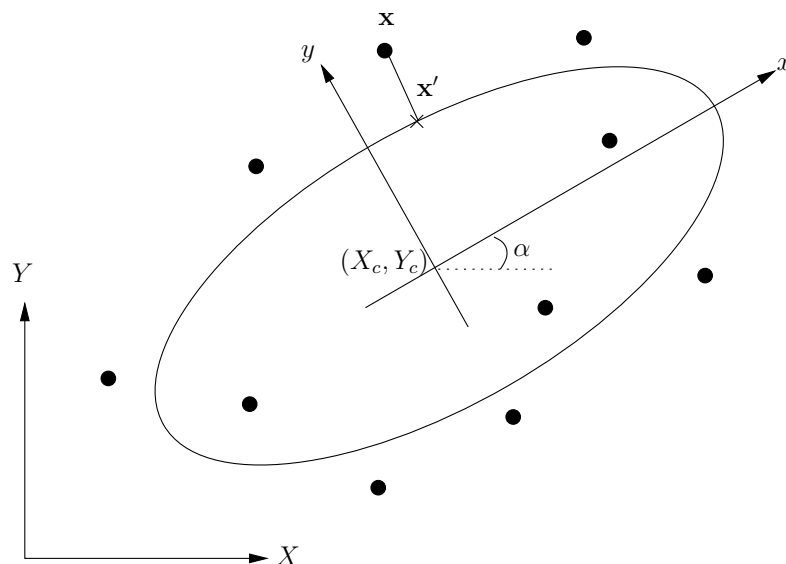


Figura A.1: Ajuste de la mejor elipse a un conjunto de puntos

El mejor algoritmo que realiza el ajuste geométrico de una elipse es el algoritmo

¹La mínima suma de distancias entre los puntos dados y la cónica a ajustar.

de ajuste de distancias ortogonales por mínimos cuadrados (LSODF por sus siglas en inglés) propuesto por Ahn et al., en [28]. Este algoritmo ejecuta los siguientes pasos:

1. Obtener un valor inicial mediante el ajuste algebraico a una elipse a partir de n puntos \mathbf{X}_i , $1 \leq i \leq n$.
2. Calcular los puntos \mathbf{x}'_i , los cuales son los puntos sobre la cónica ortogonales a los puntos \mathbf{x}_i mediante el método de Newton.
3. Mediante el método de Gauss-Newton minimizar la suma del cuadrado de la distancia euclidiana,

$$\sum d^2 = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{x}'_i\|^2. \quad (\text{A.1})$$

En lugar de los pasos 1 y 3, se propuso emplear algoritmos genéticos y la evolución diferencial para resolver el problema de optimización, sujeto a encontrar la mínima suma de las distancias, tanto euclidiana $\sum d$, como el cuadrado de ésta $\sum d^2$.

A.1.1. Cálculo de los puntos sobre la elipse ortogonales a los puntos dados

En [28] emplean la ecuación reducida de la elipse. De esta forma se garantiza que se está ajustando una elipse. La ecuación reducida de la elipse es:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1,$$

Dada una elipse general, puede tratarse con la ecuación reducida de la misma si se traslada el eje de coordenadas al centro de la elipse (X_c, Y_c) , y se rota su ángulo de pose α , como se indica en la figura A.1. Esto es:

$$\mathbf{x}_i = R(\mathbf{X}_i - \mathbf{X}_c), \quad (\text{A.2})$$

donde R es la matriz de rotación representada por

$$R = \begin{bmatrix} \cos \alpha & \text{sen } \alpha \\ -\text{sen } \alpha & \cos \alpha \end{bmatrix}.$$

El siguiente paso es hallar el punto más cercano a la elipse (la elipse ya está en su forma reducida) que sea ortogonal a ella y que pase por el punto dado. Dada la ecuación paramétrica de la elipse en su forma reducida $g(t) = (a \cos t, b \text{sen } t)$, donde t es la variable paramétrica, su derivada es $g'(t) = (-a \text{sen } t, b \cos t)$, la cual representa el vector tangente a la elipse, y el vector normal es $n(t) = (b \cos t, a \text{sen } t)$. Para calcular el valor de t , correspondiente al punto \mathbf{x}' sobre la elipse y que sea normal al punto \mathbf{x} , se debe estimar la ecuación de la línea que pase por el punto \mathbf{x} y que sea normal a la elipse, es decir, $(x - b \cos t, y - a \text{sen } t)$. Esta línea debe intersectarse a la elipse en el punto $\mathbf{x}' = (a \cos t, b \text{sen } t)$, de tal forma que al igualar ambos términos mediante el despeje de t y sustituyéndola en la otra ecuación se obtiene,

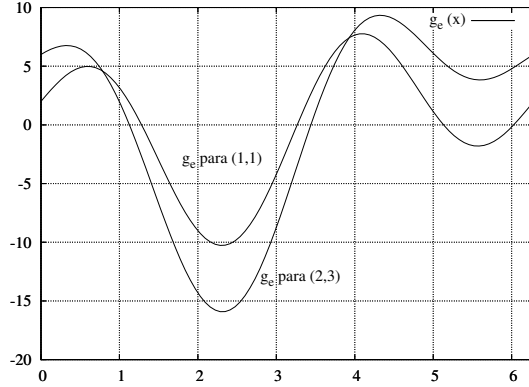


Figura A.2: Ejemplos de la curva con base en la ecuación (A.3), para una elipse con $a = 4$ y $b = 2$. En la figura se muestran los valores del punto dado (x, y) para ambas curvas.

$$g_e(t) = yb \cos t - xa \sin t + (a^2 - b^2)(\cos t)(\sin t) = 0, \quad (\text{A.3})$$

la cual, de forma muy burda, tiene dos soluciones si el punto \mathbf{x} está fuera de la elipse, o cuatro soluciones si el punto está dentro de la elipse. La curva de la función (A.3) presenta una forma muy suave, como se muestra en la figura A.2.

El punto \mathbf{x}' puede calcularse mediante el método de Newton, usando el punto inicial $t_0 = \arctan[(y/a)/(x/b)]$, el cual es calculado por la función *atan2* para obtener su valor en cualquier cuadrante. El método de Newton converge en sólo tres o cuatro iteraciones, debido a la forma muy suave de la curva, como se ve en la figura A.3.

La suma de las distancias puede calcularse en el sistema de coordenadas xy , para evitar transformar el punto calculado sobre la elipse al sistema de coordenadas global XY (ver fig. A.1).

A.2. Resultados al emplear algoritmos evolutivos

El problema de ajuste de la mejor elipse se resolvió mediante dos algoritmos evolutivos: un algoritmo genético (AG) y la evolución diferencial (ED). Se hicieron dos tipos de experimentos, el primero corresponde al ajuste de la mejor elipse sin restricciones y el segundo a una elipse con restricciones.

A.2.1. Ajuste de la mejor elipse sin restricciones

El primer experimento consta de 8 puntos²: (1, 7), (2, 6), (5, 8), (7, 7), (9, 5), (3, 7), (6, 2) y (8, 4). En ambos algoritmos se emplearon los siguientes intervalos para cada una de los parámetros a estimar: $1 \leq a, b \leq 15$, $-10 \leq X_c, Y_c \leq 10$ y $0 \leq \alpha \leq \pi/2$.

En el algoritmo genético (AG) se empleó una precisión de 13 decimales, de modo que la longitud total del cromosoma quedó de 234 bits (47 bits para la variable a , 47 bits para la variable b , X_c y Y_c requieren de 48 bits cada uno, y α necesita 44 bits). La aptitud de los individuos se calcula con la ecuación (A.1) para el cuadrado de la distancia euclidiana o la raíz de esta ecuación para la distancia euclidiana.

²Este conjunto de puntos fue tomado de los experimentos realizados en [28].

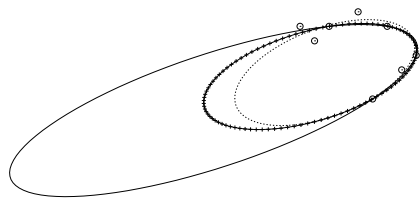


Figura A.3: Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en la distancia euclidiana (excepto el LSODF).

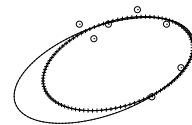


Figura A.4: Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en el cuadrado de la distancia euclidiana.

Los parámetros del AG que ofrecen mejores resultados para este problema son: 100 individuos en la población, 3000 generaciones, cruce de un punto, porcentaje de cruce de 0.8, sobrante estocástico como tipo de selección, representación binaria con código Gray y porcentaje de mutación de $1/L$, donde L es la longitud del cromosoma en bits.

La función objetivo de la evolución diferencial es la misma que la del AG, la representación de las variables es real y los parámetros empleados son: tamaño de población de 20, 300 generaciones, porcentaje de recombinación de 0.8 y constante de diferencia de 0.8. En la tabla A.1 y A.2 se observan los resultados del algoritmo propuesto por Ahn [28] (LSODF) y los algoritmos evolutivos a partir de 40 ejecuciones para el caso de la distancia euclidiana y el cuadrado de la misma, respectivamente.

Al emplear la distancia euclidiana, no puede usarse el algoritmo LSODF. El algoritmo genético, en su mejor ejecución, obtiene un valor cercano a la solución óptima global, pero con base en las demás ejecuciones y a la desviación estándar (d.e.), presenta ciertas dificultades para converger a una solución cercana al óptimo global. En cambio, la evolución diferencial es más estable y encuentra soluciones muy cercanas al óptimo global en todas las ejecuciones que se realizaron. Además, la evolución diferencial sólo necesita 6,000 evaluaciones para obtener estos resultados, mientras que el algoritmo genético necesita 300,000 evaluaciones. En las figuras A.5 y A.6 se muestran las gráficas de convergencia de ambos algoritmos evolutivos, en las cuales se observa que la evolución diferencial obtiene la solución cercana al óptimo global en menos generaciones y en un tiempo mucho menor que el algoritmo genético.

Al emplear el cuadrado de la distancia euclidiana se presenta el mismo efecto que con la distancia euclidiana para los algoritmos evolutivos. Es decir, la evolución diferencial genera mejores resultados y en menos generaciones. En cuanto al algoritmo LSODF, éste llega a la solución óptima global; esta solución también se presenta en la mejor ejecución del AG y ED, y en la media de la ED.

En las figuras A.3 y A.4 se muestran las reconstrucciones de la elipse cuando se emplea como función objetivo la distancia euclidiana y el cuadrado de ésta, respectivamente. Las elipses reconstruidas fueron obtenidas por los tres algoritmos, en relación a los valores más cercanos a la media para el caso de los algoritmos evolutivos.

En el caso de la distancia euclidiana, la cónica obtenida por los algoritmos evolutivos es totalmente diferente a la cónica que se obtiene con el algoritmo LSODF,

A.2 Resultados al emplear algoritmos evolutivos

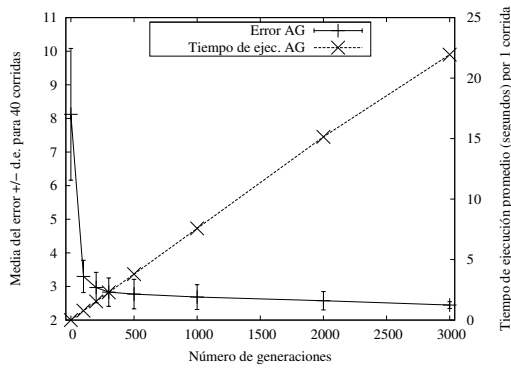


Figura A.5: Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del AG.

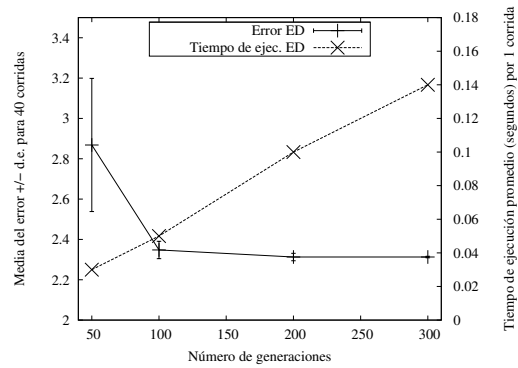


Figura A.6: Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del ED.

Algoritmo	Tipo	a	b	x_c	y_c	α	Error
LSODF		6.518	3.031	2.699	3.815	0.359	2.727
AG	Mejor	14.404	3.863	-4.664	1.364	0.314	2.316
	Media	7.589	3.083	1.701	3.543	0.286	2.447
	Peor	3.239	2.967	4.904	4.769	0.162	2.664
	d.e.	3.254	0.294	2.805	1.051	0.178	0.102
	i.c.	[7.747,9.567]	[3.185,3.348]	[-0.125,1.392]	[2.875,3.475]	[0.284,0.383]	[2.423,2.478]
ED	Mejor	14.948	3.910	-5.153	1.144	0.319	2.303
	Media	14.739	3.891	-4.956	1.199	0.320	2.308
	Peor	14.686	3.884	-4.881	1.182	0.324	2.315
	d.e.	0.230	0.024	0.214	0.080	0.001	0.004
	i.c.	[14.619,14.827]	[3.877,3.898]	[-5.036,-4.852]	[1.182,1.252]	[0.318,0.320]	[2.306,2.310]

Tabla A.1: Resultados del ajuste de la mejor elipse sin restricciones con base en la distancia euclidiana.

Algoritmo	Tipo	a	b	x_c	y_c	α	Error
LSODF		6.518	3.031	2.699	3.815	0.359	1.373
AG	Mejor	6.500	3.030	2.714	3.823	0.359	1.373
	Media	5.337	2.929	3.657	4.282	0.315	1.386
	Peor	13.850	3.786	-3.658	0.856	0.411	1.410
	d.e.	2.565	0.263	2.201	1.034	0.028	0.011
	i.c.	[8.108,9.478]	[3.199,3.343]	[0.145,1.411]	[2.614,3.194]	[0.372,0.387]	[1.383,1.390]
ED	Mejor	6.516	3.031	2.701	3.816	0.359	1.373
	Media	6.516	3.031	2.701	3.816	0.359	1.373
	Peor	6.705	3.296	2.510	3.786	0.393	1.386
	d.e.	0.654	0.066	0.554	0.260	0.008	0.0
	i.c.	[6.521,6.956]	[3.031,3.775]	[-2.992,2.677]	[1.055,3.822]	[0.359,0.420]	[1.373,1.380]

Tabla A.2: Resultados del ajuste de la mejor elipse sin restricciones con base en el cuadrado de la distancia euclidiana.

e incluso difiere de la cónica obtenida al emplear el cuadrado de la distancia euclidiana. Con este experimento se puede comprobar que sí existe una diferencia en los resultados al emplear la distancia euclidiana en lugar del cuadrado de la misma.

A.2.2. Ajuste de la mejor elipse con restricciones

En el ajuste de la mejor elipse con restricciones también se deben estimar los cinco parámetros que conforman a la elipse (X_c, Y_c, a, b, α), pero se deben cumplir las siguientes dos restricciones: $\alpha = \pi/2$ y $ab = 20$. Estas restricciones se introdujeron en

A. AJUSTE DE LA MEJOR ELIPSE A UN CONJUNTO DE PUNTOS

ambos algoritmos evolutivos de la siguiente manera: el valor del ángulo deja de ser una variable y se fija al valor requerido $\alpha = \pi/2$. Una de las dos longitudes de los ejes (a o b) se deja como variable y la otra en relación a ésta, por ejemplo, a se deja como variable y b en relación a ella mediante $b = 20/a$ para cumplir con la segunda restricción. Por lo tanto, sólo restan tres variables a , X_c y Y_c a estimar.

El conjunto de puntos para realizar este experimento es: $(8, 1)$, $(3, 6)$, $(2, 3)$, $(7, 7)$, $(6, 1)$, $(6, 10)$ y $(4, 0)$, los cuales se tomaron de los experimentos presentados en [28]. En ambos algoritmos, los límites de las variables son los mismos que en el experimento anterior, así como sus parámetros, excepto por el número de generaciones, el cual se redujo a 1000 generaciones para el algoritmo genético y a 100 generaciones para la evolución diferencial. Asimismo, la longitud del cromosoma en el algoritmo genético se decrementó a 143 bits, debido a que se eliminaron dos variables.

En la tabla A.3 y A.4 se presentan los resultados del algoritmo LSODF y los algoritmos evolutivos a partir de 40 ejecuciones para el caso de la distancia euclidiana y el cuadrado de ésta, respectivamente. Tanto el algoritmo genético como la evolución diferencial presentan muy buenos resultados para ambas distancias, con una desviación estándar muy pequeña (menor a 0.0001). Esto se traduce en que ambos algoritmos son muy estables. Sin embargo, el error que se obtiene con la evolución diferencial es menor, a nivel de decimales, que los resultados del AG. En este experimento, el algoritmo LSODF no converge a la solución óptima global o en su vecindad, debido a la inicialización que emplea.

Algoritmo	Tipo	a	b	x_c	y_c	α	Error
LSODF		4.580	4.367	6.176	4.467	1.898	49.792
AG	Mejor	8.569	2.333	5.081	2.122	1.57	3.956
	Media	8.570	2.333	5.081	2.121	1.57	3.956
	Peor	8.561	2.561	5.078	2.133	1.57	3.961
	d.e.	0.0004	0.0004	0.0004	0.0016	0.0	0.0007
	i.c.	[8.569,8.570]	[2.333,2.333]	[5.080,5.081]	[2.121,2.122]	[1.57,1.57]	[3.956,3.956]
ED	Mejor	8.569	2.333	5.081	2.122	1.57	3.935
	Media	8.569	2.333	5.081	2.122	1.57	3.935
	Peor	8.569	2.333	5.081	2.122	1.57	3.935
	d.e.	0.0	0.0	0.0	0.0	0.0	0.0
	i.c.	[8.569,8.569]	[2.333,2.333]	[5.081,5.081]	[2.122,2.122]	[1.57,1.57]	[3.935,3.935]

Tabla A.3: Resultados del ajuste de la mejor elipse sin restricciones utilizando como función la distancia euclidiana.

Algoritmo	Tipo	a	b	x_c	y_c	α	Error
LSODF		4.580	4.367	6.176	4.467	1.898	358.048
AG	Mejor	9.532	2.098	4.988	2.746	1.57	3.982
	Media	9.532	2.098	4.988	2.746	1.57	3.982
	Peor	9.532	2.098	4.988	2.746	1.57	3.982
	d.e.	0.0	0.0	0.0	0.0	0.0	0.0
	i.c.	[9.532,9.532]	[2.098,2.098]	[4.988,4.988]	[2.746,2.746]	[1.57,1.57]	[3.982,3.982]
ED	Mejor	9.394	2.128	4.947	3.144	1.57	3.950
	Media	9.394	2.128	4.947	3.144	1.57	3.950
	Peor	9.394	2.128	4.947	3.144	1.57	3.950
	d.e.	0.0	0.0	0.0	0.0	0.0	0.0
	i.c.	[9.394,9.394]	[2.128,2.128]	[4.947,4.947]	[3.144,3.144]	[1.57,1.57]	[3.950,3.950]

Tabla A.4: Resultados del ajuste de la mejor elipse sin restricciones utilizando como función el cuadrado de la distancia euclidiana.

En las figuras A.7 y A.8 se muestra la reconstrucción de la elipse cuando se emplea como función objetivo la distancia euclidiana y el cuadrado de ésta, respectivamente.

Asimismo, en las figuras A.9 y A.10 se muestran las gráficas de convergencia de ambos algoritmos evolutivos, en ellas se observa de nuevo que la evolución diferencial obtiene la solución cercana a la óptima global en menos evaluaciones y en un tiempo mucho menor que el algoritmo genético. Al observar la desviación estándar, se aprecia que hay mayor estabilidad en las soluciones obtenidas con la ED.

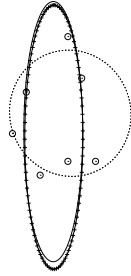


Figura A.7: Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en la distancia euclidiana (excepto el LSODF).

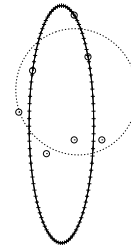


Figura A.8: Elipses reconstruidas por: LSODF (línea punteada), AG (línea con cruces) y ED (línea sólida); con base en el cuadrado de la distancia euclidiana.

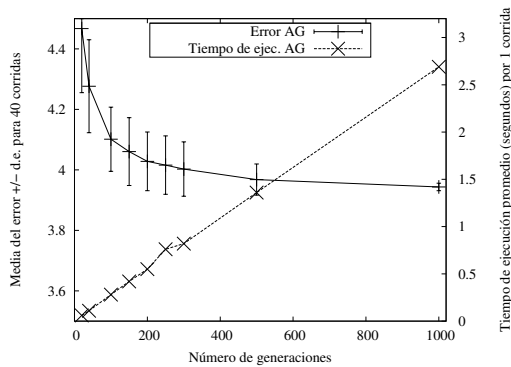


Figura A.9: Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones del algoritmo AG.

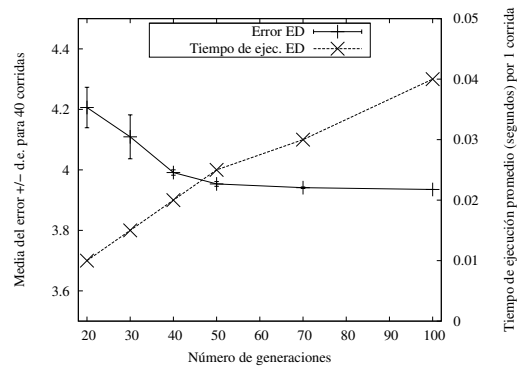


Figura A.10: Media del error \pm la desviación estándar y el tiempo de ejecución, para 40 ejecuciones de la ED.

A.3. Algoritmo híbrido

El tiempo de ejecución de los algoritmos evolutivos es alto comparado con el método de Newton, el cual es empleado en el algoritmo LSODF [28]. El problema del método de Newton es que necesita un valor inicial cercano a la solución óptima global. Los inconvenientes de ambos algoritmos (tiempo y convergencia) se reducen si se combinan ambos algoritmos, es decir, si se plantea un algoritmo híbrido; por un lado el algoritmo evolutivo se emplea para generar un valor inicial cercano a la solución óptima global y, por otro lado, el algoritmo LSODF se emplea para hacer una búsqueda local y encontrar la solución óptima global.

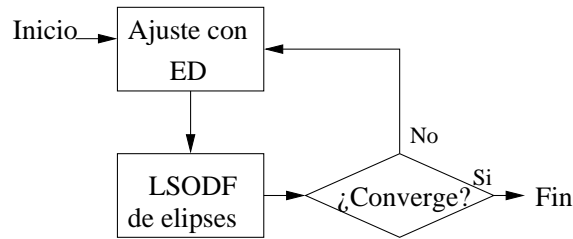


Figura A.11: Esquema del algoritmo híbrido

Esta metodología se esquematiza en el diagrama a bloques mostrado en la figura A.11, y su forma de operar es la siguiente:

1. Ejecutar la evolución diferencial cierto número de generaciones.
2. El mejor individuo enviarlo al algoritmo LSODF como valor inicial.
3. Ejecutar cierto número de iteraciones hasta que, la diferencia entre el valor anterior y el valor actual de cada variable, sea menor a 1×10^{-6} ; cuando este caso se presenta, se dice que el algoritmo ha convergido y termina la búsqueda.
4. En caso de no cumplirse la condición anterior después de cierto número de iteraciones, se ejecuta el paso 1, con la misma población.
5. Este proceso se repite hasta que el algoritmo LSODF converja.

A.4. Resultados al emplear el algoritmo híbrido

El algoritmo híbrido se probó sobre los puntos que pertenecen al experimento de la elipse con restricciones; experimento donde el algoritmo LSODF no converge.

Los parámetros que se emplearon en la evolución diferencial son los siguientes: 10 individuos, 10 generaciones, porcentaje de recombinación de 0.8 y constante de diferencia de 0.8. En el caso del algoritmo LSODF el límite de iteraciones es de 10. El algoritmo híbrido se ejecutó 50 veces para este experimento. De las 50 ejecuciones, 46 convergieron en tan sólo un ciclo del algoritmo híbrido, las 4 restantes necesitaron de dos ciclos. Las elipses estimadas en estas ejecuciones son idénticas a la elipse que se muestra en la figura A.8, la cual fue obtenida por la evolución diferencial con base en el cuadrado de la distancia euclidiana.

Apéndice B

Experimentos con la SPII en configuraciones no críticas

En el capítulo de resultados se mostraron parte de los experimentos realizados con la SPII en configuraciones no críticas. En este apéndice se presentan los experimentos restantes en configuraciones no críticas. Estos experimentos parten de correspondencias de puntos adquiridas de imágenes sintéticas con diferentes niveles de ruido y en diversas superficies (incluso en las superficies críticas). Asimismo, parten de correspondencias de puntos que se extrajeron de imágenes reales.

B.1. Reconstrucción tridimensional al usar puntos de imágenes sintéticas

Continuando con la numeración de los experimentos mostrados en el capítulo de resultados, *el tercer experimento* consiste en medir el desempeño del algoritmo al momento de calibrar la cámara. Para este experimento sólo se toma la longitud del foco como parámetro de la matriz de calibración a medir. El escenario es un conjunto de puntos que pertenecen a una esfera, con diferentes niveles de ruido y bajo un configuración no crítica.

Los valores empleados para generar las matrices de proyección se presentan en la tabla B.1. A partir de las matrices de proyección y los puntos tridimensionales de la esfera se generaron las imágenes que se muestran en las figuras B.1a y B.1b.

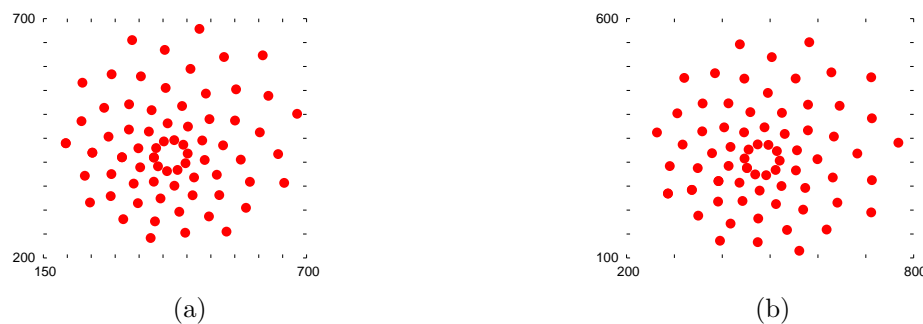


Figura B.1: Imágenes de la esfera: (a) imagen 1, (b) imagen 2.

Para este experimento, y por cada ejecución, se tomaron aleatoriamente 24 de 74 correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que va de

B.1 Reconstrucción tridimensional al usar puntos de imágenes sintéticas

Imagen	α	β	γ	t_x	t_y	t_z	f
1	2	20	10	7	12	22	700
2	30	5	20	8	5	22	700

Tabla B.1: Parámetros de la cámara utilizados para el tercer experimento.

1 a 3 píxeles. Los parámetros empleados para la evolución diferencial son los siguientes: 5,000 generaciones, 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

El número de variables que se emplean en este experimento es trece; seis variables que generan las dos matrices de rotación, seis variables que generan los dos vectores de traslación y una variable que corresponde a la longitud del foco.

El espacio de búsqueda donde opera el algoritmo de evolución diferencial está definido por el intervalo de cada variable. Para las variables que generan las dos matrices de rotación, el intervalo es de $[0, \pi]$; para las variables que conforman los dos vectores de traslación, el intervalo es de $[-1, 1]$; para la longitud del foco, el intervalo que se estableció es de $[10, 1000]$.

De este experimento se realizaron 40 ejecuciones para cada nivel de ruido. En la tabla B.2 se muestran las estadísticas del error de estas ejecuciones.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	Sin ruido	0.0	0.0	0.00002	0.0	[0.0,0.0]
	1 píxel	0.58780	0.80435	0.98706	0.10716	[0.76486,0.84341]
	2 píxeles	1.16976	1.47396	1.72679	0.16810	[1.40683,1.53503]
	3 píxeles	1.43522	1.91899	2.35499	0.24665	[1.83162,1.99680]

Tabla B.2: Estadísticas del error a partir de 40 ejecuciones de la SPII.

Puntos	Ruido	Mínimo	Media	Máximo	d.e.	i.c. de la media
24	Sin ruido	699.99	699.99	700.00	0.00	[699.99, 700.00]
	1 píxel	661.55	716.31	739.99	23.36	[708.00,724.27]
	2 píxeles	656.10	719.81	744.20	39.75	[704.32,731.94]
	3 píxeles	651.40	716.03	749.95	30.73	[705.88,725.14]

Tabla B.3: Estadísticas del parámetro intrínseco (longitud del foco).

	Sin ruido	1 píxel	2 píxeles	3 píxeles
Mínimo	0.01 %	5.49 %	6.27 %	6.9429 %
Media	0.01 %	2.33 %	2.83 %	2.2900 %
Máximo	0.0 %	5.71 %	6.31 %	7.1357 %

Tabla B.4: Porcentaje del error relativo al comparar la longitud del foco real y su valor estimado con diferentes niveles de ruido.

La finalidad del tercer experimento es mostrar la diferencia entre el valor real y el valor estimado del parámetro intrínseco de la cámara (longitud del foco). El valor estimado de la longitud del foco se extrae de la salida de la SPII. En primer lugar se dan a conocer las estadísticas del valor estimado (ver tabla B.3), el cual es muy similar al valor real, aún en presencia de ruido. Posteriormente, se muestra el porcentaje de error relativo en la tabla B.4. Este porcentaje se calcula en función de la diferencia entre el valor real y el valor estimado para los diferentes niveles de ruido sobre el intervalo de cada variable.

B. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES NO CRÍTICAS

El cuarto experimento es similar al anterior, con la diferencia de que en este experimento se miden todos los parámetros de la cámara. El escenario es un conjunto de correspondencias de puntos que pertenecen a una esfera, con diferentes niveles de ruido y bajo un configuración no crítica. Los valores empleados para generar las matrices de proyección se presentan en la tabla B.5.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	20	24	15	1	5	20	150
2	24	60	10	15	5	14	150

Tabla B.5: Parámetros de la cámara utilizados para el cuarto experimento.

Para poder realizar esta medición se deben tomar las matrices de proyección en su forma canónica, es decir, la primer matriz de proyección está dada por $M_c = K[I \mid \mathbf{0}]$ y la segunda matriz de proyección $M'_c = K[R' - R \mid s(\mathbf{t}' - \mathbf{t})]$, donde s es un factor de escala para normalizar el vector resultante. De tal forma que los parámetros de la tabla B.1, se expresan de la siguiente forma:

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	0	150
2	4	36	-5	0.7	0	-0.3	150

Tabla B.6: Parámetros de la cámara en su forma canónica.

En esta prueba se utilizaron n correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados en la evolución diferencial son los mismos que en el experimento anterior. El número de variables que se emplean en este experimento es trece y el intervalo de cada variable es el mismo que en la prueba anterior.

De este experimento se realizaron 40 ejecuciones por cada 8, 12, 16, 20 y 24 correspondencias de puntos sin ruido y con ruido gaussiano RMS de 1, 2 y 3 píxeles. Las estadísticas del error son similares a los resultados anteriores. En cuanto a los parámetros, en la tabla B.7 se presentan los valores estimados de los parámetros de la cámara de acuerdo al número de correspondencias de puntos y el nivel de ruido en las correspondencias de puntos. Estos valores son la salida del algoritmo, los cuales fueron tomados de la ejecución más cercana a la media respecto al error. De esta tabla de valores se percibe que la SPII produce resultados muy cercanos a los valores de los parámetros reales cuando se tiene una mayor correspondencia de puntos. No obstante, cuando se tiene poca correspondencia de puntos y el ruido se incrementa, la diferencia entre los valores reales y los estimados también se incrementa, aunque esto no provoca que la reconstrucción de los objetos se degrade como se mostró en los experimentos anteriores. Para visualizar mejor los resultados obtenidos en este experimento, se presentan unas gráficas (B.2 y B.3) que muestran el porcentaje de error de acuerdo a la diferencia entre los valores reales y los valores estimados.

El quinto experimento expone la comparación entre la SPII y un método del estado del arte llamado *método convencional* [5, 4]. Este método emplea el algoritmo de Sturm et al. [26] para calibrar la cámara, el cual emplea formulaciones que se derivan de la ecuaciones de Kruppa. Es posible emplear otros algoritmos que usen la matriz fundamental para calibrar la cámara, no obstante, el resultado es el mismo. El escenario es un conjunto de puntos que pertenecen a seis figuras sintéticas con diferentes

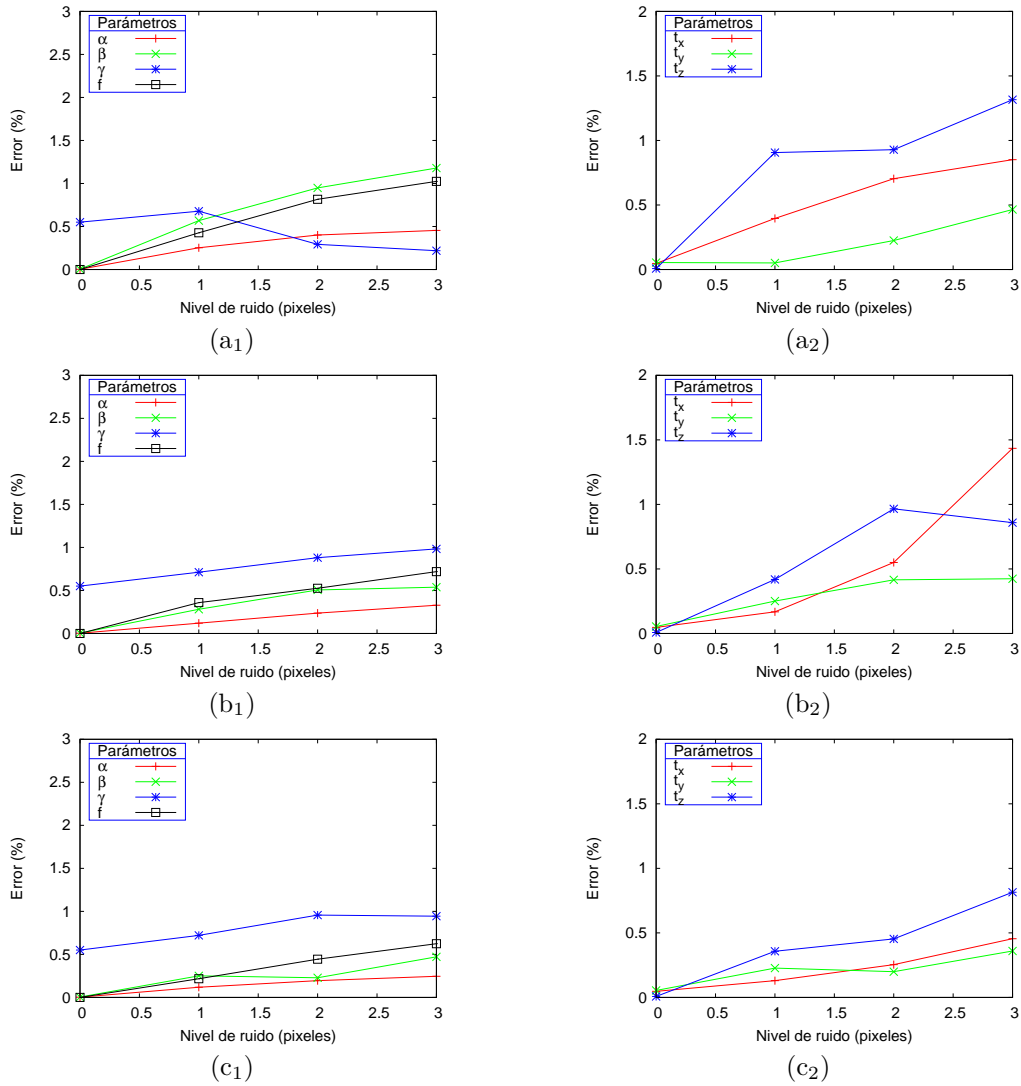


Figura B.2: Gráficas del porcentaje de error relativo de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado para: 8 (a_1 , a_2), 12 (b_1 , b_2) y 16 (c_1 , c_2) correspondencias de puntos.

B. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES NO CRÍTICAS

8 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
	Sin ruido	3.99	35.99	-5.99	0.699	0.001	-0.299	149.99
1 píxel	3.54	37.02	-6.21	0.707	0.001	-0.281	145.94	
2 píxeles	3.27	37.70	-5.52	0.714	0.004	-0.281	142.24	
3 píxeles	3.18	38.12	-5.39	0.717	0.009	-0.273	140.26	
12 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
	Sin ruido	3.99	35.99	-5.99	0.699	0.001	-0.299	149.99
1 píxel	3.78	36.50	-6.28	0.703	0.005	-0.291	146.59	
2 píxeles	3.57	36.91	-6.58	0.711	0.008	-0.280	145.00	
3 píxeles	3.40	36.96	-6.76	0.728	0.008	-0.282	143.17	
16 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
	Sin ruido	3.99	35.99	-5.99	0.699	0.001	-0.299	149.99
1 píxel	3.78	36.45	-6.29	0.702	0.004	-0.292	152.06	
2 píxeles	3.64	36.41	-6.72	0.705	0.003	-0.290	154.22	
3 píxeles	3.55	36.84	-6.69	0.709	0.007	-0.283	155.93	
20 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
	Sin ruido	3.99	35.99	-5.99	0.699	0.001	-0.299	149.99
1 píxel	3.78	36.27	-6.28	0.702	0.004	-0.294	152.02	
2 píxeles	3.67	36.38	-6.41	0.713	0.003	-0.295	154.32	
3 píxeles	3.61	36.99	-6.88	0.720	0.007	-0.288	155.67	
24 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
	Sin ruido	3.99	35.99	-5.99	0.699	0.001	-0.299	149.99
1 píxel	3.93	36.96	-6.39	0.699	0.009	-0.291	151.49	
2 píxeles	3.82	36.88	-6.63	0.707	0.007	-0.284	153.03	
3 píxeles	3.75	37.06	-6.76	0.718	0.008	-0.282	154.17	

Tabla B.7: Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido y correspondencias de puntos.

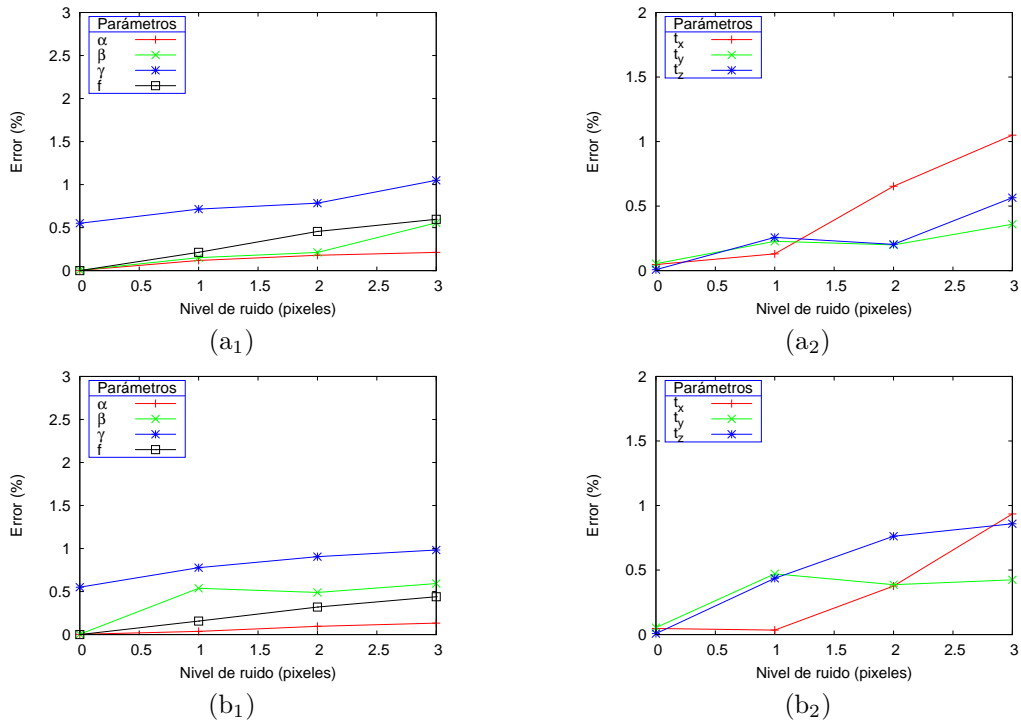


Figura B.3: Gráficas del porcentaje de error relativo de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado para: 20 (a_1, a_2) y 24 (b_1, b_2) correspondencias de puntos.

superficies y formas, sin ruido y bajo un configuración no crítica. Estas figuras son: una media esfera, un cubo, un figura regular, un barco, parte de un anillo y un objeto formado por cubos (ver fig. B.4).

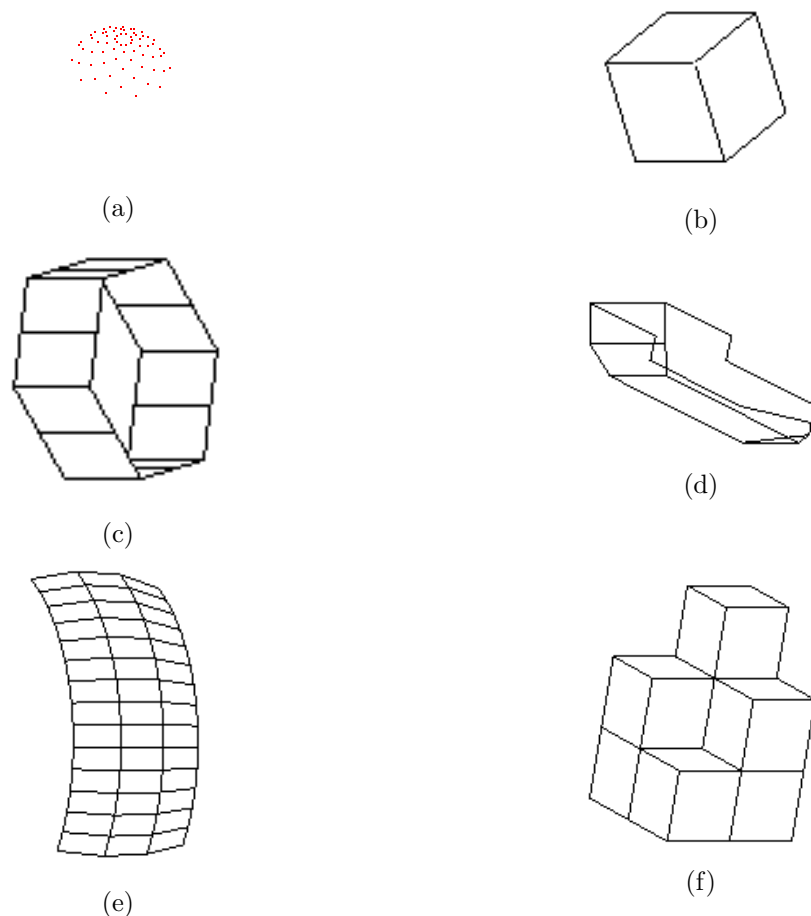


Figura B.4: Objetos empleados en los experimentos: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.

Para este experimento, la matriz de calibración que se utiliza en ambas cámara, es de la siguiente forma:

$$K = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix},$$

por lo tanto, los valores asignados son las coordenadas del punto principal (o_x, o_y) y la longitud del foco f . Los valores empleados para generar las matrices de proyección se presentan en la tabla B.1.

En la figuras B.5, B.6, B.7, B.8, B.9 y B.10 se muestran las dos imágenes sintéticas que se generaron por cada uno de los objetos.

En esta prueba se utilizaron diferente número de correspondencias de puntos sin ruido. Los parámetros para la evolución diferencial se mantienen igual a los parámetros de experimentos anteriores, aunque para esta prueba, se agregaron dos variables que conforman al punto principal y pertenecen a la matriz de calibración. Esto da un total de 15 variables. El intervalo de cada variable es el mismo que en los experimentos anteriores; para las dos variables que se agregaron, los intervalos son los siguientes: o_x es de $[250, 350]$ y o_y es de $[150, 250]$. Se eligieron estos intervalos en consecuencia de que el punto principal está en el centro de la imagen para la mayoría de las cámaras

B. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES NO CRÍTICAS

Objeto	Imagen	Puntos	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Media esfera	1	74	10	20	30	3	1	25	700	300	200
Media esfera	2	74	25	5	18	5	7	35	700	300	200
Cubo	1	8	1	21	35	6	11	30	600	300	200
Cubo	2	8	13	31	22	1	4	30	600	300	200
Fig. regular	1	24	22	33	17	2	3	25	500	300	200
Fig. regular	2	24	41 s	20	18	6	2	25	500	300	200
Barco	1	26	19	50	23	7	7	28	700	300	200
Barco	2	26	20	24	5	1	1	28	700	300	200
Anillo	1	60	5	45	9	2	4	30	350	300	200
Anillo	2	60	1	11	2	0	2	30	350	300	200
Fig. cubos	1	29	23	22	34	0	0	24	800	300	200
Fig. cubos	2	29	9	35	23	2	0	24	800	300	200

Tabla B.8: Parámetros de la cámara utilizados para generar las dos imágenes de cada una de las 6 figuras sobre configuraciones no críticas.

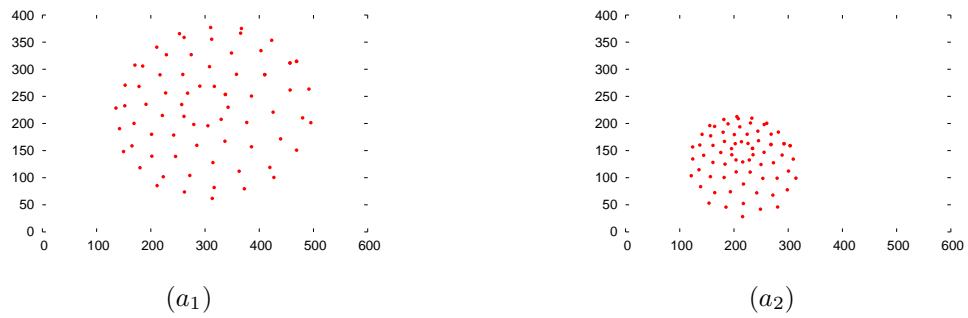


Figura B.5: Media esfera: (a₁) primera imagen y (a₂) segunda imagen.

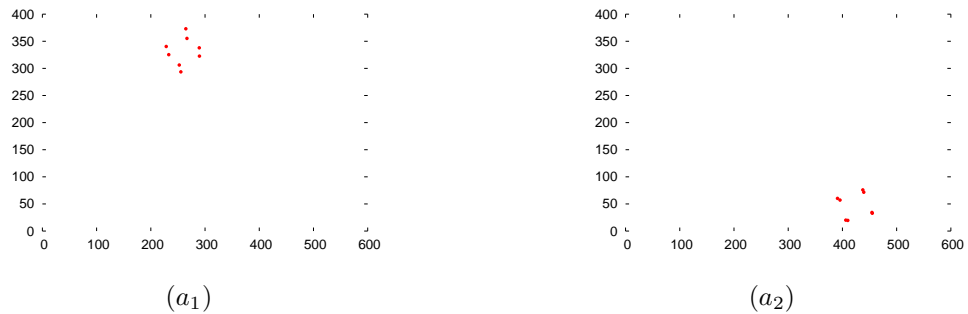


Figura B.6: Cubo: (a₁) primera imagen y (a₂) segunda imagen.

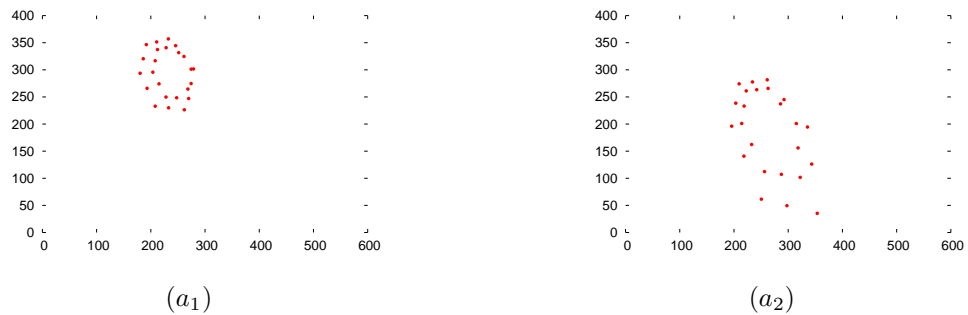


Figura B.7: Figura regular: (a₁) primera imagen y (a₂) segunda imagen.

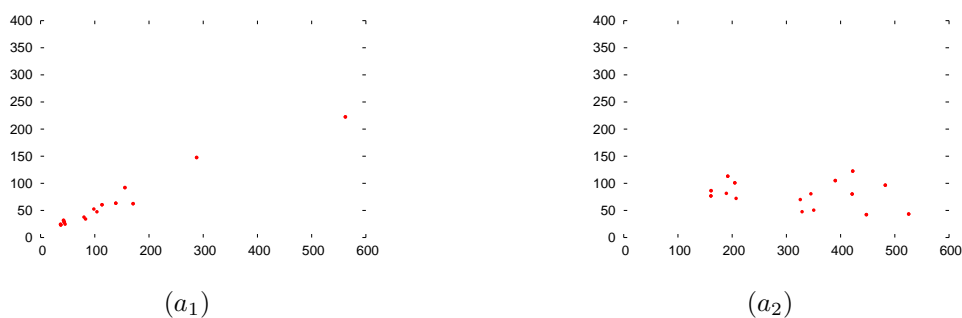


Figura B.8: Barco: (a_1) primera imagen y (a_2) segunda imagen.

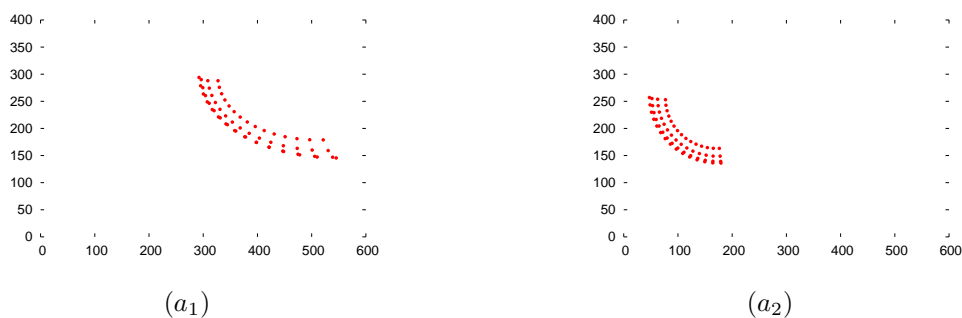


Figura B.9: Parte de un anillo: (a_1) primera imagen y (a_2) segunda imagen.

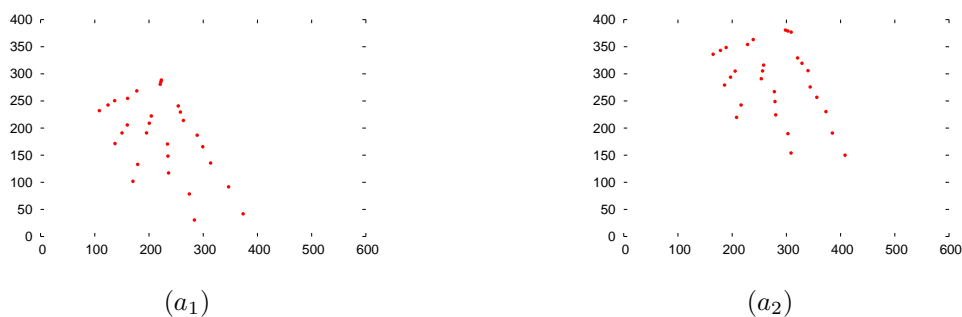


Figura B.10: Figura hecha de cubos: (a_1) primera imagen y (a_2) segunda imagen.

actuales [5, 4]. Por cada objeto, la SPII se ejecutó 30 veces y del método convencional se realizó sólo una ejecución, ya que es un algoritmo determinista. Los objetos reconstruidos por la SPII se muestran en las figuras B.11a, B.11b, B.11c, B.11d, B.11e y B.11f, y los objetos reconstruidos por el método convencional se muestran en las figuras B.12a, B.12b, B.12c, B.12d, B.12e y B.12f.

En estas figuras se observa que los objetos reconstruidos por la SPII son una reconstrucción muy similar a los objetos reales, es decir, no se distingue entre el objeto real y el reconstruido. Por parte del método convencional se percibe que, cuando se utilizan pocos puntos, la reconstrucción es deficiente como sucede con el cubo (ver fig. B.12b).

El sexto experimento muestra el desempeño de la SPII sobre puntos que yacen sobre un mismo plano (superficie crítica). Cuando esto sucede, la matriz fundamental ya no es única, y no es posible obtener una reconstrucción tridimensional métrica [4].

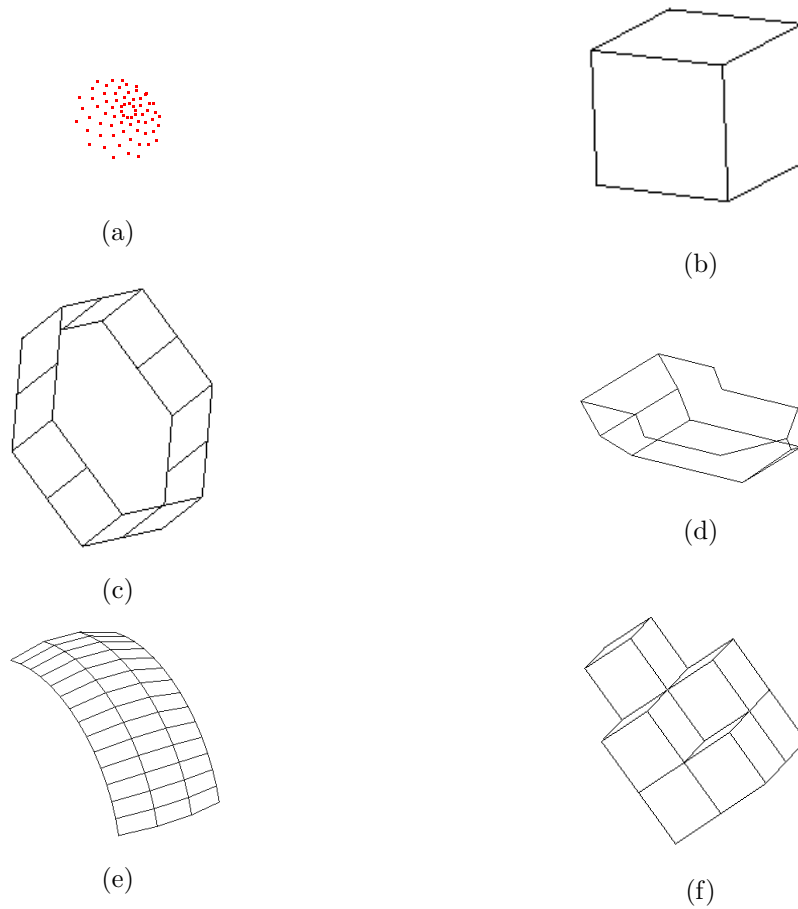


Figura B.11: Objetos tridimensionales reconstruidos por la SPII: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.

Sin embargo, la SPII no tiene este problema como se muestra en este experimento. El escenario es un conjunto de puntos que pertenecen a una cuadrícula (la cuadrícula yace sobre un plano), con diferentes niveles de ruido y bajo un configuración no crítica.

Los valores empleados para generar las matrices de proyección se presentan en la tabla B.9. A partir de las matrices de proyección y los puntos tridimensionales de la cuadrícula se generaron las imágenes que se muestran en las figuras B.13a y B.13b.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	10.000	5.000	20.000	6.000	9.000	10.000	350.000
2	6.000	18.000	7.000	11.000	5.000	12.000	350.000

Tabla B.9: Parámetros de la cámara utilizados para el sexto experimento.

El conjunto de puntos se conforma de 24 correspondencias. A estos puntos se les añadió k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados para la evolución diferencial son los mismos que en el experimento anterior, al igual que el número de variables y sus intervalos. El algoritmo se ejecutó 40 veces para cada nivel de ruido. En la tabla B.1 se muestran las estadísticas del error; los resultados son similares a los resultados obtenidos en el experimento dos.

En la tabla B.11 se muestran las estadísticas del parámetro intrínseco (longitud del foco) estimado por el algoritmo. La diferencia entre la longitud del foco real y el

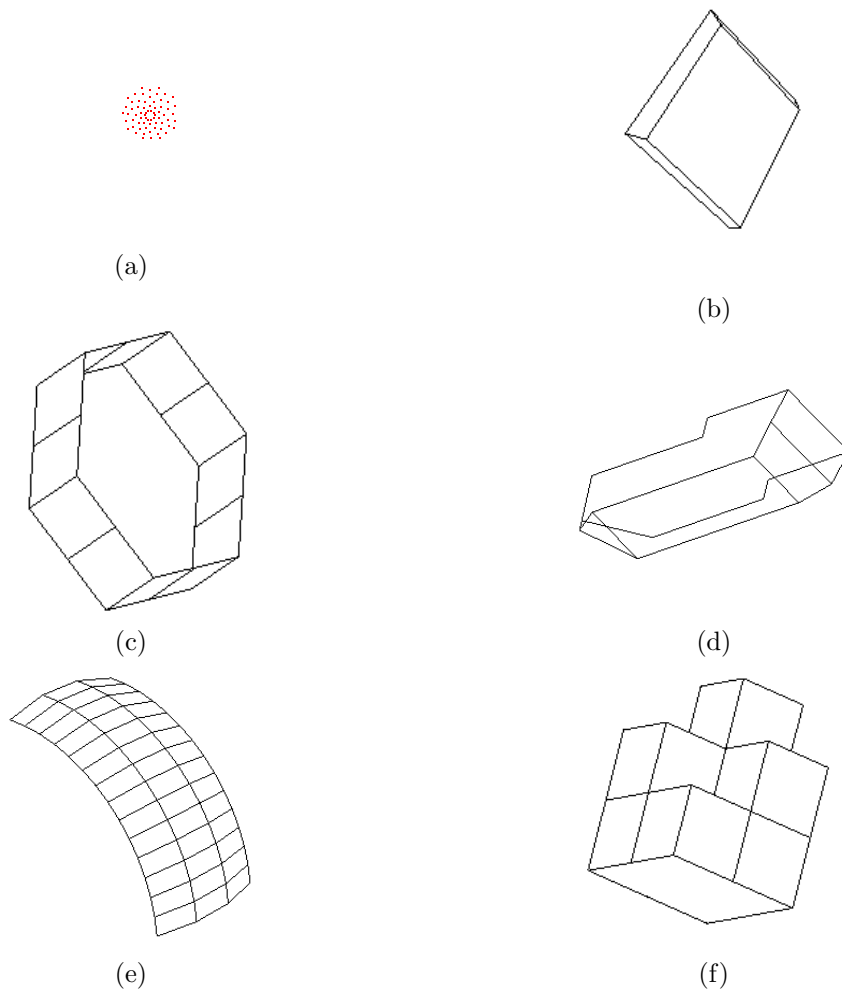


Figura B.12: Objetos reconstruidos por el método convencional: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.

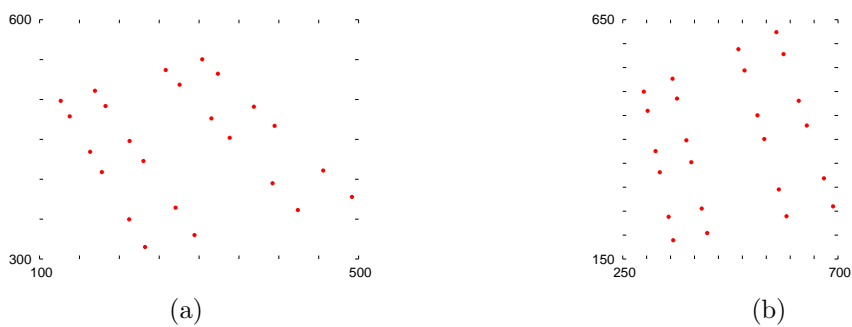


Figura B.13: Cuadrícula. (a) imagen 1 y (b) imagen 2.

estimado, se incrementó en este experimento. Sin embargo, este incremento parece no afectar la reconstrucción tridimensional métrica, como se observa en la figura B.14.

B. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES NO CRÍTICAS

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.76	0.84	0.96	0.05	[0.82,0.85]
	2	1.74	1.79	1.90	0.06	[1.77,1.81]
	3	2.54	2.61	2.79	0.06	[2.59,2.64]

Tabla B.10: Estadísticas del error a partir de 40 ejecuciones de la SPII para el sexto experimento.

Puntos	Ruido	Mínimo	Media	Máximo	d.e.	i.c. de la media
24	Sin ruido	312.91	361.37	398.19	30.77	[350.41,371.56]
	1 píxel	305.92	362.85	399.04	30.19	[351.31,372.42]
	2 píxeles	304.44	366.55	399.98	23.42	[357.83,373.67]
	3 píxeles	300.00	350.93	393.20	34.66	[339.11,360.92]

Tabla B.11: Estadísticas de la longitud del foco a partir de 40 ejecuciones de la SPII para el sexto experimento.

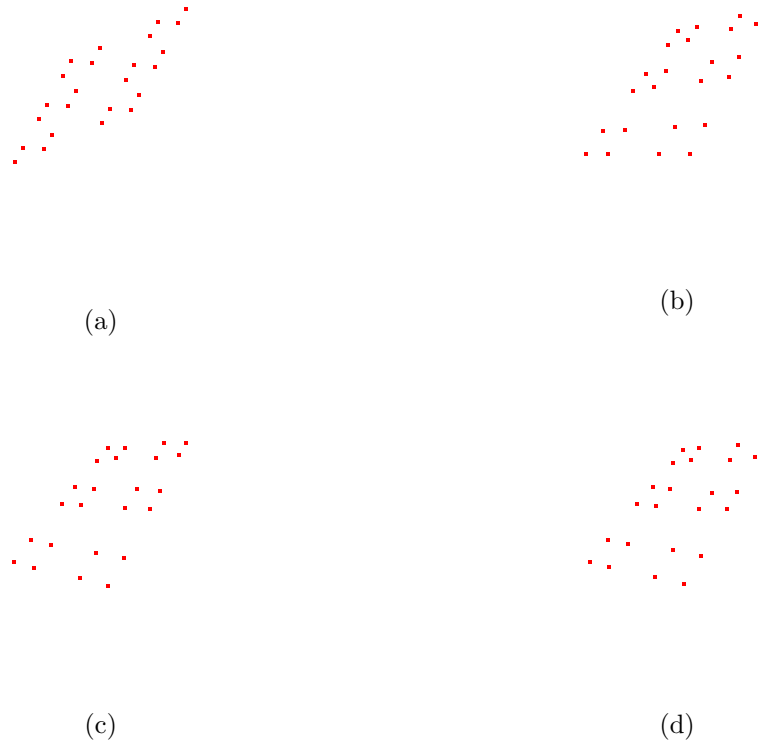


Figura B.14: Reconstrucción tridimensional métrica de la cuadrícula cuando: (a) no hay ruido, (b) un píxel, (c) dos píxeles y (d) tres píxeles de ruido gaussiano.

B.2. Reconstrucción tridimensional al usar puntos de imágenes reales

El séptimo experimento muestra la capacidad que tiene el algoritmo para operar con imágenes reales. El escenario de esta prueba parte de puntos de imágenes reales, con el nivel de ruido que éstas presentan, y bajo una configuración no crítica.

El primer objeto se conforma de 24 correspondencias de puntos, las cuales fueron

tomadas de imágenes del estado del arte [5] y se muestran en la figura B.15. Estas imágenes corresponden a una casa de campo.

El segundo objeto (un juguete) consiste de 48 correspondencias de puntos, las cuales fueron extraídas manualmente de imágenes reales. Estas imágenes se adquirieron dentro del laboratorio de pruebas y se exhiben en la figura B.16.

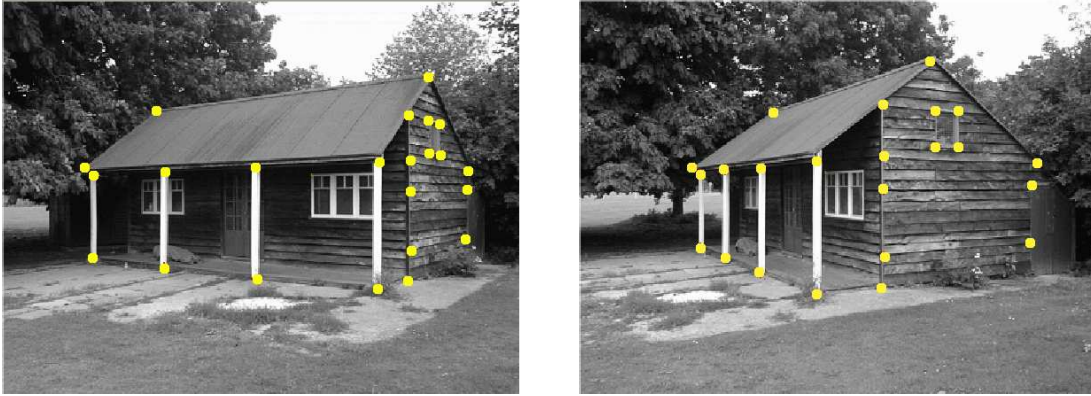


Figura B.15: Imágenes reales de una casa de campo. Los puntos que se emplearon están marcados en ambas imágenes.

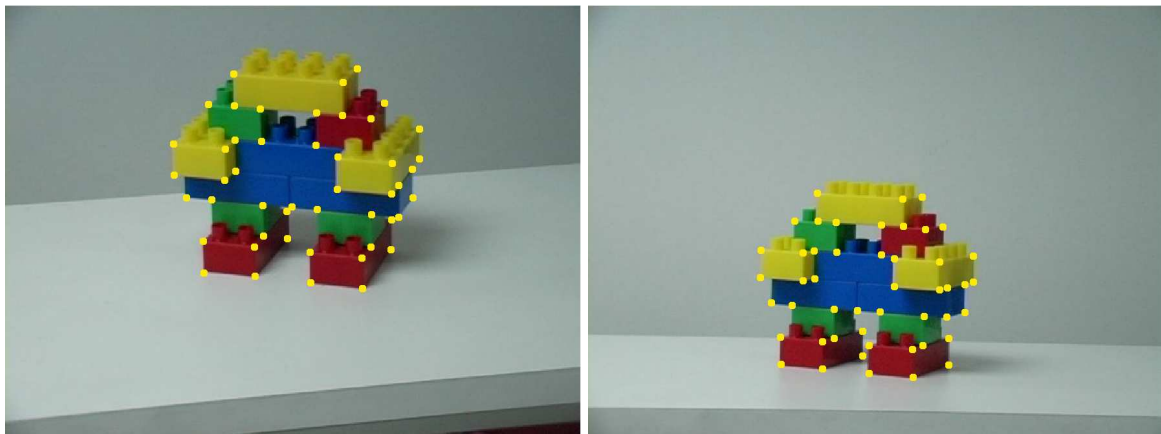


Figura B.16: Imágenes reales de un juguete. Los puntos que se emplearon están marcados en ambas imágenes.

Cabe señalar que los puntos que se emplean en todos los experimentos en escenarios reales fueron extraídos manualmente de las imágenes. Existen algoritmos robustos que obtienen la correspondencia de puntos entre imágenes [191, 192, 193], pero ese no es el objetivo de este trabajo.

Los parámetros empleados para la evolución diferencial son los mismos que los empleados en los experimentos anteriores. La excepción es el caso de la casa, en donde el número de generaciones se tuvo que aumentar al doble (10,000 generaciones) para que el algoritmo llegara a una solución óptima global o en su vecindad.

El número de variables que se emplean para la reconstrucción métrica de la casa es trece y el intervalo de cada variable es el mismo que en las pruebas anteriores, excepto por la longitud del foco, en el cual se aumentó el intervalo a [10, 1500]. Este aumento en el intervalo se debe a que existen cámaras con lentes ópticos que alcanzan estos

B. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES NO CRÍTICAS

valores. El tamaño de las imágenes que se utilizaron es de 840×640 , por lo tanto se fijó el centro óptico al centro de la imagen, es decir, $o_x = 420$ y $o_y = 320$.

Para el caso del juguete, el número de variables es de quince, debido a que se agregó el punto principal. El intervalo de cada variable de los parámetros de orientación y posición se mantiene como en los experimentos anteriores. Para el caso de la longitud del foco se incrementó el intervalo a $[10, 1500]$ y para el punto principal, los intervalos son: $o_x = [270, 370]$ y $o_y = [190, 290]$. Estos intervalos fueron tomados con base en el tamaño de la imagen que, para este caso, es de 640×480 . De tal forma que, el punto principal se encuentra en el centro de la imagen $[320, 240]$ o cerca de él.

En la tabla B.12 se muestran las estadísticas del error al realizar 30 ejecuciones de la SPII sobre el primer caso. Estas estadísticas muestran que el algoritmo da muy buenos resultados, ya que el error obtenido es muy bajo. Además, la casa reconstruida por el algoritmo es muy similar a la casa original, esto se puede observar en la figura B.17, donde se muestran tres vistas de la casa reconstruida. Esta reconstrucción se tomó de la ejecución más cercana a la media del error. Asimismo, de esta ejecución se extraen los parámetros estimados por la SPII de cada una de las cámaras, los cuales se muestran en la tabla B.13 .

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	Real	0.220	0.224	0.238	0.004	[0.223,0.226]

Tabla B.12: Estadísticas del error de 30 ejecuciones de la SPII sobre imágenes reales de la casa de campo.

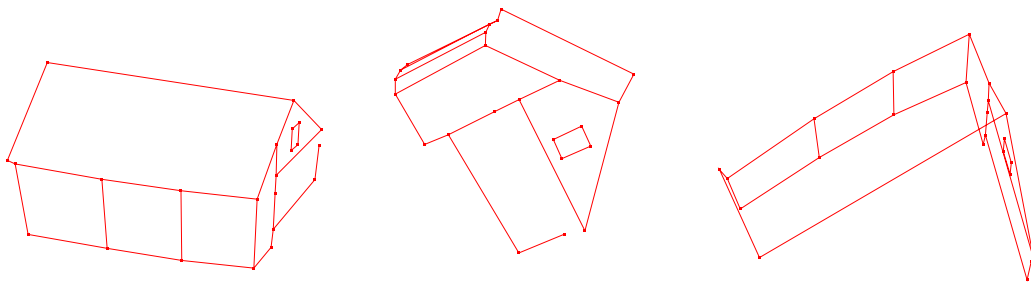


Figura B.17: Tres vistas de la casa reconstruida empleando la SPII.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	2.75	46.81	24	0.702	0.974	0.946	840.64
2	29.85	66.29	47.32	0.439	0.977	0.720	840.64

Tabla B.13: Parámetros de la cámara estimados por la SPII.

En la tabla B.14 se muestran las estadísticas del error al ejecutar la SPII 40 veces. Estas estadísticas pertenecen al juguete. Al igual que con la casa de campo, estas

B.2 Reconstrucción tridimensional al usar puntos de imágenes reales

estadísticas muestran que el algoritmo proporciona muy buenos resultados, aunque fue necesario aumentar el número de generaciones. Asimismo, el juguete reconstruido (ver fig. B.18) es muy parecido (bajo una transformación de similitud) a la estructura del juguete real, aún cuando se obtuvo de la ejecución más cerca a la media del error.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
48	Real	0.624	0.625	0.629	0.000	[0.625,0.626]

Tabla B.14: Estadísticas del error de 40 ejecuciones de la SPII sobre imágenes del juguete.

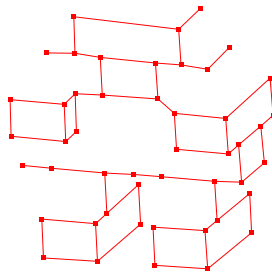


Figura B.18: Reconstrucción del juguete empleando la SPII.

Experimentos con la SPII en configuraciones críticas

Este apéndice muestra el resto de los experimentos realizados sobre la configuraciones críticas. Estos experimentos parten de correspondencias de puntos adquiridas de imágenes sintéticas con diferentes niveles de ruido y en diversas superficies (incluso en las superficies críticas), y de correspondencias de puntos adquiridas de imágenes reales.

C.1. Reconstrucción tridimensional al usar puntos de imágenes sintéticas

Siguiendo con la numeración de los experimentos mostrados en el capítulo de resultados, *el tercer experimento* muestra el desempeño de la SPII al calibrar la cámara y estimar los parámetros de orientación y posición sobre un escenario con diferentes niveles de ruido en las dos configuraciones críticas. El experimento parte de un conjunto de puntos que pertenecen a una media esfera (ver fig. C.1),

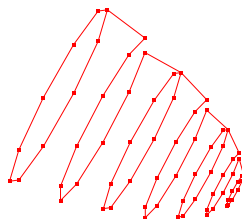


Figura C.1: Media esfera.

Los valores empleados para generar las matrices de proyección de la primera y segunda configuración crítica se muestran en las tablas C.1 y C.2, respectivamente. En las figuras C.2 y C.3 se muestran las imágenes de los puntos pertenecientes a la media esfera de la primera y segunda configuración crítica.

C.1 Reconstrucción tridimensional al usar puntos de imágenes sintéticas

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	22	700
2	0	0	0	10	0	22	700

Tabla C.1: Parámetros de la cámara para el tercer experimento (configuración crítica I).

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	20	350
2	0	20	0	0	0	20	350

Tabla C.2: Parámetros de la cámara para el tercer experimento (configuración crítica II).



Figura C.2: Imágenes de la media esfera con los 74 puntos sin ruido sobre la primera configuración crítica. (a) imagen 1 y (b) imagen 2.



Figura C.3: Imágenes de la media esfera con los 74 puntos sin ruido sobre la segunda configuración crítica. (a) imagen 1 y (b) imagen 2.

Para este experimento, se emplearon 24 correspondencias de puntos, las cuales fueron extraídas aleatoriamente de un conjunto de 74 correspondencias de puntos sin ruido y con k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados en la evolución diferencial son: 5,000 generaciones, 30 individuos en la población, la constante de recombinación de 0.9 y la constante de diferencia de 0.85.

Los intervalos son los siguientes: para las variables que conforman a las dos matrices de rotación, el intervalo es de $[0, \pi]$; para las variables que conforman los dos vectores de traslación, el intervalo es de $[-1, 1]$; para la longitud del foco, el intervalo que se estableció es de $[10, 1000]$. Las estadísticas del error de 40 ejecuciones sobre la primera y segunda configuración crítica, se muestran en las tablas C.3 y C.4, respectivamente.

El siguiente paso consiste en comparar los parámetros de la cámara estimados por la SPII con los parámetros reales. Para realizar esta comparación es necesario que ambos parámetros (reales y estimados) estén en su forma canónica. En las tablas C.5 y C.6 se muestran los parámetros reales en su forma canónica.

En la tabla C.7 están los parámetros estimados en su forma canónica a partir

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.48	0.68	0.89	0.11	[0.65,0.71]
	2	0.69	1.03	1.45	0.16	[0.98,1.08]
	3	1.16	1.78	2.52	0.26	[1.70,1.86]

Tabla C.3: Estadísticas del error a partir de 40 ejecuciones de la SPII para el tercer experimento, sobre la primera configuración crítica.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.59	0.76	0.97	0.10	[0.73,0.80]
	2	0.98	1.42	1.75	0.18	[1.36,1.48]
	3	1.50	2.00	2.48	0.23	[1.92,2.07]

Tabla C.4: Estadísticas del error a partir de 40 ejecuciones de la SPII para el tercer experimento, sobre la segunda configuración crítica.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	0	700
2	0	0	0	1	0	0	700

Tabla C.5: Parámetros de la cámara en forma canónica de la primera configuración crítica para el tercer experimento.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	0	0	0	350
2	0	20	0	0	0	0	350

Tabla C.6: Parámetros de la cámara en forma canónica de la segunda configuración crítica para el tercer experimento.

de 24 correspondencias de puntos, con diferentes niveles de ruido y bajo la primera configuración crítica y en la tabla C.8 bajo la segunda configuración crítica. Los parámetros presentados en estas tablas pertenecen: a la mejor ejecución (mejor), a la ejecución más cercana a la media (media) y a la peor ejecución (peor) en relación a las estadísticas del error.

24 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
Mejor	Sin ruido	0.00	0.00	0.00	1.00	0.00	0.00	700.00
Media	Sin ruido	0.00	0.00	0.00	1.00	0.00	0.00	700.00
Peor	Sin ruido	0.00	0.00	0.00	1.00	0.00	0.00	700.00
Mejor	1 píxel	-0.05	-0.28	-0.08	1.00	0.00	0.00	698.33
Media	1 píxel	-0.39	-0.24	0.09	0.98	-0.02	0.00	705.52
Peor	1 píxel	1.10	-1.05	0.67	0.93	0.06	-0.01	708.59
Mejor	2 píxeles	-1.00	-0.07	0.27	0.95	-0.04	0.01	705.71
Media	2 píxeles	-1.79	0.48	0.11	0.91	-0.09	0.01	690.34
Peor	2 píxeles	-1.71	2.57	-0.34	0.90	-0.10	0.00	719.95
Mejor	3 píxeles	0.96	0.01	-0.19	0.95	0.04	0.00	689.91
Media	3 píxeles	2.23	-1.75	0.06	0.91	0.09	0.03	725.27
Peor	3 píxeles	-1.01	-4.37	-0.02	0.89	-0.04	0.04	727.82

Tabla C.7: Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.

En el caso de la primera configuración crítica, todos los parámetros de la cámara que se estimaron con la SPII, presentan un valor muy cercano al valor real. En las figuras C.4a₁, C.4a₂, C.4b₁, C.4b₂, C.4c₁ y C.4c₂ se muestran las gráficas del porcentaje de error de cada uno de estos parámetros en función a los parámetros reales.

C.1 Reconstrucción tridimensional al usar puntos de imágenes sintéticas

24 Puntos	Ruido	α	β	γ	t_x	t_y	t_z	f
Mejor	Sin ruido	0.00	20.00	0.00	0.15	0.00	0.84	350.00
Media	Sin ruido	0.00	20.00	0.00	0.11	-0.01	-0.87	349.99
Peor	Sin ruido	0.00	19.99	0.00	-0.20	-0.00	-0.78	349.99
Mejor	1 píxel	-0.23	19.35	-0.08	0.20	-0.04	0.76	350.39
Media	1 píxel	0.75	19.61	0.04	0.06	-0.53	-0.42	347.53
Peor	1 píxel	1.18	21.05	0.16	0.14	-0.40	0.46	354.22
Mejor	2 píxeles	-1.21	19.21	0.20	-0.10	0.54	-0.35	348.29
Media	2 píxeles	0.56	20.47	0.02	-0.12	-0.14	-0.74	344.45
Peor	2 píxeles	-1.31	22.25	-0.61	0.27	0.09	0.64	373.11
Mejor	3 píxeles	-2.73	22.57	0.32	-0.46	0.38	0.15	339.53
Media	3 píxeles	-0.93	21.79	0.25	0.28	-0.70	-0.02	335.08
Peor	3 píxeles	-1.22	17.91	-0.25	0.07	-0.00	0.92	330.00

Tabla C.8: Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.

Con base en estos resultados y gráficas se deduce que la SPII tiene un excelente desempeño sobre la primera configuración crítica, aún cuando las correspondencias de puntos presenten ruido.

Para el caso de la segunda configuración crítica se observa que los parámetros de rotación de la cámara fueron estimados correctamente por la SPII, debido a que son muy similares a los parámetros reales; para el caso de los parámetros de traslación se manifiesta que no se obtuvieron los parámetros esperados. Esto es una consecuencia de la segunda configuración crítica, como se mencionó en el capítulo de discusión. Sin embargo, para el caso de la longitud del foco es muy interesante observar que se obtuvo un valor muy cercano al real, aún en imágenes con ruido.

En las figuras C.5a₁, C.5a₂, C.5b₁, C.5b₂, C.5c₁ y C.5c₂ se muestran las gráficas del porcentaje de error relativo de cada uno de estos parámetros en función a los parámetros reales. Gracias a estas gráficas es posible decir que la SPII es muy robusta y tiene un buen desempeño al estimar los parámetros de rotación y la longitud del foco sobre la segunda configuración crítica, aún cuando la reconstrucción tridimensional que se obtiene no sea la reconstrucción deseada.

Para finalizar con este experimento, se midió el error entre los puntos tridimensionales reconstruidos y los puntos tridimensionales reales (estos puntos no presentan ruido) para el caso de la primera configuración crítica. En el caso de la segunda configuración crítica, no es necesario aplicar esta prueba, debido a que la reconstrucción que se obtiene no es válida. Para realizar esta medición se calculó el centroide de los 24 puntos elegidos aleatoriamente para ambos puntos tridimensionales (reales y estimados). Después se trasladaron todos los puntos tridimensionales, de tal forma que, el origen sea el centroide que se calculó. Posteriormente, se normalizaron ambos puntos tridimensionales y se calculó la magnitud de cada uno de ellos. Finalmente se calculó el porcentaje de error relativo con base en la diferencia entre las magnitudes de los puntos tridimensionales reales y los puntos tridimensionales estimados. El resultado de estos cálculos se presenta en la tabla C.1.

El cuarto experimento tiene la finalidad de comparar la SPII contra el método convencional. Los resultados de este experimento aplican a los algoritmos que estimen los parámetros de la cámara mediante la matriz fundamental. El escenario es un conjunto de puntos que pertenecen a seis figuras virtuales con diferentes superficies y formas, sin ruido, y bajo las configuraciones críticas. Estas figuras son: una media

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

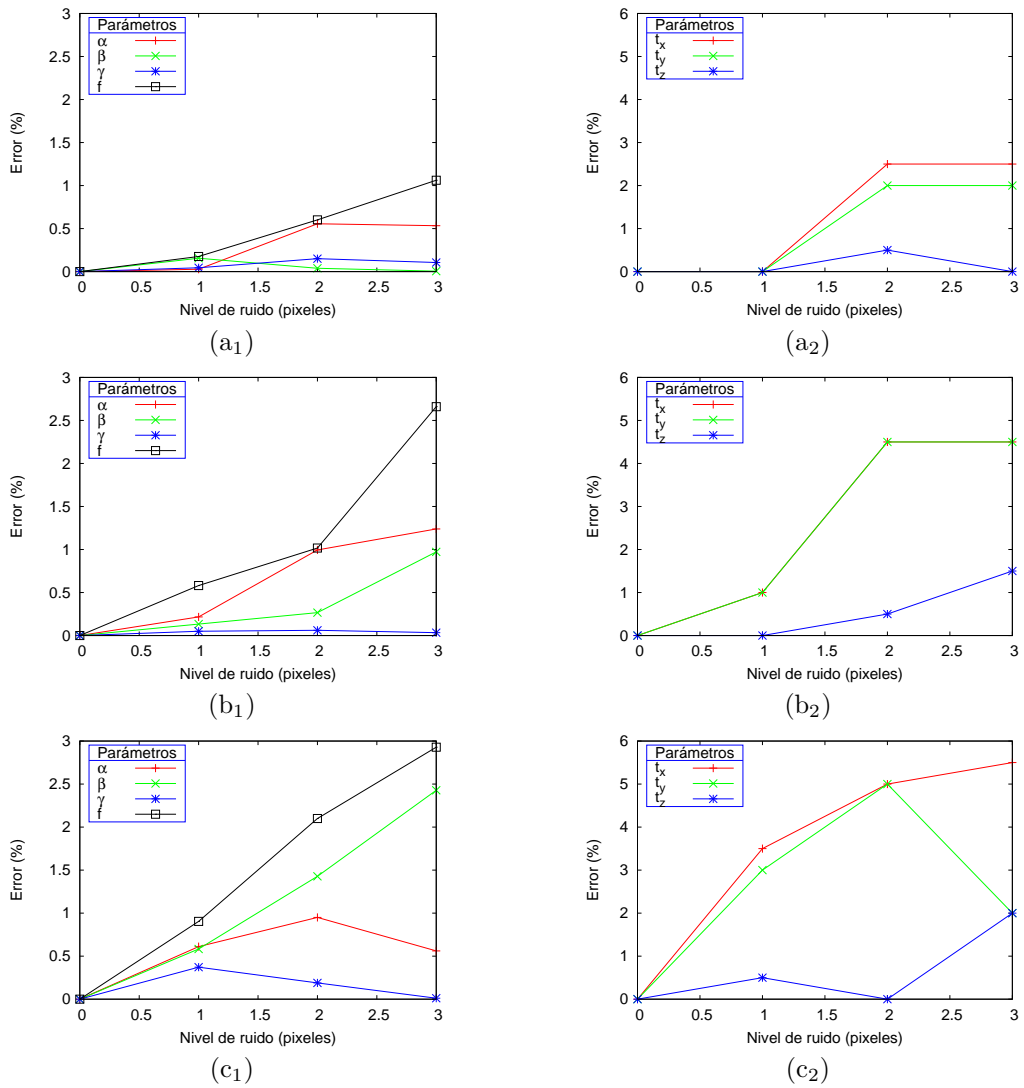


Figura C.4: Gráficas del porcentaje de error de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado al emplear 24 correspondencias de puntos, con diferentes niveles de ruido en la primera configuración crítica. (a₁, a₂) mejor ejecución, (b₁, b₂) ejecución más cercana a la media y (c₁, c₂) peor ejecución.

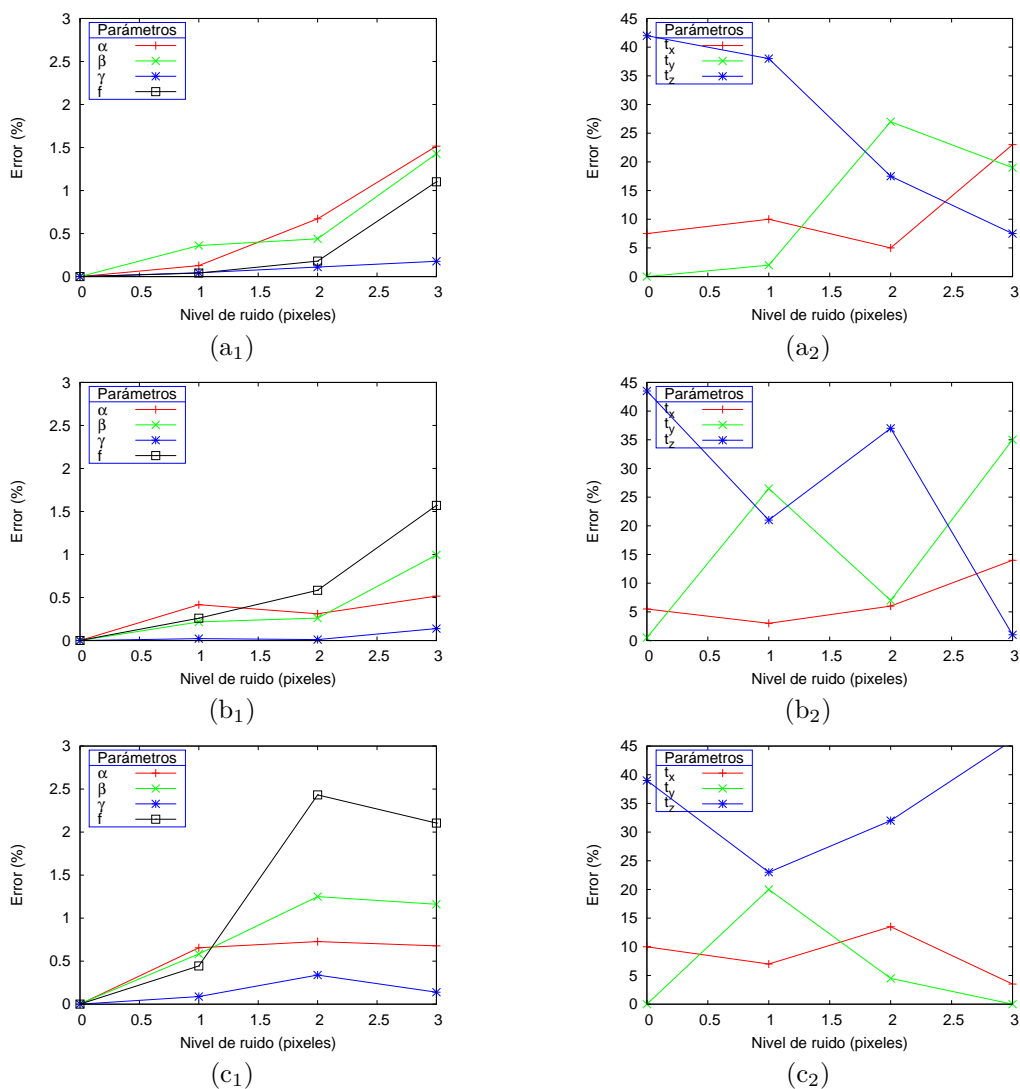


Figura C.5: Gráficas del porcentaje de error de los parámetros de la cámara con base en la diferencia entre el valor real y el valor estimado al emplear 24 correspondencias de puntos, con diferentes niveles de ruido en la segunda configuración crítica. (a₁, a₂) mejor ejecución, (b₁, b₂) ejecución más cercana a la media y (c₁, c₂) peor ejecución.

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.016	0.281	0.818	0.251	[0.126,0.567]
	1	0.824	1.100	1.392	0.171	[0.961,1.223]
	2	1.684	2.283	3.196	0.459	[2.026,2.698]
	3	2.143	2.701	3.596	0.493	[2.348,3.248]

Tabla C.9: Estadísticas del porcentaje de error relativo entre las magnitudes de los puntos tridimensionales reales y los puntos estimados a partir de 40 ejecuciones de la SPII.

esfera, un cubo, un figura regular, un barco, parte de un anillo y un objeto formado por cubos, los cuales se exhiben en el quinto experimento de las configuraciones no críticas (ver fig. B.4 en la pág. 122). Para este experimento se agregaron dos variables de los parámetros intrínsecos que pertenecen al punto principal. Los valores empleados para generar las matrices de proyección y el número de puntos que se utilizaron se presentan en la tabla C.10 para la primera configuración crítica y en la tabla C.11 para la segunda configuración crítica.

Objeto	Imagen	Puntos	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Esfera	1	74	0	0	0	5	2	25	500	300	200
Esfera	2	74	0	0	0	5	10	25	500	300	200
Cubo	1	8	0	0	0	3	0	28	650	300	200
Cubo	2	8	0	0	0	5	0	28	650	300	200
Fig. regular	1	24	0	0	0	0	0	25	700	300	200
Fig. regular	2	24	0	0	0	3	0	25	700	300	200
Barco	1	26	0	0	0	0	0	28	700	300	200
Barco	2	26	0	0	0	0	9	28	700	300	200
Anillo	1	60	0	0	0	0	0	31	550	300	200
Anillo	2	60	0	0	0	7	0	31	550	300	200
Fig. Cubos	1	29	0	0	0	0	0	24	800	300	200
Fig. Cubos	2	29	0	0	0	0	4	24	800	300	200

Tabla C.10: Parámetros de la cámara utilizados para generar las dos imágenes por cada una de las 6 figuras, sobre la primera configuración crítica.

Objeto	Imagen	Puntos	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Esfera	1	74	0	0	0	0	0	25	500	300	200
Esfera	2	74	0	25	0	0	0	25	500	300	200
Cubo	1	8	0	0	0	0	0	30	600	300	200
Cubo	2	8	10	0	0	0	0	30	600	300	200
Fig. regular	1	24	0	0	0	0	0	25	700	300	200
Fig. regular	2	24	0	33	0	0	0	25	700	300	200
Barco	1	26	0	0	0	0	0	28	700	300	200
Barco	2	26	0	0	35	0	0	28	700	300	200
Anillo	1	60	0	0	0	0	0	31	550	300	200
Anillo	2	60	0	0	25	0	0	31	550	300	200
Fig. Cubos	1	29	0	0	0	0	0	30	800	300	200
Fig. Cubos	2	29	0	5	0	0	0	30	800	300	200

Tabla C.11: Parámetros de la cámara utilizados para generar las dos imágenes por cada una de las 6 figuras, sobre la segunda configuración crítica.

Los parámetros propuestos en estas tablas, producen imágenes sintéticas que se encuentran en la dos configuraciones críticas. Las dos imágenes que fueron generadas sintéticamente por cada objeto y por cada configuración crítica se visualizan en las figuras C.6, C.7, C.8, C.9, C.10 y C.11.

El experimento sólo emplea correspondencias de puntos sin ruido. Los parámetros para la evolución diferencial y sus intervalos son iguales a los experimentos anteriores,

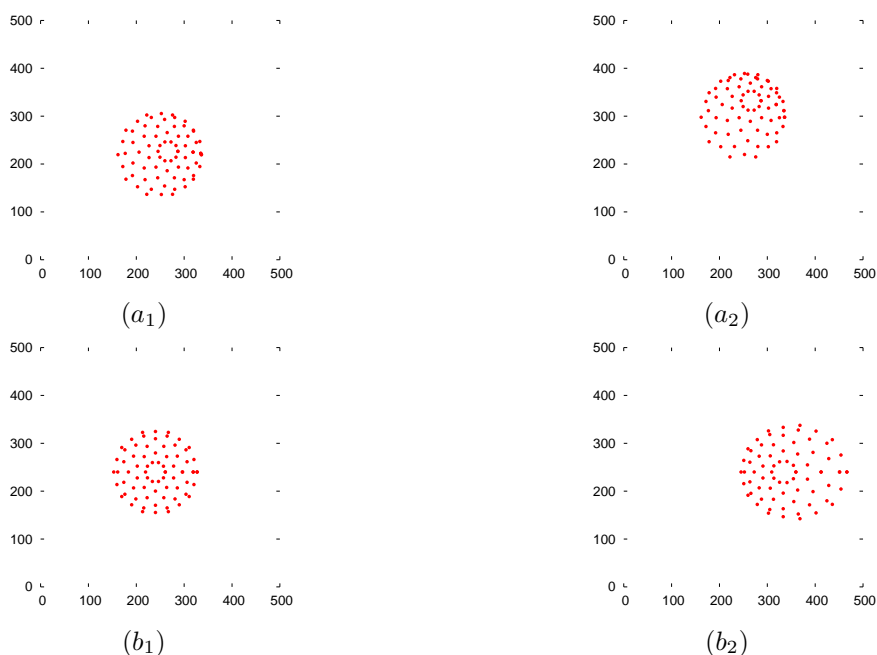


Figura C.6: Imágenes sintéticas de la media esfera. (a_1, a_2) primera y segunda imagen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, respectivamente.

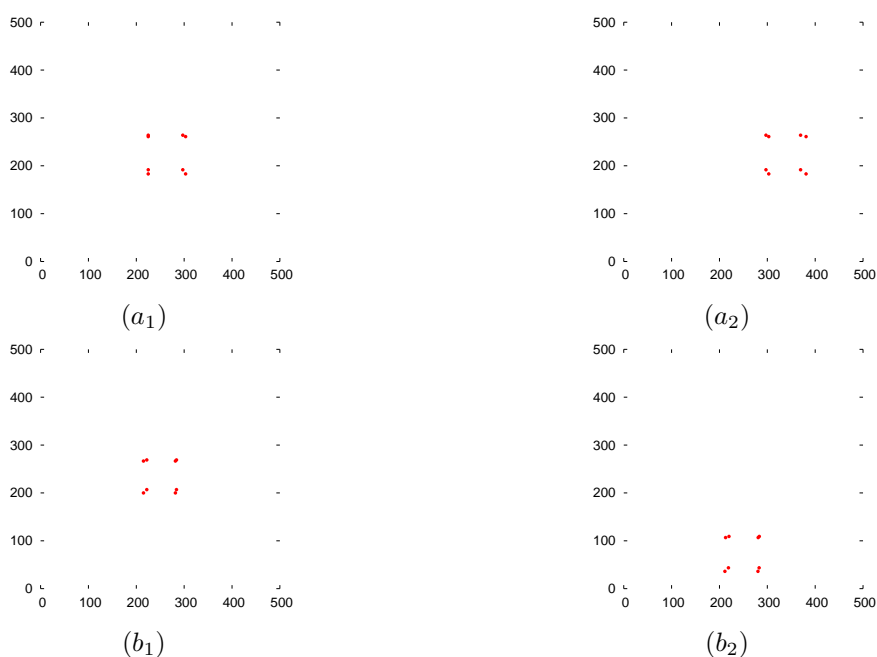


Figura C.7: Imágenes sintéticas del cubo. (a_1, a_2) primera y segunda imagen, respectivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segunda configuración crítica.

excepto porque se agregaron dos variables que conforman al punto principal. Esto da un total de 15 variables. Para las dos variables que se agregaron, los intervalos son los siguientes: o_x es de $[250, 350]$ y o_y es de $[150, 250]$. Se eligieron estos intervalos debido a que el punto principal se presupone que está en el centro de la imagen para

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

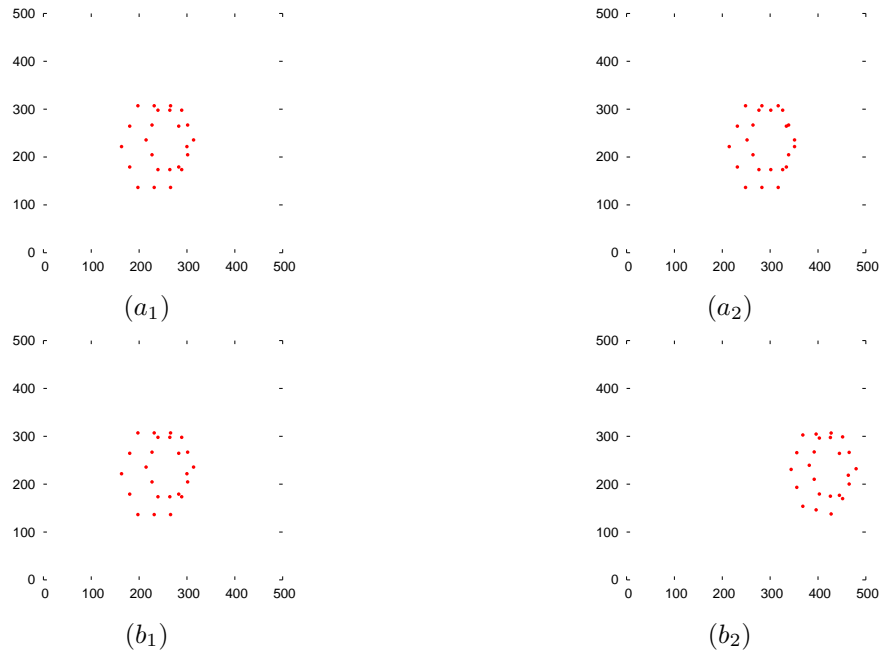


Figura C.8: Imágenes sintéticas de la figura regular. (a_1, a_2) primera y segunda imagen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, respectivamente.

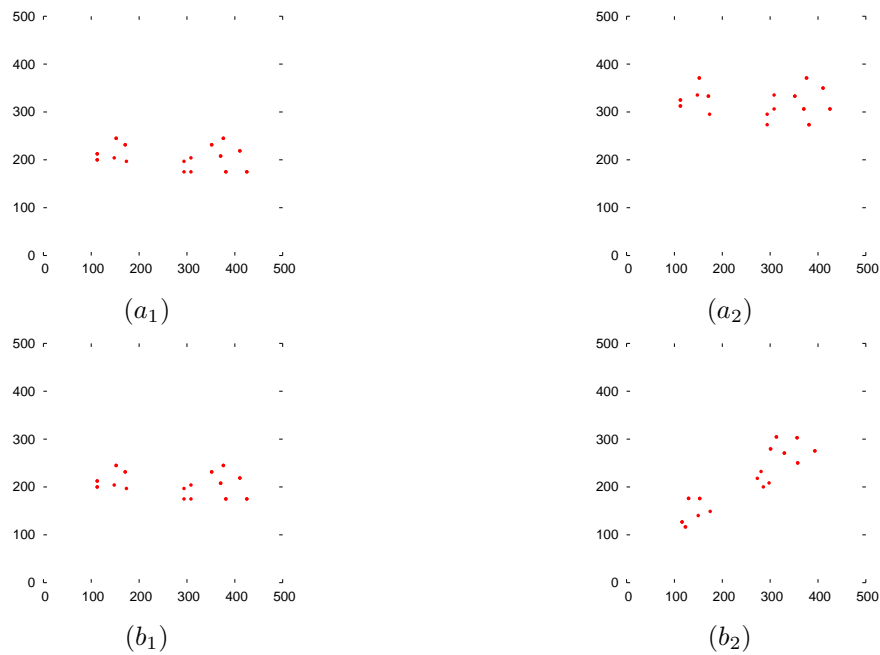


Figura C.9: Imágenes sintéticas del barco. (a_1, a_2) primera y segunda imagen, respectivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segunda configuración crítica.

la mayoría de las cámaras actuales.

Por cada objeto y cada configuración crítica se ejecutó 30 veces la SPII y sólo una vez el método convencional. Al ejecutar el método convencional en ambas configuraciones críticas no se obtuvieron resultados correctos, debido a que éste utiliza

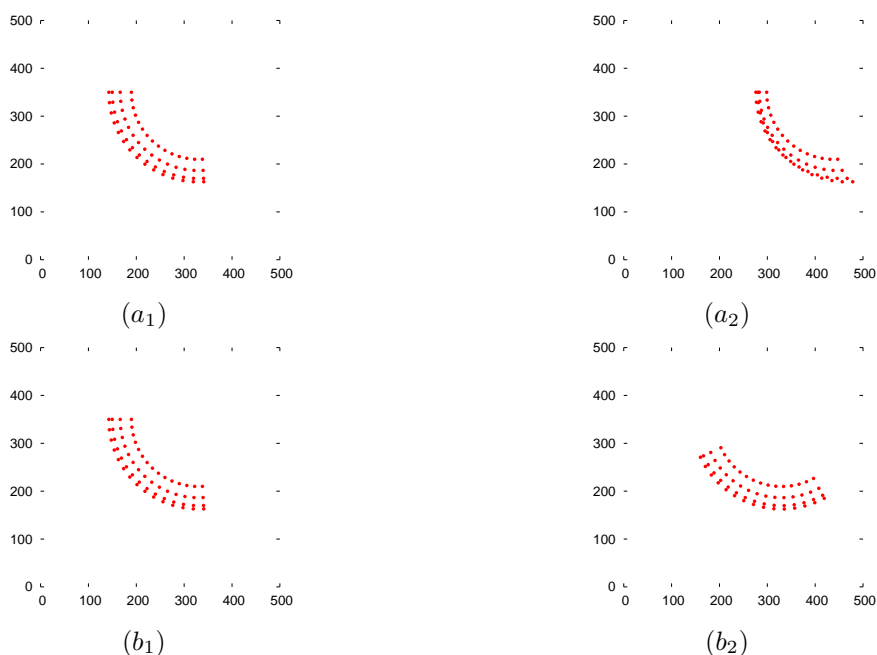


Figura C.10: Imágenes sintéticas del anillo. (a_1, a_2) primera y segunda imagen, respectivamente que pertenecen a la primera configuración crítica, y (b_1, b_2) primera y segunda imagen, respectivamente que pertenecen a la segunda configuración crítica.

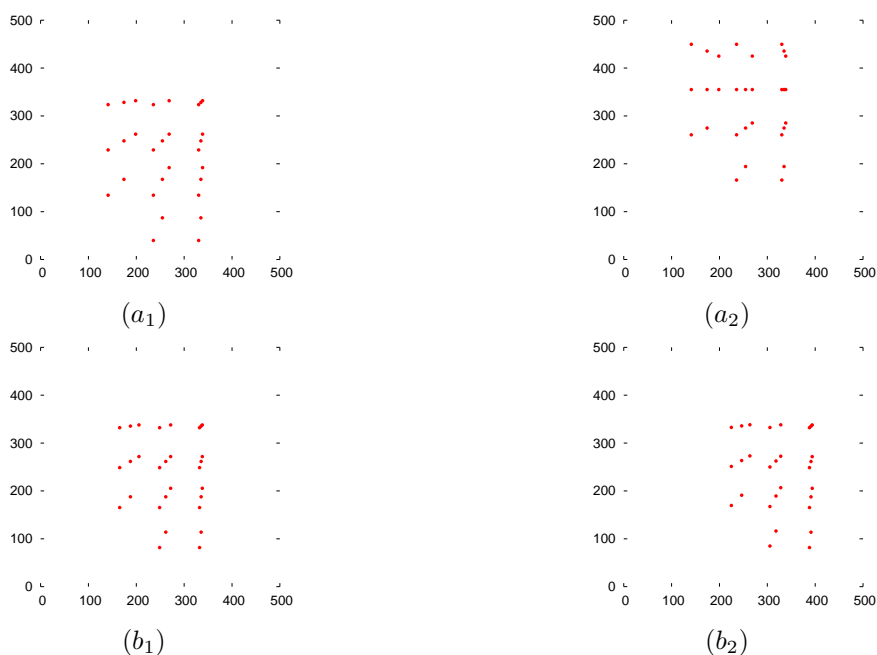


Figura C.11: Imágenes sintéticas del objeto formado por cubos. (a_1, a_2) primera y segunda imagen sobre la primera configuración crítica, respectivamente, y (b_1, b_2) primera y segunda imagen sobre la segunda configuración crítica, respectivamente.

la matriz fundamental para obtener los parámetros de la cámara. Para que éste o cualquier otro algoritmo que utilice la matriz fundamental obtenga resultados, la matriz fundamental se debe ver como una cónica no degenerada (como se muestra en el capítulo anterior). Sin embargo, las cónicas que se generan a partir de las matrices

fundamentales, para todos los objetos en ambas configuraciones críticas, son cónicas degeneradas como se muestra en las figuras C.12 y C.13. Para el caso de la primera configuración crítica, estas cónicas se ven como una línea recta y, para la segunda configuración crítica son dos líneas que se intersecan en un punto. Al calcular el rango de la matriz fundamental el valor obtenido es de 1. Por tal motivo, no es posible extraer los parámetros de la cámara con los algoritmos del estado del arte.

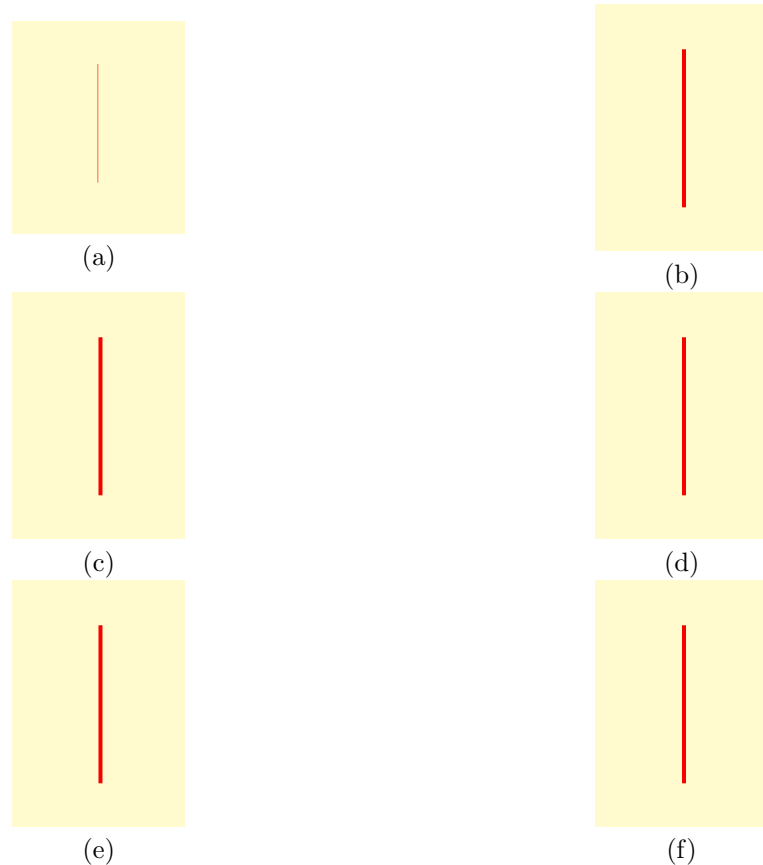


Figura C.12: Representación de la matriz fundamental como una cónica degenerada para: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos sobre la primera configuración crítica.

La SPII no utiliza la matriz fundamental, por lo cual es posible obtener los parámetros de la cámara. En las tablas C.12 y C.13 se presentan tanto los parámetros reales como los parámetros estimados (parámetros que fueron tomados de la ejecución más cercana a la media con base en las estadísticas del error) de ambas configuraciones críticas en su forma canónica con la finalidad de compararlos. En estas tablas se observa que, en la primera configuración crítica, el algoritmo obtiene los parámetros de orientación y posición idénticos a los reales, y la matriz de calibración similares a los reales. Estos resultados comprueban la robustez de la SPII. En las figuras C.14a, C.14b, C.14c, C.14d, C.14e y C.14f se muestra la reconstrucción tridimensional métrica de cada uno de los objetos.

En el caso de la segunda configuración crítica, el algoritmo recupera correctamente los parámetros de rotación y los parámetros intrínsecos de la cámara (longitud del foco y el punto principal) aunque, como había sucedido en experimentos anteriores, los parámetros de traslación no los obtiene correctamente. En las figu-

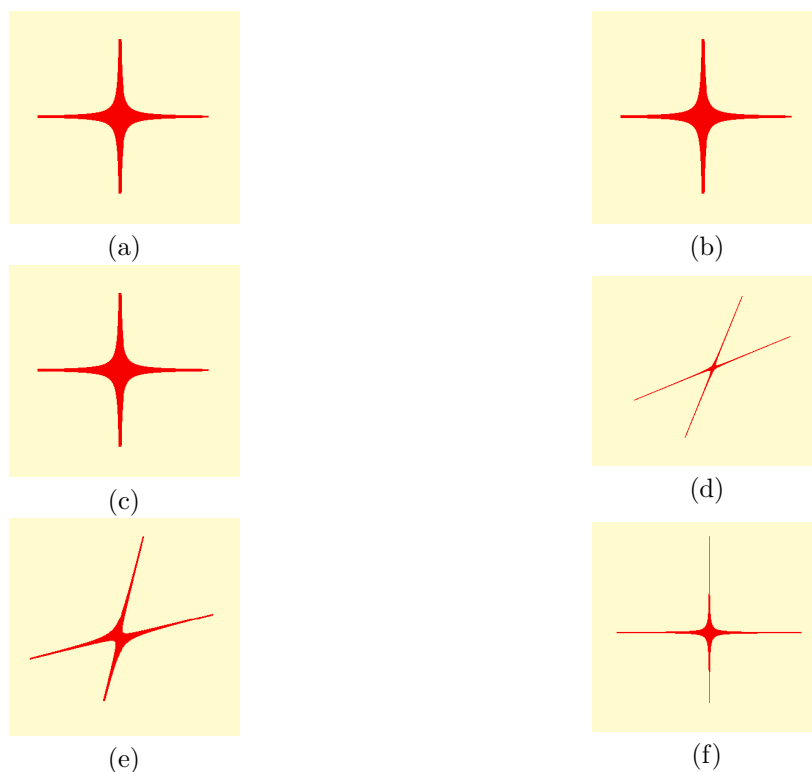


Figura C.13: Representación de la matriz fundamental como una cónica degenerada para: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos sobre la segunda configuración crítica.

Objeto	Tipo	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Esfera	Real	0	0	0	0	1	0	500	300	200
Esfera	Estimado	0.0	0.0	0.0	0.0	1	0.0	505.35	305.75	189.34
Cubo	Real	0	0	0	1	0	0	650	300	200
Cubo	Estimado	0.0	0.0	0.0	1	0.0	0.0	655.50	308.62	190.91
Fig. regular	Real	0	0	0	1	0	0	700	300	200
Fig. regular	Estimado	0.0	0.0	0.0	1	0.0	0.0	704.18	291.02	196.68
Barco	Real	0	0	0	0	1	0	700	300	200
Barco	Estimado	0.0	0.0	0.0	0.0	1	0.0	702.02	298.48	193.45
Anillo	Real	0	0	0	1	0	0	550	300	200
Anillo	Estimado	0.0	0.0	0.0	1	0.0	0.0	553.46	290.07	191.45
Fig. cubos	Real	0	0	0	0	1	0	800	300	200
Fig. cubos	Estimado	0.0	0.0	0.0	0.0	1	0.0	801.46	295.07	195.61

Tabla C.12: Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.

ras C.15a, C.15b, C.15c, C.15d, C.15e y C.15f se visualiza la reconstrucción tridimensional métrica de cada uno de los objetos, pero se sabe que estas reconstrucciones no son correctas (ver primer experimento sobre configuraciones críticas).

El quinto experimento muestra el desempeño de la SPII cuando se parte de puntos que yacen sobre un mismo plano. El escenario de tal experimento es el mismo que el sexto experimento que se realizó sobre una configuración no crítica (experimento descrito en la pág. 124), es decir, es un conjunto de puntos que pertenecen a una cuadrícula (la cuadrícula yace sobre un mismo plano), con diferentes niveles de ruido;

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

Objeto	Tipo	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
Esfera	Real	0	25	0	0	0	0	500	300	200
Esfera	Estimado	0.05	25.16	0.09	-0.820	0.080	0.099	502.55	297.71	201.94
Cubo	Real	10	0	0	0	0	0	600	300	200
Cubo	Estimado	9.75	0.36	-0.04	0.411	0.344	-0.244	605.03	291.01	209.53
Fig. regular	Real	0	33	0	0	0	0	700	300	200
Fig. regular	Estimado	0.05	33.17	-0.17	0.359	-0.264	0.375	699.95	304.54	192.23
Barco	Real	0	0	35	0	0	0	700	300	200
Barco	Estimado	0.0	0.0	35.0	0.119	-0.879	-0.001	702.64	300	200
Anillo	Real	0	0	25	0	0	0	550	300	200
Anillo	Estimado	0.0	0.01	25.01	-0.601	-0.204	-0.194	555.22	299.82	199.59
Fig. cubos	Real	0	5	0	0	0	0	800	300	200
Fig. cubos	Estimado	-0.01	5.02	-0.03	-0.475	-0.295	0.229	799.78	294.82	193.95

Tabla C.13: Parámetros de la cámara estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.

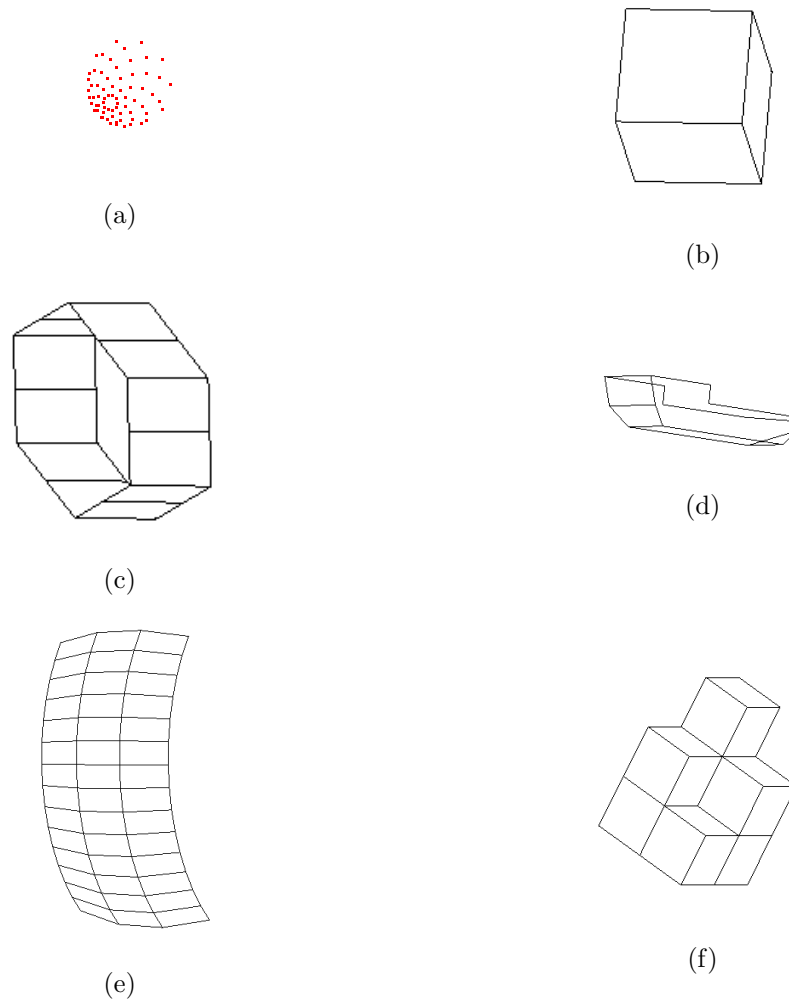


Figura C.14: Objetos reconstruidos por la SPII sobre la primera configuración crítica: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.

la diferencia radica en que este experimento está en las dos configuraciones críticas.

Los valores empleados para generar las matrices de proyección se presentan en las tablas C.14 y C.15 para la primera y segunda configuración crítica, respectivamente.

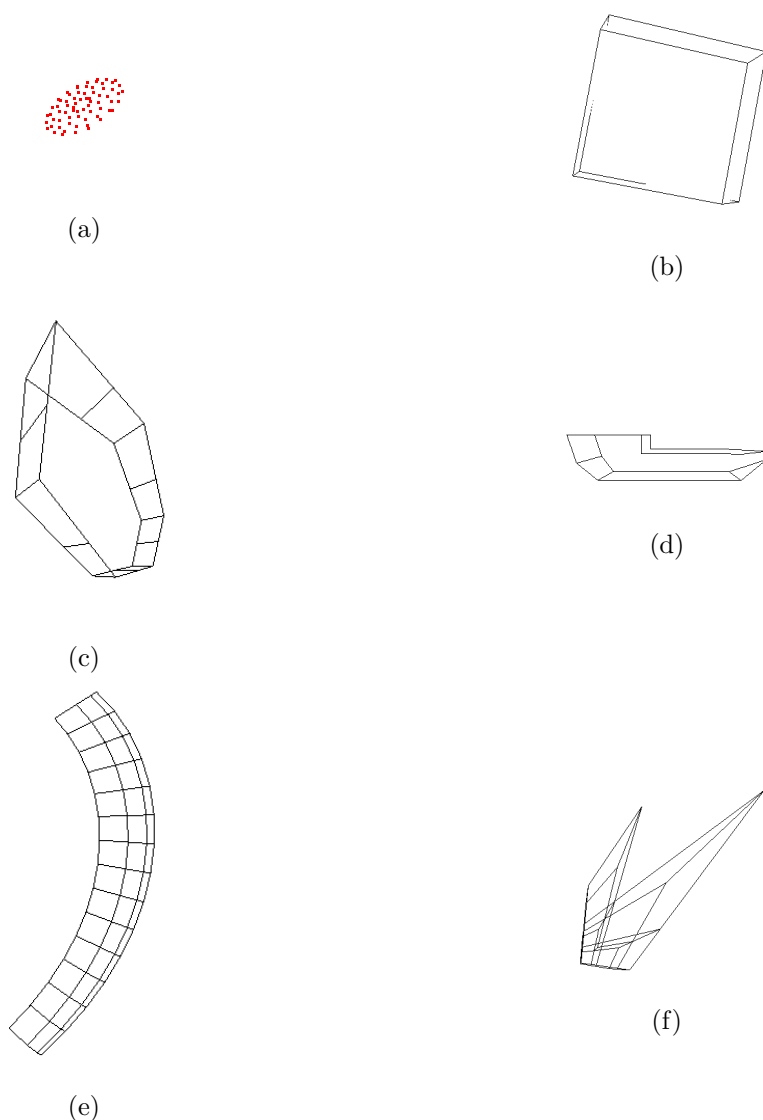


Figura C.15: Objetos reconstruidos por la SPII sobre la segunda configuración crítica: (a) media esfera, (b) cubo, (c) figura regular, (d) barco, (e) parte de un anillo y (f) objeto formado por cubos.

A partir de las matrices de proyección y los puntos tridimensionales de la cuadrícula se generaron las imágenes que se visualizan en las figuras C.16a , C.16b, C.17c y C.17d sobre la primera y segunda configuración crítica.

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0.000	0.000	0.000	0.000	0.000	10.000	350.000
2	0.000	0.000	0.000	10.000	0.000	10.000	350.000

Tabla C.14: Parámetros de la cámara utilizados en el quinto experimento sobre la primera configuración crítica.

El conjunto de puntos se conforma de 24 correspondencias. A estos puntos se les añadió k niveles de ruido gaussiano RMS que va de 1 a 3 píxeles. Los parámetros empleados para la evolución diferencial son los mismos que en el experimento anterior,

C. EXPERIMENTOS CON LA SPII EN CONFIGURACIONES CRÍTICAS

Imagen	α	β	γ	t_x	t_y	t_z	f
1	0.000	0.000	0.000	0.000	0.000	9.000	350.000
2	0.000	20.000	0.000	0.000	0.000	9.000	350.000

Tabla C.15: Parámetros de la cámara utilizados en el quinto experimento sobre la segunda configuración crítica.



Figura C.16: Imágenes de la cuadrícula generada sobre la primera configuración crítica: (a) imagen 1 y (b) imagen 2.



Figura C.17: Imágenes de la cuadrícula generada sobre la segunda configuración crítica: (a) imagen 1 y (b) imagen 2.

al igual que el número de variables y sus intervalos. El algoritmo se ejecutó 40 veces para cada nivel de ruido. En la tabla C.16 se exhiben las estadísticas del error de estas ejecuciones, las cuales muestran que el algoritmo presenta un excelente desempeño en la primera configuración crítica. Tal desempeño se replica en la segunda configuración crítica, como se muestra en la tabla C.17.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.66	0.68	0.68	0.01	[0.67,0.68]
	2	1.45	1.65	1.70	0.10	[1.60,1.67]
	3	2.46	2.63	2.97	0.23	[2.55,2.72]

Tabla C.16: Estadísticas del error a partir de 40 ejecuciones de la SPII para el quinto experimento, sobre la primera configuración crítica.

Los parámetros estimados por el algoritmo son muy similares a los parámetros reales en su forma canónica para la primera configuración crítica. Para el caso de la segunda configuración crítica, sólo los parámetros de rotación y el parámetro intrínseco se recuperaron correctamente. Esta información se visualiza en las tablas C.18 y C.19 para ambas configuraciones críticas, donde los parámetros estimados muestra-

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.46	0.46	0.46	0.00	[0.46,0.46]
	2	1.19	1.21	1.22	0.01	[1.21,1.21]
	3	1.86	1.92	2.14	0.09	[1.89,1.96]

Tabla C.17: Estadísticas del error a partir de 40 ejecuciones de la SPII para el quinto experimento, sobre la segunda configuración crítica.

dos corresponden a las ejecuciones más cercanas a la media del error. Esto comprueba de nuevo la efectividad del algoritmo, aún en imágenes que contienen puntos coplanares y bajo configuraciones críticas.

También se muestra en las figuras C.18 y C.19 que, al igual que el sexto experimento de la configuración no crítica, el incremento de la diferencia entre la longitud del foco real y el estimado no afecta la reconstrucción tridimensional obtenida para los diferentes niveles de ruido. Además, se logró obtener una reconstrucción correcta para ambas configuraciones, debido a que los puntos yacen sobre el mismo plano, por lo cual, no se percibe el problema que se tiene en los experimentos anteriores.

Tipo	Ruido	α	β	γ	t_x	t_y	t_z	f
Real	0	0	0	0	1	0	0	350
Estimado	0	0	0	0	1	0.0	0.0	353.41
Estimado	1	-0.50	0.82	-0.50	0.99	0.00	0.0	327.71
Estimado	2	-0.55	0.93	-0.37	-0.95	-0.02	0.01	323.31
Estimado	3	1.53	-0.56	0.12	-0.93	-0.01	0.06	315.00

Tabla C.18: Comparación entre los parámetros de la cámara reales y los estimados por la SPII con diferentes niveles de ruido sobre la primera configuración crítica.

Tipo	Ruido	α	β	γ	t_x	t_y	t_z	f
Real	0	0	20	0	0	0	0	350
Estimado	0	0.0	20.00	0.0	0.86	0.0	0.14	350.00
Estimado	1	0.31	19.92	-0.18	-0.24	0.22	-0.54	354.89
Estimado	2	-0.47	21.40	-0.02	0.82	0.14	-0.04	372.25
Estimado	3	-1.08	19.23	0.77	-0.41	0.26	-0.33	334.73

Tabla C.19: Comparación entre los parámetros de la cámara reales y los estimados por la SPII con diferentes niveles de ruido sobre la segunda configuración crítica.

C.2. Reconstrucción tridimensional al usar puntos de imágenes reales

El sexto experimento utiliza la cuadrícula adherida a la pared, la misma que se empleó en el octavo experimento bajo una configuración no crítica. Sin embargo, el escenario de esta prueba es bajo la primera configuración crítica. En la figura C.20 se visualizan ambas imágenes con marcas en los puntos elegidos.

Los parámetros empleados para la evolución diferencial son los mismos que los empleados en los experimentos anteriores, a diferencia del número de generaciones donde el valor que se utilizó fue de 10,000 generaciones.

El número de variables que se definieron para este experimento es de quince, debido a que se agregó el punto principal. El intervalo de cada variable de los parámetros



Figura C.18: Reconstrucción tridimensional métrica de la cuadrícula cuando hay: (a) dos píxeles y (b) tres píxeles de ruido gaussiano sobre la primera configuración crítica.



Figura C.19: Reconstrucción tridimensional métrica de la cuadrícula cuando hay: (a) dos píxeles y (b) tres píxeles de ruido gaussiano sobre la segunda configuración crítica.

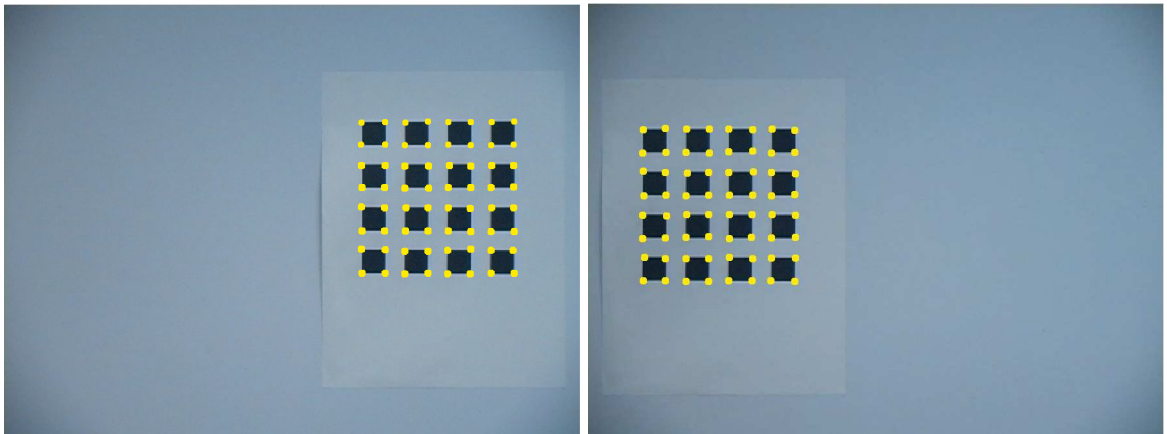


Figura C.20: Imágenes reales de una cuadrícula sobre la primera configuración crítica. Los puntos que se emplearon están marcados en ambas imágenes.

de orientación y posición se mantiene como en los experimentos anteriores. Para el caso de la longitud del foco, el intervalo es de $[10, 1500]$ y, para el punto principal, los intervalos son: $o_x = [270, 370]$ y $o_y = [190, 290]$. Estos intervalos fueron tomados con base en el tamaño de la imagen; para este caso es de 640×480 .

C.2 Reconstrucción tridimensional al usar puntos de imágenes reales

Las estadísticas del error, al realizar 40 ejecuciones de la SPII, se muestran en la tabla C.20. Estas estadísticas muestran que el algoritmo proporciona excelentes resultados y la reconstrucción que se obtiene es casi idéntica a la cuadrícula real (ver figura C.21). Asimismo, en la tabla C.21 se exhiben las estadísticas obtenidas por esta solución en relación a la longitud foco.

Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
64	Real	0.256	0.256	0.258	0.0	[0.256,0.286]

Tabla C.20: Estadísticas del error a partir de 40 ejecuciones de la SPII, al utilizar puntos de imágenes reales.

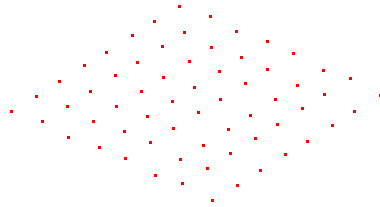


Figura C.21: Reconstrucción tridimensional de la cuadrícula empleando la SPII.

Puntos	Ruido	Mínimo	Media	Máximo	d.e.	i.c. de la media
64	Real	849.51	849.97	850.0	0.080	[849.38,849.99]

Tabla C.21: Estadísticas de la longitud foco a partir de 40 ejecuciones de la SPII, al utilizar puntos de imágenes reales.

Apéndice D

Comparación de los algoritmos evolutivos

El análisis descrito a continuación, parte de los resultados obtenidos por los diferentes algoritmos evolutivos (AE) en la sesión especial del Congreso de Computación Evolutiva (CEC'2005) sobre optimización con parámetros reales [130]. Estos algoritmos se pusieron a prueba en un conjunto de 25 funciones, cinco funciones son uni-modales y veinte multi-modales.

D.1. Análisis comparativo de los AE para la optimización con parámetros reales

Los algoritmos que se pusieron a prueba, corresponden a las siguientes heurísticas: algoritmos genéticos, estrategias evolutivas, optimización mediante cúmulos de partículas, algoritmos de estimación de distribución y evolución diferencial.

En la tabla D.1 se exhiben los algoritmos clasificados por la heurística a la que pertenecen, así como el primer autor del artículo donde se presentan.

Nombre	Heurística	Autor
BLX-GL50	Algoritmo genético	C. García-Martínez [194]
BLX-MA	Algoritmo genético con búsqueda local	D. Molina [195]
SPC-PNX	Algoritmo genético	P. Ballester [104]
K-PCX	Algoritmo genético	A. Sinha [102]
G-CMA-ES	Estrategia evolutiva	A. Auger [111]
L-CMA-ES	Estrategia evolutiva	A. Auger [109]
Co-EVO	Estrategia evolutiva	P. Pöšík [108]
DMS-L-PSO	Optimización mediante cúmulos de partículas	J. Liang [123]
EDA	Algoritmo de estimación de distribución	B. Yuan [91]
ED	Evolución diferencial	J. Rönkkönen [140]
L-SaDE	Evolución diferencial	A. Qin [196]

Tabla D.1: Algoritmos presentados en la sesión del CEC'2005.

Cada algoritmo se ejecutó en 25 ocasiones por cada función y considerando 10 y 30 variables. Los criterios de evaluación que se emplearon son los siguientes:

1. Número máximo de evaluaciones (NME): es el número máximo de evaluaciones permitidas para encontrar la solución óptima global (se adoptaron: para 10 variables = 100,000 NME y para 30 variables = 300,000 NME).
2. Número de funciones resueltas (NFR): es el número de funciones en donde, al menos en una ejecución, el algoritmo obtuvo la solución óptima global antes del número máximo de evaluaciones.
3. Tasa de éxito (TE): es el número de ejecuciones en las cuales se obtuvo la solución óptima global antes del NME, dividido por el número total de ejecuciones.
4. Número de evaluaciones (NE): es el número de evaluaciones para obtener una solución óptima global.
5. Éxito del desempeño (EDES): es el valor dado por la media del NE (sólo se toman en cuenta las ejecuciones que llegaron a la solución óptima global antes del NME), dividido por la TE.

El análisis se muestra por el tipo de las funciones (uni-modales, multi-modales resueltas) y el número de variables empleadas. Las funciones multi-modales resueltas son aquellas donde al menos en una ejecución un algoritmo obtuvo la solución óptima global. En la recopilación de resultados en este Congreso, al menos un algoritmo llegó a la solución óptima global en todas las funciones uni-modales y en 6 de 14 funciones multi-modales.

D.1.1. Funciones uni-modales con 10 variables

En relación al NFR, los algoritmos G-CMA-ES, L-CMA-ES, EDA y ED obtuvieron el valor más alto al resolver todas las funciones, en al menos una ejecución y sólo el algoritmo G-CMA-ES alcanzó la solución óptima global en todas las ejecuciones y en todas las funciones uni-modales, por lo tanto, obtuvo una TE del 100 %, seguido por el EDA con 97 % y la ED con 96 %.

Asimismo, el algoritmo G-CMA-ES obtuvo el mejor EDES para el conjunto de funciones uni-modales y su NE es menor aproximadamente en un orden de magnitud al NME. No obstante, en dos funciones fue superado por el algoritmo K-PCX, debido a que este algoritmo consiguió llegar a la solución óptima global en menos NE.

El EDA y la ED obtuvieron un EDES bajo ya que necesitaron de más evaluaciones para llegar a la solución óptima global.

D.1.2. Funciones multi-modales resueltas con 10 variables

Para este tipo de funciones el G-CMA-ES y la ED presentan los mejores resultados en el NFR, al resolver 6 de 7 funciones. Sin embargo, el G-CMA-ES tiene una TE y un EDES mejor que la ED, es decir, el G-CMA-ES obtiene la solución óptima global en más ejecuciones (mejor TE) y con un número menor de evaluaciones (mejor EDES) que la ED. De hecho la TE y el EDES de la ED está por debajo de los algoritmos L-SaDE, DMS-L-PSO y K-PCX. Aunque estos algoritmos tienen una NFR menor, es decir, son menos robustos en cuanto al número de funciones que pueden resolver.

El algoritmo EDA, que tuvo un buen desempeño en las funciones uni-modales, sólo pudo obtener la solución óptima global en 3 funciones y la TE está muy por debajo de la TE de la ED.

D.1.3. Funciones uni-modales con 30 variables

Cuando se tiene una dimensionalidad de 30 variables, el algoritmo G-CMA-ES tiene el mejor NFR, al obtener la solución óptima global en todas las funciones. También su TE de 90% y EDES es muy superior a los otros algoritmos. En su TE no alcanzó el 100% dado que no en todas las ejecuciones llegó al óptimo global.

Después del algoritmo G-CMA-ES le sigue el L-CMA-ES y la EDA, donde el L-CMA-ES resuelve 5 funciones con una TE de 83%, y la EDA 4 funciones con una TE de 67%. Estos dos algoritmos mantuvieron su buen desempeño aún cuando se aumentó la dimensionalidad, pero no pasó de la misma manera para la ED, la cual disminuyó su desempeño considerablemente, es decir, sólo encontró la solución óptima global en una función y su TE es de 17%.

D.1.4. Funciones multi-modales resueltas con 30 variables

En este tipo de funciones, el algoritmo G-CMA-ES obtuvo el mejor NFR, al registrar 6 de 7 funciones resueltas, y en una de ellas fue el único que obtuvo la solución óptima global. Además, fue el algoritmo que presentó el mejor EDES. Sin embargo, en la TE fue superado por el algoritmo K-PCX con 38%, tan sólo por 1%. Esto habla de que el G-CMA-ES mantiene su desempeño aún en funciones con mayor dimensionalidad.

La ED, que había presentado buen desempeño con 10 variables, obtuvo un desempeño pobre al aumentar el número de variables, a tal grado que sólo pudo resolver una función. En cuanto a su TE decreció por el NFR y sólo obtuvo 20%. Los algoritmos K-PCX y L-SaDE mantuvieron su desempeño en cuanto a la TE aún cuando se aumentó la dimensionalidad. Aunque la L-SaDE sólo resolvió 2 funciones.

D.1.5. Complejidad de los algoritmos

La complejidad de los algoritmos se mide de la siguiente forma:

$$(\widehat{tm}_2 - tm_1)/tm_0, \tag{D.1}$$

donde \widehat{tm}_2 es la media del tiempo al ejecutar cinco veces el algoritmo con la tercera función de prueba (función uni-modal [130]) hasta alcanzar 200,000 evaluaciones; tm_1 es el tiempo de evaluar 200,000 veces la tercera función de prueba y tm_0 es el tiempo de ejecución de las operaciones aritméticas expuestas en [130].

En la tabla D.2 se muestra la complejidad de los algoritmos que destacaron en su desempeño de acuerdo al NFR, la TE y el EDES. Los autores de los algoritmos G-CMA-ES [111] y L-CMA-ES [109] no presentan la complejidad en sus resultados de acuerdo al expresión (D.1), por ello los valores en la tabla se expresan con asteriscos.

Nombre	10 variables	30 variables
K-PCX	138	338.12
G-CMA-ES	*	*
L-CMA-ES	*	*
DMS-L-PSO	1.1386	1.2330
EDA	0.4035	0.6085
ED	1.0413	2.9379
L-SaDE	0.8264	0.8806

Tabla D.2: Complejidad de los algoritmos (medida en segundos).

D.1.6. Discusión del análisis comparativo

En las funciones uni-modales y multi-modales resueltas con 10 y 30 variables la robustez del G-CMA-ES es incomparable, dado que puede resolver más funciones que los demás algoritmos y el número de evaluaciones que necesita para resolverlas es usualmente, más bajo, aún cuando el número de variables se incrementa. Aunque no presentan la complejidad del algoritmo en sus resultados, se intuye que es mucho mayor a la complejidad de la ED, dado que el G-CMA-ES debe calcular la distribución normal multivariada y la matriz de covarianza.

En el caso de 10 variables, la ED es el segundo mejor algoritmo, con base en el número de funciones (uni-modales y multi-modales) que resuelve y el éxito del desempeño. Sin embargo, cuando se aumenta el número de variables el desempeño de la ED decrece considerablemente. No obstante, la complejidad de este algoritmo es baja (ver tabla D.2), ya que sólo emplea operaciones aritméticas simples como la multiplicación, suma y resta, en su ejecución.

En el caso de 30 variables, la segunda mejor opción para funciones uni-modales es el L-CMA-ES y para funciones multi-modales es el K-PCX. Este último presenta la complejidad más alta de todos los algoritmos mostrados en la tabla D.2.

D.2. Comparación entre el algoritmo G-CMA-ES y la ED

El algoritmo G-CMA-ES y la evolución diferencial son los algoritmos que obtuvieron mejor desempeño en funciones uni-modales y multi-modales con diez variables, en relación al análisis anterior. No obstante, el problema principal que se ataca en esta tesis, presenta trece variables si se fija el punto principal (punto que pertenece a los parámetros intrínsecos) o quince variables si también se hace una búsqueda del punto principal. De tal forma que el análisis comparativo no garantiza que la evolución diferencial tenga un desempeño similar para trece o quince variables, a diferencia del algoritmo G-CMA-ES, el cual presenta un buen desempeño hasta treinta variables. Cabe señalar que, aunque el algoritmo G-CMA-ES presenta un buen desempeño en las funciones de prueba, no necesariamente puede tener un buen desempeño en problemas reales [197], como el que se presenta en este trabajo. Es por ello que se compararon ambos algoritmos en el problema de la reconstrucción tridimensional empleando el número máximo de variables que puede tener el problema (15 variables).

D. COMPARACIÓN DE LOS ALGORITMOS EVOLUTIVOS

Los experimentos consisten en ejecutar 100 veces cada uno de los dos algoritmos para mostrar su desempeño y complejidad; en el primer experimento se ejecutaron los algoritmos hasta alcanzar las 90,000 evaluaciones, en el segundo experimento se ejecutaron los algoritmos hasta alcanzar las 150,000 evaluaciones (número de evaluaciones que se emplearon en la mayoría de los experimentos presentados en este trabajo). Al mismo tiempo se calculó el tiempo que tardan en realizar cada ejecución.

Para ambos experimentos se emplearon 24 puntos de un anillo tridimensional. Cada punto tridimensional fue transformado en un punto bidimensional por cada imagen, mediante su respectiva matriz de proyección. Las matrices de proyección M y M' fueron generadas a partir de los datos que se muestran en la tabla D.3.

Imagen	α	β	γ	t_x	t_y	t_z	f	o_x	o_y
1	10	30	20	2	1	20	200	320	240
2	5	16	2	1	5	20	200	320	240

Tabla D.3: Parámetros de la cámara para generar los puntos en las dos imágenes.

A cada punto se le añadió ruido gaussiano, con media cero y desviación estándar de un píxel.

Los parámetros empleados para el algoritmo G-CMA-ES son: un vector *sigma* de igual dimensión al número de variables, en este caso 15 y el número de individuos de la población inicial. Estos parámetros se fijaron con base en la sugerencias del autor [111], es decir, sigma igual a $0.3 * (\text{límite superior} - \text{límite inferior})$ y la población inicial igual a $4 + \lceil 3 \log(d) \rceil$, donde d es el número de variables del problema.

En el caso de la evolución diferencial, los valores de los parámetros son los siguientes: la constante de diferencia igual a 0.85, la constante de recombinación igual a 0.9 y 30 individuos en la población (valores empleados en los experimentos de esta tesis).

En ambos algoritmos se utilizaron los siguientes límites de las variables: para la longitud del foco el intervalo es de $[10, 1000]$, para el punto principal el intervalo es de $[270, 370]$ para o_x y $[190, 290]$ para o_y , en los ángulos de rotación el intervalo es de $[0, \pi]$, y para el vector de traslación el intervalo es de $[-1, 1]$.

En la tabla D.4 se presentan las estadísticas del error del primer experimento y en la tabla D.5 del segundo experimento para ambos algoritmos. En el primer experimento se observa que el desempeño de la ED es ligeramente mejor al desempeño del G-CMA-ES, ya que los valores estadísticos (mínimo, media, máximo y desviación estándar) que se obtuvieron al ejecutar la ED son menores que los valores del G-CMA-ES. En el segundo experimento sucede algo similar, la diferencia radica en que el G-CMA-ES obtiene el error con menor valor.

Algoritmo	Mejor	Media	Peor	d.e.	i.c. de la media
G-CMA-ES	0.458	0.755	1.326	0.189	[0.723,0.796]
ED	0.407	0.678	1.156	0.140	[0.652,0.713]

Tabla D.4: Estadísticas del error del primer experimento.

Algoritmo	Mejor	Media	Peor	d.e.	i.c. de la media
G-CMA-ES	0.332	0.682	0.985	0.123	[0.655,0.707]
ED	0.385	0.632	0.843	0.102	[0.610,0.655]

Tabla D.5: Estadísticas del error del segundo experimento.

En las tablas D.6 y D.7 se presentan las estadísticas de la longitud del foco obtenidas del primer experimento y segundo experimento respectivamente. En el primer experimento se observa que el valor del foco varía más al emplear el G-CMA-ES, aunque su media es casi igual a la media de la ED. En el segundo experimento, el valor del foco varía más al emplear el G-CMA-ES, sin embargo la media se acerca más al valor real del foco.

Algoritmo	Mínimo	Media	Máximo	d.e.	i.c. de la media
G-CMA-ES	110.000	172.549	301.464	54.805	[162.143,184.162]
ED	122.958	173.119	302.835	33.170	[167.381,180.114]

Tabla D.6: Estadísticas de la longitud del foco del primer experimento.

Algoritmo	Mínimo	Media	Máximo	d.e.	i.c. de la media
G-CMA-ES	161.067	201.104	243.390	21.894	[196.813,205.461]
ED	178.907	192.487	222.726	9.987	[190.491,194.934]

Tabla D.7: Estadísticas de la longitud del foco del segundo experimento.

Posteriormente se estimó la complejidad del algoritmo mediante la ecuación (D.1). Ambos algoritmos se ejecutaron en la misma computadora (procesador Intel core duo a 2.16Ghz), bajo la misma plataforma (Windows) y se empleó el mismo lenguaje de programación (Matlab 7.0), por ello se puede asignar el valor de uno a la variable tm_0 . La variable tm_1 es el valor de la media del tiempo de 100 ejecuciones, donde cada ejecución evalúa la función objetivo 90,000 veces y 150,000 veces para el primer y segundo experimento respectivamente. Por cada algoritmo, la variable \widehat{tm}_2 es la media del tiempo de ejecución del primer y segundo experimento. En la tabla D.8 se presentan los resultados de la complejidad de cada algoritmo. Estos resultados muestran que la complejidad del algoritmo G-CMA-ES es mayor por 5.2 veces a la complejidad de la ED cuando se realizan 90,000 evaluaciones, y este valor aumenta a 6.03 cuando se realizan 150,000 evaluaciones de la función objetivo.

Evaluaciones	Algoritmo	\widehat{tm}_2	tm_1	tm_0	$(\widehat{tm}_2 - tm_1)/tm_0$
90,000	G-CMA-ES	60.209	47.336	1	12.873
	ED	49.808	47.336	1	2.472
150,000	G-CMA-ES	105.153	79.327	1	25.826
	ED	83.605	79.327	1	4.278

Tabla D.8: Complejidad de los algoritmos.

D.2.1. Discusión de resultados

En los experimentos mostrados anteriormente, la ED exhibe un desempeño ligeramente mejor que el G-CMA-ES al atacar el problema de la reconstrucción tridimensional. La diferencia de este desempeño se incrementa cuando se tiene un menor número de evaluaciones (primer experimento) y esto se refleja en los valores obtenidos de la longitud del foco.

Por otro lado, la complejidad del algoritmo G-CMA-ES es mayor que la ED y cuando se incrementa el número de evaluaciones, el tiempo de ejecución del G-CMA-ES se incrementa más rápido que la ED. Con base en estos resultados es más conveniente emplear la ED que el G-CMA-ES cuando se desea resolver el problema de la reconstrucción tridimensional.

Bibliografía

- [1] J. Piaget. *The Construction of Reality in the Child*. Basic Books, Inc., New York, USA, 1954.
- [2] I. Bierderman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.
- [3] Stephan Heuel. *Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction*. Springer. Lecture Notes in Artificial Intelligence. Volume 3008, 2004.
- [4] Y. Ma, S. Soatto, J. Koěcká, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.
- [6] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, volume 26, pages 75–84, New York, NY, USA, 1998. ACM.
- [7] I-C. Lin, J.-S. Yeh, and M. Ouhyoung. Extracting 3D facial animation parameters from multiview video clips. *IEEE Computer Graphics and Applications*, 8(2):72–80, 2002.
- [8] G. Moiza, A. Tal, I. Shimshoni, D. Barnett, and Y. Moses. Image-based animation of facial expressions. *International Journal of Computer Graphics*, 18(7):445–465, 2002.
- [9] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 67–74, New York, NY, USA, 1998. ACM Press.
- [10] S. R. Musse and D. Thaimann. Hierarchical model for real time simulation of virtual human crowds. *Transactions on Visualization and Computer Graphics*, 7(2):152–164, 2001.
- [11] M. Pollerfeys and L.V. Gool. 3D recording for archaeological fieldwork. *IEEE Computer Graphics and Applications*, 27(2):20–27, 2003.

-
- [12] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D Machine Vision. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.
- [13] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.
- [14] B. Triggs. Autocalibration from planar scenes. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, volume 1, pages 89–105, London, UK, 1998. Springer-Verlag.
- [15] R. Cipolla, T. Drummond, and D. Robertson. Calibration from vanishing points in image of architectural scenes. In *The 10th British Machine Vision Conference*, volume 2, pages 382–391, 1999.
- [16] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
- [17] D. Svedberg and S. Carlsson. Calibration, pose and novel views from single images of constrained scenes. *Pattern Recognition Letters*, 21(13-14):1125–1133, 2000.
- [18] J. Koěcká and W. Zhang. Video compass. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision*, volume 4, pages 476–490, London, UK, 2002. Springer-Verlag.
- [19] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 579–587, London, UK, 1992. Springer-Verlag.
- [20] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
- [21] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 237–256, London, UK, 1994. Springer-Verlag.
- [22] P. Sturm. A case against kruppa’s equations for camera self-calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1199–1204, 2000.
- [23] K. Kanatani, A. Nakatsuji, and Y. Sugaya. Stabilizing the focal length computation for 3-D reconstruction from two uncalibrated views. *International Journal of Computer Vision*, 66(2):109–122, 2006.
- [24] M. Pollefeys, R. Koch, and L. V. Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.
- [25] P. Sturm. On focal length calibration from two views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume 2, pages 145–150. IEEE Computer Society Press, 2001.
-

- [26] P. Sturm, Z. Cheng, P. C. Chao, and A. N. Poo. Focal length calibration from two views: method and analysis of singular cases. *Computer Vision and Image Understanding*, 99(1):58–95, 2005.
- [27] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [28] S. J. Ahn, W. Rauh, and H. J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303, 2001.
- [29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
- [30] B. Triggs. Matching constraints and the joint image. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 338–343, Washington, DC, USA, 1995. IEEE Computer Society.
- [31] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 951–956, Washington, DC, USA, 1995. IEEE Computer Society.
- [32] A. Heyden. *Geometry and algebra of multiple projective transformations*. PhD thesis, Lund Institute Technology, Sweden, 1995.
- [33] D. Q. Huynh, R. I. Hartley, and A. Heyden. Outlier correction in image sequences for the affine camera. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 585–590, Washington, DC, USA, 2003. IEEE Computer Society.
- [34] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.
- [35] J. Knight and I. D. Reid. Self-calibration of a stereo rig in a planar scene by data combination. In *ICPR*, pages 1411–1414, 2000.
- [36] E. Malis and R. Cipolla. Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1268–1272, 2002.
- [37] R. I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.
- [38] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–607, 1997.

-
- [39] O. Faugeras and T. Papadopoulos. Grassmann-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. Technical Report RR-3225, INRIA, Sophia-Antipolis, France, 1997.
- [40] R. I. Hartley. Minimizing algebraic error. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, pages 469–476, Washington, DC, USA, 1998. IEEE Computer Society.
- [41] Y. Z. Ma, W. J. Liu, and Y. N. Sun. A robust self-calibration algorithm based on three views. In *CIT*, pages 741–746, 2004.
- [42] J. Koenderink and A. V. Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8:377–385, 1991.
- [43] Q. Long. Affine stereo calibration for relative affine shape reconstruction. In *Proceedings of the fourth British Machine Vision Conference, Surrey, England*, pages 659–668, 1993.
- [44] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.
- [45] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 1100–1105, Washington, DC, USA, 1997. IEEE Computer Society.
- [46] P. Sturm and S. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single images. In *Proceedings British Machine Vision Conference*, pages 265–274, Nottingham, 1999.
- [47] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.
- [48] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.
- [49] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Applications of Invariance in Computer Vision*, LNCS 588, pages 563–578. Springer-Verlag, 1992.
- [50] Y. Ma, R. Vidal, J. Koěcká, and S. Sastry. Kruppa equation revisited: its renormalization and degeneracy. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision*, volume 2, pages 561–577, London, UK, 2000. Springer-Verlag.
- [51] Y. Seo and K. Hong. About the self-calibration of a rotating and zooming camera. In *ECCV '94: Proceedings of the Third European Conference on Computer Vision*, volume 1, pages 463–470, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
-

- [52] J. Qiang and D. Songtao. Self-calibration of a rotating camera with a translational offset. *IEEE Transactions on Robotics and Automation*, 20(1):1–14, 2004.
- [53] L. Quan and R. Mohr. Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters*, 9:279–286, 1989.
- [54] R. Collins and R. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Proceedings of the Third International Conference on Computer Vision*, pages 400–403, Osaka, Japan, 1990.
- [55] Z. Yang and F. Cao. Linear six-point algorithm of camera self-calibration and 3D reconstruction from single-view or multi-views. In *International Conference on Signal Processing*, volume 2, pages 16–20, 2006.
- [56] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.
- [57] Q.-T. Luong and O. D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996.
- [58] Cyril Zeller and Olivier Faugeras. Camera self-calibration from video sequences: the kruppa equations revisited. Technical Report RR-2793, INRIA, France, 1996.
- [59] Q.-T. Luong and O. D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.
- [60] J. Ponce, D. H. Marimont, and T. A. Cass. Analytical methods for uncalibrated stereo and motion reconstruction. In *ECCV '94: Proceedings of the third European Conference on Computer Vision*, volume 1, pages 463–470, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
- [61] A. Heyden and K. Astrom. Euclidean reconstruction from constant intrinsic parameters. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, pages 339–343, Washington, DC, USA, 1996. IEEE Computer Society.
- [62] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using virtual parallax. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 1030–1036, Washington, DC, USA, 1995. IEEE Computer Society.
- [63] J. Chai and S. D. Ma. Robust epipolar geometry estimation using genetic algorithm. *Pattern Recognition Letters*, 19(9):829–838, 1998.
- [64] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, volume 4, pages 790–796, Washington, DC, USA, 1998. IEEE Computer Society.

-
- [65] Y. Ma, J. Koěcká, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.
- [66] R. I. Hartley and S.-Anan. Reconstruction from two views using approximate calibration. In *Proceedings 5th Asian Conference on Computer Vision*, volume 1, pages 338–343, Melbourne, Australia, 2002.
- [67] R. I. Hartley and F. Kahl. Critical configurations for projective reconstruction from multiple views. *International Journal of Computer Vision*, 71(1):5–47, 2007.
- [68] T. Ueshiba and F. Tomita. A closed-form solution for a two-view self-calibration problem under fixation. In *3DPVT*, pages 648–655, 2004.
- [69] J.-E. Ha and D. J. Kang. Initialization method for self-calibration using 2-views. *Pattern Recognition*, 38(1):143–150, 2005.
- [70] P. A. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure from motion. In *ECCV '94: Proceedings of the Third European Conference on Computer Vision*, volume 2, pages 85–96, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
- [71] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
- [72] L. Brand. *Vector and Tensor Analysis*. John Wiley & Sons, Inc., 10th printing edition, 1966.
- [73] P. L. Rosin. A note on the least squares fitting of ellipses. *Pattern Recognition Letters*, 14(10):799–808, 1993.
- [74] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
- [75] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [76] S. S. Rao. *Engineering Optimization Theory and Practice*. New York: Wiley, ISBN: 0-471-55034-5, 3rd edition, 1996.
- [77] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
- [78] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis. *Engineering Optimization Methods and (A)pplications*. New Jersey: Wiley, ISBN: 978-0-471-55814-9, 2nd edition, 2006.

- [79] D. A. V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
- [80] C. A. Coello and G. B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004. ISBN 981-256-106-4.
- [81] C. A. Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.
- [82] F. C. Domínguez. *Métodos Numéricos Aplicados a la Ingeniería*. México D.F.: CECSA, ISBN: 978-26-1260-8, 1st edition, 1996.
- [83] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
- [84] D. D. McCracken and W. S. Dorn. *Metodos Numéricos y Programacion FORTRAN: con Aplicaciones en Ingeniería y Ciencias*. Ed. Limusa, 1966.
- [85] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Presss Syndicate of the University of Cambridge, 2002.
- [86] W. Lao, Z. Cheng, A. H. Kam, T. Tan, and A. A. Kassim. Focal length self-calibration based on degenerated kruppa's equations method and evaluation. In *ICIP*, pages 3391–3394, 2004.
- [87] G. J. Mendel. Experiments in plant hybridisation. *Journal of Royal Horticultural Society*, 26:1–32, 1901.
- [88] A. Wismann. *The Germ Plasm: A Theory of Heredity*. Scott, London, UK, 1893.
- [89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.
- [90] R. Landa. *Use of Domain Information to Improve the Performance of an Evolutionary Algorithm*. PhD thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico, June 2007.
- [91] B. Yuan and M. Gallagher. Experimental results for the special session on real-parameter optimization at CEC 2005: A simple, continuous EDA. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 1792–1799, Edinburgh, Scotland, 2005. IEEE.
- [92] J. H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 9(3):297–314, 1962.

- [93] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg, Natural Computing Series, New York, USA, 2003.
- [94] D. E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5:139–167, 1991.
- [95] T. Higuchi, S. Tsutsui, and M. Yamamura. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 365–374, London, UK, 2000. Springer-Verlag.
- [96] K. Deb. A population-based algorithm generator for real-parameter optimization. *Soft Computing*, 9(4):236–253, April 2005.
- [97] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1996.
- [98] T. Krink, B. Filipič, G. B. Fogel, and R. Thomsen. Noisy optimization problems - A particular challenge for differential evolution? In *Proceedings of the Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 332–339. IEEE Service Center, June 2004.
- [99] H. Kita. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. *Evolutionary Computation*, 9(2):223–241, 2001.
- [100] P. J. Ballester and J. N. Carter. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In E. Cantú-Paz, J. A. Foster, K. Deb, D. Lawrence, and R. Roy, editors, *Genetic and Evolutionary Computation Conference*, pages 706–717, Berlin, 2003. Springer-Verlag. Lecture Notes in Computer Science. Volume 2723.
- [101] M. Rocha, J. P. Pinto, I. Rocha, and E. C. Ferreira. Evaluating evolutionary algorithms and differential evolution for the online optimization of fermentation processes. In E. Marchiori, J. H. Moore, and J. C. Rajapakse, editors, *EvoBIO*, pages 236–246. Springer. Lecture Notes in Computer Science. Volume 4447, 2007.
- [102] A. Sinha, S. Tiwari, and K. Deb. A population-based, steady-state procedure for real-parameter optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 514–521, Edinburgh, UK, September 2005. IEEE.
- [103] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
- [104] P. J. Ballester, J. Stephenson, J. N. Carter, and K. Gallagher. Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 498–505, Edinburgh, UK, September 2005. IEEE.

- [105] W. E. Hart and R. K. Belew. Optimizing an arbitrary function is hard for the genetic algorithm. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 190–195, San Mateo CA, 1991. Morgan Kaufmann Publishers, Inc.
- [106] H. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *International Journal of Natural Computing*, 1(1):3–52, 2002.
- [107] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, USA, 1995.
- [108] P. Pošík. Real-parameter optimization using the mutation step co-evolution. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 872–879, Edinburgh, UK, September 2005. IEEE.
- [109] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 1777–1784, Edinburgh, UK, 2005. IEEE.
- [110] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):345–369, 2002.
- [111] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 1769–1776, Edinburgh, UK, 2005. IEEE.
- [112] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966.
- [113] D. B. Fogel. *Evolutionary computation: Toward a new philosophy of machine intelligence*. IEEE Press, Piscataway, NJ, USA, 1995.
- [114] X. Yao, Y. Liu, and G. Liu. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.
- [115] C.-Y. Lee and X. Yao. Evolutionary programming using mutations based on the levy probability distribution. *IEEE Transactions on Evolutionary Computation*, 8(1):1–13, 2004.
- [116] T. Bäck, G. Rudolph, and H.-P. Schwefel. Evolutionary programming and evolution strategies: Similarities and differences. In *Proceedings of the second Annual Conference on Evolutionary Programming*, pages 11–22, La Jolla, CA, USA, 1993. Evolutionary Programming Society.
- [117] R. C. Eberhart and X. Hu. Human tremor analysis using particle swarm optimization. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzalá, editors, *Proceedings of the Congress on Evolutionary Computation (CEC'1999)*, volume 3, pages 1927–1930, Mayflower Hotel, Washington D.C., USA, 1999. IEEE Press.

-
- [118] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, June 2004.
- [119] J. Koza. *Genetic Programming II*. MIT Press, Cambridge, MA, USA, 1994.
- [120] Y. Fukuyama, S. Takayama, Y. Nakanishi, and H. Yoshida. A particle swarm optimization for reactive power and voltage control in electric power systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1523–1528, Orlando, Florida, USA, July 1999. Morgan Kaufmann.
- [121] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi. A particle swarm optimization for reactive power and voltage control considering voltage stability. In *Proceedings of the International Conference on Intelligent System Application to Power System*, pages 117–121, Rio de Janeiro, Brazil, 1999.
- [122] J. Vesterstrom and R. Thomsen. A comparative study of differential evolution particle swarm optimization and evolutionary algorithms on numerical benchmark problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2004)*, volume 3, pages 1980–1987. IEEE Service Center, 2004.
- [123] J. J. Liang and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 522–528, Edinburgh, UK, 2005. IEEE.
- [124] A. Acan, A. Unveren, and M. Bodur. The latest vs. averaged recent experience: Which better guides a PSO algorithm? In G. G. Yen, S. M. Lucas, G. Fogel, G. Kendall, R. Salomon, B.-T. Zhang, C. A. Coello, and T. P. Runarsson, editors, *Proceedings of the Congress on Evolutionary Computation (CEC'2006)*, pages 414–419, Vancouver, BC, Canada, July 2006. IEEE Press.
- [125] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 178–187, London, UK, 1996. Springer-Verlag. Lecture Notes in Computer Science. Volume 1141.
- [126] H. Mühlenbein, J. Bendisch, and H.-M. Voigt. From recombination of genes to the estimation of distributions II. Continuous parameters. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 188–197, London, UK, 1996. Springer-Verlag. Lecture Notes in Computer Science. Volume 1141.
- [127] Q. Zhang, J. Sun, E. Tsang, and J. Ford. Hybrid estimation of distribution algorithm for global optimization. *Engineering Computations*, 21(1), January 2004.
- [128] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena. Optimization in continuous domains by learning and simulation of Gaussian networks. In *Proceeding of the Genetic and Evolutionary Computation Conference Workshop Program*, pages 201–204, Las Vegas, Nevada, USA, July 2000.

- [129] T. K. Paul and H. Iba. Optimization in continuous domain by real-coded estimation of distribution algorithm. *Design and Application of Hybrid Intelligent Systems*, pages 262–271, 2003.
- [130] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2005.
- [131] B. Yuan, M. Gallagher, and S. Crozier. MRI magnet design: Search space analysis, EDAs and a real-world problem with significant dependencies. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO'05)*, pages 2141–2148, New York, NY, USA, 2005. ACM.
- [132] K. V. Price. An introduction to differential evolution. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 79–108. McGraw-Hill, London, UK, 1999.
- [133] E. Mezura, J. Velázquez, and C. A. Coello. A comparative study of differential evolution variants for global optimization. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 485–492, New York, NY, USA, 2006. ACM Press.
- [134] V. Feoktistov and S. Janaqi. Generalization of the strategies in differential evolution. In *In Proceedings of the 18th International Parallel and Distributed Processing Symposium*, volume 7, page 165a, Santa Fe, New Mexico, USA, 2004. IEEE Computer Society.
- [135] O. Soliman, L. T. Bui, and H. Abbass. The effect of a stochastic step length on the performance of the differential evolution algorithm. In *Proceedings of the Congress on Evolutionary Computation (CEC'2007)*, pages 2850–2857. IEEE Press, September 2007.
- [136] N. Noman and H. Iba. Enhancing differential evolution performance with local search for high dimensional function optimization. In *GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 967–974, New York, NY, USA, 2005. ACM Press.
- [137] J. Alvarez, C. A. C. Villar, and E. A. P. Flores. Evolutionary dynamic optimization of a continuously variable transmission for mechanical efficiency maximization. In A. Gelbukh, A. de Albornoz, and H. Terashima-Marín, editors, *MICAI 2005: Advances in Artificial Intelligence*, pages 1093–1102. Springer-Verlag. Lecture Notes in Artificial Intelligence. Volume 3789, 2005.
- [138] J. Lampinen. A constraint handling approach for the differential evolution algorithm. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 2, pages 1468–1473. IEEE Service Center, 2002.
- [139] R. Storn. Differential evolution design of an IIR-filter with requirements for magnitude and group delay. In *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 268–273. IEEE, 1996.

- [140] J. Rönkkönen, S. Kukkonen, and K. V. Price. Real-parameter optimization with differential evolution. In *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'2005)*, volume 1, pages 567–574. IEEE Service Center, 2005.
- [141] B. V. Babu and R. Angira. Optimization of non-linear functions using evolutionary computation. In *Proceedings of 12th ISME Conference*, pages 153–157, January 2001.
- [142] R. Thomsen. Flexible ligand docking using differential evolution. In *Proceedings of the Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2354–2361. IEEE Press, 2003.
- [143] Y. Shi, H. Teng, and Z. Li. Cooperative co-evolutionary differential evolution for function optimization. In L. Wang, K. Chen, and Y.-S. Ong, editors, *Proceedings of the First International Conference on Natural Computation*, pages 1080–1088. Springer-Verlag. Lecture Notes in Computer Science. Volume 3611, August 2005.
- [144] R. Mendes, M. Rocha, I. Rocha, and E. C. Ferreira. A comparison of algorithms for the optimization of fermentation processes. In *Proceedings of the Conference on Evolutionary Computation (CEC'2006)*, pages 2018–2025. IEEE Press, July 2006.
- [145] N. Cruz-Cortés. *Sistema Inmune Artificial para Solucionar Problemas de Optimización*. PhD thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico, Octubre 2004.
- [146] E. Michielssen, J.-M. Sajer, S. Ranjithan, and R. Mittra. Design of Lightweight, Broad-Band Microwave Absorbers Using Genetic Algorithms. *IEEE Transactions on Microwave Theory and Techniques*, 41(6/7):1024–1031, 1993.
- [147] B. J. Ritzel, J. W. Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30(5):1589–1603, may 1994.
- [148] M. A. Lee and H. Esbensen. Fuzzy/Multiobjective Genetic Systems for Intelligent Systems Design Tools and Components. In W. Pedrycz, editor, *Fuzzy Evolutionary Computation*, pages 57–80. Kluwer Academic Publishers, Boston, Massachusetts, 1997.
- [149] F. Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, pages 193–197, Berlin, Germany, October 1991. Springer-Verlag. Lecture Notes in Computer Science. Volume 496.
- [150] S. R. Norris and W. A. Crossley. Pareto-Optimal Controller Gains Generated by a Genetic Algorithm. In *AIAA 36th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1998. AIAA Paper 98-0010.

- [151] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.
- [152] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kaufman Publishers.
- [153] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.
- [154] M. Erickson, A. Mayer, and J. Horn. The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 681–695. Springer-Verlag. Lecture Notes in Computer Science. No. 1993, 2001.
- [155] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.
- [156] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.
- [157] J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [158] J. D. Knowles, D. W. Corne, and M. J. Oates. The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 839–848, Berlin, September 2000. Springer.
- [159] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In L. Spector, E. D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 283–290, San Francisco, California, 2001. Morgan Kaufmann Publishers.

-
- [160] C. A. Coello and G. Toscano. A micro-genetic algorithm for multiobjective optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, LNCS 1993, pages 126–140. Springer-Verlag, 2001.
- [161] C. A. Coello and G. Toscano. Multiobjective optimization using a micro-genetic algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.
- [162] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
- [163] D. S. Weile, E. Michielssen, and D. E. Goldberg. Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms. In *1996 IEEE Antennas and Propagation Society International Symposium Digest*, volume 1, pages 592–595, Baltimore, Maryland, July 1996.
- [164] A. L. Blumel, E. J. Hughes, and B. A. White. Fuzzy Autopilot Design using a Multiobjective Evolutionary Algorithm. In *2000 Congress on Evolutionary Computation*, volume 1, pages 54–61, Piscataway, New Jersey, July 2000. IEEE Service Center.
- [165] P. M. Reed, B. S. Minsker, and D. E. Goldberg. Designing a New Elitist Nondominated Sorted Genetic Algorithm for a Multiobjective Long Term Groundwater Monitoring Application. In *Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, pages 352–358, San Francisco, California, July 2001.
- [166] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science. No. 1917.
- [167] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann Publishers, 2001.
- [168] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing local optima in single-objective problems by multi-objectivization. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, LNCS 1993, pages 269–283, Berlin, 2001. Springer-Verlag.

- [169] M. T. Jensen. Guiding single-objective optimization using multi-objective methods. In G. Raidl et al., editor, *Applications of Evolutionary Computing. Evoworkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM*, LNCS 2611, pages 199–210, Essex, UK, 2003. Springer-Verlag.
- [170] K. Deb, M. Mohan, and S. Mishra. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of pareto optimal solutions. *Evolutionary Computation*, 13(4):501–525, 2005.
- [171] L. V. Santana-Quintero and C. A. Coello Coello. An algorithm based on differential evolution for multi-objective problems. *International Journal of Computational Intelligence Research*, 1(2):151–169, 2005.
- [172] K. Mitra and R. Gopinath. Multiobjective Optimization of an Industrial Grinding Operation Using Elitist Nondominated Sorting Genetic Algorithm. *Chemical Engineering Science*, 59(2):385–396, 2004.
- [173] Tea Robič and Bogdan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In C. A. Coello, editor, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science. Volume 3410.
- [174] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [175] G. Arfken. *Mathematical Methods for Physicists 3rd ed.* Academic Press, Orlando, FL, USA, 1985.
- [176] S. Agarwal, M. K. Chandraker, F. Kahl, D. J. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. In *ECCV 2006: Proceedings of the Ninth European Conference on Computer Vision*, pages 592–605. Springer. Lecture Notes in Computer Science. Volume 3951, 2006.
- [177] G. Olague and E. Dunn. Development of a practical photogrammetric network design using evolutionary computing. *The Photogrammetric Record*, 22(117):22–38, March 2007.
- [178] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, 2004.
- [179] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 978–985, Washington, DC, USA, 2005. IEEE Computer Society.
- [180] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 1, pages 686–693, Washington, DC, USA, 2005. IEEE Computer Society.

- [181] I. Vite, L. G. de la Fraga, N. Cruz-Cortés, and G. Toscano. Optimal triangulation in 3D Computer Vision using a multi-objective evolutionary algorithm. In M. Giacobini et al., editor, *EvoWorkshops 2007 (EvoIASP)*, LNCS 4448, pages 330–339. Springer-Verlag, Berlin Heidelberg. Valencia, España, 2007.
- [182] S. I. Grossman. *Álgebra Lineal*. McGraw-Hill, Naucalpan, Edo. de México, México, 1996.
- [183] K.-Y. K. Wong, P. R. S. Mendonça, and R. Cipolla. Camera calibration from surfaces of revolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):147–161, 2003.
- [184] L. G. de la Fraga and I. Vite. Direct 3D metric reconstruction from two views using differential evolution. In V. Ciesielski et al., editor, *WCCI '08: Proceedings of the IEEE World Congress on Computational Intelligence and Evolutionary Computer Vision*. Springer-Verlag, Berlin Heidelberg. Hong Kong, China, 2008.
- [185] L. G. de la Fraga and I. Vite. Direct 3D metric reconstruction from multiple views using differential evolution. In M. Giacobini et al., editor, *EvoWorkshops 2008 (EvoIASP)*, LNCS 4974, pages 341–346. Springer-Verlag, Berlin Heidelberg. Napoli, Italy, 2008.
- [186] L. G. de la Fraga, I. Vite, and N. Cruz-Cortés. Euclidean distance fit of ellipses with a genetic algorithm. In M. Giacobini et al., editor, *EvoWorkshops 2007 (EvoIASP)*, LNCS 4448, pages 359–366. Springer-Verlag, Berlin Heidelberg. Valencia, España, 2007.
- [187] R. Safaee-Rad, K. C. Smith, B. Benhabib, and I. Tchoukanov. Application of moment and fourier descriptors to the accurate estimation of elliptical shape parameters. *Pattern Recognition Letters*, pages 497–508, 1992.
- [188] K. U. Kasemir and K. Betzler. Detecting ellipses of limited eccentricity in images with high noise levels. *Image and Vision Computing*, 21(2):221–227, 2003.
- [189] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
- [190] P. O’Leary and P. Zsombor-Murray. Direct and specific least-square fitting of hiperbolae and ellipses. *Journal of Electronic Imaging*, 13(3):492–503, 2004.
- [191] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.
- [192] Y. Kanazawa and K. Kanatani. Robust image matching under a large disparity. In *Proceedings 6th Asian Conference Computer Vision*, volume 2, pages 1128–1133, 2004.

- [193] Y. Kanazawa and Y. Ito. Image matching by ransac using multiple non-uniform distributions computed from images. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP 2006), Setubal, Portugal*, volume 2, pages 377–382, 2006.
- [194] C. García-Martínez and M. Lozano. Hybrid real-coded genetic algorithms with female and male differentiation. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 896–903, Edinburgh, UK, September 2005. IEEE.
- [195] D. Molina, F. Herrera, and M. Lozano. Adaptive local search parameters for real-coded memetic algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 888–895, Edinburgh, UK, September 2005. IEEE.
- [196] A. K. Qin and P. N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC'2005)*, pages 1785–1791, Edinburgh, UK, September 2005. IEEE.
- [197] A. E. Eiben and M. Jelasity. A critical note on experimental research methodology in EC. In *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, pages 582–587. IEEE Press, May 2002.