



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco
Departamento de Computación

**Ajuste Robusto de Múltiples Elipses Usando
Algoritmos Genéticos**

Tesis que presenta

César Cruz Díaz

para obtener el Grado de

Maestro en Ciencias en Computación

Directores de la Tesis

Dr. Luis Gerardo de la Fraga

Dr. Oliver Steffen Schütze

México, Distrito Federal

Febrero, 2012

Resumen

El problema de ajuste de elipses consiste en encontrar la mejor elipse que se ajusta a un conjunto de puntos, éste es un problema que surge en diversas áreas, por ejemplo en visión computacional, Robótica, detección de rostros, Astronomía, entre otras.

En esta tesis, nosotros usamos un algoritmo genético para la detección y ajuste de múltiples elipses que están dadas implícitamente por un conjunto de datos que contiene ruido. El objetivo general es detectar rápidamente todo el conjunto de elipses, sin información adicional sobre el tamaño y forma de las elipses, o la cantidad de ruido en el conjunto de datos, pero conociendo el número de elipses y proporcionando un valor de umbral.

En este trabajo, nosotros desarrollamos e investigamos —basados en un algoritmo genético estándar— tres diferentes funciones de aptitud que tenían diferentes ventajas y desventajas. A partir de los resultados numéricos, verificamos que estamos en condiciones de calcular de manera fiable y eficiente el conjunto de elipses en ciertas situaciones.

Abstract

The problem of ellipse fitting consists of finding the best fit ellipse to a set of points, it is a problem that arises in many areas, e.g. Computer Vision, Robotics, Face Detection, Astronomy, among others.

In this thesis, we use a genetic algorithm for the detection and fitting of multiple ellipses which are implicitly given by a data set which contains noisy data. The overall aim is to quickly detect the entire set of ellipses, without additional information about the sizes, and shapes of the ellipses, or the amount of noise in the data set, but knowing the number of ellipses and providing a threshold value.

In this work, we develop and investigate —based on a standard genetic algorithm— three different fitness functions which have different advantages and disadvantages. From numerical results we verify that we are yet able to reliably and efficiently compute the set of ellipses in certain situations.

Agradecimientos

Agradezco a mis padres **Efigenio Cruz Mora** y **Olga Díaz Vázquez** por todo el apoyo incondicional que me han dado desde que tengo memoria, sin el cual no hubiera podido cumplir mis metas.

Agradezco a mi hermano **Jiovanni Cruz Díaz** por todas las veces que me ha apoyado y ser mi compañero de juego desde que eramos pequeños.

Agradezco a mis hermanas **Yulisa Cruz Díaz** y **Diana Iris Cruz Díaz** por darme esperanza en los malos momentos.

Agradezco al **Dr. Luis Gerardo de la Fraga** por su valiosa ayuda, así como su interminable paciencia para conmigo.

Agradezco al **Dr. Oliver Schütze** por la paciencia que me tuvo, a pesar de mi mal inglés y mis limitaciones.

Agradezco a la **Dra. Nareli Cruz Cortés** y **Dra. Xiaoou Li** por tomarse el tiempo de revisar esta tesis y ayudarme a mejorar este trabajo.

Agradezco a mis **compañeros de generación** por la camaradería y la ayuda proporcionada cuando la necesite.

Agradezco a **Fernando Gutiérrez Méndez** por su amistad, sin su ayuda y consejos, mi estancia en la maestría habría sido muy difícil.

Agradezco a **Sofía Reza Cruz** por ser la mejor administrativa que he tenido el gusto de conocer, ser amable y atenta con cada uno de los estudiantes, espero nunca cambie.

Agradezco a **Felipa Rosas López** por ser una buena persona y explicarme como usar el material de papelería que estaba a disposición de los estudiantes.

Agradezco a **Carlos Daniel Vázquez Rosas** por ser un gran amigo y apoyarme en las buenas y en las malas.

Agradezco a **Alejandro Gonzalez Huitron** por el apoyo brindado.

Agradezco a **Internet** por existir.

Agradezco a los **videojuegos** por quitarme el estrés.

Agradezco al **CINVESTAV** por ser la institución que me dio la oportunidad de estudiar una maestría, además de poner a mi disposición las herramientas necesarias para la realización de esta tesis.

Por último agradezco al **CONACYT**, por el apoyo económico brindado durante los años en los que realicé mis estudios de maestría.

Índice general

Resumen	III
Abstract	V
Agradecimientos	VII
Índice de figuras	X
Índice de tablas	XIII
Índice de algoritmos	XVI
1. Introducción	1
1.1. Descripción del problema	1
1.2. Motivación	2
1.3. Metodología	4
1.4. Contribuciones	4
1.5. Organización de la tesis	5
2. Marco teórico	7
2.1. Elipse	7
2.2. Ajuste de curvas	8
2.3. Distancia	12
2.4. Optimización	13
2.5. Algoritmos evolutivos	14
2.6. Estado del arte	18
2.6.1. Detección de círculos en imágenes usando algoritmos genéticos	18
2.6.2. Ajuste directo de mínimos cuadrados	19
2.6.3. Transformada de Hough	19
2.6.4. Extracción de elipses robusto y rápido usando un algoritmo genético con múltiple población	19
2.6.5. Detección de elipses por EM	19
2.6.6. Heurísticas tipo RANSAC	20
2.6.7. Obtención de múltiples elipses mediante Evolución Diferencial	20

3. Ajuste robusto de una elipse	23
3.1. Ajuste por métodos analíticos	24
3.1.1. Ajuste algebraico	25
3.1.2. Regresión no lineal	28
3.1.3. Comportamiento de los métodos analíticos con una elipse . . .	29
3.2. Ajuste por métodos probabilistas	30
3.2.1. RANSAC + MMC	30
3.2.2. Evolución diferencial + suma de distancias ortogonales	32
3.2.3. PSO + SD	34
4. Ajuste robusto de múltiples elipses	37
4.1. RANSAC + Algoritmo genético	38
4.2. Funciones de aptitud	40
4.2.1. Número de puntos	41
4.2.2. Número de puntos en la elipse	41
4.2.3. Suma de distancias de Hausdorff	42
4.3. Características extras	42
4.3.1. Operador local	43
4.3.2. Condición de paro	44
4.4. Extracción de múltiples elipses	44
5. Resultados	49
5.1. Diseño de experimentos para el ajuste de una elipse	49
5.1.1. Experimentos con diferentes niveles de ruido	50
5.1.2. Experimentos con diferentes niveles de puntos atípicos	54
5.2. Diseño de experimentos para el ajuste de múltiples elipses	60
5.2.1. Experimentos con dos elipses	62
5.2.2. Experimentos con ruido en los puntos	66
5.2.3. Experimentos con diferentes niveles de puntos atípicos	69
5.3. Discusión	73
6. Aplicaciones	77
6.1. Calibración de cámaras	77
6.1.1. Calibración de la cámara usando planos	79
6.1.2. Auto-calibración de la cámara usando cuboides	81
6.1.3. Calibración de cámaras usando círculos concéntricos	82
6.2. Aplicación de la calibración de cámaras en realidad aumentada	86
6.2.1. Círculos concéntricos para calibración de la cámara	86
7. Conclusiones	91
Bibliografía	94

Índice de figuras

1.1. Parámetros de la elipse	2
1.2. Puntos que representan múltiples elipses	3
2.1. Puntos atípicos	10
2.2. Ruido estadístico	10
3.1. Puntos extraídos de una imagen de un patrón de dos círculos concéntricos. Claramente se nota que los dos círculos se proyectan como dos elipses.	24
3.2. Elipses encontradas en un conjunto de puntos con una elipse sin puntos atípicos	29
3.3. Elipses encontradas en un conjunto de puntos con una elipse y tres puntos atípicos	30
4.1. <i>Vecindad</i> $-\epsilon$ de una elipse candidata.	42
4.2. Múltiples elipses inmersas en ruido	45
4.3. Extracción de múltiples elipses en un ejemplo con cuatro elipses	48
5.1. Elipses generadas para los experimentos con diferentes niveles de ruido gaussiano	51
5.2. Gráficas de los porcentajes de error en cada parámetro con diferentes niveles de ruido	55
5.3. Elipses generadas para los experimentos, con diferentes niveles de puntos atípicos	56
5.4. Gráficas de los porcentajes de error en cada parámetro con diferentes porcentajes de puntos atípicos	61
5.5. Caso 1. Elipses disjuntas, la elipse 1 está formada por 100 puntos y la elipse 2 varía	62
5.6. Caso 2. Elipses anidadas, la elipse 1 está formada por 100 puntos y la elipse 2 varía	63
5.7. Caso 3. Elipses superpuestas, la elipse 1 está formada por 100 puntos y la elipse 2 varía	63
5.8. Gráfica de la elipse reconocida por GA-RANSAC $+ F_2$ en el caso 1, para los diferentes números de puntos en la elipse 2	64

5.9. Gráfica de la elipse reconocida por GA-RANSAC + F_2 en el caso 2, para los diferentes números de puntos en la elipse 2	64
5.10. Gráfica de la elipse reconocida por GA-RANSAC + F_2 en el caso 3, para los diferentes números de puntos en la elipse 2	64
5.11. Gráfica de la elipse reconocida por ED + SD en el caso 1, para los diferentes números de puntos en la elipse 2	65
5.12. Gráfica de la elipse reconocida por ED + SD en el caso 2, para los diferentes números de puntos en la elipse 2	65
5.13. Gráfica de la elipse reconocida por ED + SD en el caso 3, para los diferentes números de puntos en la elipse 2	66
5.14. Caso 1. Múltiples elipses disjuntas con diferentes niveles de ruido gaussiano	66
5.15. Caso 2. Múltiples elipses anidadas con diferentes niveles de ruido gaussiano	67
5.16. Caso 3. Múltiples elipses superpuestas con diferentes niveles de ruido gaussiano	67
5.17. Gráfica del número de elipses reconocidas por GA-RANSAC + F_2 en el caso 1, para los diferentes valores de σ	68
5.18. Gráfica del número de elipses reconocidas por GA-RANSAC + F_2 en el caso 2, para los diferentes valores de σ	68
5.19. Gráfica del número de elipses reconocidas por GA-RANSAC + F_2 en el caso 3, para los diferentes valores de σ	69
5.20. Gráfica del número de elipses reconocidas por ED + SD en los casos 1,2 y 3, para los diferentes valores de σ	69
5.21. Caso de prueba uno, las elipses son disjuntas con 20 por ciento de puntos atípicos.	70
5.22. Caso de prueba dos, las elipses están superpuestas, con 60 por ciento de puntos atípicos.	71
5.23. Caso de prueba tres, las elipses están anidadas, con 100 por ciento de puntos atípicos.	71
5.24. Resultados del primer caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.	72
5.25. Resultados del segundo caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.	73
5.26. Resultados del tercer caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.	73
5.27. Resultados para la heurística sin GA-RANSAC . Las líneas representan el promedio de elipses reconocidas por la heurística.	75
6.1. Planos para hacer la calibración	80
6.2. Cuboides para la auto-calibración	81

6.3. Dibujado de los cuboides a partir de los parámetros de la cámara . .	82
6.4. Patrón de círculos concéntricos	83
6.5. Patrón de círculos después de la umbralización y la obtención de sus bordes	83
6.6. Patrón de círculos concéntricos rojo	87
6.7. Segmentación de las figuras	88
6.8. Mapa de bits obtenido para las figuras	88
6.9. Elipses obtenidas con GA-RANSAC	89

Índice de tablas

4.1. Parámetros para el algoritmo genético	38
4.2. Parámetros extras para el algoritmo genético	44
5.1. Resultados del Ajuste algebraico con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	52
5.2. Resultados de Levenberg-Marquadt con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	52
5.3. Resultados de RANSAC + MMC con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	52
5.4. Resultados de PSO + SD con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	53
5.5. Resultados de ED + SD con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	53
5.6. Resultados de GA-RANSAC + F_1 con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	53
5.7. Resultados de GA-RANSAC + F_2 con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	54
5.8. Resultados de GA-RANSAC + F_3 con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	54
5.9. Resultados del Ajuste algebraico con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	57
5.10. Resultados de Levenberg-Marquadt con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	57

5.11. Resultados de RANSAC + MMC con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	58
5.12. Resultados de PSO + SD con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	58
5.13. Resultados de ED + SD con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	58
5.14. Resultados de GA-RANSAC + F_1 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	59
5.15. Resultados de GA-RANSAC + F_2 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	59
5.16. Resultados de GA-RANSAC + F_3 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.	59
5.17. Resultados del algoritmo ED + SD y GA-RANSAC + F_2 con 200 ejecuciones para el primer caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.	72
5.18. Resultados del algoritmo ED + SD y GA-RANSAC + F_2 con 200 ejecuciones para el segundo caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.	74
5.19. Resultados del algoritmo ED + SD y GA-RANSAC + F_2 con 200 ejecuciones para el tercer caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.	74

Índice de algoritmos

1.	Algoritmo genético	15
2.	Programación evolutiva	16
3.	Algoritmo básico de la ED	17
4.	Heurística RANSAC + MMC	31
5.	Heurística para ajustar una elipse	32
6.	Heurística ED + SD	33
7.	Heurística PSO + SD	35
8.	Algoritmo GA-RANSAC	39
9.	Operador local	43
10.	Algoritmo para extraer múltiples elipses	45
11.	Algoritmo para extraer múltiples elipses automáticamente	47
12.	Cálculo de la homografía	85
13.	Segmentación por color	89

Capítulo 1

Introducción

1.1. Descripción del problema

Ajuste de elipses.

El problema de ajuste de elipses consiste en lo siguiente:

Dado un conjunto de puntos en un sistema de coordenadas global $p_i \in \mathbb{R}^2 |_{1 \leq i \leq n}$, se debe encontrar la mejor elipse que representa ese conjunto de puntos.

Una elipse puede representarse por 5 parámetros que son:

- Semieje mayor a
- Semieje menor b
- Coordenadas del centro x_c y y_c
- Angulo de rotación α

En la figura 1.1 podemos ver un ejemplo de los parámetros que definen una elipse.

Ahora definiremos brevemente dos conceptos importantes, que serán explicados con más detalles la sección 2.2, en la pág. 8:

- **Punto atípico** es un punto que pertenece a otra distribución, en nuestro caso a otra u otras elipses.
- **Ruido** es un punto que no pertenece a ninguna distribución. En el caso de procesamiento de imágenes este ruido se genera en la misma captura de la imagen.

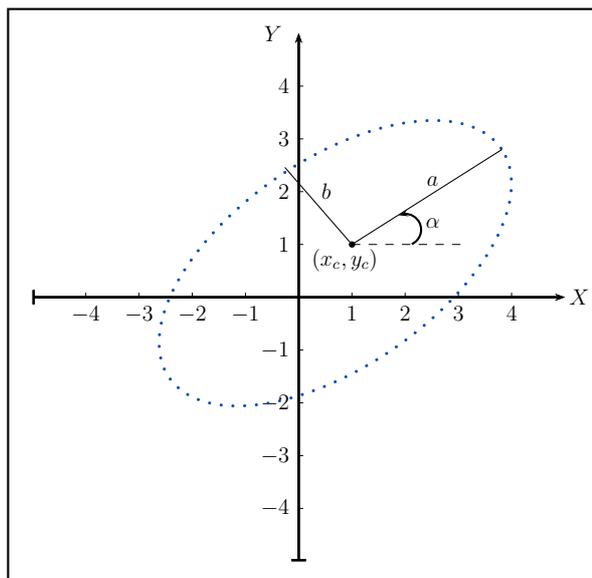


Figura 1.1: Parámetros de la elipse

Entonces, el ajuste de elipses se encarga de encontrar los parámetros que definen una elipse en un conjunto de datos dados, los cuales pueden contener ruido. En la práctica, difícilmente se da el caso en el que se obtengan solo las muestras de los puntos que pertenecen a una elipse, además existen puntos que no pertenecen a la elipse conocidos como puntos atípicos.

Debido a que en visión y procesamiento de imágenes, es común que se tenga un conjunto de datos con varias elipses, es necesario encontrar los parámetros de las distintas elipses en el conjunto de datos. Dada la complejidad de encontrar múltiples elipses en un conjunto de datos, es necesaria una heurística para resolver el problema, ya que los métodos convencionales usualmente no son capaces de resolverlo.

Entonces, el ajuste de múltiples elipses consiste en encontrar los parámetros de cada una de las elipses dadas implícitamente en un conjunto de datos que contiene ruido. En la figura 1.2 podemos ver un ejemplo de un conjunto de datos con múltiples elipses.

1.2. Motivación

El problema de ajuste de elipses es concurrente en diversas áreas del conocimiento, principalmente en procesamiento de imágenes y visión computacional, aunque tiene diversas aplicaciones en otras áreas, como son Robótica[1], Astronomía, detección de rostros[2], Física [3], Biología, control de calidad, etc.

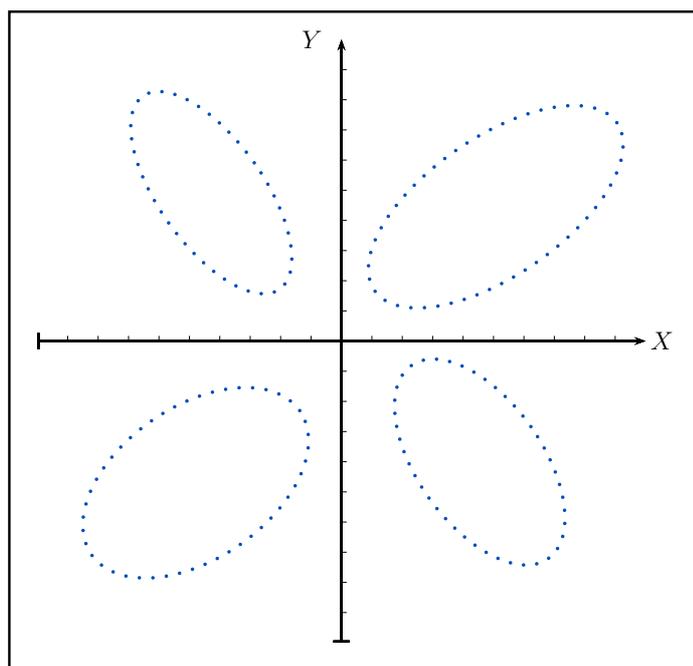


Figura 1.2: Puntos que representan múltiples elipses

Existe una gran cantidad de métodos para ajustar elipses, algunos son métodos analíticos, mientras otros son heurísticas, los algoritmos más sencillos son los que se basan en el ajuste algebraico, estos métodos no son iterativos, además de que son sencillos de implementar y con un costo computacional bajo, en general funcionan bien para ajustar círculos y elipses en ausencia de ruido, aunque este problema es en general no-lineal y muy propenso a tener ruido.

Otro método determinista es el ajuste por mínimos cuadrados, un método muy popular debido a que se puede resolver por métodos numéricos tales como el método de Gauss-Newton [4], búsqueda lineal *search line*-, descenso empinado *steepest descent*-, Levenberg Marquadt [5], etc. o incluso con un ajuste directo de mínimos cuadrados [6], todos estos métodos son más robustos que los métodos algebraicos, aunque con un costo computacional un poco mayor por ser iterativos (excepto el método directo), son menos sensibles al ruido, pero aun tienen el inconveniente de ser sensibles a los puntos atípicos.

Otra manera de resolver el problema de ajuste de elipses, es usando métodos con la transformada de Hough, que son métodos estadísticos con el inconveniente de dar problemas para figuras geométricas más complejas que líneas y círculos. Por último tenemos los métodos robustos, que pueden ser heurísticas o híbridos de alguna heurística con un método determinista. Existen varios métodos robustos, pero uno de los más populares es el de consenso de muestras aleatorias *Random Sample Consensus* (RANSAC) [7], así como métodos usando algoritmos evolutivos.

Existen diversos métodos para el problema de la extracción de los parámetros de múltiples figuras geométricas, en nuestro caso elipses. El ajuste de múltiples elipses tiene además variadas aplicaciones, como por ejemplo en [8], en donde se extraen varias elipses a partir de la silueta de una persona. En este caso los autores usaron el algoritmo esperanza minimización –*Expectation Minimization*– (EM) primero para acercarse, y después el método Levenberg Marquadt para encontrar los parámetros.

El procedimiento más popular para ajustar múltiples elipses es una combinación de RANSAC con algún ajuste por mínimos cuadrados [9]. Cabe mencionarse que existen otras heurísticas para obtener múltiples elipses como [10], donde se propone un método parecido a RANSAC. En [11] se propone otro algoritmo que utiliza una heurística para buscar los centros de las elipses y después utilizar ajuste algebraico para encontrar la elipse. En [12] se usa la transformada de Hough y un algoritmo genético para obtener múltiples elipses.

Los métodos para ajustar múltiples elipses tienen la finalidad de ser robustos, sin embargo presentan problemas para algunos casos, y solo son capaces de encontrar elipses con cierta cantidad de ruido y puntos atípicos, siendo necesaria una heurística más robusta que las mencionadas, con la finalidad de ajustar elipses en conjuntos de datos con mayor ruido.

1.3. Metodología

En la presente tesis se propone una solución al problema de ajuste de múltiples elipses en un conjunto de datos con condiciones altas de ruido. El método propuesto consiste en minimizar una función de aptitud robusta que resuelve el problema de ajustes de elipses usando un algoritmo genético (AG).

Se proponen tres funciones de aptitud para el ajuste de múltiples elipses, que son robustas a los puntos atípicos. Las funciones propuestas fueron minimizadas con un algoritmo genético.

1.4. Contribuciones

Las principales contribuciones fueron:

- Se propuso un algoritmo robusto capaz de resolver la extracción de múltiples elipses.

- Se implementaron diversos métodos para la solución del problema de ajuste de elipses.
- Se obtuvieron mejores resultados en algunos casos que con los métodos ya existentes.
- Se hizo la detección de múltiples elipses sin la necesidad de conocimiento específico sobre cada elipse en los datos de entrada.

1.5. Organización de la tesis

La tesis está organizada en 7 capítulos que se describen a continuación:

- Capítulo 1. Introducción. En este capítulo se da el panorama de lo que se realizó en la tesis, se presentan las primeras ideas sobre el ajuste de elipses y el alcance que tuvo la tesis.
- Capítulo 2. Marco teórico. En este capítulo se abordan formalmente los fundamentos y conceptos necesarios, como lo son el ajuste de curvas, la distancia, la optimización, métodos de ajuste de elipses conocidos y algoritmos evolutivos. Además se presentan trabajos previos que resuelven el problema de ajuste de múltiples elipses.
- Capítulo 3. Ajuste robusto de una elipse. En este capítulo se presentan algunos métodos que resuelven el problema de ajuste de elipses, cuando solo se busca una elipse como lo es el ajuste directo, el ajuste por mínimos cuadrados y el ajuste usando heurísticas, al final se presentan algunos resultados.
- Capítulo 4. Ajuste robusto de varias elipses. En este capítulo se explica de que se trata la extracción de múltiples elipses, las aplicaciones que tiene para diversos usos, así como el método para resolver dicho problema. Además se describe formalmente la solución propuesta para resolver el problema usando un algoritmo genético con distintas funciones de aptitud propuestas.
- Capítulo 5. Resultados. En este capítulo se muestran los experimentos realizados y los resultados obtenidos en el método propuesto. Se compara el método propuesto con el trabajo presentado en [9]. Al final se tiene la discusión sobre los resultados obtenidos.
- Capítulo 6. Aplicaciones. En este capítulo se presenta una aplicación de la extracción de múltiples elipses para la calibración de cámaras usando círculos concéntricos.

- Capítulo 7. Conclusiones. En este capítulo se presentan las conclusiones y el trabajo a futuro para esta tesis.

Capítulo 2

Marco teórico

En este capítulo se explican algunos conceptos básicos que son de interés para este trabajo de tesis. En primer lugar se define que es una elipse, debido a que el trabajo está enfocado en la extracción de elipses, es importante saber que es una elipse. Después se habla del ajuste de curvas, pues el ajuste de elipses es un caso particular de éste.

Luego se habla de la distancia, debido a que éstas son comúnmente usadas como función a minimizar en el ajuste de elipses. En la siguiente sección se define brevemente lo que es la optimización, debido a que el ajuste de elipses puede plantearse como un problema de optimización.

Después se habla de los algoritmos evolutivos, porque engloban varias técnicas para la solución de diversos problemas, entre los cuales se encuentran los algoritmos genéticos. Por último se mencionan algunos trabajos relacionados con el trabajo presentado en esta tesis.

2.1. Elipse

Una elipse es la curva cerrada que resulta de la intersección de un plano y un cono. Una elipse es también el lugar geométrico de todos los puntos en el plano cuyas distancias a dos puntos fijos suman lo mismo. Las elipses son de interés, debido a que al proyectar un círculo se forma una elipse, y los círculos son objetos muy comunes.

Elementos de una elipse

La elipse es una curva plana y cerrada, simétrica a dos ejes perpendiculares entre sí: el semieje mayor a y el semieje menor b , que son la mitad del eje mayor y el eje menor respectivamente.

Los focos de la elipse son dos puntos equidistantes del centro en el eje mayor, cuya suma de las distancias desde cualquier punto en la elipse a los focos es constante.

Ecuaciones de la elipse

Existen diversas formas de representar una elipse por medio de ecuaciones, a continuación se mencionan las más comunes.

La ecuación de la elipse en coordenadas cartesianas, con centro en el origen es:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2.1)$$

y con centro fuera del origen:

$$\frac{(x - x_c)^2}{a^2} + \frac{(y - y_c)^2}{b^2} = 1 \quad (2.2)$$

La ecuación de la elipse en coordenadas polares:

$$r(\theta) = \frac{b}{\sqrt{\frac{\cos(\theta)^2}{a^2} + \frac{\sen(\theta)^2}{b^2}}} \quad (2.3)$$

La ecuación paramétrica de la elipse con centro en (x_c, y_c) es:

$$\begin{cases} x = x_c + a \cos \alpha \\ y = y_c + b \sen \alpha \end{cases} \quad (2.4)$$

donde $\alpha \in [0, 2\pi)$.

La ecuación general de las secciones cónicas es:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (2.5)$$

y describe una elipse cuando $B^2 < 4AC$.

2.2. Ajuste de curvas

El ajuste de curvas es el proceso por el cual se reconstruye una curva, o función matemática, esta curva es la que se ajusta mejor a un conjunto de puntos dados, los cuales posiblemente tengan una restricción. El ajuste de curvas puede considerarse como una interpolación, si se busca un ajuste exacto en los datos, o solo como una función suave que se ajusta a los puntos.

Un área relacionada es el análisis de regresión, que se enfoca más en temas de inferencia estadística tales como ajustar una curva presente en un conjunto de datos con errores aleatorios. El ajuste de curvas puede ser utilizado como una ayuda para la visualización de datos, inferir valores de una función donde no hay información disponible sobre los datos, para deducir las relaciones entre dos variables, etc.

El ajuste de curvas es un tema importante en procesamientos de imágenes y visión computacional, sin embargo también es utilizado fuera de estos ámbitos.

A continuación se definirán dos conceptos que son de importancia, el ruido y los puntos atípicos, debido a que al tomar muestras de algún experimento, es común que dichas muestras contengan ruido y puntos atípicos. Por lo tanto para llevar a cabo el ajuste de un conjunto de datos, hay que tener en cuenta que éstos pueden estar alterados por diversas razones.

Puntos atípicos

En estadística, un punto atípico [13] es una observación que está numéricamente distante del resto de datos de un conjunto dado. En la figura 2.1 se puede ver un ejemplo de puntos atípicos, en la cual se pueden observar siete puntos muy distantes de la distribución que forma la elipse.

En la práctica el conjunto de datos con el que se está trabajando puede tener una gran cantidad de ruido, y al existir elementos muy alejados del modelo original, éstos repercuten sobre el ajuste que se esté haciendo, los datos que se encuentran alejados del modelo original se conocen como puntos atípicos.

Los puntos atípicos son consecuencia de errores de almacenamiento, errores de captura, errores de redondeo, errores de comunicación, entre otros. Estos puntos son comunes en datos obtenidos de problemas reales, y en sistemas automáticos es común que estos puntos pasen desapercibidos.

Ruido estadístico

En computación, ruido son los datos sin significado, que se producen simplemente como un subproducto no deseado de otras actividades. En particular, en procesamiento de imágenes este ruido se genera al capturar la imagen. En la figura 2.2 se puede ver un ejemplo de ruido estadístico, en la figura hay puntos que asemejan una elipse, pero no se ven uniformes sino distorsionados, a dicha distorsión se le conoce como ruido.

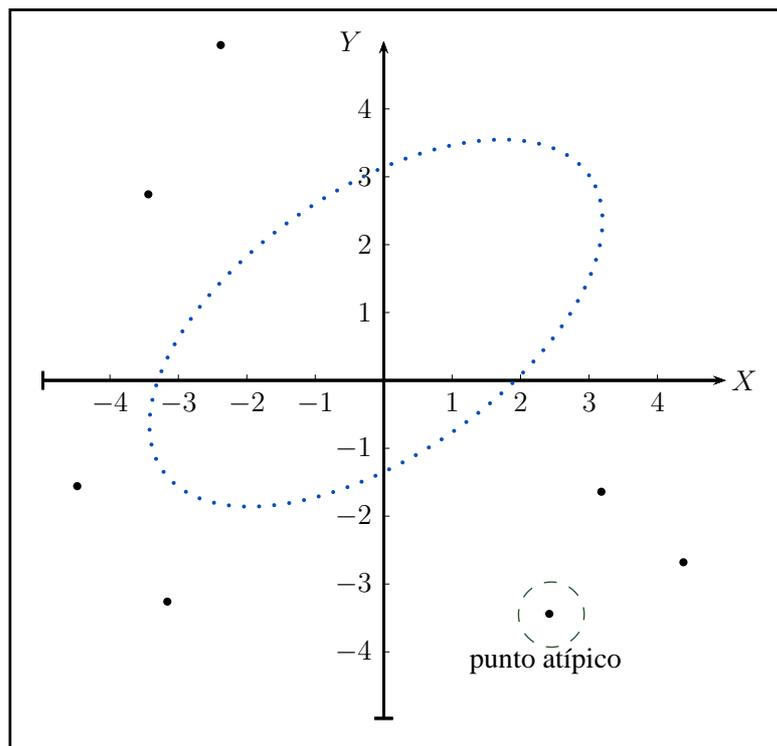


Figura 2.1: Puntos atípicos

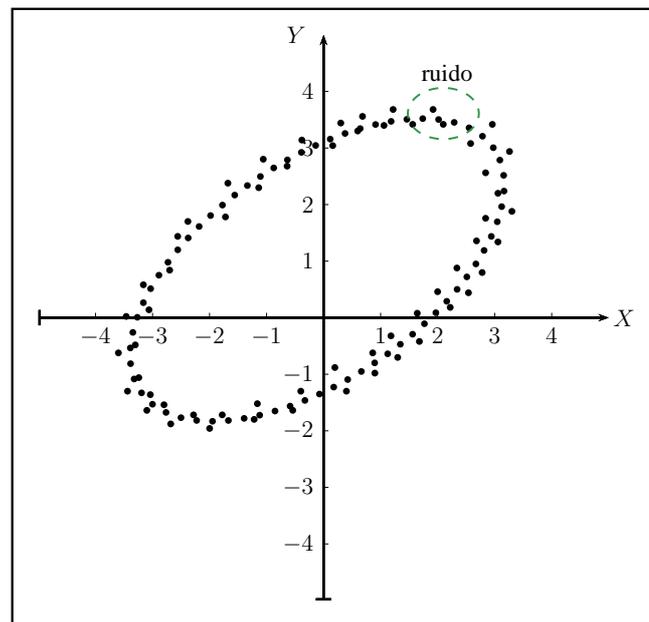


Figura 2.2: Ruido estadístico

Mínimos cuadrados

El método de mínimos cuadrados es un enfoque común para aproximar soluciones en un sistema sobre determinado, esto es, un sistema de ecuaciones donde hay más

ecuaciones que incógnitas. Los mínimos cuadrados significan que se reducirá al mínimo la suma de los cuadrados de los errores cometidos al resolver cada ecuación.

Los mínimos cuadrados fueron descritos por primera vez por Gauss en 1794 [14]. La aplicación más importante de los mínimos cuadrados, es en el ajuste de datos. El mejor ajuste usando mínimos cuadrados, minimiza la suma de los errores residuales al cuadrado. Un error residual es la diferencia entre el valor dado y el valor obtenido mediante un modelo matemático.

Los problemas de mínimos cuadrados pueden ser de dos tipos: lineales y no lineales, dependiendo si los residuales son lineales en todas las incógnitas. Los mínimos cuadrados lineales son un problema que ocurre en análisis de regresión; éstos tienen una solución exacta (involucra la inversión de una matriz). Los métodos no lineales no tienen una solución exacta y son resueltos por métodos iterativos con aproximaciones sucesivas.

Mínimos cuadrados no lineales

Los mínimos cuadrados no lineales es la forma de mínimos cuadrados que se usa para ajustar datos cuando se tiene un modelo no lineal, esto es, se tiene un sistema con m ecuaciones, n incógnitas y $m > n$. Ésta es usada en análisis de regresión no lineal.

En [15] se define el problema de mínimos cuadrados no lineal como sigue:

Definición 1 *Problemas por mínimos cuadrados:*
Encontrar x^* , un local mínimo para:

$$F(x) = \frac{1}{2} \sum_{i=1}^m (f_i(x))^2 \quad (2.6)$$

donde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ son las funciones dadas y $m \geq n$

Para resolver estos problemas se utilizan diversos métodos iterativos, como el descenso empinado, el algoritmo de Gauss-Newton, entre otros. El método más popular para ajustar los mínimos cuadrados no lineales, es el algoritmo de Levenberg-Marquadt.

Mínimos por valores absolutos

Debido a que pueden existir puntos atípicos, dichos puntos pueden alterar en gran medida la solución del problema al usar el cuadrado de la diferencia, entonces un

método más robusto es el uso de la suma de los valores absolutos, que es menos sensible al ruido. Por lo tanto la función a minimizar será ahora:

$$F(x) = \sum_{i=1}^m |f_i(x)| \quad (2.7)$$

El problema con este tipo de función es que no es derivable en el mínimo, lo que no la hace viable para usarse con los métodos para mínimos cuadrados, sino con métodos de optimización no lineal, como los mencionados para los mínimos cuadrado no lineales. Aunque los métodos iterativos no nos garantizaran que lleguemos al óptimo global y el problema puede quedar atrapado en un óptimo local.

2.3. Distancia

Se define a la distancia o métrica como:

Definición 2 Una función binaria $d(a, b)$ en $X \times X \in \mathbb{R}$ que cumple con las siguientes condiciones:

1. No negatividad: $d(a, b) \geq 0 \forall a, b \in X$
2. Simetría: $d(a, b) = d(b, a) \forall a, b \in X$
3. Desigualdad del triángulo: $d(a, b) \leq d(a, c) + d(c, b) \forall a, b, c \in X$
4. $\forall x \in X : d(x, x) = 0$
5. Si $x, y \in X$ cumplen que $d(x, y) = 0 \Rightarrow x = y$

Distancia Euclidiana

La distancia o métrica euclidiana es la distancia **ordinaria** entre dos puntos y está dada por una fórmula Pitagórica.

La distancia euclidiana entre dos puntos x e y es el largo del segmento de línea que los conecta. En coordenadas cartesianas, si $x = (x_1, x_2)$ e $y = (y_1, y_2)$ son dos puntos en el espacio euclidiano de dos dimensiones, entonces la distancia de x a y , o de y a x está dada por:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.8)$$

Distancia de Hausdorff

Se define la distancia de Hausdorff como:

Definición 3 Sean $u, v \in \mathbb{R}^n$ y $A, B \subset \mathbb{R}^n$. La distancia normal máxima d_∞ , las semi-distancias $dist(.,.)$, y la distancia de Hausdorff $d_H(.,.)$ definidas como sigue:

1. $d_\infty(u, v) := \max_{i=1, \dots, n} |u_i - v_i|$
2. $dist(u, A) := \inf_{v \in A} d_\infty(u, v)$
3. $dist(B, A) := \sup_{u \in B} dist(u, A)$
4. $d_H(A, B) := \max\{dist(A, B), dist(B, A)\}$

Cabe mencionarse que en conjuntos discretos (con los que se trabaja en computación) el ínfimo y el supremo son equivalentes al mínimo y máximo respectivamente.

2.4. Optimización

La optimización es el área de las matemáticas que intenta dar respuesta a los problemas donde se desea elegir el mejor entre un conjunto de elementos. En su forma más simple, el problema puede definirse como:

$$\min(\max) f(x) \tag{2.9}$$

donde $x = \{x_1, \dots, x_m\} \in X \in \mathbb{R}^m$ es un vector con las variables de decisión y $f(x)$ es la función objetivo, la cual determina la calidad de las posibles soluciones.

Entonces el problema de optimización se puede ver como encontrar el mínimo (o máximo) valor que tomará la función, con las variables de decisión en el espacio de soluciones factibles X .

Los problemas de optimización pueden tener restricciones, dichas restricciones se pueden definir con un sistema de igualdades o desigualdades, de la siguiente forma:

$$g(x) \leq 0$$

$$h(x) = 0$$

2.5. Algoritmos evolutivos

Los algoritmos evolutivos son técnicas inspiradas en la biología, dichas heurísticas simulan el comportamiento evolutivo de las especies desde distintos puntos de vista, y se usan para solucionar diversos problemas, en general cumplen con los siguientes requerimientos:

1. Función de aptitud: Ésta es la función objetivo.
2. Codificar las estructuras: Para poder trabajar con las variables de decisión es necesario trabajar con una representación en particular de los individuos.
3. Operaciones que afecten a los individuos: Para poder modificar a los individuos en cada generación es necesario el uso de operadores.
4. Mecanismo de selección: Se necesita de una manera específica para seleccionar a los mejores individuos para asegurar que la solución siempre mejorará.

Existen diversos tipos de algoritmos evolutivos, aunque guardan muchas semejanzas entre ellos que los hacen muy similares.

Existen tres paradigmas de algoritmos evolutivos principales:

- Programación evolutiva
- Estrategias evolutivas
- Algoritmos genéticos

Para esta tesis solo es de interés la última por ser con la que se trabajará. A continuación se hará una breve mención de los algoritmos genéticos, las estrategias evolutivas, la programación evolutiva, la evolución diferencial (ED) y la optimización por acumulación de partículas (PSO).

Algoritmos genéticos

Los algoritmos genéticos fueron desarrollados por Holland a principios de 1960 para resolver problemas de aprendizaje de máquina.

El funcionamiento de los algoritmos genéticos se describe en el algoritmo 1.

En los algoritmos genéticos es usual que se use representación binaria. Las partes que definen a un individuo en un algoritmo genético son los cromosomas y el fenotipo, los cromosomas son la representación de los individuos con la cual se trabaja dentro del algoritmo, mientras que el fenotipo es la representación de los individuos en el

Algoritmo 1 Algoritmo genético

Generar una población inicial de individuos de manera aleatoria.

Calcular la función de aptitud de cada individuo en la población

repeat

 Seleccionar individuos en base al valor de aptitud

 Aplicar operador de cruza

 Aplicar operador de mutación

 Calcular la función de aptitud de cada individuo en la población

 Conservar el mejor individuo para la siguiente generación

until La condición de paro se cumpla

return Mejor individuo

contexto del problema a resolver. La aptitud por otro lado es el valor que tiene cada individuo y con el cual se sabe qué tan bueno es.

Cada que iteramos el algoritmo obtenemos una nueva generación, dicha generación es el conjunto de individuos que van cambiando en cada iteración debido a los operadores.

El operador de cruza se encarga de tomar dos individuos de la generación actual y combinarlos para generar un nuevo individuo, mientras que el operador de mutación se encarga de modificar el individuo para crear uno nuevo. Cabe resaltar que los operadores se aplican dependiendo de una probabilidad, al igual que se necesita de una selección para poder conservar al mejor individuo.

Aplicaciones

Los algoritmos genéticos tienen aplicaciones en diversas áreas:

- Gestores de bases de datos
- Optimización
- Aprendizaje de máquina
- Visión
- Generación de gramáticas para lenguajes formales
- Trayectorias de robots
- Entre otras.

Estrategias evolutivas

Estas estrategias simulan la evolución biológica a nivel de individuo. En las estrategias evolutivas existen los operadores de cruce, sin embargo el operador de mutación es el que tiene mayor peso dentro de dichos algoritmos. Una característica peculiar es que solo los mejores individuos pasan a las siguientes generaciones. Las estrategias evolutivas usualmente usan representación real.

Aplicaciones

Las estrategias evolutivas tienen aplicaciones en diversas áreas:

- Diseño en ingeniería
- Magnetismo
- Bioquímica
- Óptica
- Entre otras.

Programación evolutiva

La programación evolutiva pone énfasis en la relación entre padres e hijos, en lugar de emular operadores genéticos (como los algoritmos genéticos). El algoritmo básico de la programación evolutiva se describe en el algoritmo 2.

Algoritmo 2 Programación evolutiva

Generar una población inicial de manera aleatoria.

Aplicar mutación

Se calcula la aptitud de cada hijo y se usa un proceso de selección mediante torneo para decidir cuáles serán las soluciones que se conservarán.

La programación evolutiva abstrae la evolución al nivel de las especies, y no requiere de un operador de recombinación. De la misma manera, utiliza selección probabilística.

ED

La ED es una rama de la computación evolutiva, cuyo algoritmo básico se describe en el algoritmo 3.

En la ED, las variables se representan mediante números reales. Para generar la población inicial debe estar dentro de límites requeridos, y en cada iteración se

Algoritmo 3 Algoritmo básico de la ED

Generar una población inicial.
Aplicar mutación
Aplicar recombinación
Aplicar selección.

seleccionan tres individuos como padres, y se aplican los operadores con los cuales dependiendo de una probabilidad se genera un nuevo individuo, que solo sobrevivirá si es mejor que un individuo seleccionado de manera aleatoria.

La ED difiere de los algoritmos genéticos en que:

- Usa representación real en vez de binaria
- Se usan 3 padres en vez de 2
- Se genera un solo hijo de la cruce y éste pasa solo con la perturbación de uno de los padres
- El nuevo individuo reemplaza a otro individuo elegido aleatoriamente solo si es mejor.

PSO

PSO es una heurística de búsqueda (que puede considerarse algoritmo evolutivo) que simula los movimientos de una bandada de aves que aspiran encontrar comida. La relativa simplicidad de PSO y el hecho de que es una técnica basada en población, la hacen muy popular. Al igual que ED utiliza una representación real.

Originalmente fue adaptado para el equilibrio de pesos en redes neuronales, y pronto llegó a ser un optimizador global muy popular, principalmente en problemas en que las variables de decisión eran números reales.

Operadores locales

Como se ha mencionado en los algoritmos evolutivos existen diversos operadores que hacen cambiar a los individuos de generación en generación, con la intención de mejorar la búsqueda. Los operadores locales son métodos que ayudan dependiendo del problema, a buscar mejores soluciones.

La idea principal del operador local es hacer una búsqueda con algún algoritmo evolutivo, y dada una probabilidad se aplica un operador local según algún criterio dependiente del contexto del problema a resolver, para buscar soluciones locales.

2.6. Estado del arte

Hay muchos trabajos relacionados con esta tesis, pero se mencionarán solo algunos. Para empezar, los trabajos más importantes para esta tesis. En [16] los autores presentan un método para ajustar elipses, que es un método similar al presentado en esta tesis, con la diferencia de que solo sirve para ajustar círculos. El siguiente trabajo es [9] donde los autores presentan un método de RANSAC con la mediana de los mínimos cuadrados como función de aptitud, y otro método usando ED con la suma de distancias euclidianas como función de aptitud. Además está el popular método de Fitzgibbon [6] para ajuste de elipses, que será usado en este trabajo.

En las siguientes secciones se describen algunos métodos existentes sobre el ajuste de elipses.

2.6.1. Detección de círculos en imágenes usando algoritmos genéticos

En [16] los autores presentan un método para extraer círculos de una imagen usando algoritmos genéticos. La idea fundamental es como sigue: A partir de un conjunto de puntos dado $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$, se eligen tres puntos aleatoriamente $\{p_i, p_j, p_k\} \mid p_i = (x_i, y_i) \quad p_j = (x_j, y_j) \quad p_k = (x_k, y_k)$, y se usa una ecuación para aproximar un círculo que cruce por estos tres puntos, la ecuación es la siguiente:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (2.10)$$

donde

$$x_0 = \frac{\begin{vmatrix} x_j^2 + y_j^2 - (x_i^2 + y_i^2) & 2(y_j - y_i) \\ x_k^2 + y_k^2 - (x_i^2 + y_i^2) & 2(y_k - y_i) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \quad (2.11)$$

$$y_0 = \frac{\begin{vmatrix} 2(x_j - x_i) & x_j^2 + y_j^2 - (x_i^2 + y_i^2) \\ 2(x_k - x_i) & x_k^2 + y_k^2 - (x_i^2 + y_i^2) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \quad (2.12)$$

y

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \quad (2.13)$$

después se revisa el número de puntos en la imagen que están en la circunferencia del círculo.

Un algoritmo genético mejora la búsqueda del mejor círculo que se ajusta a estos puntos dependiendo de una función de aptitud, que cuenta los puntos en la imagen que están en la circunferencia encontrada con las ecuaciones anteriores. Este algoritmo usa como cromosomas los índices de los puntos y el método de selección de ruleta.

Además usa un operador de cruza de 1-punto y un operador de mutación por bit.

2.6.2. Ajuste directo de mínimos cuadrados

En [6] los autores presentan un método para ajustar elipses. Este método es una referencia popular en el problema de ajuste de elipses, porque es muy eficiente, aunque tiene el problema de no ser robusto en conjuntos de datos que contengan puntos atípicos.

2.6.3. Transformada de Hough

Existe una gran cantidad de trabajos relacionados con la transformada de Hough debido a su popularidad para resolver ajuste de datos. Todos ellos son robustos en conjuntos de datos con ruido, pero no son buenos para problemas de varias dimensiones, porque tienen un alto costo computacional para un desempeño robusto.

Varios investigadores han propuesto modificaciones a la transformada de Hough original para resolver el problema de ajuste de elipses, como la transformada de Hough aleatoria [17], transformada de Hough probabilística [18], transformada de Hough restringida [19], transformada de Hough difusa [20], transformada de Hough generalizada [21], entre otros.

2.6.4. Extracción de elipses robusto y rápido usando un algoritmo genético con múltiple población

Este método, publicado en [12], propone un algoritmo para obtener múltiples elipses a partir de imágenes, no sus parámetros. Pero usando las elipses obtenidas, se puede meter en un algoritmo que obtenga múltiples elipses y obtener los parámetros de todas ellas, mostrando así una aplicación en la que se puede usar el ajuste de múltiples elipses.

Para resolver el problema usa un algoritmo genético con múltiples poblaciones, que es fácilmente paralelizable, y que permite hacer más rápido el procedimiento.

2.6.5. Detección de elipses por EM

En [8], se reporta una aplicación del ajuste de múltiples elipses, proponiéndose como solución un algoritmo EM sin restricciones, que busca elipses en una imagen y

selecciona un subconjunto de puntos, que se ajusta utilizando el algoritmo de Levenberg Marquadt.

2.6.6. Heurísticas tipo RANSAC

En [10] y en [11], se proponen dos heurísticas parecidas a RANSAC que sirven para la detección de elipses. RANSAC funciona básicamente tomando muestras aleatorias, que después se ajustan utilizando alguna técnica tradicional, como ajuste algebraico, y se continúa el proceso eliminando puntos para que el ajuste se tome sólo una de una parte.

En [11], el algoritmo propuesto busca los centros de las elipses y después los ajusta con alguna técnica determinista.

2.6.7. Obtención de múltiples elipses mediante Evolución Diferencial

En [9] se describe un método para encontrar múltiples elipses, el cual utiliza como función a minimizar las distancias euclidianas, las cuales son no lineales y hacen difícil su tratamiento por métodos deterministas, por lo cual se usa un algoritmo evolutivo, en este caso Evolución Diferencial.

Éste minimiza las sumas de distancias euclidianas descritas en [22], y usa como restricción el número de puntos utilizados. Además propone una heurística para encontrar múltiples elipses.

La función de aptitud que utiliza es:

$$g_1(x) = \sum d_i(p'_i, p''_i) \quad (2.14)$$

Donde d_i es la distancia euclidiana entre el punto p'_i de los datos y el punto p''_i en la elipse.

Dos modificaciones se agregan para mejorar el ajuste: sólo las distancias menores que un umbral T serán tomadas en cuenta, y al mismo tiempo, el número de puntos usados para el ajuste será maximizado.

Entonces, el problema es minimizar la función:

$$g_2(x) = \frac{2T(n-m)}{n} + \frac{1}{m} \sum d_i(p'_i, p''_i) | d_i < T \quad (2.15)$$

El problema con este método es que puede fallar para un número grande de elipses y que necesita que se sepa información de las elipses para restringir el espacio de búsqueda, lo cual para métodos automáticos, es un problema.

Capítulo 3

Ajuste robusto de una elipse

Tal como se describió en la introducción, el ajuste de elipses consiste en encontrar los parámetros que definen una elipse, el número mínimo de parámetros necesarios para representar una elipse en un sistema coordenado de dos dimensiones es 5, que pueden ser el semi eje mayor, semi eje menor, las coordenadas del centro de la elipse y el ángulo del semi eje mayor con respecto al eje de las abscisas. Otros parámetros que también sirven para describir una elipse son los 5 coeficientes de la ecuación general de las secciones cónicas (2.5).

En diversas áreas del conocimiento es común trabajar con elipses, en visión particularmente se trabaja con elipses debido a que la proyección de un círculo es una elipse, un ejemplo se muestra en la figura 3.1, y un círculo se encuentra comúnmente en los objetos hechos por el hombre. Al momento de extraer elipses se da el caso en el que las muestras o puntos extraídos contienen ruido y puntos atípicos, lo que puede complicar el ajuste de la elipse dependiendo del nivel de ruido y/o puntos atípicos.

En la figura 3.1 podemos ver un ejemplo de los puntos de dos elipses extraídas de una imagen, la figura 3.1(a) es una imagen tomada de un patrón de círculos concéntricos y la figura 3.1(b) son los puntos extraídos de la imagen, los cuales contienen puntos atípicos.

Existen dos maneras de resolver el ajuste de elipses, por métodos analíticos o por métodos probabilísticos. Los métodos analíticos o deterministas, se caracterizan por ser completamente predecibles, en los cuales no está involucrado el azar, por lo cual una misma entrada producirá la misma salida. Los métodos probabilistas a diferencia de los deterministas obtiene el resultado tomando algunas decisiones al azar, por lo cual a partir de una misma entrada se pueden obtener diferentes resultados.

Los métodos deterministas incluyen el popular método de la transformada de Hough [23], ajuste euclidiano [24], ajuste algebraico [6], entre otros. Los métodos probabilistas incluyen la técnica RANSAC [25], algoritmos evolutivos [9], recocido simulado [26], entre otros.

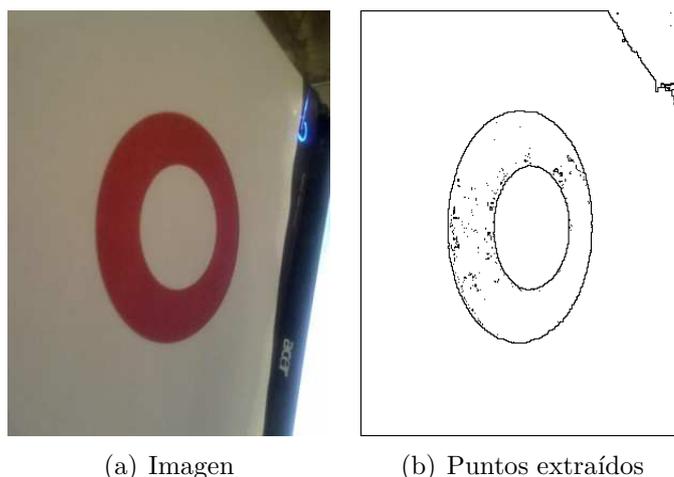


Figura 3.1: Puntos extraídos de una imagen de un patrón de dos círculos concéntricos. Claramente se nota que los dos círculos se proyectan como dos elipses.

Los métodos deterministas tienen la desventaja de ser sensibles a los puntos atípicos, por lo que la precisión de sus resultados se ve disminuida si hay una cantidad considerable de puntos atípicos. No obstante tienen la ventaja de ser eficientes en términos de consumo de tiempo computacional, sobre todo en un conjunto de datos con ausencia de puntos atípicos.

Los métodos probabilistas tienen la ventaja de ser robustos a los puntos atípicos, sin embargo, son usualmente lentos en términos de tiempo, y hay aplicaciones que requieren de una ejecución rápida.

El enfoque del trabajo en esta tesis, es probabilista. Sin embargo es importante mencionar a los métodos analíticos, ya que una parte de este trabajo hace uso de estos métodos.

En la sección 3.1 se mencionarán algunos métodos deterministas, mientras que en la sección 3.2 se mencionarán algunos métodos probabilistas populares para el ajuste de elipses.

3.1. Ajuste por métodos analíticos

Como se mencionó existen diversos métodos deterministas para resolver problema de ajuste de elipses, los más comunes son los métodos basados en ajuste por mínimos cuadrados, que se centran en encontrar el conjunto de parámetros que minimiza alguna métrica entre el conjunto de datos y la elipse.

Dentro de las métricas que se manejan para hacer el ajuste por mínimos cuadrados, se tienen la distancia euclidiana, la distancia ortogonal y la distancia algebraica principalmente.

A continuación se describirán muy brevemente tres de los métodos más populares y sencillos para resolver el ajuste de elipses, que son el ajuste algebraico, el ajuste por mínimos cuadrados usando el método de Gauss-Newton y finalmente el ajuste por mínimos cuadrados usando el método de Levenberg-Marquadt.

3.1.1. Ajuste algebraico

El ajuste algebraico o directo, es probablemente el método más simple para resolver el problema de ajuste de elipses, este método está basado en la minimización de la suma de distancias algebraicas del conjunto de datos dado a la elipse y fue propuesto en [6] y mejorado en [27].

Como se mencionó en el capítulo 2, en la pág. 8, una elipse puede ser representada implícitamente por sus coeficientes en la ecuación general de segundo grado de las cónicas (2.5), que podemos expresar en forma vectorial de la siguiente manera:

$$F(A, V) = A \cdot V = a_0x^2 + b_0xy + c_0y^2 + d_0x + e_0y + f_0 = 0, \quad (3.1)$$

donde $A = [a_0, b_0, c_0, d_0, e_0, f_0]^T$ y $V = [x^2, xy, y^2, x, y, 1]$.

$F(A; V_i)$ es llamada la distancia algebraica de un punto $p_i = (x, y)$ a la elipse $F(A; V) = 0$, ésto quiere decir que para calcular la distancia algebraica de un punto a una elipse, solo hay que sustituir el punto en la ecuación que define a la elipse.

El ajuste algebraico consiste en minimizar la suma de las distancias algebraicas al cuadrado

$$g(A) = \sum_{i=1}^N (A \cdot V_i)^2 \quad (3.2)$$

de la elipse a los N puntos V_i . La ecuación general de las secciones cónicas, describe una elipse cuando se cumple la condición $b^2 - 4ac < 0$. El problema puede resolverse si se fija la restricción, por ejemplo a $b^2 - 4ac = -1$, que es igual a $4ac - b^2 = 1$. Esta condición se puede representar en forma matricial de la siguiente manera

$$A^T \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} A = 1 \quad (3.3)$$

y se puede utilizar $A^T C A = 1$ como restricción en el problema a minimizar, donde C es la matriz de restricción.

En [28] se mostró que la minimización (3.2) puede ser resuelto considerando un sistema generalizado de eigenvalores

$$D^T D A = \lambda C A, \quad (3.4)$$

donde $D = [V_1, V_2, \dots, V_n]^T$ es llamada la matriz de diseño.

Entonces el problema de ajuste de elipses se reduce a

$$\begin{aligned} F &= \|DA\|^2 \\ \text{sujeto a } &A^T C A = 1 \end{aligned} \quad (3.5)$$

Usando multiplicadores de Lagrange y diferenciando, se obtiene un sistema de ecuaciones simultaneas

$$\begin{aligned} 2D^T D A - 2\lambda C A &= 0 \\ \text{sujeto a } &A^T C A = 1 \end{aligned} \quad (3.6)$$

Que se puede reescribir como

$$\begin{aligned} S A &= \lambda C A \\ \text{sujeto a } &A^T C A = 1 \end{aligned} \quad (3.7)$$

La ecuación (3.7) es el problema generalizado de eigenvalores, pero si se reordena las matriz S en sub-matrices de tamaño 3×3 como en [27]

$$\begin{aligned} \begin{pmatrix} S_1 & S_2 \\ S_2^T & S_4 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} &= \lambda \begin{pmatrix} C_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \\ \text{sujeto a } &A^T C A = 1 \end{aligned} \quad (3.8)$$

donde

$$C_1 = \begin{pmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{pmatrix} \quad (3.9)$$

Desarrollando la ecuación (3.8) se obtiene

$$\begin{aligned} S_1 A_1 + S_2 A_2 &= \lambda C_1 A_1 \\ S_2^T A_1 + S_3 A_2 &= 0 \\ A_2 &= -S_3^{-1} S_2^T A_1 \\ (S_1 - S_2 S_3^{-1} S_2^T) A_1 &= \lambda C_1 A_1 \end{aligned} \quad (3.10)$$

que sigue siendo el problema generalizado de eigenvalores y eigenvectores. Si se desarrolla y sustituye con nuevas variables queda como

$$\begin{aligned} C_1^{-1} (S_1 - S_2 S_3^{-1} S_2^T) A_1 &= \lambda A_1 \\ M A_1 &= \lambda A_1 \\ \text{sujeto a } A_1^T C_1 A_1 &= 1 \end{aligned} \quad (3.11)$$

donde $M = C_1^{-1} (S_1 - S_2 S_3^{-1} S_2^T)$ es una matriz dispersa reducida de tamaño 3×3 , con lo que el sistema se puede resolver como se detalla en [28] y es equivalente a encontrar las raíces de una ecuación cúbica.

Una vez encontrada λ , se calcula A , y se transforman los parámetros implícitos en parámetros explícitos; esto es, se transforman los coeficientes $A = [a_0, b_0, c_0, d_0, e_0, f_0]^T$ en los parámetros (a, b, x_c, y_c, α) . Sin perder generalidad el vector A puede ser expresado como $A = [a', 2b', c', 2d', 2f', g']$, y los parámetros explícitos se calculan usando las siguientes ecuaciones

$$a = \sqrt{\frac{2(a'f'^2 + c'd'^2 + g'b'^2 - 2b'd'f' - a'c'g')}{(b'^2 - a'c')[\sqrt{(a' - c')^2 + 4b'^2} - (a' + c')]}} \quad (3.12)$$

$$b = \sqrt{\frac{2(a'f'^2 + c'd'^2 + g'b'^2 - 2b'd'f' - a'c'g')}{(b'^2 - a'c')[-\sqrt{(a' - c')^2 + 4b'^2} - (a' + c')]}} \quad (3.13)$$

$$x_c = \frac{c'd' - b'f'}{b'^2 - a'c'} \quad (3.14)$$

$$y_c = \frac{a'f' - b'd'}{b'^2 - a'c'} \quad (3.15)$$

$$\alpha = \begin{cases} 0 & \text{para } b' = 0 \text{ y } a' < c' \\ \frac{\pi}{2} & \text{para } b' = 0 \text{ y } a' > c' \\ \frac{\arctan(\frac{2b'}{a'-c'})}{2} & \text{para } b' \neq 0 \text{ y } a' < c' \\ \frac{\pi}{2} + \frac{\arctan(\frac{2b'}{a'-c'})}{2} & \text{para } b' \neq 0 \text{ y } a' > c' \end{cases} \quad (3.16)$$

Este método es muy eficiente en cuanto a consumo de tiempo de ejecución se refiere, aunque tiene el inconveniente de no ser lo suficiente robusto en datos con muchos puntos atípicos.

3.1.2. Regresión no lineal

Gauss-Newton

El ajuste por mínimos cuadrados para ajustar elipses con el método de Gauss-Newton utiliza como métrica o función a minimizar las distancias ortogonales a diferencia del ajuste algebraico, donde se usaba la distancia algebraica. Un ejemplo del método de Gauss Newton para resolver el problema de ajuste de elipses es presentado en [4] donde además habla de otras secciones cónicas, que no son de interés para este trabajo.

El método de Gauss-Newton se utiliza para resolver problemas no lineales de mínimos cuadrados cuya función sea derivable. Es una modificación del método de optimización de Newton que no usa segundas derivadas.

El problema general que resuelve es:

dadas m funciones f_1, \dots, f_m de n parámetros p_1, \dots, p_n con $m \geq n$, queremos minimizar la suma

$$F(P) = \sum_{i=1}^m (f_i(P))^2 \quad (3.17)$$

donde, P se refiere al vector (p_1, \dots, p_n) .

El método de Gauss-Newton está basado en una aproximación lineal a los componentes de la función F en el vecindario de P , Gauss-Newton usa la primera derivada de los componentes de la función a minimizar. Gauss-Newton es un método iterativo, en casos especiales este método entrega una convergencia cuadrática.

Levenberg-Marquadt

El algoritmo de Levenberg-Marquardt proporciona una solución numérica al problema de minimizar una función, generalmente no lineal, sobre un espacio de parámetros de la función. Estos problemas surgen especialmente en el ajuste de curvas por mínimos cuadrados y programación no lineal.

El algoritmo intercala entre el algoritmo de Gauss-Newton y el método de descenso de gradiente. Éste es más robusto que Gauss Newton, lo que significa que en muchos casos se encuentra una solución, incluso si se inicia muy lejos del mínimo. Por otra parte, para funciones con buen comportamiento y parámetros iniciales razonables, el algoritmo de Levenberg Marquadt suele ser más lento que el de Gauss Newton.

Levenberg Marquadt es un algoritmo muy popular para ajuste de curvas utilizado en muchas aplicaciones de software para resolver problemas genéricos de ajuste de curvas. Es de interés para esta tesis porque también puede ser utilizado para resolver el problema de ajuste de elipses [8] de una manera determinista.

3.1.3. Comportamiento de los métodos analíticos con una elipse

Como se mencionó, los métodos analíticos obtienen buenos resultados en el ajuste de una elipse, en la figura 3.2 se puede observar un ejemplo de la elipse encontrada por el método algebraico, de Gauss-Newton y de Levenberg-Marquadt. Sin embargo, los métodos analíticos tienen problemas con conjuntos de datos que contienen puntos atípicos, como se puede observar en la figura 3.3, donde se muestra la elipse encontrada en un ejemplo con la misma elipse de la figura 3.2, pero con tres puntos atípicos.

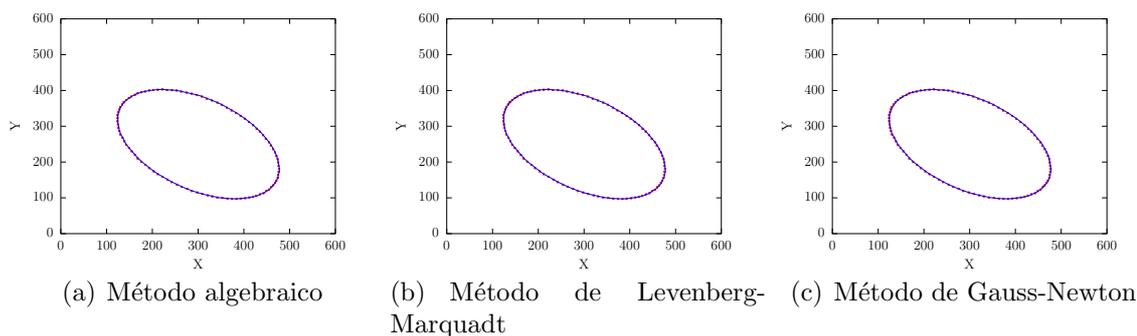


Figura 3.2: Elipses encontradas en un conjunto de puntos con una elipse sin puntos atípicos

En las figuras 3.2(a), 3.2(b) y 3.2(c), se puede ver que en ausencia de ruido y puntos atípicos, los métodos deterministas obtienen una elipse que se ajusta muy bien a

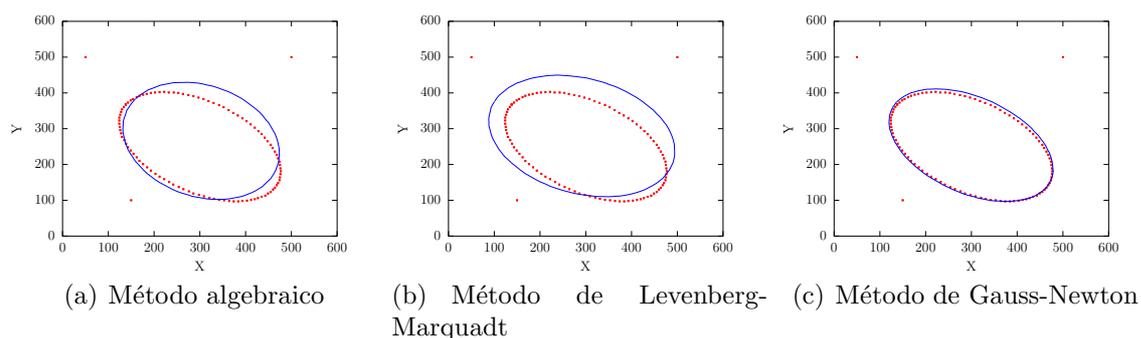


Figura 3.3: Elipses encontradas en un conjunto de puntos con una elipse y tres puntos atípicos

los puntos del conjunto de datos. En cambio, en las figuras 3.3(a), 3.3(b) y 3.3(c), se observa la elipse resultante con los mismos puntos de las elipses anteriores, pero con tres puntos atípicos.

Los resultados detallados se muestran en el capítulo 5, sección 5.1 (en la pág. 49).

3.2. Ajuste por métodos probabilistas

Como se dijo, además de los métodos deterministas, existen los métodos probabilistas, que son métodos que usan alguna heurística con toma de decisiones arbitrarias en favor de mejorar la búsqueda de una solución para el problema de ajuste de elipses. Los métodos deterministas son muy sensibles a puntos atípicos, por lo que surge la necesidad de algoritmos robustos para resolver problemas donde los datos que se tienen están inmersos en puntos atípicos.

En las siguientes secciones se mencionarán algunos métodos probabilistas populares para resolver el ajuste de elipses.

3.2.1. RANSAC + MMC

RANSAC es un método probabilista para estimar parámetros de un modelo matemático a partir de un conjunto de datos que contiene puntos atípicos. RANSAC fue publicado por primera vez en [25].

En [9] los autores presentan dos métodos robustos para ajustar elipses. El primer método es RANSAC + MMC (donde MMC es la abreviación de ‘mínima mediana de los cuadrados’). El funcionamiento de RANSAC + MMC lo podemos ver en el

algoritmo 4.

La función que se utilizó para resolver el problema fue:

$$g_3 : \mathbb{R}^5 \rightarrow \mathbb{R}$$

$$g_3(x) = \text{mediana} \sum_{i=1}^n d(p'_i, p''_i)^2 \quad (3.18)$$

donde $d()$ es la distancia euclidiana entre el punto del conjunto de datos y el contorno de la elipse.

RANSAC se encarga de buscar el mínimo de esta función, o sea que entregará la mejor elipse cuya suma de distancias ortogonales entre los puntos y la elipse al cuadrado es mínima.

Algoritmo 4 Heurística RANSAC + MMC

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$

Entrada: Probabilidad de encontrar puntos atípicos en el conjunto de datos ϵ_1

Entrada: Probabilidad de que los puntos a ajustar sean puntos de la elipse ϵ_0

Salida: Parámetros de la elipse $E = \{a, b, x_c, y_c, \alpha\}$

Determinar el número de iteraciones a realizar con la ecuación $m = \frac{\log(1-\epsilon_0)}{\log(1-(1-\epsilon_1)^{\epsilon_1})}$
 $i = 0$

while $i \neq m$ **do**

 Seleccionar cinco puntos aleatoriamente de P

 Ajustar los puntos y obtener una elipse candidata E

 Obtener las distancias ortogonales $R = \{r_1, r_2, \dots, r_n\}$ de cada punto en P a E

 Obtener la mediana del conjunto R

 Calcular el factor de escala $\hat{\sigma}_0 = 1.4826(1 + \frac{5}{n-5}) \cdot \text{med}$.

 Calcular el conjunto de pesos iniciales $W_0 = \{w_1, \dots\}$ con

$$w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_0} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_0} > 2.5 \end{cases}$$

if la mediana actual es menor a la de la iteración anterior **then**

 Calcular el factor de escala final usando: $\hat{\sigma}_f = \sqrt{\frac{\sum_{i=1}^n w_i r_i^2}{\sum_{i=1}^n w_i^{-5}}}$

 Calcular el conjunto de pesos finales W_f con

$$w_f = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_f} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_f} > 2.5 \end{cases}$$

 Realizar el ajuste con los puntos de P que tengan un peso final $w_i = 1$

 Almacenar los parámetros de la elipse E

$i \leftarrow i + 1$

return E

Ajuste tipo RANSAC

Si se quisiera encontrar el mínimo de la función descrita en la ecuación (3.18), o alguna otra función de aptitud, se necesitaría calcular $c = \binom{n}{5}$ combinaciones. Para valores grandes de n , c es un número muy grande, por lo que una heurística como RANSAC reducirá considerablemente la búsqueda.

En general una forma probabilista que reduce dicha búsqueda es la descrita en el algoritmo 5, que es el funcionamiento básico de RANSAC, aunque RANSAC utiliza fórmulas propuestas en la literatura para mejorar la búsqueda. Por lo tanto, a partir del algoritmo 5 se pueden proponer nuevas formas de mejorar la búsqueda, que puede variar dependiendo del enfoque que se de al problema. En esta tesis se propuso mejorar dicha búsqueda con ayuda de un algoritmo genético, dicha técnica será detallada en la sección 4.1, en la pág. 38.

Algoritmo 5 Heurística para ajustar una elipse

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$ **Salida:** Parámetros de la elipse $E = \{a, b, x_c, y_c, \alpha\}$

B = 0

repeat Seleccionar cinco puntos aleatoriamente de P

Ajustar los puntos usando el ajuste algebraico y obtener una elipse candidata

Evaluar la elipse candidata mediante una función de aptitud

if El valor de aptitud es mejor que B **then**

B = valor de aptitud

E = los parámetros de la elipse candidata

until La condición de paro fue alcanzada **return** E

3.2.2. Evolución diferencial + suma de distancias ortogonales

El segundo método presentado en [9] es llamado ED +SD (donde SD es la abreviación de ‘suma de distancias ortogonales’). Este método resuelve el problema usando la ED para encontrar la mejor solución con suma de distancias ortogonales como función de aptitud y se puede ver su funcionamiento en el algoritmo 6.

El ajuste de elipses usando la suma de distancias ortogonales, es una forma de hacer más robusto el ajuste de elipses, al ser menos sensibles a los puntos atípicos. La función de aptitud será entonces:

$$g_1 : \mathbb{R}^5 \rightarrow \mathbb{R}$$

$$g_1(x) = \sum_{i=1}^n d_i \quad (3.19)$$

donde d_i es la distancia euclidiana entre un punto del conjunto de datos y la elipse.

Algoritmo 6 Heurística ED + SD

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$

Entrada: Valor de umbral T

Entrada: Tamaño de población μ

Entrada: Número de generaciones máximo G_{max}

Entrada: Factor de recombinación CR

Entrada: Factor de diferencia F

Entrada: Valor de diferencia entre el máximo y mínimo s

Entrada: Límites superior e inferior de los parámetros (a, b, x_c, y_c, α)

Salida: Parámetros de la elipse $E = \{a, b, x_c, y_c, \alpha\}$

Generar la población inicial $X = \{x_1, \dots, x_n\}$

Calcular el valor de aptitud de cada individuo con la función de aptitud (en este caso suma de distancias ortogonales)

$k = 0$

repeat

for $i = 1$ **to** μ **do**

 Se seleccionan tres índices $\{I_1, I_2, I_3\} \in \mu$ diferentes de manera aleatoria

 Sea $j_{rand} \in [1, 5]$ un entero aleatorio

for $j = 1$ **to** 5 **do**

$$x'_{i,j} = \begin{cases} x_{i,I_3} + F(x_{i,I_1} - x_{i,I_2}) & : \text{ si } rand() < CR \text{ ó } j = j_{rand} \\ 0 & : \text{ cualquier otro caso} \end{cases}$$

if el valor de aptitud de x'_i es menor que x_i **then**

$$x_i = x'_i$$

min = valor de aptitud más pequeño de los individuos de X

max = valor de aptitud más grande de los individuos de X

$k \leftarrow k + 1$

until $max - min < s$ o $k > G_{max}$

return E

Para hacer más robusto el ajuste de elipses se agregaron algunas restricciones y se modificó la función de aptitud quedando de la siguiente forma:

$$g_2 : \mathbb{R}^5 \rightarrow \mathbb{R}$$

$$g_2(x) = \frac{2U(n-m)}{n} + \frac{1}{m} \sum_{\text{si } d_i < U} d_i \quad (3.20)$$

donde n es el número total de puntos y m el número de puntos cuya distancia es menor que el umbral U .

3.2.3. PSO + SD

Por último, se tiene el algoritmo de optimización por acumulación de partículas, cuyo funcionamiento se puede ver en el algoritmo 7 y que trabaja igual que ED + SD, pero con el algoritmo PSO en lugar de ED, para la búsqueda del mínimo del problema.

Ésto fue con la intención de comparar las diferencias entre estos algoritmos evolutivos, muy populares en la literatura especializada.

PSO fue propuesto originalmente en [29] para resolver problemas de optimización. Es un algoritmo basado en poblaciones que a su vez se basa en la simulación del comportamiento del vuelo de las aves.

A continuación están algunos términos usados en el algoritmo PSO[30]:

- **Acumulación.** Es la población del algoritmo.
- **Partícula.** Son los miembros o individuos de la acumulación. Cada partícula representa una posible solución del problema que se está resolviendo.
- ***pbest*.** Es la mejor posición que ha tenido una partícula dada (esto es, la posición donde el valor de aptitud fue el más alto).
- ***gbest*.** Es la mejor posición en toda la acumulación.
- **Líder.** Es la partícula con la mejor posición.
- **Velocidad.** Es el vector que determina en que dirección necesita moverse una partícula, para que mejore su posición.
- **Ponderación de inercia.** Denotado por W , la ponderación de inercia se emplea para controlar la variación de la velocidad debida a las velocidades anteriores en una partícula dada.
- **Factor de aprendizaje.** El factor de aprendizaje, son las constantes C_1 y C_2 . C_1 representa el factor de aprendizaje *cognitivo*, mientras que C_2 representa el factor de aprendizaje *social*.

Sea $\vec{x}_i(t)$ la posición de cada partícula en la posición i , en el paso t , que cambia dependiendo de la velocidad $\vec{v}_i(t)$. La posición $\vec{x}_i(t)$ se calcula con la siguiente ecuación:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (3.21)$$

El vector velocidad $\vec{v}_i(t)$ refleja la información del cambio posición y se calcula con la siguiente ecuación:

$$\vec{v}_i(t) = W\vec{v}_i(t-1) + C_1r_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + C_2r_2(\vec{x}_{gbest} - \vec{x}_i(t)) \quad (3.22)$$

donde $r_1, r_2 \in [0, 1]$ son valores aleatorios.

Algoritmo 7 Heurística PSO + SD

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$

Entrada: Valor de umbral T

Entrada: Tamaño de población μ

Entrada: Número de generaciones máximo G_{max}

Entrada: Constantes positivas C_1 y C_2

Entrada: Ponderación de inercia W

Entrada: Valor de diferencia entre el máximo y mínimo s

Entrada: Límites superior e inferior de los parámetros (a, b, x_c, y_c, α)

Salida: Parámetros de la elipse $E = \{a, b, x_c, y_c, \alpha\}$

Generar la población inicial $X = \{x_1, \dots, x_n\}$

Calcular el valor de aptitud de cada partícula o individuo con la función de aptitud (en este caso, suma de distancias ortogonales)

Calcular la velocidad de cada partícula

Obtener el líder

$k = 0$

repeat

for $i = 1$ **to** μ **do**

 Obtener la nueva posición de cada partícula

 Evaluar y obtener su función de aptitud

 Obtener la mejor posición de la partícula

 Obtener su nueva velocidad

 Obtener el nuevo líder

min = valor de aptitud más pequeño de los individuos de X

max = valor de aptitud más grande de los individuos de X

$k \leftarrow k + 1$

until $max - min < s$ o $k > G_{max}$

return E

Los resultados detallados de las heurísticas se muestran en el capítulo 5, sección 5.1 (en la pág. 49).

Capítulo 4

Ajuste robusto de múltiples elipses

Para llevar a cabo el ajuste de elipses es necesario encontrar los 5 parámetros que definen una elipse, una manera de hacerlo es con RANSAC, como está descrito en el algoritmo 5 (en la pág. 32). Si se hace la búsqueda de una manera determinista por fuerza bruta tomaría mucho tiempo llegar a la solución, sin embargo, si utilizamos un procedimiento probabilista como el del algoritmo 5, el tiempo de ejecución para obtener resultados aceptables se reduciría considerablemente.

En este trabajo se propone agregar un algoritmo genético al algoritmo 5, para mejorar la búsqueda de la solución. Se eligió un algoritmo genético por las características del problema, el algoritmo 5 usa cinco puntos tomados de manera de aleatoria, los cuales se ajustan con algún método determinista y se evalúan con una función de aptitud. Entonces el algoritmo genético puede mejorar la búsqueda, debido a que éstos han mostrado tener buen desempeño para aplicaciones de optimización. Si se toman los cinco puntos como índices, éste será un enfoque propicio para utilizar representación binaria, la cual es utilizada por los algoritmos genéticos.

El ajuste de múltiples elipses es un problema derivado del ajuste de una elipse, la diferencia radica en que el conjunto de datos representa no una, sino varias elipses. Entonces el ajuste de múltiples elipses puede verse como la obtención de los parámetros de cada elipse en un conjunto de datos con varias elipses dadas implícitamente. La forma en que esto se lleva a cabo es ajustando una elipse en el conjunto de datos, después se borran los puntos que describen dicha elipse y se hace el ajuste de otra elipse con los demás puntos, repitiendo este proceso hasta encontrar todas las elipses.

En la sección 4.1 se describirá el método propuesto en esta tesis para la resolución del problema de ajuste de elipses. En la sección 4.2 se explicarán las funciones de aptitud propuestas para resolver el problema de ajustes de elipses. En la sección 4.3 se mostrarán las características extras agregadas al algoritmo genético. Por último en la sección 4.4 se describirá el método propuesto para ajustar varias elipses en un conjunto de puntos dado.

Tabla 4.1: Parámetros para el algoritmo genético

Parámetro	Valor
Tamaño de población	70
Número de generaciones	5000
Probabilidad de mutación	0.15
Probabilidad de cruza	0.6

4.1. RANSAC + Algoritmo genético

Para mejorar la heurística presentada en el algoritmo 5 y de una manera similar a como se hace con RANSAC, se propone agregar un algoritmo genético para solucionar el problema de encontrar los cinco mejores puntos que representan a una elipse.

El funcionamiento básico de un algoritmo genético se puede ver en el algoritmo 1, en el que se genera una población a la que se aplican diferentes operadores para generar nuevas poblaciones hasta obtener un buen resultado.

El desempeño del algoritmo genético depende de los parámetros del mismo y de la robustez de la función de aptitud elegida. En este trabajo se utilizó un algoritmo genético genérico con las instancias y métodos comúnmente usadas en el área, un operador de cruza y uno de mutación, cabe mencionarse que se propuso un operador de búsqueda local que se explicará en la sección 4.3.1. Los parámetros usados en el algoritmo genético propuesto pueden verse en la tabla 4.1.

Dado que el algoritmo propuesto toma como referencia un algoritmo tipo RANSAC, con un algoritmo genético para mejorar la búsqueda, se referirá al algoritmo propuesto en esta tesis como GA-RANSAC. El algoritmo propuesto GA-RANSAC puede verse en el algoritmo 8.

A grandes rasgos GA-RANSAC funciona de la siguiente forma, se genera una población inicial de manera aleatoria, donde cada individuo se representa por los índices de 5 puntos del conjunto de datos. Los individuos pueden ser vistos como elipses candidatas, es decir, para cada individuo se realiza el ajuste algebraico de los cinco puntos, y se evalúa la función de aptitud, para determinar cual individuo corresponde a la elipse que mejor representa al conjunto de datos.

En cada generación se aplica un operador de cruza y de mutación, conservando al mejor individuo para la siguiente generación, el proceso es repetido hasta que se alcance un número máximo de generaciones o que se cumpla con la condición de paro.

Algoritmo 8 Algoritmo GA-RANSAC

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$ **Salida:** Parámetros de la elipse $E = \{a, b, x_c, y_c, \alpha\}$

Generar una población inicial de individuos de manera aleatoria, cada individuo se genera como la concatenación de los índices de 5 puntos aleatorios entre 1 y n .

Calcular la función de aptitud de cada individuo en la población, ajustando la elipse que define los puntos de los índices de cada individuo.

repeat

 Seleccionar individuos en base al valor de aptitud

 Aplicar operador de cruce

 Aplicar operador de mutación

 Calcular la función de aptitud de cada individuo en la población

 Conservar el mejor individuo para la siguiente generación

 Generar la siguiente población

until La condición de paro se cumpla**return** Los parámetros de la elipse que define los puntos del mejor individuo en la población.

El algoritmo genético usado en GA-RANSAC, tiene las características explicadas en las siguientes secciones.

Representación individual

Los puntos $P = \{p_1, p_2, \dots, p_n\}$ dados se almacenan en un archivo, representados por sus coordenadas $p_i = (x_i, y_i)$ en un espacio bidimensional.

Cada individuo usa los índices de cinco puntos en representación decimal como fenotipo y en representación binaria como cromosomas. Estos puntos se pueden ajustar con distintos métodos, en este caso se usa el método algebraico para ajustar los puntos.

Los cinco puntos son decodificados en representación binaria, con precisión variable dependiendo del número de puntos de entrada. La representación binaria es la concatenación de los índices $\{i_1, i_2, i_3, i_4, i_5\}$ de cada punto usado. En cada iteración del algoritmo se decodifican los índices y se ajusta una elipse candidata, dependiendo de su valor de aptitud, se busca el individuo con el mejor valor de aptitud en la población.

La población es un conjunto de individuos y en la codificación en lenguaje C está almacenada como un arreglo de estructuras, donde cada estructura almacena básicamente la cadena binaria (de tipo `char`) y el valor de la función objetivo (en un `double`).

Mecanismo de selección

El operador de selección es una rueda de ruleta, cada individuo es asignado a una sección de la ruleta. Esta estrategia le da alta prioridad al mejor individuo pero permite sobrevivir a los peores individuos. Además se aplica elitismo, esto es, el mejor individuo siempre será incluido en la siguiente generación.

Operador de cruza

El operador de cruza es aplicado en pares dependiendo de una probabilidad de cruza, es decir, la probabilidad determina si se lleva a cabo o no la cruza. En caso que no se lleve a cabo la cruza, los dos hijos resultantes son iguales a los padres, esto es, el hijo uno es igual al padre uno y el hijo dos es igual al padre dos.

En caso de que la cruza se lleve a cabo, se selecciona un punto de cruza de manera aleatoria, se toman los primeros bits del padre uno hasta el punto de cruza y los últimos bits del padre dos desde el punto de cruza para formar el primer hijo. El segundo hijo es formado de manera inversa, tomando los primeros bits del padre dos y los últimos bits del padre uno.

Operador de mutación

El operador de mutación se aplica a los individuos de la población después de que han pasado por el operador de cruza, al igual que con el operador de cruza, se usa un factor de probabilidad para aplicar o no el operador de mutación. La mutación consiste en modificar cada bit del individuo dependiendo del factor de probabilidad, se recorre la cadena y se invierten los bits cuando el factor de probabilidad es uno o se dejan intactos cuando es cero. En caso de que algún bit de la cadena que forma al individuo se haya modificado en la operación de cruza y/o mutación, se recalcula el valor fenotípico y el valor de aptitud del individuo.

4.2. Funciones de aptitud

En las siguientes secciones se proponen y discuten tres funciones que pueden ser útiles y robustas para el trabajo presentado en esta tesis. Estas funciones de aptitud son:

- Número de puntos
- Número de puntos en la elipse
- Suma de distancias Hausdorff

4.2.1. Número de puntos

La primera función de aptitud propuesta, es establecida como el número de elementos $p \in P$ que están localizados en una *vecindad* $-\epsilon$ de una elipse dada E . Para ser más precisos, se define de la siguiente manera:

Definición 4 Para una elipse E , se define una *vecindad* $-\epsilon$ como:

$$D_\epsilon(E) = \{x \in \mathbb{R}^2 \mid \text{dist}(x, E) \leq \epsilon\}, \quad (4.1)$$

donde $\epsilon > 0$ es un umbral dado.

En la figura 4.1 se puede ver un ejemplo de la *vecindad* $-\epsilon$.

Definición 5 Se define N_E como el número de puntos que están en $D_\epsilon(E)$:

$$N_E = |\{p \in P : p \in D_\epsilon(E)\}| \quad (4.2)$$

Lo que conlleva a la primera función de aptitud F_1 , esto es

Definición 6 Para una elipse E y un conjunto de datos $P = \{p_1, p_2, \dots, p_n\}$, se calcula F_1 como:

$$F_1 = N_E(E, P) \quad (4.3)$$

Esta función de aptitud obtiene en varios casos resultados satisfactorios. Aunque, tiene el problema de dar preferencia a elipses grandes (incluyendo elipses falsas).

4.2.2. Número de puntos en la elipse

La segunda función de aptitud es el cociente de dos funciones. Estas funciones son N_E , definida de la misma manera que en la sección anterior, y P_E , el perímetro de una elipse E . El perímetro P_E de una elipse se calcula con la aproximación de Ramanujan y se define a continuación:

Definición 7 Aproximación del perímetro de una elipse.

Para una elipse E definida por sus parámetros $\{a, b, x_c, y_c, \alpha\}$, la aproximación del perímetro P_E es:

$$P_E \approx \pi[3(a+b) - \sqrt{(a+3b)(3a+b)}] \quad (4.4)$$

Entonces F_2 se define como:

Definición 8 Para una elipse E definida por sus parámetros $\{a, b, x_c, y_c, \alpha\}$ y un conjunto de datos $P = \{p_1, p_2, \dots, p_n\}$, se calcula F_2 como:

$$F_2 = \frac{N_E(a, b, x_c, y_c, \alpha, P)}{P_E(a, b)} \quad (4.5)$$

Esta función de aptitud le da preferencia a elipses pequeñas con más puntos, por lo cual es una buena función de aptitud para el propósito de esta tesis.

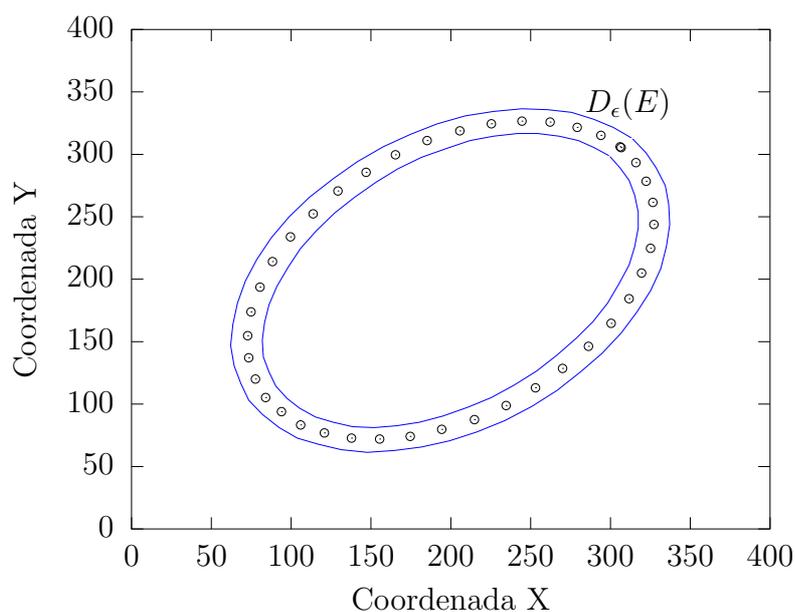


Figura 4.1: *Vecindad* $-\epsilon$ de una elipse candidata.

4.2.3. Suma de distancias de Hausdorff

La distancia de Hausdorff definida en la sección 2.3 (en la pág. 13), se utiliza como métrica en varios escenarios donde la distancia de dos conjuntos tiene que ser evaluada, lo cual es útil para el contexto de este trabajo. De manera particular, se busca minimizar la distancia de Hausdorff entre los puntos dados y la elipse candidata. Para hacer ésto, se generan los puntos correspondientes a la elipse candidata, y después se calcula la distancia de Hausdorff entre estos puntos y los puntos del conjunto de datos. De tal manera que la función es definida como sigue:

Definición 9 Sea X_ϵ el conjunto de puntos dentro de una vecindad $-\epsilon$ (definida en la ecuación (4.1))

$$X_\epsilon = \{x \in P : x \in D_\epsilon(E)\}, \quad (4.6)$$

Definición 10 Sea E_ϵ el conjunto de puntos generado en la frontera de una elipse dada.

Se define F_3 como la distancia de Hausdorff entre X_ϵ y E_ϵ :

$$F_3 = d_H(X_\epsilon, E_\epsilon) \quad (4.7)$$

4.3. Características extras

Además de los operadores de selección, mutación y cruce, se propuso un operador de búsqueda local, con el propósito de mejorar el desempeño del algoritmo genético.

Asimismo se agregó una condición de paro extra al algoritmo genético, para que regrese el resultado en una cantidad menor de evaluaciones de la función de aptitud.

4.3.1. Operador local

Para mejorar la búsqueda de soluciones se agregó un operador local. El operador local, es una inspección en una región específica del espacio de búsqueda, en el caso de este trabajo, el operador local busca elipses dentro de una región en particular.

Al igual que con los operadores de cruce y mutación, el operador local se aplica dependiendo de un factor de probabilidad. En esta tesis se usó un factor de probabilidad igual a 0.1. Cuando el operador local es aplicado, se toma uno de los puntos que conforman el individuo a modificar y se reasignan los otros 4 puntos, de manera que la distancia euclidiana entre ellos, sea menor a un valor determinado δ .

El funcionamiento del operador local puede verse en el algoritmo 9.

Algoritmo 9 Operador local

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$.

Entrada: Índices del individuo a modificar $I = [i_1, \dots, i_5]$.

Entrada: Valor de tolerancia δ .

Entrada: Factor de probabilidad PO

Salida: Individuo modificado.

```
if Prob( $PO$ ) then
  {/*la función Prob regresa 1 o 0 dependiendo del argumento*/}
   $N \leftarrow \emptyset$ 
  Elegir  $i_0 \in I$  aleatoriamente
  for  $k \leftarrow 0$  to  $n$  do
    if ( $\|p_k - p_{i_0}\|_2 < \delta$ ) & ( $k \neq i_0$ ) then
       $N \leftarrow N \cup k$ 
      if  $|N| = 4$  then
        Detener el ciclo
  if  $|N| < 4$  then
    return  $I$ 
  else
     $I \leftarrow i_0 \cup N$ 
    return  $I$ 
else
  return  $I$ 
```

Tabla 4.2: Parámetros extras para el algoritmo genético

Parámetro	Valor
Número de repeticiones del mejor valor	50
Número de generaciones máximo	5000
Valor de tolerancia δ	15
Probabilidad de operador local	0.1

4.3.2. Condición de paro

El algoritmo propuesto GA-RANSAC requiere de un número máximo de generaciones. Usualmente, un número en el rango de miles de generaciones es usado, y en este caso se usaron cinco mil generaciones. Sin embargo, en muchas aplicaciones es requerido un algoritmo rápido, por lo que se agregó una condición de paro para interrumpir el ciclo principal del algoritmo genético.

El algoritmo se detiene cuando el mejor individuo de cada generación no mejora en un número dado de iteraciones. Esto es, en cada ciclo del algoritmo genético, se conserva al individuo con mejor valor de aptitud en la población, que es sustituido solo cuando un nuevo individuo tiene mejor valor de aptitud en las siguientes generaciones, entonces si después de un número dado de generaciones no hay un mejor individuo, el algoritmo se detiene. Cabe hacer la mención de que éste es un criterio usualmente usado en la literatura especializada.

Los parámetros extras del algoritmo genético se pueden ver en la tabla 4.2.

4.4. Extracción de múltiples elipses

Existen diversas aplicaciones en donde es necesario el ajuste de múltiples elipses, como en calibración de cámaras por círculos concéntricos [31] o para ajustar un modelo en términos de elipses conectadas [8], solo por mencionar algunas.

La extracción de múltiples elipses consiste en obtener los parámetros de cada elipse en un conjunto de datos con varias elipses dadas implícitamente, en la figura 4.2 se puede ver un ejemplo de cuatro elipses inmersas en ruido.

Para llevar a cabo la extracción de múltiples elipses, el método utilizado consistió en buscar la elipse con mejor valor de aptitud en el conjunto de datos dado, después se guardaban los parámetros de dicha elipse y se eliminaban los puntos que pertenecieran a la elipse encontrada, el proceso se repitió con los puntos restantes,

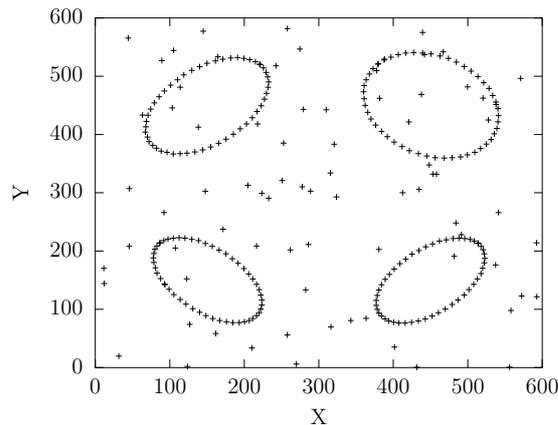


Figura 4.2: Múltiples elipses inmersas en ruido

hasta encontrar el número de elipses que se buscaban o que el número de puntos fuera menor a 5.

El algoritmo 10 proporciona el pseudocódigo de los pasos que se utilizaron para extraer múltiples elipses.

Algoritmo 10 Algoritmo para extraer múltiples elipses

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$

Entrada: Número de elipses que serán extraídas E .

Entrada: Parámetros para el algoritmo GA-RANSAC

Entrada: Definir un valor de umbral T

Salida: Parámetros de cada elipse extraída.

```

 $m \leftarrow n$ 
for  $i \leftarrow 1$  to  $E$  do
  Llamar a GA-RANSAC y extraer una elipse
  Almacenar los parámetros de la elipse
  for  $j \leftarrow 1$  to  $n$  do
    Calcular la distancia  $d$ , del punto  $p_j$  a la elipse
    if  $d < T$  then
      Eliminar el punto  $p_j$  del conjunto de datos
       $m \leftarrow m - 1$ 
    if  $m < 5$  then
      Terminar ejecución
   $n \leftarrow m$ 
return Parámetros de las elipses

```

En la figura 4.3 se ilustra un ejemplo de como funciona la extracción de múltiples elipses utilizando el número de puntos en la elipse como función de aptitud. En

la figura 4.3(a) están los puntos del conjunto de datos inicial, después de aplicar el algoritmo GA-RANSAC, se obtiene la mejor elipse (según la función de aptitud F_2) que se ve en la figura 4.3(b).

El siguiente paso es guardar los parámetros de la elipse con el mejor valor de aptitud y después eliminar los puntos que pertenecen a dicha elipse como se ve en la figura 4.3(c). El proceso se repite y se busca la elipse con el mejor valor de aptitud como se puede ver en la figura 4.3(d), de nuevo se guardan los parámetros y se eliminan los puntos que pertenecían a esa elipse, quedando los puntos de la figura 4.3(e).

El proceso se repite y se obtiene la elipse con el mejor valor de aptitud como se ve en la figura 4.3(f), después de guardar los parámetros de la elipse, se eliminan los puntos y quedan los puntos de la figura 4.3(g). Por último se busca la mejor elipse en el conjunto de datos restantes (figura 4.3(h)) y se guardan los parámetros de la elipse. Como ya se encontraron las cuatro elipses, el algoritmo se detiene.

Los resultados detallados de la extracción de múltiples elipses se muestran en el capítulo 5, sección 5.2 (en la pág. 60).

Búsqueda automática

Debido a que en ocasiones es necesario un proceso completamente automático, si no se conoce el número de elipses en el conjunto de datos, es requerido un método para encontrar las elipses de manera autónoma. En esta tesis se propone un método para hacer la extracción automáticamente, el proceso consiste en buscar las elipses como en el algoritmo 10, y en cada iteración guardar el valor de aptitud de la primer elipse encontrada, después el algoritmo se detendrá si el valor de aptitud de las elipses es menor al 20 % del valor de aptitud de la primer elipse encontrada. El algoritmo 11 proporciona el pseudocódigo de los pasos que se utilizaron para extraer múltiples elipses automáticamente.

Algoritmo 11 Algoritmo para extraer múltiples elipses automáticamente

Entrada: Un conjunto de puntos dado $P = \{p_1, p_2, \dots, p_n\}$ **Entrada:** Parámetros para el algoritmo GA-RANSAC**Entrada:** Definir un valor de umbral T **Salida:** Parámetros de cada elipse extraída.

Llamar a GA-RANSAC y extraer una elipse

Almacenar los parámetros de la elipse

 $V \leftarrow$ valor de aptitud de la elipse encontrada $m \leftarrow n$ **repeat** **for** $j \leftarrow 1$ **to** n **do** Calcular la distancia d , del punto p_j a la elipse **if** $d < T$ **then** Eliminar el punto p_j del conjunto de datos $m \leftarrow m - 1$ $n \leftarrow m$

Llamar a GA-RANSAC y extraer una elipse

 $v_1 \leftarrow$ valor de aptitud de la elipse encontrada **if** $v_1 < (0.2 * V)$ **then**

Terminar ejecución

Almacenar los parámetros de la elipse

until $m < 5$ **return** Parámetros de las elipses

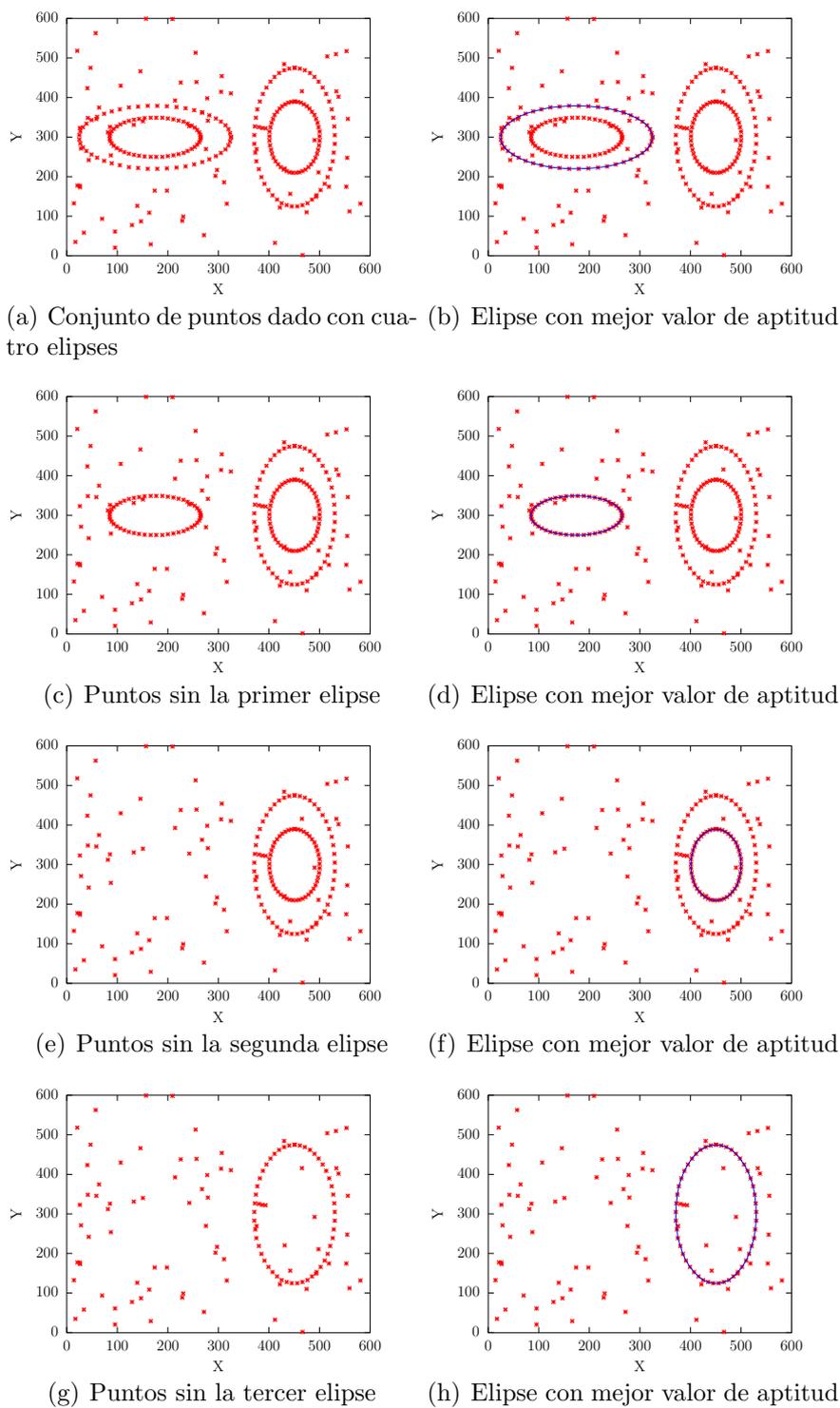


Figura 4.3: Extracción de múltiples elipses en un ejemplo con cuatro elipses

Capítulo 5

Resultados

Las pruebas que se realizaron, fueron para el ajuste de una elipse y para el ajuste de múltiples elipses, siendo estas últimas las de mayor interés para esta tesis. En el ajuste de una elipse se hicieron las pruebas con métodos deterministas y probabilistas, mientras que para el ajuste de múltiples elipses se hicieron las pruebas con las heurísticas GA-RANSAC y ED+SD.

En el ajuste de una elipse, se comparó el comportamiento para diferentes niveles de ruido y diferentes cantidades de puntos atípicos. En total se compararon ocho algoritmos, dos de los cuales fueron deterministas y los otros seis fueron probabilistas.

El objetivo fue ver cual algoritmo se acercaba mejor a la solución en condiciones altas de ruido, y comparar las tres funciones de aptitud propuestas en este trabajo, contra los métodos existentes para el ajuste de elipses.

En la parte de ajuste de múltiples elipses, solo se comparó GA-RANSAC y ED+SD, debido a que ED+SD es una técnica con resultados mejores a los del método RANSAC, que es el método más popular para ajuste de múltiples elipses.

Del mismo modo se experimento con el comportamiento de GA-RANSAC y ED+SD, en el ajuste de dos elipses sin ruido, cuatro elipses con ruido y cuatro elipses con diferentes niveles de puntos atípicos.

5.1. Diseño de experimentos para el ajuste de una elipse

Las pruebas se realizaron con algunos de los métodos deterministas y probabilistas que existen para resolver el ajuste de una elipse, en concreto con los siguientes métodos:

- Ajuste algebraico
- Levenberg-Marquadt
- RANSAC + MMC
- PSO + SD
- ED + SD
- GA-RANSAC

Las pruebas se dividieron en 2 partes, primero las pruebas con diferentes niveles de ruido y después las pruebas con diferentes niveles de puntos atípicos. Para las pruebas se generaron conjuntos de datos con los cuales llevar a cabo los experimentos.

5.1.1. Experimentos con diferentes niveles de ruido

En la figura 5.1 se pueden ver el tipo de elipses generadas que se utilizaron para llevar a cabo los experimentos con los diferentes niveles de ruido. Se usaron seis niveles ruido gaussiano con $\sigma \in \{0, 1, 2, 4, 8, 16\}$. Se hizo hasta un valor de $\sigma = 16$ porque valores mayores no son muy comunes dentro de los experimentos de visión y procesamiento de imágenes.

Se generaron 50 archivos por cada nivel de ruido, tomando como referencia una elipse con los siguientes valores:

- $a = 200$
- $b = 120$
- $x_c = 300$
- $y_c = 250$
- $\alpha = 2.51 \approx 0.8\pi$

en cada archivo se agregó ruido gaussiano aleatorio dependiendo del valor de σ . Se ejecutaron los métodos con cada uno de los archivos como datos de entrada, y en cada ejecución se guardaron los resultados.

En las tablas 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 y 5.8, se pueden ver los valores mínimos, máximos y promedio obtenidos en cada parámetro de la elipse para los diferentes niveles de ruido.

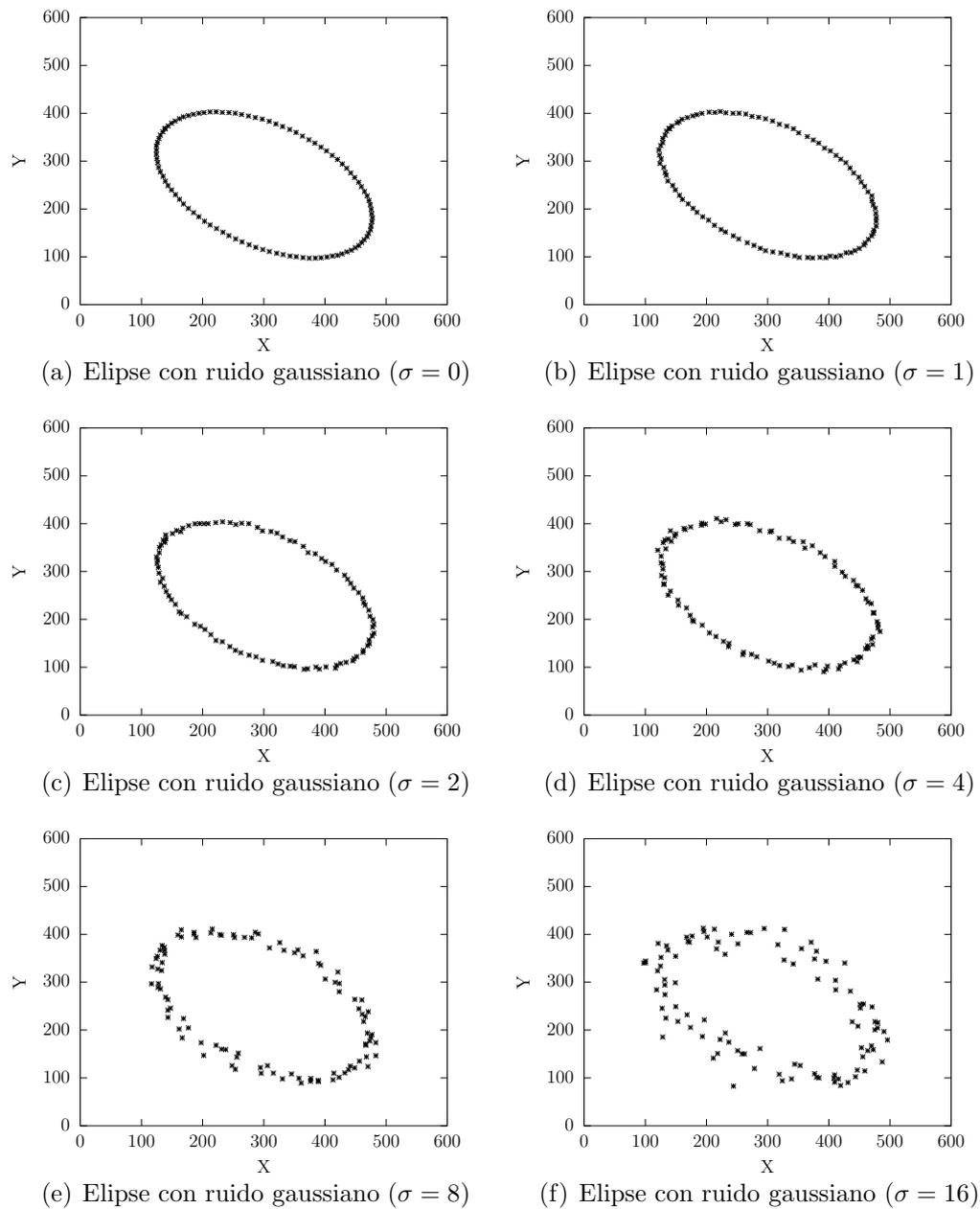


Figura 5.1: Elipses generadas para los experimentos con diferentes niveles de ruido gaussiano

Se puede ver en las tablas 5.1 y 5.2, los resultados de los métodos deterministas, mientras que en las tablas 5.3, 5.4, 5.5, 5.6, 5.7 y 5.8, están los resultados de los métodos probabilistas.

La tabla 5.1 muestra los resultados del ajuste algebraico, la tabla 5.2 los resultados del algoritmo de Levenberg-Marquadt, la tabla 5.3 los resultados de RANSAC + MMC, la tabla 5.4 los resultados de PSO+SD, la tabla 5.5 los resultados de ED+SD,

Tabla 5.1: Resultados del **Ajuste algebraico** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.09	200.09	200.09	119.91	119.91	119.91	300.12	300.12	300.12	249.79	249.79	249.79	2.51	2.51	2.51
1	199.73	200.25	200.01	119.80	120.25	120.05	299.94	300.41	300.21	249.29	249.76	249.59	2.51	2.52	2.51
2	199.07	200.36	199.84	119.40	120.42	120.01	299.65	300.71	300.24	249.21	249.93	249.59	2.51	2.52	2.51
4	198.47	200.94	199.86	119.09	121.03	120.06	299.50	301.54	300.47	248.78	250.48	249.72	2.51	2.52	2.51
8	197.10	201.50	199.44	118.81	122.84	121.42	298.18	302.12	300.27	247.73	251.56	249.97	2.49	2.54	2.52
16	193.85	202.62	199.01	118.81	126.36	122.97	296.97	304.03	299.74	247.50	253.32	250.40	2.47	2.54	2.51

Tabla 5.2: Resultados de **Levenberg-Marquadt** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.16	200.16	200.16	119.93	119.93	119.93	300.10	300.10	300.10	249.75	249.75	249.75	2.51	2.51	2.51
1	199.69	200.31	200.00	119.79	120.27	120.05	299.91	300.48	300.20	249.26	249.78	249.58	2.51	2.52	2.51
2	199.02	200.55	199.94	119.47	120.37	119.90	299.58	300.78	300.23	249.09	250.02	249.63	2.51	2.52	2.51
4	198.55	200.92	199.97	118.96	120.83	119.86	299.57	301.46	300.44	248.85	250.47	249.74	2.50	2.52	2.51
8	197.99	202.20	200.57	118.52	122.78	120.51	298.40	302.29	300.59	247.35	251.10	249.70	2.49	2.55	2.52
16	198.18	206.84	202.62	113.97	124.09	119.94	295.44	302.58	299.22	246.22	252.85	249.89	2.47	2.56	2.52

Tabla 5.3: Resultados de **RANSAC + MMC** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	197.56	200.14	199.16	118.54	120.29	119.39	299.90	301.37	300.49	247.08	250.39	248.76	2.51	2.53	2.52
1	188.43	205.77	198.74	102.24	120.93	116.72	281.88	308.01	297.57	238.52	250.39	248.06	2.44	2.56	2.51
2	179.74	202.80	196.11	91.75	121.83	115.72	257.83	301.63	293.22	241.95	255.12	250.92	2.29	2.53	2.47
4	172.23	216.03	205.87	75.65	122.26	113.68	252.76	302.98	302.28	226.56	252.34	245.39	2.39	2.55	2.52
8	169.37	222.40	203.93	62.64	134.81	115.02	260.72	311.28	299.58	221.79	255.68	245.68	2.21	2.67	2.51
16	187.49	220.90	205.56	88.56	120.29	111.80	285.21	304.58	299.21	209.24	255.68	246.88	2.38	2.64	2.53

y las tablas 5.6, 5.7 y 5.8 los resultados de GA-RANSAC con las tres funciones de aptitud propuestas.

En general los ocho métodos son robustos al ruido con un valor de σ de hasta 16, aunque GA-RANSAC+ F_3 , es el que encuentra valores más alejados del valor real de los parámetros.

Para visualizar con más detalle los resultados de los diferentes métodos, en la figura 5.2 están los porcentajes de error obtenidos en cada parámetro de la elipse. Se puede observar el desempeño de cada método, siendo el método de GA-RANSAC +

Tabla 5.4: Resultados de **PSO + SD** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.04	200.04	200.04	119.84	119.84	119.84	300.19	300.19	300.19	249.78	249.79	249.79	2.51	2.51	2.51
1	199.70	200.40	199.99	119.73	120.22	120.03	299.86	300.50	300.26	249.28	249.79	249.60	2.51	2.52	2.51
2	199.02	200.63	199.91	119.05	120.58	119.87	299.61	300.66	300.22	249.13	250.11	249.61	2.51	2.52	2.51
4	176.11	201.52	197.68	118.61	121.63	123.07	299.36	301.72	300.38	248.45	250.45	249.65	1.24	2.52	2.36
8	170.87	201.78	196.66	118.04	148.05	125.00	298.06	303.04	300.51	247.57	252.32	250.27	1.25	2.54	2.36
16	193.88	205.86	200.66	116.73	123.41	120.51	295.85	304.76	300.29	246.63	253.97	250.04	2.47	2.55	2.52

Tabla 5.5: Resultados de **ED + SD** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	199.96	199.96	199.96	119.85	119.85	119.85	300.16	300.16	300.16	249.82	249.82	249.82	2.51	2.51	2.51
1	199.59	200.21	199.95	119.71	120.22	120.04	299.87	300.48	300.24	249.31	249.80	249.60	2.51	2.52	2.51
2	198.84	200.52	199.80	119.07	120.59	119.88	299.43	300.66	300.20	249.18	250.12	249.64	2.51	2.52	2.51
4	198.30	201.22	199.87	118.63	121.16	119.93	299.38	301.93	300.47	248.80	250.48	249.71	2.50	2.53	2.52
8	195.43	202.23	199.73	117.39	123.50	120.50	297.23	303.63	300.08	247.70	251.70	249.80	2.47	2.54	2.51
16	189.88	207.49	200.04	110.87	128.81	121.96	292.62	305.13	300.17	246.09	256.82	252.00	2.39	2.59	2.48

Tabla 5.6: Resultados de **GA-RANSAC + F_1** con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	199.96	199.96	199.96	119.43	119.43	119.43	300.25	300.25	300.25	250.12	250.12	250.12	2.52	2.52	2.52
1	196.41	202.45	200.16	118.51	121.24	120.13	298.57	302.56	300.71	247.87	251.48	249.70	2.49	2.54	2.52
2	197.01	203.52	200.16	117.55	122.80	119.82	298.23	301.52	300.31	247.66	251.51	249.97	2.50	2.53	2.52
4	197.59	201.90	199.75	117.19	122.41	119.86	298.85	302.19	300.62	247.26	250.95	249.61	2.49	2.54	2.52
8	196.54	203.42	200.65	115.88	121.45	119.47	297.34	303.15	300.25	247.28	252.37	249.65	2.47	2.54	2.51
16	187.85	210.72	199.65	110.23	129.01	121.04	293.43	306.84	300.61	242.09	259.41	250.99	2.36	2.61	2.51

F_3 el menos eficiente.

La figura 5.2(a) muestra la gráfica de los porcentajes de error obtenidos en el parámetro a , por los ocho métodos, donde GA-RANSAC + F_3 es el que tiene mayores porcentajes de error, para valores de $\sigma > 4$.

La figura 5.2(b) muestra la gráfica de los porcentajes de error obtenidos en el parámetro b , por los ocho métodos, RANSAC + MMC, GA-RANSAC + F_3 y PSO + SD, son los que tienen mayores porcentajes de error.

Tabla 5.7: Resultados de **GA-RANSAC** + F_2 con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	192.93	192.93	192.93	119.72	119.72	119.72	300.35	300.35	300.35	250.09	250.09	250.09	2.49	2.49	2.49
1	191.72	194.64	193.10	117.95	120.94	119.55	299.04	301.00	300.07	247.94	250.48	249.31	2.47	2.55	2.51
2	193.40	196.72	194.77	115.32	119.29	118.06	298.92	301.24	300.02	248.08	250.96	249.73	2.48	2.55	2.52
4	196.04	199.49	197.62	116.37	120.69	118.79	297.99	302.01	300.59	247.84	251.72	249.88	2.49	2.52	2.51
8	195.41	202.50	198.98	117.55	122.12	120.29	296.85	303.04	300.05	247.09	251.12	249.67	2.47	2.54	2.51
16	184.95	207.29	197.90	108.43	126.35	119.26	294.63	307.38	300.67	246.01	260.70	251.86	2.37	2.60	2.51

Tabla 5.8: Resultados de **GA-RANSAC** + F_3 con 50 ejecuciones de cada nivel de σ . Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

σ	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.37	200.37	200.37	119.52	119.52	119.52	299.94	299.94	299.94	249.91	249.91	249.91	2.51	2.51	2.51
1	197.07	213.44	202.38	114.97	121.12	119.14	298.27	302.14	300.26	248.29	250.17	249.47	2.47	2.60	2.54
2	192.28	219.69	204.19	115.19	122.76	119.41	294.61	301.84	299.24	246.80	252.19	249.60	2.28	2.60	2.47
4	193.83	233.35	208.97	110.20	122.80	118.30	293.44	306.71	301.19	247.85	253.88	251.02	2.24	2.64	2.52
8	174.98	311.41	238.26	103.28	128.91	117.24	284.19	310.18	298.45	240.74	260.57	249.69	2.03	2.83	2.52
16	165.46	299.29	267.45	97.34	129.94	114.27	232.34	316.23	285.74	235.23	264.56	259.15	1.61	2.93	2.40

La figura 5.2(c) muestra la gráfica de los porcentajes de error obtenidos en el parámetro x_c , por los ocho métodos, donde GA-RANSAC + F_3 es el que tiene mayores porcentajes de error, para valores de $\sigma > 8$.

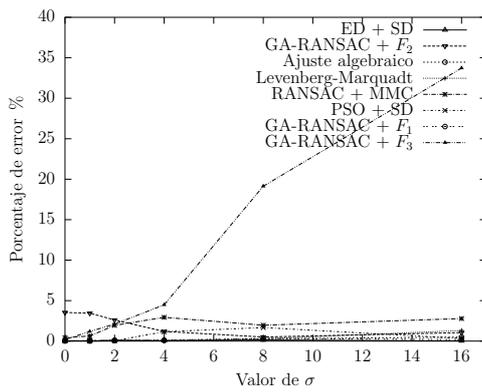
La figura 5.2(d) muestra la gráfica de los porcentajes de error obtenidos en el parámetro y_c , por los ocho métodos, RANSAC + MMC y GA-RANSAC + F_3 , son los que tiene mayores porcentajes de error.

La figura 5.2(e) muestra la gráfica de los porcentajes de error obtenidos en el parámetro α , por los ocho métodos, donde PSO+SD es el que tiene mayores porcentajes de error.

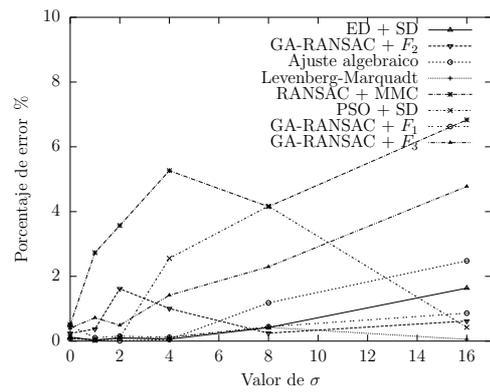
5.1.2. Experimentos con diferentes niveles de puntos atípicos

Las siguientes pruebas se realizaron con diferentes porcentajes de puntos atípicos, donde el porcentaje se refiere a la relación entre los puntos que pertenecen a la elipse dada implícitamente y los puntos que no pertenecen a ella.

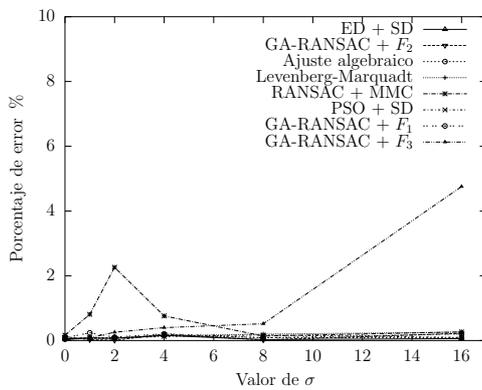
En la figura 5.3 se pueden ver el tipo de elipses generadas que se utilizaron para llevar a cabo los experimentos, se usaron seis porcentajes de puntos atípicos



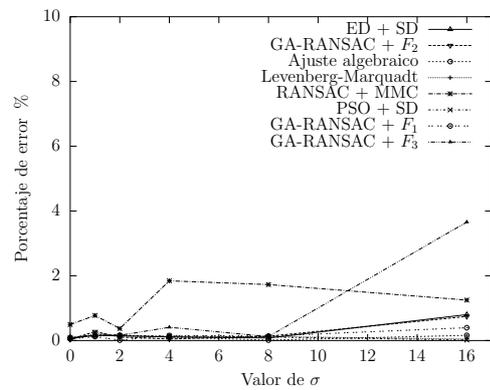
(a) Pruebas con ruido (Parámetro a)



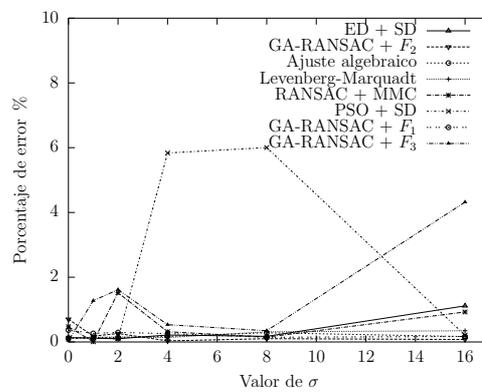
(b) Pruebas con ruido (Parámetro b)



(c) Pruebas con ruido (Parámetro x_c)



(d) Pruebas con ruido gaussiano (Parámetro y_c)



(e) Pruebas con ruido (Parámetro α)

Figura 5.2: Gráficas de los porcentajes de error en cada parámetro con diferentes niveles de ruido

{0%, 20%, 40%, 60%, 80%, 100%}. Se hizo hasta un valor de 100% para ver si son algoritmos robustos.

Se generaron 50 archivos por cada porcentaje de puntos atípicos, tomando como

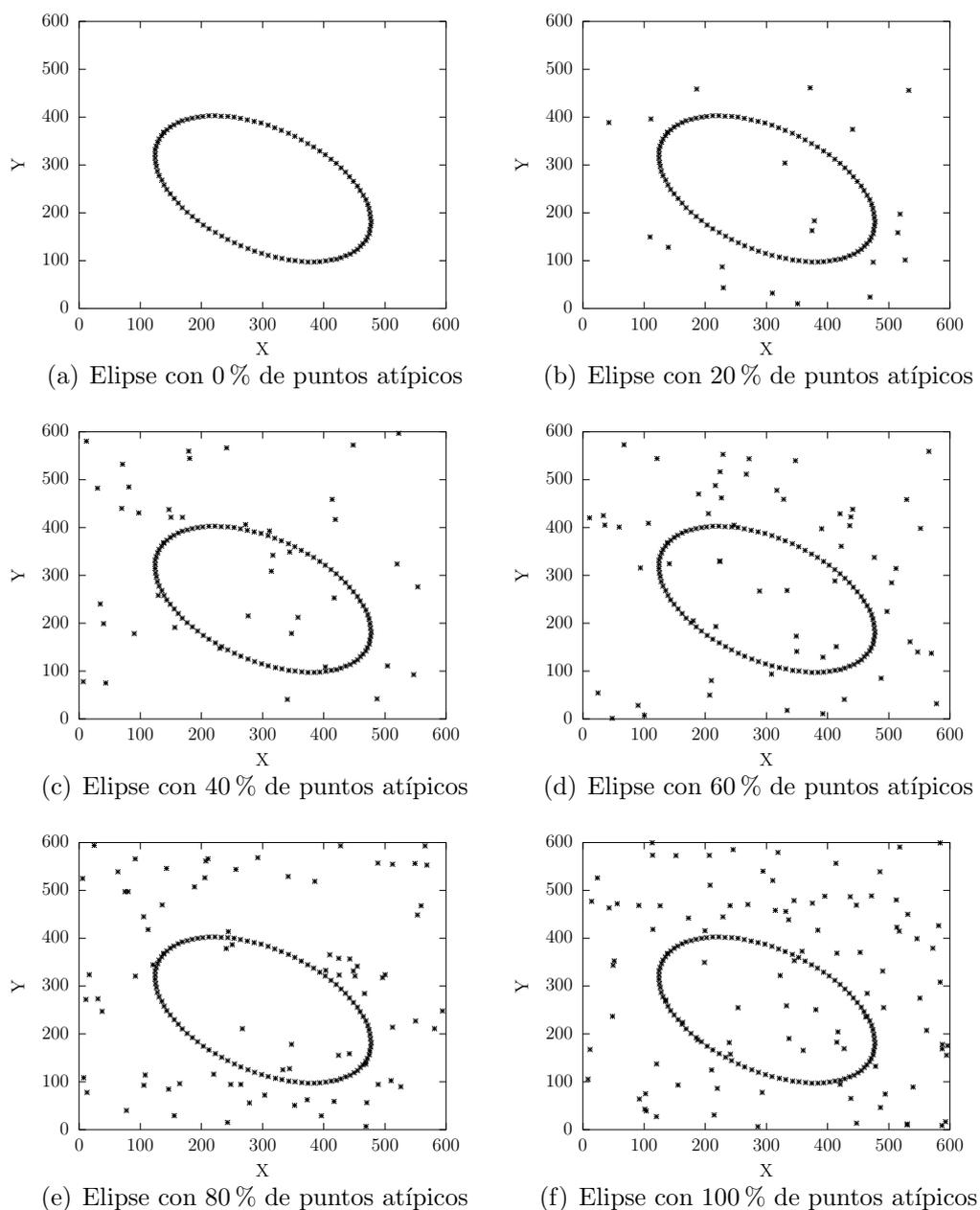


Figura 5.3: Elipses generadas para los experimentos, con diferentes niveles de puntos atípicos

referencia una elipse con los siguientes valores:

- $a = 200$
- $b = 120$
- $x_c = 300$
- $y_c = 250$

- $\alpha = 2.51 \approx 0.8\pi$

en cada archivo se agregaron puntos aleatorios dependiendo del porcentaje. Se ejecutaron los métodos con cada uno de los archivos como datos de entrada, y en cada ejecución se guardaron los resultados.

En las tablas 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15 y 5.16, se pueden ver los valores mínimos, máximos y promedio obtenidos en cada parámetro de la elipse para los diferentes porcentajes de puntos atípicos.

Tabla 5.9: Resultados del **Ajuste algebraico** con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.09	200.09	200.09	119.91	119.91	119.91	300.12	300.12	300.12	249.79	249.79	249.79	2.51	2.51	2.51
20	184.10	211.00	195.39	140.30	181.79	165.70	286.94	321.31	304.20	248.38	289.88	274.79	0.57	2.78	2.05
40	196.43	213.14	204.39	174.13	192.60	183.83	282.73	322.61	304.02	257.01	303.22	283.32	0.74	2.55	1.83
60	205.12	222.93	212.09	182.26	201.03	191.96	289.23	322.30	305.88	274.97	315.06	291.46	0.83	2.60	1.68
80	206.06	225.47	216.01	189.87	208.87	199.45	296.12	320.20	308.79	273.17	306.02	292.33	0.40	2.72	1.60
100	208.17	223.90	216.37	191.39	208.36	202.46	293.15	319.49	307.12	279.92	306.50	293.56	0.24	2.32	1.57

Tabla 5.10: Resultados de **Levenberg-Marquadt** con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.16	200.16	200.16	119.93	119.93	119.93	300.10	300.10	300.10	249.75	249.75	249.75	2.51	2.51	2.51
20	202.24	275.16	239.20	114.95	182.69	161.00	270.25	318.22	298.78	246.98	288.22	271.63	0.85	2.81	2.33
40	203.13	253.69	237.38	158.83	192.22	180.63	272.57	306.35	292.90	256.96	283.40	272.26	0.07	2.73	1.98
60	216.48	248.91	234.42	156.05	201.34	184.46	279.45	311.34	295.48	262.65	291.52	278.65	0.30	2.78	1.85
80	219.39	254.16	241.14	161.82	200.94	187.32	286.79	315.36	301.11	261.21	288.97	276.41	0.89	2.97	2.31
100	215.91	273.49	252.22	165.07	213.01	190.29	276.32	315.88	310.36	263.74	400.07	315.04	0.05	3.10	2.23

Se puede ver en las tablas 5.9 y 5.10, los resultados para los métodos deterministas, y en las tablas 5.11, 5.12, 5.13, 5.14, 5.15 y 5.16, los resultados para los métodos probabilistas.

La tabla 5.9 muestra los resultados del ajuste algebraico, la tabla 5.10 los resultados del algoritmo de Levenberg-Marquadt, la tabla 5.11 los resultados de RANSAC + MMC, la tabla 5.12 los resultados de PSO+SD, la tabla 5.13 los resultados de ED+SD, y las tablas 5.14, 5.15 y 5.16 los resultados de GA-RANSAC con las tres

Tabla 5.11: Resultados de **RANSAC + MMC** con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.09	211.04	201.75	119.44	137.14	122.29	299.56	314.59	302.02	249.52	260.51	251.58	2.51	2.55	2.52
20	198.86	201.83	199.44	116.36	122.46	118.81	286.98	301.13	297.53	247.39	252.84	250.50	2.48	2.52	2.51
40	198.86	200.94	199.99	115.20	121.51	119.29	297.64	300.66	299.41	245.67	251.39	249.56	2.50	2.53	2.51
60	198.86	201.27	200.07	117.76	120.82	119.68	299.70	301.22	300.36	247.54	250.41	249.33	2.51	2.53	2.52
80	198.88	200.77	200.18	117.14	121.78	119.81	298.83	300.77	300.08	247.55	251.68	249.64	2.50	2.53	2.51
100	198.88	200.55	200.58	117.88	120.41	120.22	299.38	301.26	300.79	247.75	250.54	249.91	2.50	2.53	2.52

Tabla 5.12: Resultados de **PSO + SD** con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	177.18	200.04	197.03	119.84	152.34	124.27	300.10	300.19	300.16	249.61	249.79	250.00	0.00	2.51	2.21
20	182.59	200.15	197.71	119.86	155.27	125.58	295.48	300.23	299.99	249.73	253.09	250.27	0.00	2.51	2.11
40	185.19	200.22	197.08	119.91	164.82	130.77	294.68	304.19	300.61	249.75	254.01	251.43	0.00	2.51	1.91
60	183.92	200.31	196.12	119.93	183.55	140.28	299.92	312.39	303.47	249.83	272.99	256.76	0.00	2.51	1.71
80	190.48	206.23	202.97	120.08	182.80	141.24	291.99	316.64	303.76	249.86	286.38	260.60	0.00	2.51	1.67
100	186.15	209.98	200.64	120.19	190.49	145.07	298.37	319.55	305.24	249.82	287.96	261.70	0.00	2.51	1.68

Tabla 5.13: Resultados de **ED + SD** con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	199.96	199.96	199.96	119.85	119.85	119.85	300.16	300.16	300.16	249.82	249.82	249.82	2.51	2.51	2.51
20	199.96	200.07	200.01	119.80	119.86	119.84	300.13	300.21	300.17	249.77	249.84	249.81	2.51	2.51	2.51
40	199.99	200.06	200.04	119.80	119.87	119.85	300.14	300.21	300.19	249.77	249.84	249.81	2.51	2.51	2.51
60	199.99	200.09	202.20	119.80	119.87	124.24	300.12	300.25	302.58	249.75	249.83	253.43	2.51	2.51	2.51
80	199.95	200.13	200.05	119.70	119.87	119.82	300.12	300.22	300.18	249.66	249.85	249.79	2.51	2.51	2.51
100	199.95	200.07	200.03	119.80	119.89	119.84	300.16	300.22	300.19	249.75	249.86	249.79	2.51	2.51	2.51

funciones de aptitud propuestas.

Como se puede ver en los resultados, para niveles de puntos atípicos grandes, los métodos deterministas son los que obtienen resultados menos acertados, mientras que los métodos probabilistas obtienen mejores resultados. Lo que justifica el uso de heurísticas probabilistas para la resolución del problema de ajuste de elipses, dada su robustez para resolver el problema en conjuntos de datos con altos niveles de puntos

Tabla 5.14: Resultados de **GA-RANSAC** + F_1 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	192.93	192.93	192.93	119.72	119.72	119.72	300.35	300.35	300.35	250.09	250.09	250.09	2.49	2.49	2.49
20	191.72	197.83	193.80	115.01	121.15	118.96	297.55	302.29	300.21	246.23	251.15	249.37	2.48	2.55	2.52
40	191.51	197.70	195.11	114.23	121.61	118.04	295.78	303.95	299.90	247.76	253.67	249.89	2.46	2.54	2.50
60	191.25	198.41	195.08	114.07	120.32	117.90	296.75	304.24	300.54	247.41	251.19	249.79	2.46	2.56	2.52
80	191.76	200.03	195.52	113.52	120.75	118.11	296.29	303.89	299.81	246.06	251.28	249.47	2.49	2.56	2.53
100	191.52	200.45	195.49	112.77	122.55	118.57	296.48	302.43	300.23	247.22	252.90	249.98	2.47	2.56	2.52

Tabla 5.15: Resultados de **GA-RANSAC** + F_2 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	199.96	199.96	199.96	119.43	119.43	119.43	300.25	300.25	300.25	250.12	250.12	250.12	2.52	2.52	2.52
20	195.69	204.76	200.07	117.30	122.89	120.25	296.28	302.15	299.41	245.76	252.70	250.03	2.49	2.55	2.51
40	197.85	203.71	200.85	116.21	122.50	119.57	295.79	303.41	299.96	248.36	253.53	250.83	2.45	2.53	2.50
60	195.80	203.24	200.35	115.24	122.78	118.96	296.40	303.46	300.56	245.73	252.96	250.59	2.48	2.54	2.52
80	197.39	203.02	199.71	115.51	121.55	118.91	295.86	303.30	300.16	246.53	251.92	249.67	2.48	2.55	2.52
100	195.81	204.21	200.35	114.09	123.72	120.51	296.16	303.57	300.54	245.88	253.20	249.94	2.48	2.57	2.52

Tabla 5.16: Resultados de **GA-RANSAC** + F_3 con 50 ejecuciones de cada porcentaje de puntos atípicos. Las columnas m , M y \bar{x} son el mínimo, máximo y valor promedio de cada parámetro obtenido.

%	Parámetro a			Parámetro b			Parámetro x_c			Parámetro y_c			Parámetro α		
	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}	m	M	\bar{x}
0	200.37	200.37	200.37	119.52	119.52	119.52	299.94	299.94	299.94	249.91	249.91	249.91	2.51	2.51	2.51
20	197.39	440.56	309.25	108.24	182.52	152.36	232.90	379.02	309.79	239.54	349.37	299.57	0.16	2.77	1.67
40	231.90	627.45	365.28	142.18	212.75	175.66	246.87	414.88	327.19	1.20	393.52	285.76	0.08	2.83	1.34
60	242.13	608.81	520.41	142.90	213.17	184.29	144.49	414.38	296.24	229.78	499.59	448.89	0.16	2.44	1.53
80	228.18	399.68	315.89	167.85	206.36	181.01	245.56	339.40	306.62	210.18	367.09	297.26	0.27	3.03	1.59
100	232.05	728.27	480.38	145.83	200.03	175.43	264.48	345.38	302.37	208.62	415.40	297.59	0.31	3.07	1.71

atípicos.

Sin embargo **GA-RANSAC** + F_3 , que es una heurística probabilista, no regresa buenos resultados, al igual que en la sección anterior con ruido en los puntos, lo que muestra que la función F_3 , no es una buena función de aptitud para la resolución del problema de ajuste de elipses.

Para visualizar con más detalle los resultados de los diferentes métodos, en la figura 5.4 están los porcentajes de error obtenidos en cada parámetro de la elipse. Se puede observar el desempeño de cada método para diferentes números de puntos atípicos.

La figura 5.4(a) muestra la gráfica de los porcentajes de error obtenidos en el parámetro a , por los ocho métodos, donde GA-RANSAC + F_3 es el que tiene mayores porcentajes de error. Mientras que ED + SD y GA-RANSAC+ F_2 son los que muestran los mejores resultados para porcentajes arriba de 60 %.

La figura 5.4(b) muestra la gráfica de los porcentajes de error obtenidos en el parámetro b , por los ocho métodos, el ajuste algebraico, GA-RANSAC + F_3 y Levenberg-Marquadt, son los que tiene mayores porcentajes de error.

La figura 5.4(c) muestra la gráfica de los porcentajes de error obtenidos en el parámetro x_c , por los ocho métodos, donde todos obtuvieron bajos porcentajes de error.

La figura 5.4(d) muestra la gráfica de los porcentajes de error obtenidos en el parámetro y_c , por los ocho métodos, Levenberg-Marquadt, el ajuste algebraico y GA-RANSAC + F_3 , son los que tiene mayores porcentajes de error.

La figura 5.4(e) muestra la gráfica de los porcentajes de error obtenidos en el parámetro α , por los ocho métodos, donde GA-RANSAC + F_3 , PSO+SD, Levenberg-Marquadt y el ajuste algebraico, son los que tienen mayores porcentajes de error.

En conclusión los algoritmos que mejor se comportaron, fueron GA-RANSAC+ F_2 y ED+SD, debido a que fueron los que presentaron menor porcentaje de error, los algoritmos deterministas Levenberg-Marquadt y ajuste algebraico, son robustos al ruido, pero no a los puntos atípicos, mientras que GA-RANSAC+ F_1 , PSO+SD y RANSAC+MMC, tienen resultados mejores que los de los algoritmos deterministas, el algoritmo con menor desempeño fue GA+RANSAC+ F_3 .

5.2. Diseño de experimentos para el ajuste de múltiples elipses

En la sección anterior se mostraron las pruebas para el ajuste de una elipse, en la que se pudo observar que la función de aptitud F_3 , no es la mejor opción para resolver el problema, mientras que las funciones F_1 y F_2 son mejores opciones para ajustar elipses.

En el problema de ajuste de múltiples elipses se necesita de un algoritmo robusto,

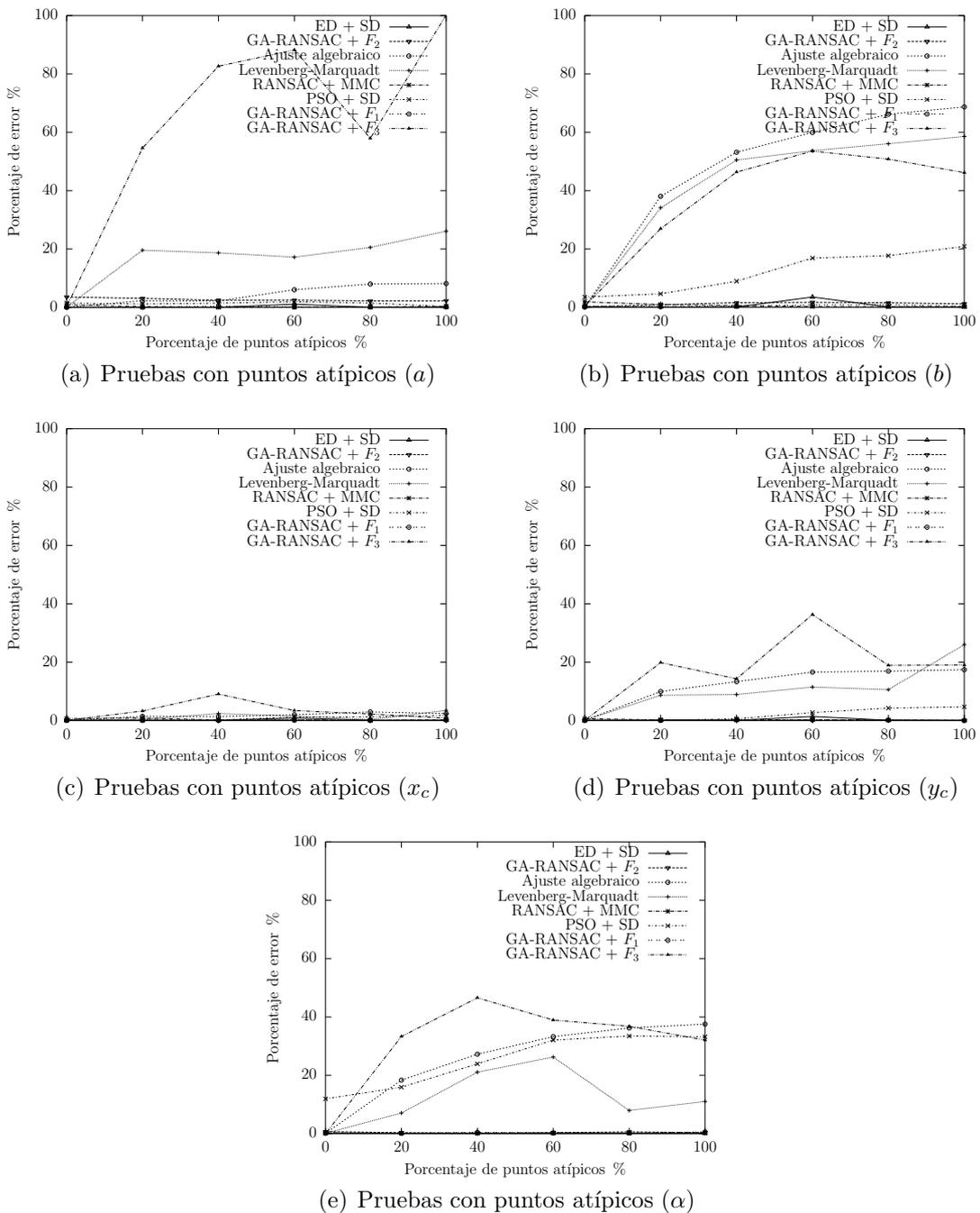


Figura 5.4: Gráficas de los porcentajes de error en cada parámetro con diferentes porcentajes de puntos atípicos

debido a que los puntos atípicos son muy comunes en diversas aplicaciones de visión computacional y procesamiento de señales. En la figura 5.4(b) se mostró que los métodos deterministas son sensibles a los puntos atípicos, mientras que los algoritmos menos sensibles a los puntos atípicos fueron GA-RANSAC + F_1 , GA-RANSAC + F_2

y ED + SD.

Debido al buen desempeño en las pruebas, los experimentos con múltiples elipses se realizaron usando los métodos de GA-RANSAC y ED + SD.

Las pruebas que se hicieron del comportamiento de GA-RANSAC y ED+SD fueron: pruebas con dos elipses, en la que se variaba el número de puntos en la segunda elipse, el segundo con cuatro elipses, variando los niveles de ruido, y el último variando los niveles de puntos atípicos.

5.2.1. Experimentos con dos elipses

Para hacer las pruebas del comportamiento de GA-RANSAC, se usó la función F_2 como función de aptitud y se usaron los siguientes casos de prueba:

- Elipses disjuntas
- Elipses anidadas
- Elipses superpuestas

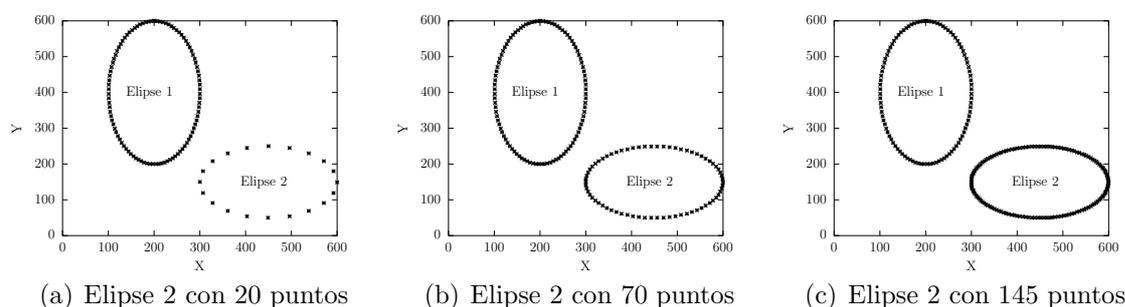


Figura 5.5: Caso 1. Elipses disjuntas, la elipse 1 está formada por 100 puntos y la elipse 2 varía

En la figura 5.5 están 3 de los conjuntos que se generaron para hacer las pruebas con elipses disjuntas, se generaron 35 archivos con dos elipses como las de la figura 5.5(a), con 100 puntos que pertenecían a la elipse 1, mientras la elipse 2 tenía diferentes números de puntos ($\#puntos \in \{5, 10, 15, \dots, 160, 165, 170\}$).

En la figura 5.6 está el segundo caso prueba con elipses anidadas, al igual que con el primer caso se generaron 35 archivos con dos elipses, en la que la primer elipse estaba formada por 100 puntos y la segunda elipse tenía diferentes números de puntos ($\#puntos \in \{5, 10, 15, \dots, 160, 165, 170\}$).

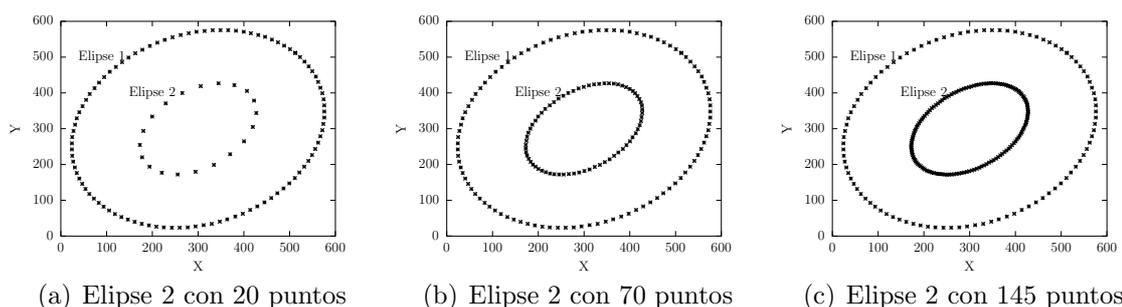


Figura 5.6: Caso 2. Elipses anidadas, la elipse 1 está formada por 100 puntos y la elipse 2 varía

En la figura 5.7 está el tercer caso con elipses superpuestas, al igual que con el primer y segundo caso se generaron 35 archivos con dos elipses, en la que la primera elipse estaba formada por 100 puntos y la segunda elipse tenía diferentes números de puntos ($\#puntos \in \{5, 10, 15, \dots, 160, 165, 170\}$).

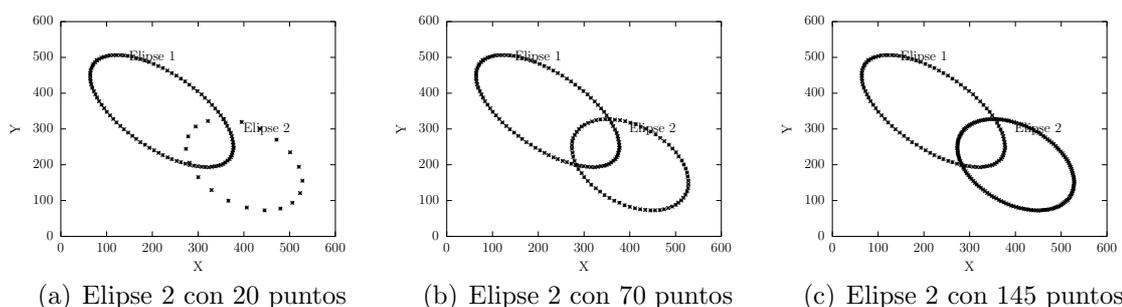


Figura 5.7: Caso 3. Elipses superpuestas, la elipse 1 está formada por 100 puntos y la elipse 2 varía

En las figuras 5.8, 5.9 y 5.10, están los resultados de GA-RANSAC + F_2 , para los tres casos.

Los resultados muestran cual fue la elipse encontrada por el algoritmo en una ejecución para cada caso, que dependiendo del número de puntos en la elipse 2, encontró una u otra.

La heurística ED + SD presentada en [32], es una técnica robusta, para la solución del problema de elipses, como se mostró en la sección 5.1. Por lo tanto se hicieron pruebas con ED + SD, para comparar contra la heurística propuesta en esta tesis.

Las pruebas que se realizaron con ED + SD para el ajuste de múltiples elipses, fueron las mismas que se realizaron para GA-RANSAC. Se usaron los mismos casos de prueba de GA-RANSAC.

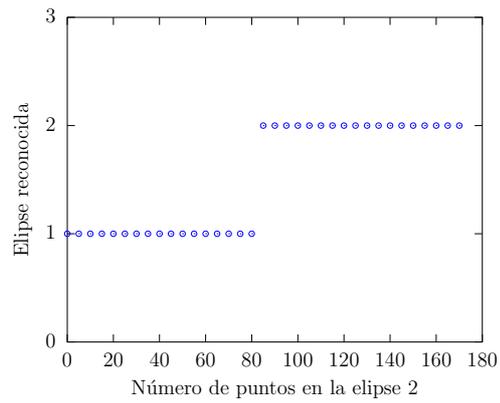


Figura 5.8: Gráfica de la elipse reconocida por **GA-RANSAC** + F_2 en el caso 1, para los diferentes números de puntos en la elipse 2

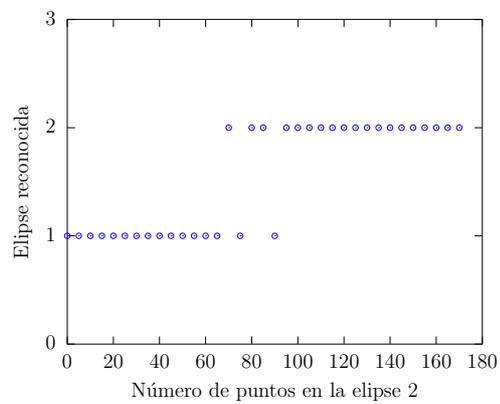


Figura 5.9: Gráfica de la elipse reconocida por **GA-RANSAC** + F_2 en el caso 2, para los diferentes números de puntos en la elipse 2

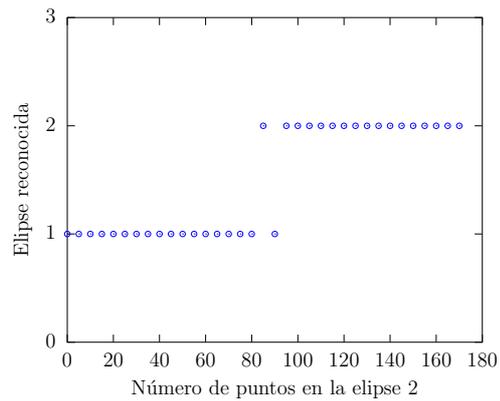


Figura 5.10: Gráfica de la elipse reconocida por **GA-RANSAC** + F_2 en el caso 3, para los diferentes números de puntos en la elipse 2

Los casos fueron las elipses disjuntas de la figura 5.5, las elipses anidadas de la figura 5.6 y las elipses superpuestas de la figura 5.16. Igualmente se generaron 35 archivos con dos elipses, en la que la primer elipse estaba formada por 100 puntos y la segunda elipse tenía diferentes números de puntos ($\#puntos \in \{5, 10, 15, \dots, 160, 165, 170\}$).

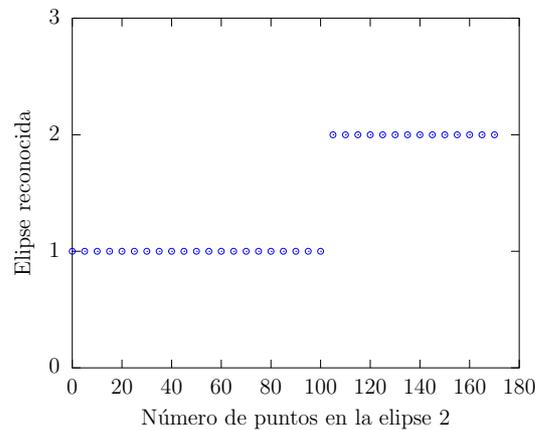


Figura 5.11: Gráfica de la elipse reconocida por **ED + SD** en el caso 1, para los diferentes números de puntos en la elipse 2

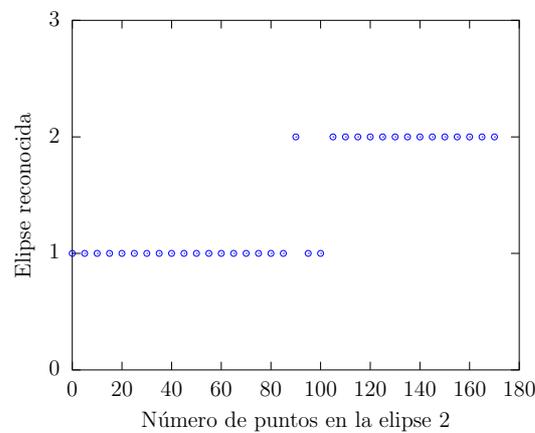


Figura 5.12: Gráfica de la elipse reconocida por **ED + SD** en el caso 2, para los diferentes números de puntos en la elipse 2

En las figuras 5.11, 5.12 y 5.13, están los resultados de ED + SD, para los tres casos.

Los resultados muestran cual fue la elipse encontrada por el algoritmo en una ejecución para cada caso, que dependiendo del número de puntos en la elipse 2, encontró una u otra.

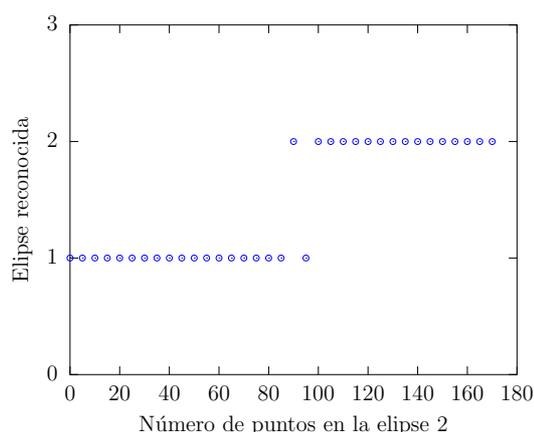


Figura 5.13: Gráfica de la elipse reconocida por **ED + SD** en el caso 3, para los diferentes números de puntos en la elipse 2

5.2.2. Experimentos con ruido en los puntos

Las siguientes pruebas se realizaron con conjuntos de datos que contenían 4 elipses en diferentes condiciones de ruido. Al igual que en la sección anterior los casos de prueba fueron elipses disjuntas, elipses anidadas y elipses superpuestas, en las figuras 5.14, 5.15 y 5.16 están los tipos de elipses utilizadas para las pruebas.

En la figura 5.14 está el primer caso de elipses disjuntas, con diferentes niveles de ruido. Se generaron 40 archivos para cada valor de σ , con 100 puntos para definir cada una de las cuatro elipses.

En la figura 5.15 están algunas elipses del segundo caso de elipses anidadas, con diferentes niveles de ruido. Al igual que en el primer caso se generaron 40 archivos para cada valor de σ , con 100 puntos para cada elipse.

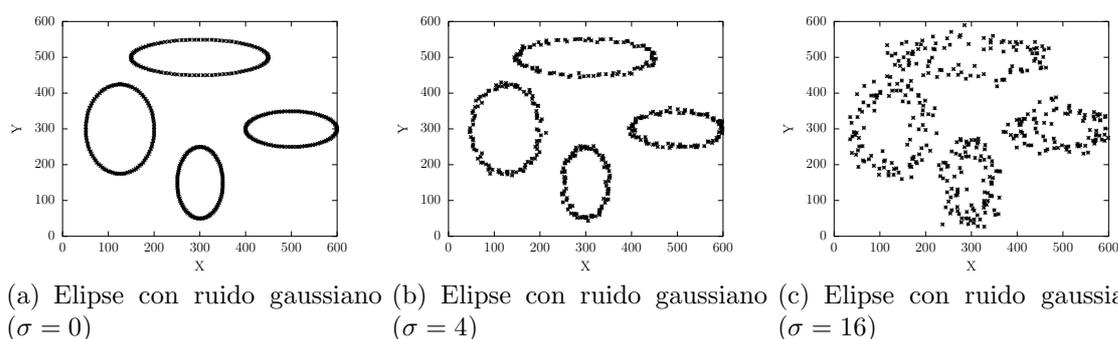


Figura 5.14: Caso 1. Múltiples elipses disjuntas con diferentes niveles de ruido gaussiano

Por último en la figura 5.16 están algunas elipses del tercer caso de elipses superpuestas, con diferentes niveles de ruido. Al igual que en el primer y segundo caso se

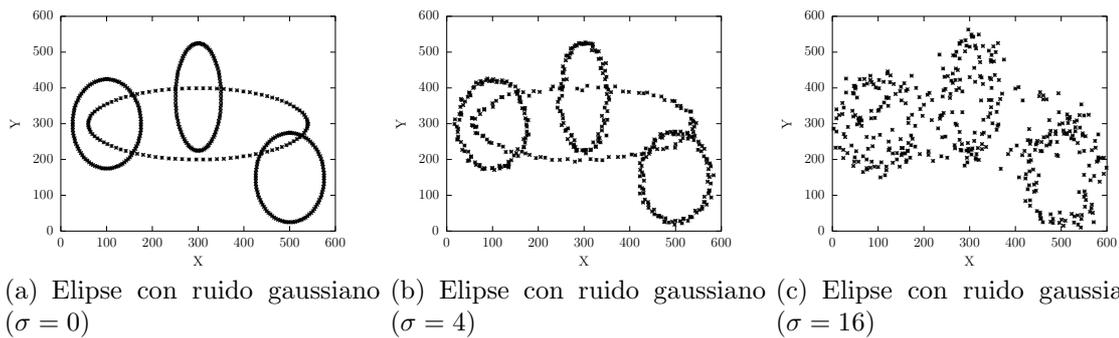


Figura 5.15: Caso 2. Múltiples elipses anidadas con diferentes niveles de ruido gaussiano

generaron 40 archivos para cada valor de σ , con 100 puntos para cada elipses.

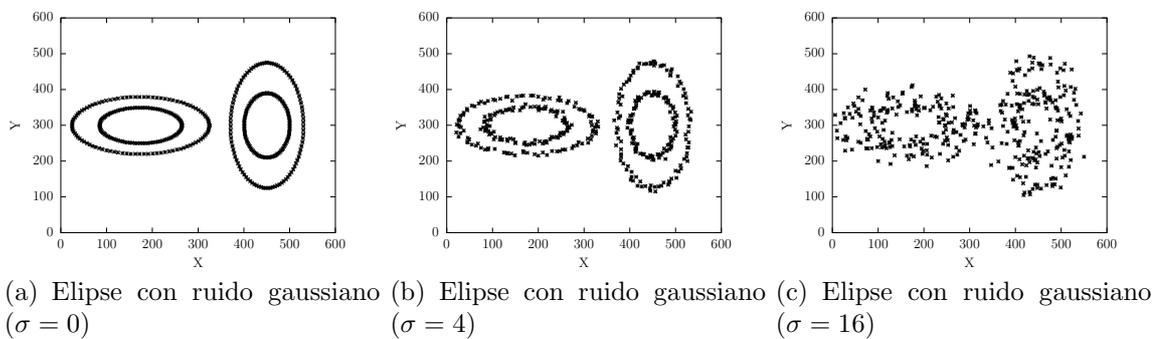


Figura 5.16: Caso 3. Múltiples elipses superpuestas con diferentes niveles de ruido gaussiano

El algoritmo se ejecutó en cada uno de los archivos, se almacenaron los resultados y se graficó el promedio de elipses encontradas en cada caso. En las figuras 5.17, 5.18 y 5.19, están las gráficas del promedio de elipses encontradas en cada nivel de ruido para cada caso.

Los valores de sigma que se probaron fueron: $\sigma \in \{0, 1, 2, 4, 8, 16\}$, para cada valor de sigma se ejecutó el algoritmo en los 40 archivos generados. En cada ejecución el algoritmo buscaba cuatro elipses y se almacenaba el número de elipses que se encontraron en dicha ejecución (verificando cuantas de las elipses encontradas coincidían con las generadas).

Para ED+SD, al igual que con GA-RANSAC los casos de prueba fueron elipses disjuntas, elipses anidadas y elipses superpuestas, vistos en las figuras 5.14, 5.15 y 5.16. Se generaron 40 archivos para cada valor de σ , con 100 puntos para cada elipses.

El algoritmo se ejecutó en cada uno de los archivos, se almacenaron los resultados y se graficó el promedio de elipses encontradas en cada caso. En la figura 5.20, están

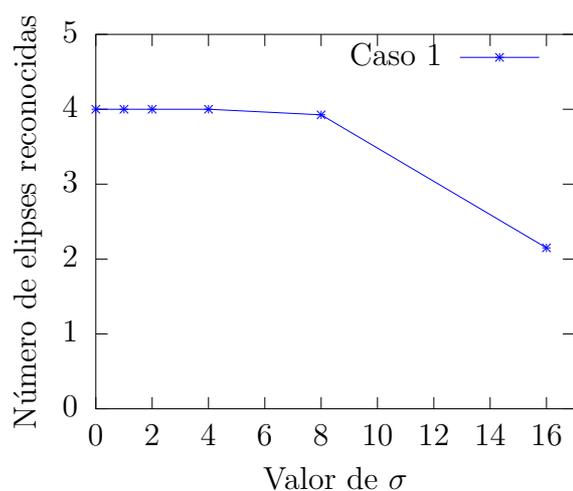


Figura 5.17: Gráfica del número de elipses reconocidas por **GA-RANSAC** + F_2 en el caso 1, para los diferentes valores de σ

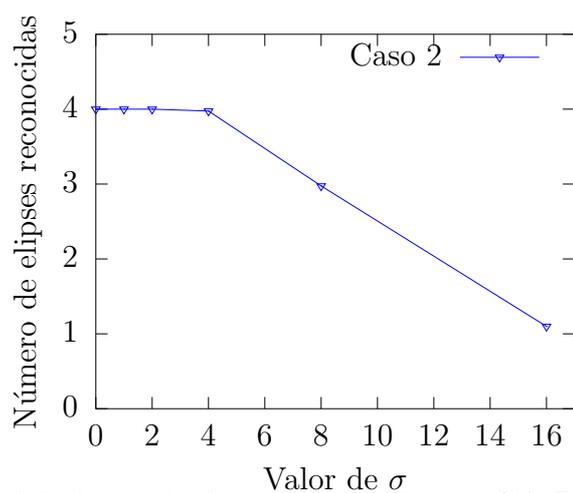


Figura 5.18: Gráfica del número de elipses reconocidas por **GA-RANSAC** + F_2 en el caso 2, para los diferentes valores de σ

las gráficas del promedio de elipses encontradas en cada nivel de ruido para cada caso por ED + SD.

Los valores de sigma que se probaron fueron los mismos de GA-RANSAC ($\sigma \in \{0, 1, 2, 4, 8, 16\}$), para cada valor de sigma se ejecutó el algoritmo en los 40 archivos generados. En cada ejecución el algoritmo buscaba cuatro elipses y se almacenaba el número de elipses que se encontraron en dicha ejecución.

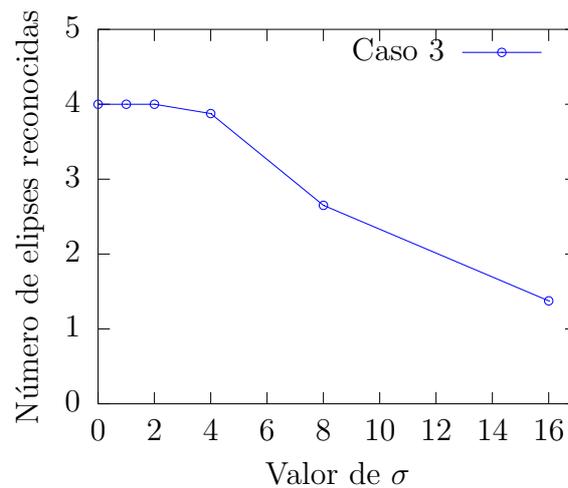


Figura 5.19: Gráfica del número de elipses reconocidas por **GA-RANSAC** + F_2 en el caso 3, para los diferentes valores de σ

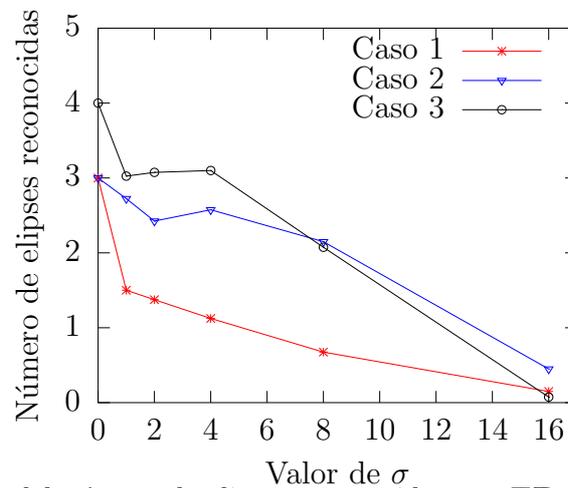


Figura 5.20: Gráfica del número de elipses reconocidas por **ED** + **SD** en los casos 1, 2 y 3, para los diferentes valores de σ

5.2.3. Experimentos con diferentes niveles de puntos atípicos

Para el trabajo experimental se generaron puntos como conjuntos de datos, usando OCTAVE¹. Los puntos se generaron alrededor de una elipse imaginaria y se almacenaron en un archivo, después se generaron puntos atípicos aleatorios y se agregaron al archivo.

En las pruebas se verificó que la segunda función de aptitud —esto es, el número de puntos en el perímetro— fue la mejor función de aptitud, las siguientes pruebas fueron realizadas usando la segunda función de aptitud.

¹<http://www.gnu.org/software/octave/>

Casos de prueba

Para esta tesis el enfoque recayó en tres casos; cuando las elipses son disjuntas, cuando las elipses están superpuestas y cuando están anidadas.

Elipses disjuntas

Éste es el caso más común que puede presentarse en datos reales, y el más sencillo de resolver. Es posible ver un ejemplo en la figura 5.21.

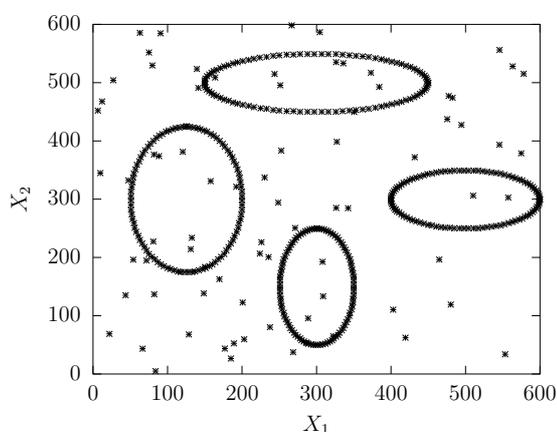


Figura 5.21: Caso de prueba uno, las elipses son disjuntas con 20 por ciento de puntos atípicos.

Elipses superpuestas

Éste caso es el más complicado de resolver de los tres por que las intersecciones de las elipses pueden interpretarse como elipses y los algoritmos pueden llegar a contar las intersecciones como elipses. Se puede ver en la figura 5.22 un ejemplo de este caso, con sesenta por ciento de puntos atípicos.

Elipses anidadas

Finalmente, el tercer caso de prueba es el de elipses anidadas, que se puede ver en la figura 5.23. Éste caso es de particular interés por servir para la calibración de cámaras [33] que se discutirá en el siguiente capítulo.

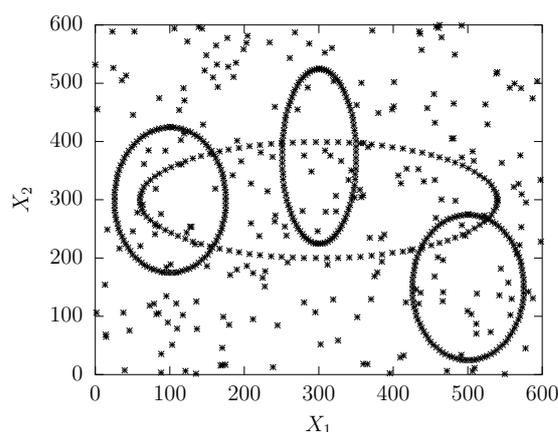


Figura 5.22: Caso de prueba dos, las elipses están superpuestas, con 60 por ciento de puntos atípicos.

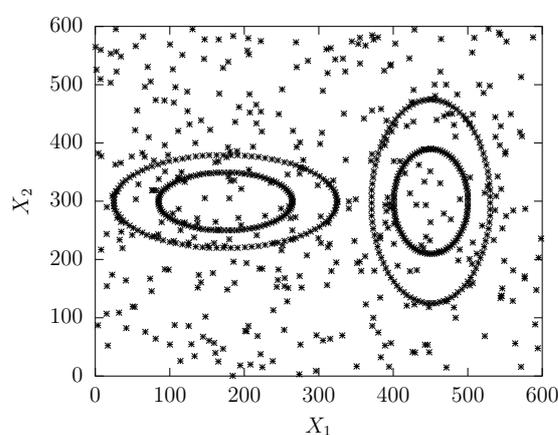


Figura 5.23: Caso de prueba tres, las elipses están anidadas, con 100 por ciento de puntos atípicos.

Resultados

Se ejecutó el programa doscientas veces para cada uno de los tres casos mencionados anteriormente, en cada caso se usó un conjunto de datos generado con diferentes niveles de puntos atípicos. El algoritmo fue comparado con ED + SD presentado en [9, 32], tal como se mencionó en la sección 3.2.2, ED + SD usa la suma de distancias euclidianas como función de aptitud, y se usaron los mismos parámetros usados en [9].

Cada caso estuvo compuesto de cuatro elipses, se ejecutó GA-RANSAC cuatro veces. Después se almacenaron los resultados y se revisó cuántas elipses fueron encontradas.

Se graficaron los resultados de GA-RANSAC + F_2 (segunda función de aptitud) para los tres casos de prueba, los cuales pueden verse en las figuras 5.24, 5.25 y 5.26.

En la tabla 5.18, 5.17 y 5.19 se pueden ver los resultados y el número promedio de iteraciones para los tres casos.

Se hizo una prueba usando la segunda función de aptitud F_2 sin GA-RANSAC, con la intención de verificar el desempeño de GA-RANSAC. Se ejecutó en un ciclo con el número de evaluaciones promedio usado en GA-RANSAC, y los resultados pueden verse en la figura 5.27.

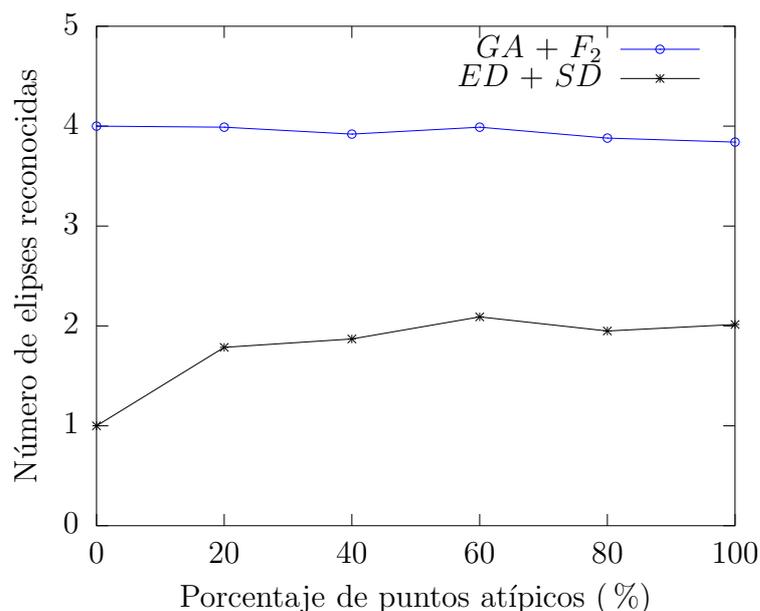


Figura 5.24: Resultados del primer caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.

Tabla 5.17: Resultados del algoritmo **ED + SD** y **GA-RANSAC + F_2** con 200 ejecuciones para el primer caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.

%	Caso 1 GA + F_2				Caso 1 ED + SD			
	m	M	\bar{x}	n	m	M	\bar{x}	n
0	4.00	4.00	4.00	53	1.00	1.00	1.00	353
20	4.00	4.00	3.99	74	0.00	3.00	1.78	378
40	3.00	4.00	3.92	79	0.00	4.00	1.87	380
60	2.00	4.00	3.95	83	0.00	4.00	2.09	390
80	2.00	4.00	3.87	89	0.00	4.00	1.95	390
100	2.00	4.00	3.84	90	0.00	4.00	2.02	389

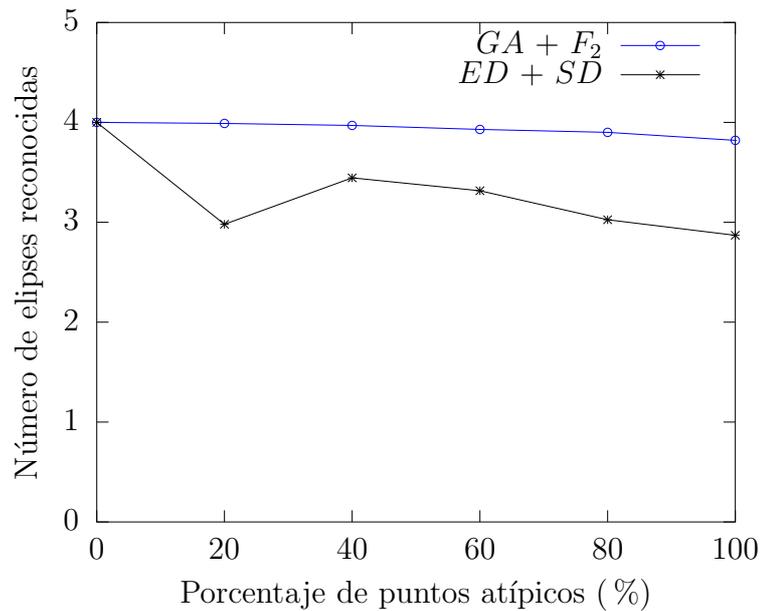


Figura 5.25: Resultados del segundo caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.

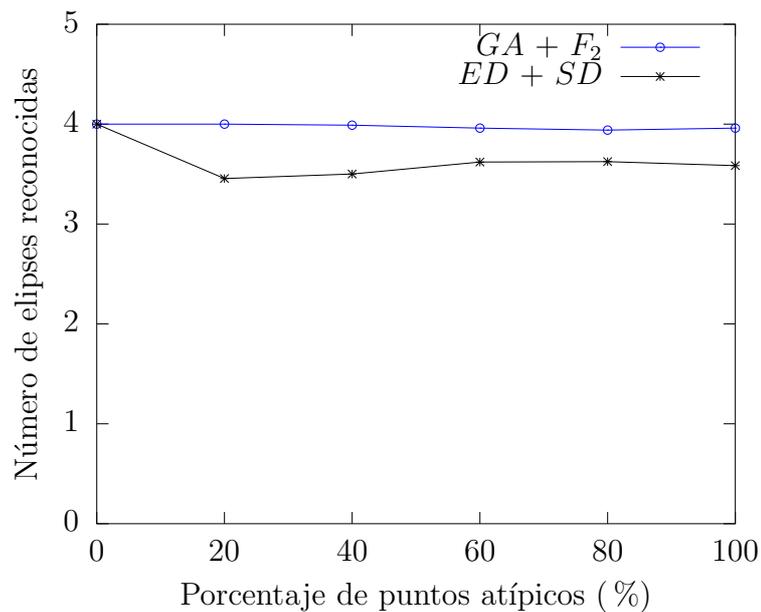


Figura 5.26: Resultados del tercer caso. Las líneas representan el promedio de elipses reconocidas para un conjunto de datos con cuatro elipses inmersas en diferentes niveles de ruido.

5.3. Discusión

ED + SD es una buena técnica pero tiene problemas con el primer caso (elipses disjuntas), como podemos ver en la figura 5.24. Dependiendo de los parámetros de

Tabla 5.18: Resultados del algoritmo **ED + SD** y **GA-RANSAC** + F_2 con 200 ejecuciones para el segundo caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.

	Caso 2 GA + F_2				Caso 2 ED + SD			
%	m	M	\bar{x}	n	m	M	\bar{x}	n
0	4.00	4.00	4.00	72	4.00	4.00	4.00	344
20	3.00	4.00	3.97	72	0.00	4.00	2.98	356
40	3.00	4.00	3.97	78	0.00	4.00	3.44	391
60	3.00	4.00	3.94	79	1.00	4.00	3.31	429
80	3.00	4.00	3.97	83	1.00	4.00	3.02	449
100	2.00	4.00	3.82	88	1.00	4.00	2.87	456

Tabla 5.19: Resultados del algoritmo **ED + SD** y **GA-RANSAC** + F_2 con 200 ejecuciones para el tercer caso; m , M y \bar{x} son el mínimo, máximo y promedio del número de elipses reconocidas; n es el promedio de número de evaluaciones.

	Caso 3 GA + F_2				Caso 3 ED + SD			
%	m	M	\bar{x}	n	m	M	\bar{x}	n
0	4.00	4.00	4.00	90	4.00	4.00	4.00	296
20	4.00	4.00	4.00	75	1.00	4.00	3.46	314
40	4.00	4.00	4.00	78	1.00	4.00	3.50	318
60	3.00	4.00	3.96	80	1.00	4.00	3.62	318
80	2.00	4.00	3.88	82	1.00	4.00	3.62	329
100	3.00	4.00	3.94	88	1.00	4.00	3.58	329

entrada ED + SD no se comporta bien con altos niveles de ruido y puntos atípicos, y necesita información específica sobre las elipses que busca, por otro lado GA-RANSAC no necesita información específica sobre el tamaño y las formas de las elipses.

Se usó la segunda función de aptitud, el número de puntos en la elipse por unidad de su perímetro, porque mostró tener mejor desempeño para resolver problemas más comunes que las otras funciones.

GA-RANSAC obtuvo buenos resultados para los tres casos, y mostró ser un algoritmo robusto incluso en conjuntos de datos con 100 % de ruido y puntos atípicos.

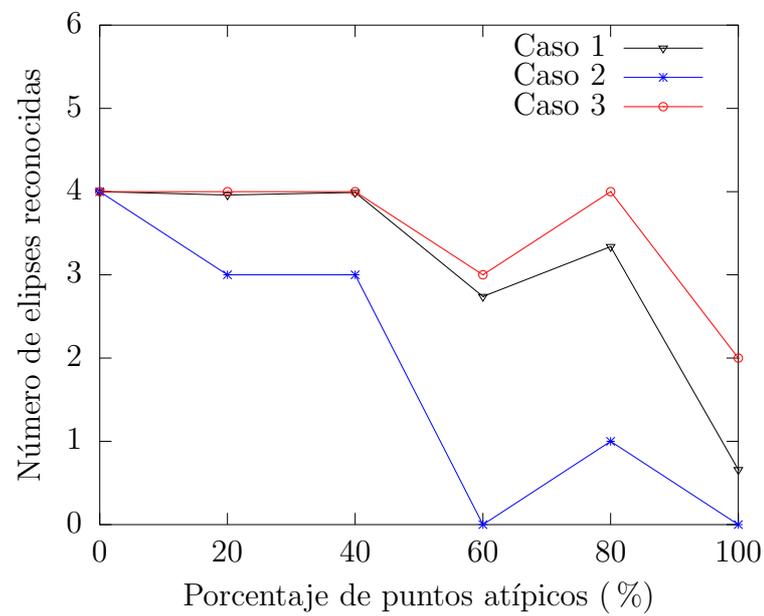


Figura 5.27: Resultados para la heurística sin GA-RANSAC. Las líneas representan el promedio de elipses reconocidas por la heurística.

Capítulo 6

Aplicaciones

Como se mencionó anteriormente existen diversas aplicaciones en las que se utiliza el ajuste de elipses para resolver un problema, en particular en visión es común utilizarlo porque las elipses pueden verse como una proyección de un círculo.

Dentro de las aplicaciones de visión que utilizan el ajuste de elipses, el problema calibración de cámaras puede resolverse usando círculos concéntricos, los cuales en una foto (o video dependiendo de la cámara a calibrar) se pueden encontrar ajustando elipses. A continuación se explicarán algunos métodos de calibración de cámaras, entre los cuales está el método por medio de círculos concéntricos, para mostrar una aplicación de visión en la que se utiliza el ajuste de elipses.

6.1. Calibración de cámaras

La calibración de cámaras es muy importante en visión computacional 3D a fin de que se pueda extraer información de imágenes 2D. Existen muchos trabajos sobre calibración de cámaras ([34, 35, 36, 37, 33] por mencionar algunos), que utilizan diversos métodos para realizar la calibración de cámaras, los cuales se diferencian en como se capturan los datos, o en el tipo proceso (automático o manual) [38].

La calibración de cámaras consiste en determinar los parámetros internos o intrínsecos de una cámara en particular, que son los siguientes:

- Distancias focales α y β
- Factor de distorsión u oblicuidad γ
- Puntos centrales del plano imagen u_0 y v_0

La calibración de cámaras es necesaria, para extraer información del mundo 3D a partir de imágenes 2D, cuya información es útil para diversas aplicaciones ya sean

realidad aumentada, reconocimiento, reconstrucción 3D, seguimiento, etc. Aunque existen técnicas para inferir la información desde imágenes captadas por cámaras no calibradas, a través de procesos iterativos de autocalibración, el proceso de calibración de cámara abre la posibilidad de realizar aplicaciones efectivas en visión.

Cuando se tiene una imagen de un objeto en el mundo real, el objeto está representado en la imagen mediante píxeles, los cuales usualmente no pueden dar información exacta de la distancias en el mundo real del objeto, debido a las transformaciones y proyecciones que sufrió el objeto cuando fue capturado por la cámara para formar la imagen. El objeto en el mundo real está posicionado en un marco de referencia mientras que el objeto en la imagen está en un marco de referencia distinto. Por consiguiente se necesita de ecuaciones para pasar de un marco de referencia a otro, y así tener una relación entre las coordenadas de los puntos en un espacio tridimensional (el mundo real) y las coordenadas en un espacio bidimensional (la imagen).

Como no es posible hacer la relación directamente entre el marco de referencia del mundo real y el marco de referencia en la imagen por las restricciones que conlleva el problema, se usa un marco de referencia intermedio, que se conoce como marco de referencia de la cámara. Entonces, se deben tener ecuaciones para pasar del marco de referencia de la cámara al marco de referencia de la imagen, y ecuaciones para pasar del marco de referencia del mundo real al marco de referencia de la cámara. Todo lo anterior es equivalente a encontrar los parámetros intrínsecos de la cámara, que son los que relacionan el marco de referencia de la imagen con el marco de referencia de la cámara, mientras que los parámetros extrínsecos (rotaciones y traslaciones) son los que relacionan el marco de referencia de la cámara con el marco de referencia del mundo real.

Según Zhang [36] es posible clasificar las técnicas de calibración de cámaras en dos categorías:

- **Calibración fotogramétrica.** Se realiza mediante la observación de patrones cuya geometría en el espacio 3D es conocida con un buen nivel de precisión. Los patrones de calibración normalmente están posicionados en dos o tres planos ortogonales entre ellos. En algunos casos, basta con un único plano, cuya traslación es perfectamente conocida. Este tipo de calibración requiere una configuración elaborada, pero sus resultados son eficientes.
- **Autocalibración.** Este método se basa en el movimiento de la cámara observando una escena estática, a partir de sus desplazamientos y usando únicamente la información de la imagen. La rigidez de la escena impone en general restricciones sobre los parámetros de cámara. Tres imágenes tomadas por una misma cámara con parámetros intrínsecos fijos son suficientes para obtener tanto los parámetros extrínsecos como intrínsecos.

Es posible calcular los parámetros de la cámara directa o indirectamente dependiendo del método que utilicemos para hacer la calibración de la cámara, en los métodos de calibración fotogramétrica se hace directamente si se conoce el centro de la imagen, o indirectamente si no se conoce.

Existen diversos métodos para resolver el problema de calibración de cámaras, como son [34] donde los autores utilizan círculos en el mismo plano como patrón de referencia, en [35, 33] los autores usan círculos concéntricos como patrón de referencia, en [36] el autor utiliza planos como patrón de referencia y en [37] los autores utilizaban planos pero con el centro de un círculo como restricción del problema.

A continuación se describirán algunos métodos para resolver el problema de calibración de cámaras. Siendo el método por círculos concéntricos el más relevante para esta tesis, dado que utiliza ajuste de elipses para resolver el problema de calibración de cámaras.

6.1.1. Calibración de la cámara usando planos

La calibración de cámaras usando planos, es uno de los métodos más comunes para resolver el problema de calibración de cámaras, siendo el trabajo presentado en [36] el más popular de este tipo. El método de Zhang necesita al menos 4 puntos en el mismo plano, un patrón común para realizar la calibración es una cuadrícula como la de la figura 6.1(a) donde tenemos 16 puntos en el mismo plano.

Para llevar a cabo la calibración por este método se necesitan 3 imágenes, como las de la figura 6.1.

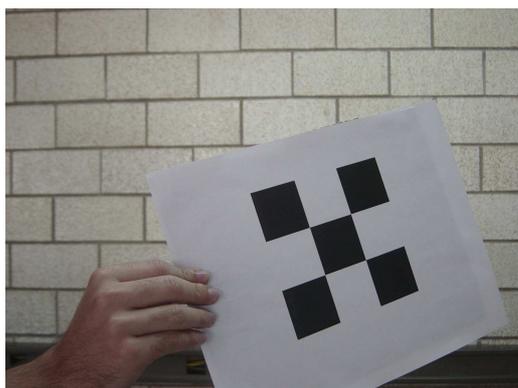
Lo primero que se hace en el método de Zhang es calcular la homografía H (que es la relación que nos permite pasar del marco de referencia de la imagen al marco de referencia de la cámara), la cual es una matriz cuadrada de 3×3 , y se calcula con la siguiente ecuación:

$$H = K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] \quad (6.1)$$

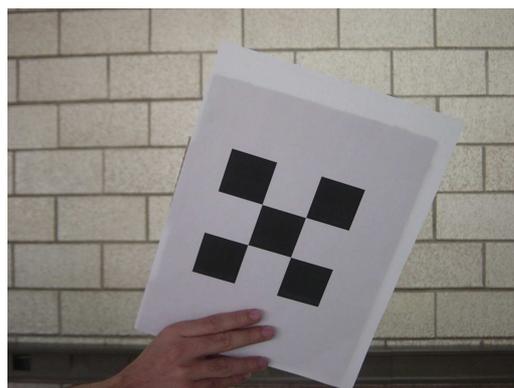
donde K es la matriz de calibración o de parámetros intrínsecos de la cámara, \mathbf{r}_1 y \mathbf{r}_2 son las dos primeras columnas de la matriz de rotación R , y \mathbf{t} es el vector de traslaciones.

La matriz K , está representada de la siguiente forma:

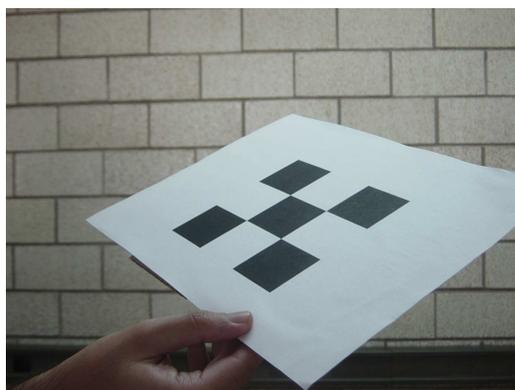
$$K = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.2)$$



(a) Primera imagen del plano



(b) Segunda imagen del plano



(c) Tercera imagen del plano

Figura 6.1: Planos para hacer la calibración por el método de Zhang

donde u_0 y v_0 son el centro de la imagen, α y β las distancias focales y γ el factor de distorsión.

Sin perder generalidad podemos considerar $\alpha = \beta = f$ y $\gamma = 0$ para una cámara con píxeles cuadrados y sin distorsión.

Para calcular la matriz H se usan los puntos de las tres imágenes y los puntos en el mundo real, con un método por descomposición en valores singulares para encontrar cada columna de H .

Después se calcula K a partir de la matriz H , que son los parámetros intrínsecos de la cámara y con la cual podemos calcular la matriz de rotación R y vector de traslación \mathbf{t} , que son los parámetros extrínsecos de la cámara.

En términos generales, este método es bueno para calibrar cámaras, aunque tiene algunos detalles, como la necesidad de buscar las esquinas en los planos, lo cual usualmente se hace manualmente, y para procesos automáticos se necesita agregar un algo-

ritmo para encontrar las esquinas. Además, al no ser un método de auto-calibración, se necesita agregar una pequeña modificación para que realice dicha calibración.

6.1.2. Auto-calibración de la cámara usando cuboides

En [39] se presenta un método para la auto-calibración de cámaras utilizando cuboides como los de la figura 6.2 y que funciona como se explica a continuación.

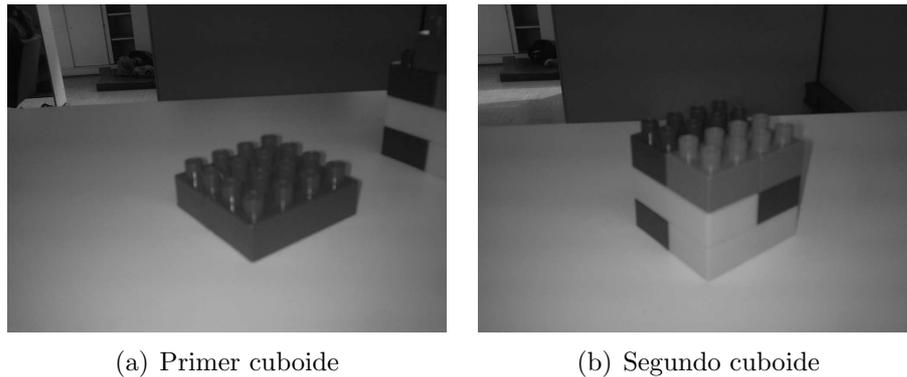


Figura 6.2: Cuboides para la auto-calibración

Un cuboide es un paralelepípedo con ángulos internos de 90° , el cual podemos parametrizar para obtener los valores del tamaño del cuboide a partir de sus lados

$$\tilde{\Delta} = \begin{bmatrix} \frac{l_1}{2} & 0 & 0 & 0 \\ 0 & \frac{l_2}{2} & 0 & 0 \\ 0 & 0 & \frac{l_3}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.3)$$

Los puntos en la imagen $\mathbf{p}_i = [x_i, y_i, 1]^T \mid_{i=1, \dots, 8}$ satisfacen la ecuación:

$$\begin{pmatrix} \rho_1 x_1 & \rho_2 x_2 & \cdots & \rho_8 x_8 \\ \rho_1 y_1 & \rho_2 y_2 & \cdots & \rho_8 y_8 \\ \rho_1 & \rho_2 & \cdots & \rho_8 \end{pmatrix} = \tilde{X} \cdot \begin{pmatrix} 1 & 1 & \cdots & -1 \\ 1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad (6.4)$$

donde $\tilde{X} \in \mathbb{R}^{3 \times 4}$ es la matriz de proyección y define al factor de escala como:

$$\tilde{X} \sim M \cdot \tilde{\Delta} \sim K \cdot [R|t] \cdot \tilde{\Delta}. \quad (6.5)$$

A partir de la cual se puede derivar la siguiente ecuación:

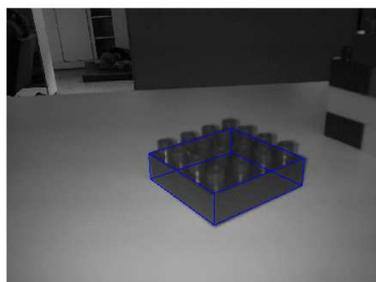
$$X^T \cdot K^{-T} \cdot K^{-1} \cdot X \sim \Delta^T \cdot \Delta \quad (6.6)$$

Después se estiman los parámetros intrínsecos de la cámara con la restricción:

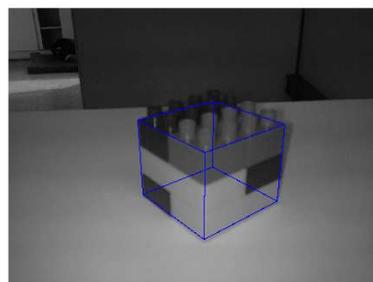
$$\begin{aligned}\omega &= K^{-T} \cdot K^{-1} \\ X_i^T \cdot \omega \cdot X_j &= 0\end{aligned}\tag{6.7}$$

Al tener la matriz de parámetros intrínsecos K , se pueden obtener los parámetros extrínsecos.

Como se dijo antes para llevar acabo este tipo de calibración se necesitan cuboides, en la figura 6.2 pueden verse bloques que sirven para hacer esta calibración, y en la figura 6.3 pueden verse los paralelepípedos formados a partir de los parámetros obtenidos de la cámara.



(a) Primer cuboide



(b) Segundo cuboide

Figura 6.3: Dibujado de los cuboides a partir de los parámetros de la cámara

6.1.3. Calibración de cámaras usando círculos concéntricos

En [33] se presenta un algoritmo para calibración de cámaras usando círculos concéntricos, para realizar este tipo de calibración, se puede utilizar un patrón como el de la figura 6.4.

Para usar el patrón de la figura 6.4, primero hay que obtener las elipses del patrón. Una manera de obtenerlas es usando un umbral¹, para extraer solo el círculo que representa el patrón.

¹Para extraer el círculo se necesita que el círculo sea de color negro y esté en un fondo de color blanco, primero se convierte la imagen de color en una imagen en escala de grises. Después se recorre la imagen píxel por píxel y los valores menores a cierto umbral se toman como blanco, los demás como negro

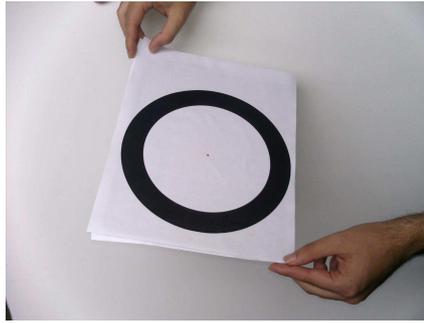
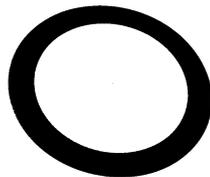
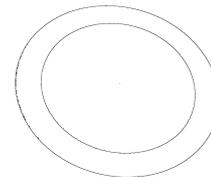


Figura 6.4: Patrón de círculos concéntricos

En la figura 6.5(a) puede observarse el círculo obtenido del patrón, al cual se aplican dos operaciones morfológicas erosión² y después dilatación³, para obtener los bordes del círculo restando la imagen erosionada a la imagen dilatada, los cuales pueden verse en la figura 6.5(b) y forman dos elipses.



(a) Patrón obtenido mediante umbralización



(b) Bordes del círculo

Figura 6.5: Patrón de círculos después de la umbralización y la obtención de sus bordes

El siguiente paso es ajustar las elipses, existen diversas formas de llevar a cabo este paso, el primero es extrayendo y separando las elipses con alguna técnica de visión y después ajustando cada elipse con alguno de los métodos presentados en el capítulo 3, o de una manera más robusta con el método presentado en esta tesis en la sección 4.1.

Una vez que se tienen los parámetros de las elipses, se calcula la homografía, con la que se obtiene la matriz de parámetros intrínsecos de la cámara, tal como se obtuvo en las secciones anteriores. El proceso para obtener la homografía es el que se detalla

²La erosión es el proceso de eliminar los puntos que se encuentran en el borde de un objeto, reduciendo su área

³La dilatación es el proceso de incorporar puntos en el borde de un objeto, incrementando su área

a continuación.

Se deben obtener los parámetros implícitos de las elipses (coeficientes en la ecuación general de las cónicas $[a_0, b_0, c_0, d_0, e_0, f_0]$) a partir de los explícitos $((a, b, x_c, y_c, \alpha))$, con las siguientes ecuaciones:

$$a_0 = a^2 \cdot \text{sen}^2 \alpha + b^2 \cdot \text{cos}^2 \alpha \quad (6.8)$$

$$b_0 = \text{cos} \alpha \cdot \text{sen} \alpha \cdot (b^2 - a^2) \quad (6.9)$$

$$c_0 = b^2 \cdot \text{cos} \alpha \cdot (-y_c \cdot \text{sen} \alpha - x_c \cdot \text{cos} \alpha) - a^2 \cdot \text{sen} \alpha \cdot (x_c \cdot \text{cos} \alpha - y_c \cdot \text{cos} \alpha) \quad (6.10)$$

$$d_0 = b^2 \cdot \text{sen} \alpha + a^2 \cdot \text{cos} \alpha \quad (6.11)$$

$$e_0 = b^2 \cdot \text{sen} \alpha \cdot (-y_c \cdot \text{sen} \alpha - x_c \cdot \text{cos} \alpha) - a^2 \cdot \text{cos} \alpha \cdot (x_c \cdot \text{cos} \alpha - y_c \cdot \text{cos} \alpha) \quad (6.12)$$

$$f_0 = b^2 \cdot (-y_c \cdot \text{sen} \alpha - x_c \cdot \text{cos} \alpha)^2 + a^2 \cdot (x_c \cdot \text{cos} \alpha - y_c \cdot \text{cos} \alpha)^2 - a^2 \cdot b^2 \quad (6.13)$$

Con estos valores se forman las matrices de las cónicas A_1 y A_2 como:

$$A_1 = \begin{pmatrix} a_{01} & b_{01} & c_{01} \\ b_{01} & d_{01} & e_{01} \\ c_{01} & e_{01} & f_{01} \end{pmatrix} \quad A_2 = \begin{pmatrix} a_{02} & b_{02} & c_{02} \\ b_{02} & d_{02} & e_{02} \\ c_{02} & e_{02} & f_{02} \end{pmatrix} \quad (6.14)$$

Se realiza el cálculo de las inversas de A_1 y A_2 , y se multiplica A_1 por la inversa de A_2 para formar la matriz B . Se obtienen los eigenvalores de B , y se multiplican los valores más cercano y alejado de la mediana de eigenvalores con la diferencia de las inversas A_1^{-1} y A_2^{-1} . El valor que resulta se descompone en valores singulares $[USV]$ y se hace la multiplicación $U \cdot \sqrt{S}$, con la que se obtiene la homografía como se muestra en el algoritmo 12.

Después con la homografía se obtiene la matriz de parámetros intrínsecos de la cámara.

Este método regresa buenos resultados, sobre todo con círculos de radio mayor. Tiene el problema de necesitar de un círculo negro en un ambiente donde no haya más negros, porque otros objetos negros pueden generar ruido, que no podrá ajustar un método convencional, por lo que una heurística como la presentada en esta tesis es una buena solución.

Algoritmo 12 Cálculo de la homografía

Entrada: matrices de cónicas A_1 y A_2 **Salida:** Homografía H

$$A_{inv1} = inversa(A_1)$$

$$A_{inv2} = inversa(A_2)$$

$$B = A_{inv2} * A_1$$

$$e = eigenvalores(B)$$

$$m = \min(\text{abs}(\text{mediana}(e) - e))$$

$$[U, S, V] = \text{descomposicion en valores singulares} (m \cdot A_{inv1} - A_{inv2})$$

$$J_1 = U \cdot \sqrt{S}$$

$$m = \max(\text{abs}(\text{mediana}(e) - e))$$

$$[U, S, V] = \text{descomposicion en valores singulares} (m \cdot A_{inv1} - A_{inv2})$$

$$J_2 = U \cdot \sqrt{S}$$

$$H(1, 1) = J_2(1, 1)$$

$$H(1, 2) = J_2(1, 2)$$

$$H(1, 3) = J_1(1, 1)$$

$$H(2, 1) = J_2(2, 1)$$

$$H(2, 2) = J_2(2, 2)$$

$$H(2, 3) = J_1(2, 1)$$

$$H(3, 1) = J_2(3, 1)$$

$$H(3, 2) = J_2(3, 2)$$

$$H(3, 3) = J_1(3, 1)$$

6.2. Aplicación de la calibración de cámaras en realidad aumentada

En [31] se presenta un trabajo de realidad aumentada, donde un bloque del sistema necesita la calibración de cámara para poder tener las características intrínsecas y extrínsecas de la cámara y así poder generar objetos virtuales dentro la imagen o video, de una manera coherente. El trabajo presenta dos métodos de calibración de cámara, uno con un patrón de círculos en un plano y otro con círculos concéntricos, su finalidad es construir una herramienta para la creación de sistemas de realidad aumentada con el uso de video digital.

La herramienta que se propuso consta de la captura de video, procesamiento de los marcos de video, reconocimiento de patrones, calibración de cámaras y despliegado de un objeto virtual. Para la captura de video en el trabajo, modificaron la librería DVGrab ⁴, con la que se obtenía el video por el puerto *fireware*.

Después de tener los marcos de video, se procesaba cada marco dependiendo del tipo de patrón que se utilizaría para la calibración de cámara. En caso de utilizar un patrón blanco y negro como el de la sección 6.1.3, el procedimiento era convertir el marco a tono de grises, suavizar la imagen, detectar los bordes con mascararas de Sobel y finalmente binarizar el marco. En caso de utilizar un patrón a color, solo se hace la segmentación por color del marco, el cual se explicará a detalle en la sección 6.2.1.

Una vez procesados los marcos, se extraen las figuras geométricas requeridas para la calibración de cámaras (en este caso círculos concéntricos que forman elipses cuando están proyectados), con el algoritmo ED + SD de la sección 3.2.2. Después se procedía a hacer la calibración de cámara, extrayendo los parámetros extrínsecos e intrínsecos de la cámara.

Por último se lleva a cabo la adición de objetos al video, esto es, se grafican los objetos con *OpenGL* ⁵. Todo trabajando en una interfaz en *Qt* ⁶.

6.2.1. Círculos concéntricos para calibración de la cámara

Como se mencionó en la sección 6.1.3, el método de calibración de cámaras se puede realizar utilizando círculos concéntricos como patrón. En el trabajo presentado en [31], se usaban dos tipos de patrones por círculos concéntricos, uno blanco y negro, y otro a color como los patrones de la figura 6.6.

⁴<http://www.kinodv.org/>

⁵<http://www.opengl.org/>

⁶<http://qt.nokia.com/>



(a) Primer patrón

(b) Segundo patrón

Figura 6.6: Patrón de círculos concéntricos rojo

Para procesar el patrón se usa segmentación por color, dicho color puede variar, aunque en este caso es el rojo, el procedimiento consiste en convertir los píxeles en el modelo de color RGB (del inglés *Red, Green, Blue*) a HSI (del inglés *Hue, Saturation, Intensity*), con lo que se extraen los objetos de un solo color en particular. El modelo HSI, define un modelo de color en términos de su matiz (H), saturación (S) e intensidad (I).

Del modelo HSI solo es de interés los parámetros H y S, para pasar de términos RGB a HSI se utilizan las siguientes ecuaciones:

$$r = \frac{R}{R + G + B}, g = \frac{G}{R + G + B}, b = \frac{B}{R + G + B} \quad (6.15)$$

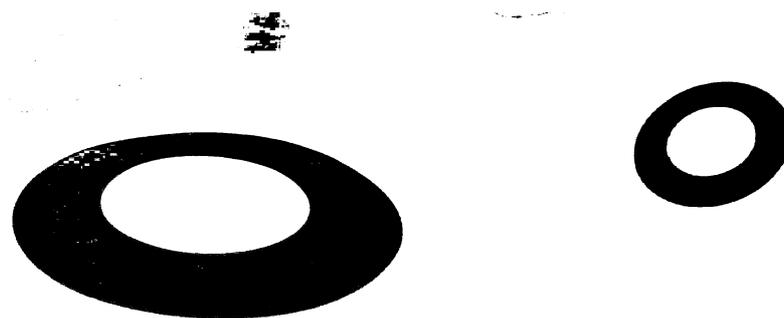
$$h = \begin{cases} \cos^{-1} \left[\frac{0.5 \cdot [(r-g) + (r-b)]}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right] & \text{si } b \leq g \\ 2\pi - \cos^{-1} \left[\frac{0.5 \cdot [(r-g) + (r-b)]}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right] & \text{si } b > g \end{cases} \quad (6.16)$$

$$s = 1 - 3 \cdot \min(r, g, b) \quad s \in [0, 1] \quad (6.17)$$

Una vez que se tienen los parámetros H y S del modelo HSI, se hace la binarización de la imagen, para tener solo los objetos del color rojo (o el color que se elija). Para binarizar se toman los píxeles con valor de H entre 300° y 360° .

En el algoritmo 13, se puede ver el procedimiento para hacer la segmentación por color, primero obteniendo los parámetros H y S del modelo RGB, y después binarizando la imagen.

El resultado de la segmentación por color para los patrones de la figura 6.6, se puede ver en la figura 6.7.

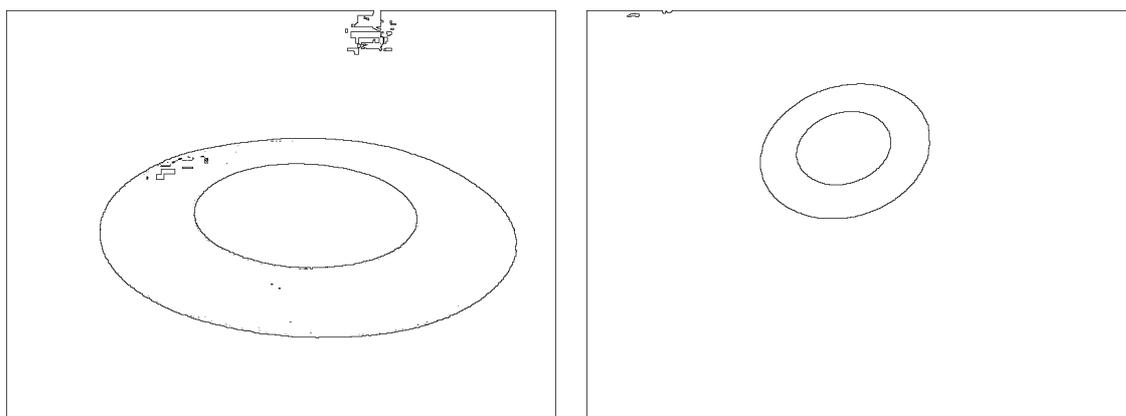


(a) Primer patrón

(b) Segundo patrón

Figura 6.7: Segmentación de las figuras

Después de obtener solo los píxeles de los objetos de color rojo, se procede a hacer la extracción de las elipses. Primero se extraen los bordes de los objetos, restando la imagen erosionada a la imagen original, quedando solo los bordes del objeto como en la figura 6.8.



(a) Primer patrón

(b) Segundo patrón

Figura 6.8: Mapa de bits obtenido para las figuras

Una vez extraídos los bordes de los objetos, se hace el ajuste de elipses en los puntos que conforman los bordes. Para realizar el ajuste, en [31], se usó ED + SD, que como se vio en el capítulo 5 puede fallar en algunos casos, dependiendo del número de puntos atípicos. En la figura 6.8(a) está un ejemplo de elipses con puntos atípicos, dependiendo del fondo en donde se encuentre el patrón es posible tener un mayor número de puntos atípicos.

ED + SD, se puede sustituir por el algoritmo propuesto en esta tesis GA-RANSAC que es un poco más robusto que ED + SD, además de necesitar menos parámetros de

Algoritmo 13 Segmentación por color**Entrada:** Elementos del marco en formato RGB $im.R, im.G$ y $im.B$ **Entrada:** Base y altura del marco $base, altura$ **Salida:** Marco binarizado hsi

```

for 1  $\leftarrow$  0 to  $base * altura$  do
     $suma \leftarrow im.R[i] + im.G[i] + im.B[i]$ 
     $r \leftarrow im.R[i] / suma$ 
     $g \leftarrow im.G[i] / suma$ 
     $b \leftarrow im.B[i] / suma$ 
    if  $r \leq b$  then
         $h \leftarrow \cos^{-1} \left[ \frac{0.5 \cdot [(r-g) + (r-b)]}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right]$ 
    else
         $h \leftarrow 2 * \pi - \cos^{-1} \left[ \frac{0.5 \cdot [(r-g) + (r-b)]}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right]$ 
     $s = 1 - 3 * \min(r, g, b)$ 
    if  $(h > 300 \&\& h < 360) \&\& (s > 0.2)$  then
         $hsi[i] \leftarrow 255$ 
    else
         $hsi[i] \leftarrow 0$ 

```

entrada relacionados con las elipses. Las elipses extraídas para los patrones de la figura 6.6 con GA-RANSAC, se pueden ver en las figuras 6.9(a) y 6.9(b) en color verde.



(a) Elipses obtenidas en la figura 1



(b) Elipses obtenidas en la figura 2

Figura 6.9: Elipses obtenidas con GA-RANSAC

Finalmente, con las elipses extraídas, se puede obtener la homografía y la matriz de parámetros intrínsecos de la cámara como en la sección 6.1.3.

Capítulo 7

Conclusiones

En esta tesis se propuso el método robusto GA-RANSAC para ajustar múltiples elipses en un conjunto de datos. Este método básicamente introduce un algoritmo genético para seleccionar el mejor subconjunto de cinco puntos, o el llamado conjunto de consenso en RANSAC, que mejor representa a una elipse dentro de un conjunto de puntos. Con las pruebas realizadas, se mostró que la propuesta presentada es más robusta que el método de RANSAC + MMC y el método ED + SD presentado en [32].

En el método presentado se propusieron tres funciones de aptitud que mostraron ser adecuadas para el problema de ajuste de elipses y robustas a los puntos atípicos. Sin embargo, en nuestras pruebas la segunda función de aptitud, que mide el número de puntos por unidad de perímetro de la elipse, mostró el mejor desempeño.

La primera función de aptitud propuesta –número de puntos en una zona sobre el perímetro de la elipse– tenía problemas pues encontraba elipses falsas que rodeaban a las elipses correctas, mientras que la segunda función de aptitud –número de puntos en la elipse– arreglaba este problema, al tomar en cuenta el perímetro de la elipse. La tercera función de aptitud –suma de distancias de Hausdorff– mostró menor desempeño en comparación con las otras dos, además de necesitar más tiempo de ejecución para resolver el problema.

El algoritmo GA-RANSAC tiene la ventaja con respecto al algoritmo ED + SD, de no necesitar parámetros de entrada implícitos relacionados con las elipses, como son los límites de los parámetros de la elipse $[a, b, x_c, y_c, \alpha]$, pues utiliza ajuste algebraico para calcular las elipses candidatas, en vez de proponer parámetros aleatorios de elipses candidatas.

El algoritmo genético, se introdujo con la intención de mejorar la búsqueda de la solución. Como se mostró en el capítulo 5, haciendo la búsqueda de la solución, con una búsqueda puramente aleatoria usando el número promedio de evaluaciones que se usaron en el algoritmo genético, se obtienen resultados menos eficaces que los del algoritmo con GA-RANSAC, lo cual muestra la mejora en el desempeño del procedi-

miento al usar un algoritmo genético.

En el trabajo se propuso un operador de búsqueda local para el algoritmo genético, con el objetivo de mejorar la búsqueda de la solución, sin embargo el operador propuesto no mostró muchas mejoras y aumentaba un poco el tiempo de ejecución del programa.

Para la extracción automática de múltiples elipses, se propuso el método de la sección 4.4, que funciona para ciertos conjuntos de datos, aunque necesita de conocimiento empírico sobre las elipses a buscar, para dar mejores resultados.

La eficacia del método presentado puede incrementarse para problemas específicos dependiendo del contexto de aplicación. La principal contribución de esta tesis es la detección de múltiples elipses sin la necesidad de conocimiento específico sobre cada elipse en los datos de entrada.

En términos generales el trabajo presentado en esta tesis es un método robusto para el ajuste de múltiples elipses inmersas en ruido, con resultados mejores a los métodos RANSAC + MMC y ED +SD. Aunque tiene la desventaja de tener un tiempo de ejecución costoso en comparación con los métodos deterministas para el ajuste de elipses.

En cuanto a las aplicaciones que puede tener el algoritmo presentado, existen diversas aplicaciones donde puede ser utilizado. Como se mostró en el capítulo 6, una de las utilidades que tiene el ajuste de elipses, es para la calibración de cámaras, que a su vez tiene aplicaciones en otras áreas como lo son la realidad aumentada.

El algoritmo presentado es robusto, aunque no inmune a los puntos atípicos, por lo cual es una buena opción en la calibración de cámaras por círculos concéntricos. Por otro lado es un poco costoso en términos de tiempo de ejecución, así que para aplicaciones de tiempo real que demanden un algoritmo rápido es mejor usar algoritmos deterministas, sin embargo sí es una excelente opción en aplicaciones que necesiten de mayor robustez.

Este trabajo de tesis produjo el siguiente artículo:

- César Cruz-Díaz, Luis Gerardo de la Fraga and Oliver Schütze. Fitness Function Evaluation for the Detection of Multiple Ellipses Using a Genetic Algorithm. *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2011)*, Merida City, Mexico, October 26 - 28, 2011, pp. 946-951.

Trabajo a futuro

El trabajo a futuro que se puede realizar es el siguiente:

- Se puede paralelizar el algoritmo, para buscar reducir el tiempo de ejecución del algoritmo, pues los algoritmos genéticos son altamente paralelizables.
- Se puede probar la segunda función de aptitud con otras heurísticas de optimización y ver si alguna resuelve el problema en un número menor de evaluaciones.
- El trabajo presentado puede mejorar significativamente, con la integración de alguna técnica de búsqueda local especializada. Con lo cual se esperaría una reducción en el tiempo de ejecución así como una aplicación exitosa en datos con gran cantidad de ruido y puntos atípicos.
- Proponer un método para la búsqueda automática de múltiples elipses, más efectivo que el presentado en la sección 4.4.
- Realizar más pruebas con datos en circunstancias diferentes a las presentadas.
- Hacer la comparación con otros métodos que existan para resolver el problema de ajustes de elipses.

Bibliografía

- [1] N. Greggio, L. Manfredi, C. Laschi, P. Dario, and M.C. Carrozza. Robotcub implementation of real-time least-square fitting of ellipses. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 174–181, 2008.
- [2] Richard Y. D. Xu and Jesse S. Jin. Camera control and multimedia interaction using individual object recognition. *Journal of Multimedia*, 2(3):77–85, 2007.
- [3] G. Ososkov, I. Silin, and N. Chernov. Robust fitting of ellipses to non-complete and contaminated data. *Czechoslovak Journal of Physics*, 50:347–354, 2000.
- [4] Sung Joon Ahn, Wolfgang Rauh, and Hans Jurgen Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303, 2001.
- [5] N. Chernov and C. Lesort. Least squares fitting of circles. *Journal of Mathematical Imaging and Vision*, 23:239–252, 2005.
- [6] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480, May 1999.
- [7] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6:59–70, 1991.
- [8] R.Y. Da Xu and M. Kemp. Fitting multiple connected ellipses to an image silhouette hierarchically. *Image Processing, IEEE Transactions on*, 19(7):1673–1682, july 2010.
- [9] L.G. de la Fraga and G.M.L. Dominguez. Robust fitting of ellipses with heuristics. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, 2010.
- [10] Watcharin Kaewapichai and Pakorn Kaewtrakulpong. Robust ellipse detection by fitting randomly selected edge patches. *World Academy of Science, Engineering and Technology*, 48(5):30–33, December 2008.

- [11] Armando Manuel Fernandes. Detection of a large number of overlapping ellipses immersed in noise. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, pages 1–10. Springer-Verlag, 2008.
- [12] Jie Yao, Nawwaf Kharma, and Peter Grogono. A multi-population genetic algorithm for robust and fast ellipse detection. *Pattern Analysis and Applications*, 8:149–162, 2005.
- [13] V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 3rd edition, 1994.
- [14] Otto Bretscher. *Linear Algebra with Applications*. Prentice Hall, 3rd edition, July 2004.
- [15] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for Non-Linear Least Squares Problems*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2nd edition, 2004.
- [16] Victor Ayala-Ramirez, Carlos H. Garcia-Capulin, Arturo Perez-Garcia, and Raul E. Sanchez-Yanez. Circle detection on images using genetic algorithms. *Pattern Recognition Letters*, 27(6):652–657, 2006.
- [17] C.A. Basca, M. Talos, and R. Brad. Randomized hough transform for ellipse detection with result clustering. In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, volume 2, pages 1397–1400, 2005.
- [18] Y.C. Cheng. The probabilistic hough transform with localized search guided by evidence clusters. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 16–20, March 2006.
- [19] Z. Cheng and Y. Liu. Efficient technique for ellipse detection using restricted randomized hough transform. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 714–718, 2004.
- [20] Joon H. Han, LaszloT. Koczy, and Timothy Poston. Fuzzy hough transform. *Pattern Recognition Letters*, 15(7):649–658, 1994.
- [21] E. R. Davies. Finding ellipses using the generalised Hough transform. *Pattern Recognition Letters*, 9(2):87–96, February 1989.
- [22] Luis de la Fraga, Israel Silva, and Nareli Cruz Cortés. Euclidean distance fit of conics using differential evolution. In Stefano Cagnoni, editor, *Evolutionary Image Analysis and Signal Processing*, volume 213 of *Studies in Computational Intelligence*, pages 171–184. Springer Berlin / Heidelberg, 2009.
- [23] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

- [24] P. Faber and R.B. Fisher. Pros and cons of euclidean fitting. In Bernd Radig and Stefan Florczyk, editors, *Pattern Recognition*, volume 2191 of *Lecture Notes in Computer Science*, pages 414–420. Springer Berlin Heidelberg, 2001.
- [25] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [26] Ender Ozcan and Chilukuri K. Mohan. Simulated annealing and genetic algorithms for partial shape matching. In *In Proceedings of the Eleventh International Symposium on Computer and Information Sciences*, pages 173–181, 1996.
- [27] Radim Halir and Jan Flusser. Numerically stable direct least squares fitting of ellipses, 1998.
- [28] Fred L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9(1):56–71, 1979.
- [29] J. Kennedy and R. Eberhart. Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 4:1942–1948 vol.4, November 1995.
- [30] Margarita Reyes-sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE RESEARCH*, 2(3):287–308, 2006.
- [31] Rosa A. Vázquez del Angel. Sistema de desarrollo para aplicaciones de realidad aumentada. Master’s thesis, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, 2010.
- [32] Gustavo M. López-Domínguez. Ajuste robusto usando heurísticas. Master’s thesis, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, 2010.
- [33] Jun-Sik Kim, P. Gurdjos, and In-So Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(4):637–642, 2005.
- [34] Qian Chen, Haiyuan Wu, and Toshikazu Wada. Camera calibration with two arbitrary coplanar circles. In *Proc. European Conf. Computer Vision*, pages 521–532. Inc, 2004.
- [35] Francisco Abad, Emilio Camahort, and Roberto Vivó. Camera calibration using two concentric circles. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 3211 of *Lecture Notes in Computer Science*, pages 688–696. Springer Berlin / Heidelberg, 2004.

- [36] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, nov 2000.
- [37] Pierre Gurdjos, Alain Crouzil, René Payrissat, and Université Paul Sabatier. Another way of looking at plane-based calibration: The centre circle constraint. In *Proc. ECCV '02*, pages 252–266. Springer, 2002.
- [38] Diego Aracena Pizarro, Pedro Campos, and Clésio Luis Tozzi. Comparación de técnicas de calibración de cámaras digitales. *Revista Facultad de Ingeniería-Universidad de Tarapacá*, 13:57–67, 04 2005.
- [39] Luis de la Fraga and Oliver Schütze. Direct calibration by fitting of cuboids to a single image using differential evolution. *International Journal of Computer Vision*, 81:119–127, 2009.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentara el **Sr. César Cruz Díaz**, declaramos que hemos revisado la tesis titulada:

Ajuste robusto de múltiples elipses usando algoritmos Genéticos

Y consideramos que cumple con los requisitos para obtener el Grado de Maestría en Ciencias en Computación.

Atentamente,

Dr. Luis Gerardo de la Fraga

Dr. Oliver Schütze

Dra. Nareli Cruz Cortés

Dra. Xiaou Li Zhang