



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

**Departamento de Computación**

## **Ajuste robusto usando heurísticas**

Tesis que presenta

**Gustavo Miguel López Domínguez**

para obtener el Grado de

**Maestro en Ciencias en Computación**

Director de la Tesis

**Dr. Luis Gerardo de la Fraga**

México, D.F.

Octubre 2010



# Resumen

---

El ajuste de formas geométricas es un problema importante en Visión por Computadora que tiene aplicación en muchos campos de la ciencia y la ingeniería, principalmente en aquellos donde es necesario extraer y analizar información útil de una imagen observada, de un conjunto de imágenes o de una secuencia de éstas.

Cuando se trabaja con imágenes reales es muy común encontrar puntos atípicos en ellas. Éste tipo de puntos se encuentran en total desacuerdo con el modelo presentado inicialmente, afectando considerablemente el proceso de ajuste de la forma geométrica que se desea extraer.

El propósito de esta tesis es realizar un método de ajuste robusto que utilice la minimización de la suma de las distancias perpendiculares. El método debe ser capaz de extraer varias de formas geométricas (líneas, círculos, elipses, triángulos y cuadriláteros) presentes en un mismo conjunto de puntos.

El método propuesto aborda un problema de ajuste no lineal que se resolvió utilizando la meta heurística llamada evolución diferencial. Se compararon los resultados obtenidos contra la heurística más representativa de la literatura (MMC + RANSAC), la cual utiliza la minimización de la mediana de las distancias cuadráticas (MMC) complementado con una técnica de muestreo aleatorio (RANSAC), con la finalidad de demostrar que se ha conseguido un algoritmo de ajuste que proporciona un punto de ruptura de más del 50%, v.g., que es capaz de extraer las formas geométricas propuestas aún cuando el conjunto de puntos conocidos contenga más del 50% de puntos atípicos.



# Abstract

---

The fitting of geometrical forms is an important task in Computer Vision that has applications in many fields within science and engineering, especially in those in which is necessary to extract relevant information from an observed images, a set of image or a sequence of them.

When working with real images is very common to find outliers in them. These points are in total disagreement with the initial model significantly affecting the fitting of the shape that we want to extract.

The aim of this thesis is to propose a robust fitting method based on the minimization of the sum of orthogonal distances. The method should be able to extract several geometric shapes (lines, circles, ellipses, triangles and quadrilaterals) present in the same set of points.

The proposed method deals with a non-linear fitting problem that was solved using a metaheuristic called differential evolution. Results were compared against the most representative heuristic (LMS + RANSAC). This heuristic uses the least median of squares (LMS) supplemented with a random sampling technique (RANSAC), in order to show that we have achieved a fitting algorithm with a break point over 50%, i.e., it can extract the proposed geometric shapes even when the known set of points contains more than 50% of outliers.



# Agradecimientos

---

*A mis padres, Miguel López Hernandez y Amelia Domínguez Ortiz por amarme y apoyarme incondicionalmente en cada una de mis decisiones.*

*A mi novia, Rosita por ser siempre para mí una fuente de amor e inspiración que me llevó a concluir este trabajo.*

*A mi director, el Dr. Luis Gerardo de la Fraga por haber sido un director excepcional y el mejor profesor que he conocido, gracias.*

*A Sofia Reza, por sus consejos en los momentos más difíciles, pero sobre todo por contagiarnos siempre toda su felicidad.*

*Al CINVESTAV, por haber sido más que un segundo hogar y fuente interminable de conocimiento.*

*Al CONACyT, por el apoyo económico que me proporcionó, sin el cual no habría sido posible concluir este trabajo, por su interés en la actividad científica de nuestro país y por el apoyo que proporcionó a los proyectos 80965 y 60240.*





# Índice general

---

Índice de figuras	x
Índice de tablas	xii
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	3
1.2. Objetivo de la tesis . . . . .	4
1.3. Organización de la tesis . . . . .	4
<b>2. Marco teórico</b>	<b>7</b>
2.1. Método de los mínimos cuadrados . . . . .	8
2.2. Método de los mínimos valores absolutos ( $L_1$ ) . . . . .	11
2.3. Estimadores M . . . . .	12
2.4. Método de la mínima mediana de los residuales cuadráticos (MMC) . . . . .	13
2.5. Estimador de muestreo aleatorio (RANSAC) . . . . .	16
2.6. Ajuste según la suma de las distancias ortogonales . . . . .	17
2.7. Sumario . . . . .	18
<b>3. Ajuste de líneas</b>	<b>19</b>
3.1. Descripción del problema . . . . .	19
3.1.1. Ajuste algebraico . . . . .	21
3.1.2. Ajuste según la suma de las distancias ortogonales . . . . .	22
3.2. Ajuste MMC y RANSAC . . . . .	25
3.3. Suma de las distancias ortogonales y evolución diferencial . . . . .	28
3.4. Extracción de varias líneas . . . . .	32
3.5. Pruebas . . . . .	33
3.6. Sumario . . . . .	35
<b>4. Ajuste de elipses</b>	<b>43</b>
4.1. Descripción del problema . . . . .	43
4.1.1. Ajuste algebraico . . . . .	44
4.1.2. Ajuste según la suma de las distancias ortogonales . . . . .	46
4.2. Ajuste MMC y RANSAC . . . . .	47
4.3. Suma de las distancias ortogonales y evolución diferencial . . . . .	49

4.4.	Extracción de varias elipses . . . . .	51
4.5.	Pruebas . . . . .	52
4.6.	Sumario . . . . .	53
<b>5.</b>	<b>Ajuste de triángulos</b>	<b>61</b>
5.1.	Descripción del problema . . . . .	61
5.2.	Cálculo de las distancias perpendiculares . . . . .	61
5.3.	Solución al problema de optimización . . . . .	64
5.4.	Extracción de varios triángulos . . . . .	70
5.5.	Pruebas . . . . .	70
5.6.	Sumario . . . . .	73
<b>6.</b>	<b>Ajuste de cuadriláteros</b>	<b>75</b>
6.1.	Descripción del problema . . . . .	76
6.2.	Cálculo de las distancias perpendiculares . . . . .	76
6.3.	Solución al problema de optimización . . . . .	77
6.4.	Extracción de varios cuadriláteros . . . . .	78
6.5.	Pruebas . . . . .	79
6.6.	Sumario . . . . .	81
<b>7.</b>	<b>Detección de círculos y elipses utilizando la evolución diferencial</b>	<b>83</b>
7.1.	Descripción del problema . . . . .	83
7.1.1.	Ajuste de círculos . . . . .	84
7.1.2.	Ajuste de elipses . . . . .	85
7.2.	Algoritmos . . . . .	86
7.2.1.	KRANSAC . . . . .	86
7.3.	Pruebas y resultados . . . . .	87
7.4.	Sumario . . . . .	93
<b>8.</b>	<b>Conclusiones</b>	<b>95</b>
8.1.	Trabajo futuro . . . . .	97

# Índice de figuras

---

1.1.	Distancia (o residual) ortogonal. . . . .	3
2.1.	(a) Residual en $y$ . (b) Residual en $x$ . (c) Residual en ambos. (d) Residual ortogonal. . . . .	9
2.2.	(a) Conjunto de cinco puntos que definen correctamente la línea. (b) Mismo conjunto de puntos presentado en (a) pero con un punto atípico en la dirección de $y$ . El ajuste de la línea se realizó utilizando la diferencia de valores (los residuales) en $y$ . . . . .	11
2.3.	Resultado entregado $L_1$ en presencia de un punto atípico en dirección de $y$ . . . . .	12
2.4.	(a) presenta un conjunto de cinco puntos que definen una línea. (b) Mismo conjunto de puntos presentado en (a) pero con un punto atípico en la dirección de $x$ . . . . .	12
2.5.	(a) Presenta un conjunto de cinco puntos con un punto contaminado en la dirección de $y$ . (b) Presenta un conjunto de seis puntos con dos puntos atípicos en la dirección de $x$ . . . . .	15
3.1.	Modelo utilizado de la línea. . . . .	20
3.2.	Caso de ejemplo. . . . .	21
3.3.	Modelo utilizado de la línea para el caso 1. . . . .	22
3.4.	Modelo utilizado de la línea para el caso 2. . . . .	23
3.5.	Modelo utilizado de la línea para representación vectorial. . . . .	24
3.6.	Modelo utilizado para la representación del problema en la Evolución diferencial. . . . .	28
3.7.	Caso de ejemplo para explicar la decodificación de las variables en la Evolución diferencial. . . . .	29
3.8.	Resultados de la primera prueba con ruido Gaussiano sobre los puntos de la recta. . . . .	36
3.9.	Resultados de la segunda prueba con ruido Laplaciano sobre los puntos de la recta. . . . .	37
3.10.	Resultados de la tercera prueba con ruido Gaussiano sobre los puntos de la recta. . . . .	38
3.11.	Resultados de la cuarta prueba con ruido Laplaciano sobre los puntos de la recta. . . . .	39

3.12. Extracción de varias líneas con $b_y$ fijo. . . . .	40
3.13. Extracción de varias líneas con el ángulo de inclinación respecto a $x$ y $y$ fijo. . . . .	40
3.14. Ajuste obtenido de los experimentos 1 y 3. . . . .	41
3.15. Ajuste obtenido de los experimentos 2 y 4. . . . .	41
3.16. Ajuste obtenido de los experimentos 5 y 6. . . . .	42
4.1. Modelo utilizado de la elipse. . . . .	44
4.2. Conjunto de 90 puntos que forman una elipse con parámetros $(100,40,300,200,40^0)$ con cinco puntos atípicos, Ajuste según la suma de las distancias ortogonales (línea continua) y ajuste según la suma de las distancias algebraicas (línea punteada). . . . .	45
4.3. Resultados de la primera prueba. . . . .	55
4.4. Ajuste resultante del primer experimento. . . . .	56
4.5. Resultados de la segunda prueba. . . . .	57
4.6. Ajuste resultante del segundo experimento. . . . .	58
4.7. Resultados de la tercera prueba. . . . .	58
4.8. Ajuste resultante del tercer experimento. . . . .	59
5.1. (a) Distancias del punto $\mathbf{p}$ a los vértices de un triángulo. (b) Distancias perpendiculares del punto $\mathbf{p}$ a los tres lados del triángulo. . . . .	62
5.2. (a) Caso de ejemplo en el que se tiene solo una distancia perpendicular válida, la cual es mucho mayor que la distancia al vértice. (b) Ejemplo en el que no existen distancias perpendiculares válidas. La distancia considerada es la menor del punto a los vértices del triángulo. . . . .	63
5.3. Distancia perpendicular de $\mathbf{p}$ al segmento de recta $\overline{\mathbf{v}_2\mathbf{v}_3}$ . . . . .	63
5.4. Ajuste utilizando el problema de optimización propuesto en (5.10). . . . .	65
5.5. Ajuste utilizando el problema de optimización propuesto en (5.10). . . . .	66
5.6. (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4 . . . . .	72
6.1. (a) Distancias del punto $\mathbf{p}$ a los vértices de un cuadrilátero. (b) Distancias perpendiculares del punto $\mathbf{p}$ a los cuatro lados del cuadrilátero. . . . .	76
6.2. (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4 . . . . .	80
7.1. (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4. . . . .	92
7.2. (a) Prueba 5 con 50% de puntos atípicos. (b) Prueba 6 con 50% de puntos atípicos y ruido Gaussiano sobre los puntos de la elipse con $\sigma = 2$ . (c) Prueba 7 con 70% de puntos atípicos. (d) Prueba 8 con 50% de puntos atípicos . . . . .	93

# Índice de cuadros

---

5.1. Se muestra el mínimo, máximo y la media de los triángulos extraídos exitosamente en cada una de las pruebas presentadas, así como la media y desviación estándar del número de evaluaciones de la función objetivo.	73
6.1. Resultados de la extracción de cuadriláteros con SD + ED, se muestra el mínimo, máximo y media de cuadriláteros extraídos exitosamente, así como la media y desviación estándar del número de evaluaciones de la función objetivo. . . . .	81
7.1. Estadísticas de los resultados obtenidos por KRANSAC, SD + ED y RHT para la extracción de círculos. Muestra el mínimo, máximo y media de los círculos extraídos exitosamente. También se puede observar la media y desviación estándar del número de evaluaciones de la función objetivo. . . . .	90
7.2. Estadísticas de los resultados obtenidos por KRANSAC y SD + ED para la extracción de elipses. Muestra el mínimo, máximo y media de las elipses extraídas exitosamente. También se puede observar la media y desviación estándar del número de evaluaciones de la función objetivo. Para el caso de HT sólo se muestra el número de extracciones exitosas y las iteraciones, ya que se ejecuta sólo una vez. . . . .	91



# Capítulo 1

## Introducción

---

El ajuste de formas geométricas es un problema importante dentro de Visión por Computadora que tiene aplicación en muchos campos de la ciencia y la ingeniería, principalmente en aquellos donde es necesario extraer y analizar información útil de una imagen observada, de un conjunto de imágenes o de una secuencia de éstas. Para realizar la extracción de dicha información se utilizan conceptos estadísticos como la regresión y el ajuste de modelos.

El método de ajuste más antiguo fue ideado por Gauss y proporcionó las bases para lo que hoy se conoce como análisis de mínimos cuadrados (MC). Este método fue adoptado por tradición y facilidad de implementación, principalmente porque surgió en una época en la que no se contaba con el poder y las herramientas de cómputo con las que se cuentan actualmente [1].

La idea principal de MC consiste en minimizar la suma de los valores cuadráticos de error (residual) que existe entre el conjunto de datos conocido y un modelo propuesto. Una vez resuelto el problema de ajuste, el método entrega el modelo que más se parece al conjunto de datos original. MC presenta una clara desventaja al utilizar los residuales cuadráticos, ya que si dentro del conjunto de datos conocido existe un elemento que se encuentre lo suficientemente alejado del modelo original, éste repercutirá en gran medida sobre el ajuste que MC proporcione, debido a que el elemento contribuirá en gran medida al error que se trata de minimizar, proporcionando un ajuste sesgado en dirección del elemento indeseado. Este tipo de elementos se conocen en la literatura como datos atípicos.

Los datos atípicos son muy comunes al trabajar con datos reales. Éstos provienen de diversas fuentes, p.ej. errores de captura, mal posicionamiento de puntos decimales, errores de almacenamiento, errores de transmisión etc. En la mayoría de los casos este tipo de datos pasan desapercibidos, debido a que resulta muy complicado realizar un análisis preliminar. Muchas veces los datos son procesados automáticamente y no se tiene acceso a ellos [1].

Para remediar este problema, las nuevas técnicas de ajuste han sido desarrolladas de forma que se no se vean afectadas tan fácilmente al trabajar sobre conjuntos de datos acompañados de elementos contaminados. Este tipo de métodos se conocen como robustos (o resistentes), y son capaces de proporcionar un buen ajuste aún cuando se trabajan con conjuntos de datos con una cierta cantidad de elementos atípicos [1, 2]. Algunos piensan que las técnicas de ajuste robustos esconden los datos atípicos, lo cual no sucede, ya que un dato atípico puede ser detectado por el enorme residual que éste genera al compararlo con el modelo estimado. De tal forma, al detectarlos se puede realizar una estandarización de residuales y realizar el ajuste incluso con una técnica como MC, ya que el conjunto de datos a ajustar se encontrará libre de elementos atípicos.

Una de las mejores técnicas de ajuste robusto es la mínima mediana de los cuadrados (MMC) la cual obtiene los parámetros desconocidos del ajuste resolviendo un problema de optimización no lineal que consiste en minimizar  $\min\{\text{mediana}(r_i^2)\}$ , donde  $r_i$  es el residual entre el  $i$ -ésimo elemento del conjunto de datos y el ajuste propuesto. El problema planteado por MMC se resuelve buscando el mejor ajuste dentro de un espacio de ajustes candidatos, el cual se generó a partir del conjunto de puntos conocido. Dicho espacio resulta muy grande y no es posible analizarlo en su totalidad en un tiempo relativamente corto, por lo que es necesario utilizar una técnica de muestreo aleatorio denominada RANSAC (*RANdom SAmple Consensus*), el cual pretende reducir el espacio de búsqueda seleccionando aleatoriamente un subconjunto de datos del conjunto conocido de datos y a partir de dicho subconjunto generar un ajuste candidato que será evaluado para determinar si representa o no al conjunto de datos dado [1, 2, 3, 4].

Para determinar la fortaleza de los métodos, se ha establecido una métrica denominada punto de ruptura, la cual indica la porción mínima de datos atípicos necesarios para que el método proporcione un ajuste no deseado [3, 1, 5]. Entre los métodos más poderosos se encuentra MMC, el cual es capaz de proporcionar un punto de ruptura del 50 %, v.g., que la mitad de los datos conocidos deben estar fuera de los rangos permisibles para que éste proporcione un ajuste erróneo, concluyendo que un punto de ruptura del 50 % es el valor máximo conseguido [1, 2, 4, 3].

Dentro de la técnicas de ajuste se ha observado que se obtienen mejores resultados utilizando la suma de los valores absolutos de los residuales ortogonales, pero esta técnica implica resolver un problema de optimización no lineal, v.g., que la derivada de la función de costo que se utilizará para obtener los parámetros desconocidos del ajuste, presenta una discontinuidad (se vuelve infinita) en el mínimo de la función de costo, por lo que no es posible utilizar técnicas clásicas de optimización que requieran información de la primera derivada [6, 7, 8].

En virtud de lo anterior, el objetivo final de este trabajo de tesis consiste en es-



tablecer como un problema de optimización no lineal el ajuste de líneas, elipses, círculos, triángulos y cuadriláteros utilizando como función de costo (o error) la suma de las distancias ortogonales entre el conjunto de datos conocido y el ajuste propuesto. Los parámetros desconocidos del ajuste se obtendrán al resolver dicho problema de optimización utilizando la evolución diferencial y los resultados se compararán con el ajuste de las formas geométricas mencionadas anteriormente utilizando MMC + RANSAC.

## 1.1. Descripción del problema

Dado un conjunto de puntos  $\mathbf{p}_i \in \mathbb{R}^2$ , para  $1 \leq i \leq n$ , en un sistema de coordenadas globales  $xy$ , se buscará la forma o formas geométricas que mejor representen al conjunto de puntos conocido bajo condiciones de mucho ruido, v.g., cuando el porcentaje de puntos atípicos supere el 50%.

Se desea realizar el ajuste de las siguiente formas geométricas:

- Línea
- Elipse
- Círculo
- Triángulo
- Cuadrilátero

Se utilizará como función de costo la suma de las distancias ortogonales entre los puntos  $\mathbf{p}_i$  y el ajuste propuesto. En la figura 1.1 se ilustran las distancias ortogonales (en este caso son perpendiculares) de los puntos a una línea (que sería el modelo de ajuste).

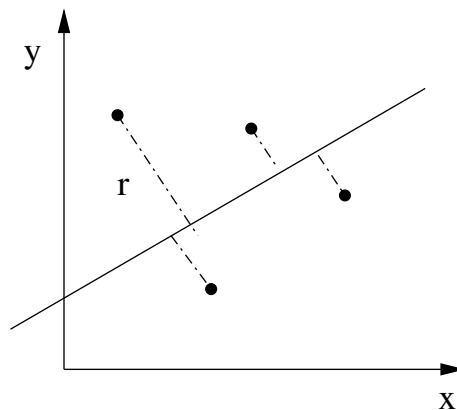


Figura 1.1: Distancia (o residual) ortogonal.

## 1.2. Objetivo de la tesis

El objetivo de la tesis es realizar un método de ajuste robusto por medio de la minimización de la suma de las distancias ortogonales. El método debe ser capaz de extraer varias formas geométricas (líneas, círculos, elipses, triángulos o cuadrados) presentes en un mismo conjunto de puntos.

El problema de ajuste será no lineal y se plantea resolverlo utilizando la metaheurística llamada evolución diferencial. Se compararán los resultados obtenidos contra la heurística MMC + RANSAC con la finalidad de demostrar que se ha conseguido un punto de ruptura de más del 50 %, esto es, que el método es capaz de extraer las formas geométricas aún cuando existan más del 50 % de puntos atípicos.

## 1.3. Organización de la tesis

En el capítulo 2 se mencionan algunas de las técnicas de ajuste robusto más representativas de la literatura utilizada para el desarrollo de esta tesis. También se da una descripción del funcionamiento de cada una de las técnicas y se hace mención de sus ventajas y desventajas.

El capítulo 3 comprende el ajuste robusto de líneas. En éste se menciona de forma detallada cómo se realiza el ajuste algebraico de la línea, la obtención de la distancia perpendicular (o geométrica) entre los puntos del conjunto conocido y la línea propuesta. También se detalla la propuesta del problema de optimización que se resuelve para obtener los parámetros del ajuste, la representación que se le dio al problema para ocupar la evolución diferencial y resolver el ajuste. Se detalla la implementación de MMC + RANSAC para el ajuste de líneas y se realizan las pruebas necesarias para comparar el nuevo método propuesto (al que llamaremos SD + ED, Suma de las Distancias más Evolución Diferencial) vs. MMC + RANSAC.

El capítulo 4 está dedicado al ajuste robusto de elipses. En él se detalla el ajuste algebraico de la elipse, el uso de la distancia ortogonal para resolver el problema de ajuste y el problema de optimización propuesto para determinar los parámetros desconocidos. También se detalla el uso de la evolución diferencial, y se menciona la implementación de MMC + RANSAC. Finalmente se realizan las pruebas necesarias para comparar MMC + RANSAC vs. SD + ED.

En el capítulo 5 se presenta el ajuste robusto de triángulos como un problema de optimización no lineal. Éste consiste en minimizar la suma de las distancias perpendiculares de cada uno de los puntos conocidos a las aristas del triángulo propuesto. Se consideran también las distancias de los puntos conocidos a los vértices de dicho triángulo y se explica con detalle cómo se distinguen las llamadas distancias perpendiculares válidas y la nueva restricción que se agregó al problema de optimización.

Éste problema se resolvió utilizando evolución diferencial. Finalmente, se presentan las pruebas realizadas para determinar el desempeño del algoritmo propuesto.

El capítulo 6 está dedicado al ajuste robusto de cuadriláteros. En él se plantea la extracción de cuadriláteros como un problema de optimización no lineal, el cual se resolvió utilizando la metaheurística llamada evolución diferencial. El ajuste de cuadriláteros se plantea como una extensión del ajuste de triángulos presentado en el capítulo 4, ya que en cuadrilátero se representa con un vértice más. Finalmente, se presentan las pruebas a las que se sometió el algoritmo para analizar su comportamiento.

En el capítulo 7 se presenta el ajuste de círculos y se retoma el de elipses. El problema de optimización que se plantea para ambos se modifica de tal forma que puede verse como la combinación de un método de ajuste y uno de agrupamiento, ya que ahora el problema consiste en minimizar la suma de las distancias perpendiculares de los puntos conocidos al círculo/elipse. Se maximiza el número de puntos que forman parte de dicho ajuste y se observa la densidad de éste. También se distinguen los puntos de ajuste de los atípicos utilizando la mediana de las desviaciones absolutas. Así mismo, se presenta una versión “acelerada” del algoritmo MMC + RANSAC de la cual se mencionan ventajas y desventajas. Finalmente, se realizan pruebas para comparar la nueva versión de MMC + RANSAC vs. SD + ED vs. los métodos de extracción de círculos y elipses basados en la transformada de Hough.

El capítulo 8 contiene las conclusiones generales de este trabajo de tesis y el trabajo futuro.



# Capítulo 2

## Marco teórico

---

El propósito del ajuste robusto es realizar una buena aproximación de las ecuaciones que definen un modelo a partir de un conjunto de datos conocidos. El modelo lineal clásico se puede definir de la siguiente forma.

$$y_i = x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{im}\theta_m + e_i \quad (2.1)$$

Para  $i = 1, \dots, n$  donde  $n$  es el número de casos conocidos. Las variables  $x_{i1}, \dots, x_{im}$  son llamadas variables exploratorias, mientras que la variable  $y_i$  es conocida como la variable de respuesta. En la teoría clásica se asume que el término de error  $e_i$  se encuentra normalmente distribuido con media cero y desviación estándar  $\sigma$  desconocida. De tal forma que el objetivo es realizar una estimación del vector de parámetros desconocidos del ajuste

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix},$$

para el conjunto de datos conocidos

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

y el vector de variables de respuesta

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Al realizar la estimación de los parámetros desconocidos  $\theta$  se obtendrá

$$\hat{\theta} = \begin{bmatrix} \hat{\theta}_1 \\ \vdots \\ \hat{\theta}_m \end{bmatrix}$$

donde los estimados  $\hat{\theta}_j$  son conocidos como los coeficientes de ajuste, a partir de los cuales se puede realizar una aproximación de las variables de respuesta dada por

$$\hat{y}_i = x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{im}\theta_m + e_i, \quad (2.2)$$

donde  $\hat{y}_i$  es el valor estimado de  $y_i$ . Una vez que se obtiene el valor de la predicción se puede realizar el cálculo del residual, el cual medirá la similitud que existe entre el ajuste estimado y el real.

$$r_i = y_i - \hat{y}_i$$

El residual  $r_i$  se puede obtener de diferentes formas, tal y como se menciona a continuación:

- Residual en  $x$
- Residual en  $y$
- Residual en  $x$  y en  $y$
- Residual ortogonal

La figura 2.1 muestra de forma gráfica los tipos de residual mencionados anteriormente.

La elección del residual que se utilizará para determinar el error del ajuste respecto al conjunto de puntos, ha sido uno de los temas más estudiados dentro de la estadística robusta, ya que muchos autores atribuyen la debilidad de las técnicas de ajuste a la elección del residual. Se ha llegado a la conclusión de que los residuales que van tanto en el eje  $x$ ,  $y$  ó ambos representan la llamada “distancia algebraica”, mientras que la distancia real entre un elemento del conjunto y el ajuste propuesto está representada por el residual ortogonal [1, 8].

## 2.1. Método de los mínimos cuadrados

El método de mínimos cuadrados (MC) es una de las técnicas de ajuste lineal más famosas y ha sido utilizada en muchos campos de la ciencia durante mucho tiempo. El método resuelve el problema de ajuste de la siguiente manera:

Retoma el modelo lineal clásico descrito en (2.1)

$$y_i = x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{im}\theta_m + e_i \quad (2.3)$$

donde  $y_i$  representa las variables de respuesta de los  $n$  casos conocido y  $x_i$  las variables exploratorias del problema de ajuste. Por lo tanto, se puede generar un vector  $Y$  que

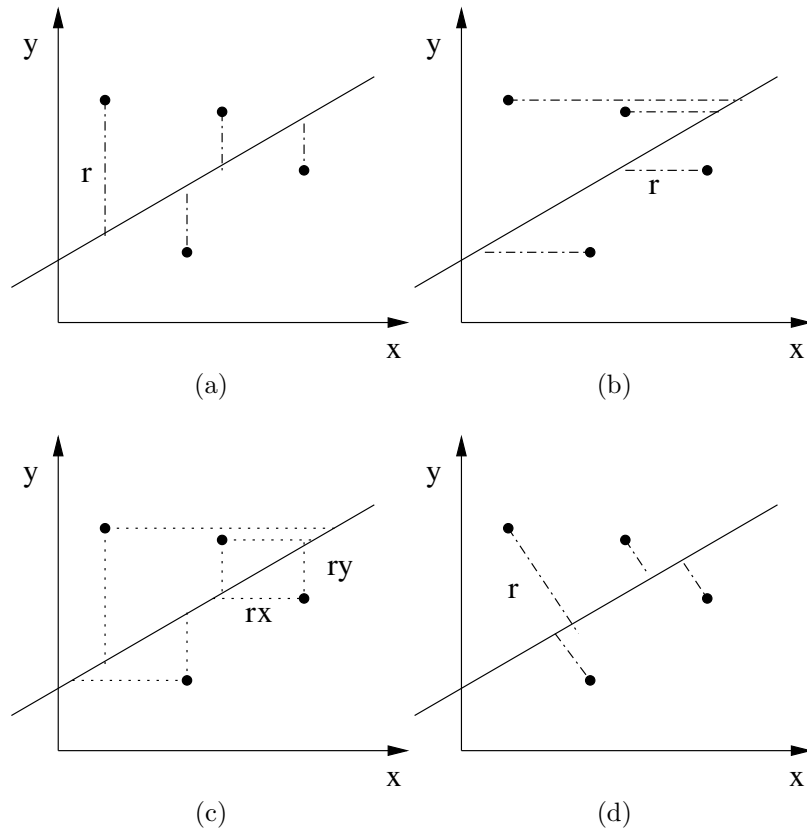


Figura 2.1: (a) Residual en  $y$ . (b) Residual en  $x$ . (c) Residual en ambos. (d) Residual ortogonal.

contenga todas las variables de respuesta y una matriz  $X$  que contenga las variables exploratorias

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

La diferencia entre el ajuste propuesto y el original se obtiene a partir de

$$r_i = y_i - \hat{y}_i$$

donde  $\hat{y}_i$  representa el ajuste propuesto de  $y_i$ , tal y como se definió en (2.2). Por lo tanto, los parámetros desconocidos  $\hat{\theta}$  del ajuste se pueden obtener al resolver el siguiente problema de optimización

$$\min_{\hat{\theta}} \sum_{i=1}^n r_i^2 \quad (2.4)$$

Se necesita optimizar el problema propuesto en (2.4) para obtener los parámetros desconocidos del ajuste  $\hat{\theta}$ . Para reducir el trabajo, se puede reescribir (2.3) en su forma matricial como:

$$Y = X\hat{\theta} \quad (2.5)$$

se tiene que

$$J(\hat{\theta}) = \sum_{i=1}^n r_i^2 \quad (2.6)$$

entonces tenemos que

$$J(\hat{\theta}) = \mathbf{r}^T \mathbf{r} = (Y - X\hat{\theta})^T (Y - X\hat{\theta}) = Y^T Y + \hat{\theta}^T X^T X \hat{\theta} - 2\hat{\theta}^T X^T Y \quad (2.7)$$

se puede observar que la función  $J(\hat{\theta})$  es cuadrática y tiene un sólo mínimo, por lo tanto para minimizar dicha función se debe cumplir que

$$\frac{\partial J(\hat{\theta})}{\partial \hat{\theta}} = 0 \quad (2.8)$$

es decir

$$2X^T X \hat{\theta} - 2X^T Y = 0 \quad (2.9)$$

por lo tanto los parámetros desconocidos del ajuste puede ser estimados de la siguiente forma

$$\hat{\theta} = (X^T X)^{-1} X^T Y = 0 \quad (2.10)$$

MC proporciona excelentes resultados incluso bajo condiciones de ruido gaussiano sobre los elementos del ajuste, pero es extremadamente sensible a la presencia de datos atípicos. En la figura 2.2 se puede observar cómo MC proporciona un excelente ajuste cuando los datos no están contaminados. Sin embargo, al encontrarse con un punto que se encuentra lo suficientemente alejado de la línea que se desea ajustar, éste repercute significativamente en la función de error que se maneja, por lo que el ajuste se sesga en dirección del punto contaminado.



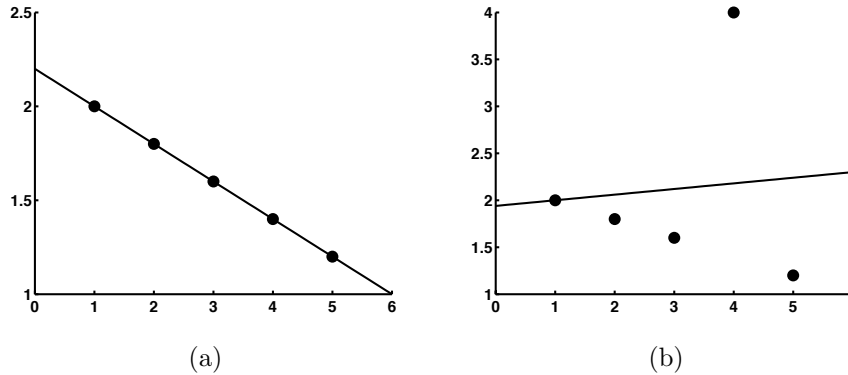


Figura 2.2: (a) Conjunto de cinco puntos que definen correctamente la línea. (b) Mismo conjunto de puntos presentado en (a) pero con un punto atípico en la dirección de  $y$ . El ajuste de la línea se realizó utilizando la diferencia de valores (los residuales) en  $y$ .

## 2.2. Método de los mínimos valores absolutos ( $L_1$ )

El primer paso hacia el ajuste robusto fue dado por Edgeworth (1887) al proponer lo que se conoce como  $L_1$ . Edgeworth argumentó que los puntos atípicos tenían efectos notorios sobre el ajuste debido a que el valor del residual es cuadrático, por lo que propuso utilizar el valor absoluto del residual.

Para obtener los parámetros desconocidos del ajuste  $\hat{\theta}$  es necesario resolver el siguiente problema de optimización

$$\min_{\hat{\theta}} \sum_{i=1}^n |r_i| \quad (2.11)$$

El problema propuesto en (2.11) es no lineal debido a que la derivada del valor absoluto no está definida en su mínimo, por lo cual no se puede obtener una solución cerrada como en el caso de MC. Para resolver este problema, es necesario utilizar algún método iterativo de optimización. Cabe mencionar que los métodos iterativos no garantizan el óptimo global y dependen del punto inicial de búsqueda, el cual puede ocasionar que el método quede atrapado en un mínimo local.

Cabe mencionar que  $L_1$  brinda robustez sobre los puntos atípicos que se encuentran sobre la dirección del residual, pero no en el caso contrario, v.g., que si se elige un residual sobre el eje  $y$ , sólo brindará protección contra los puntos atípicos en dicha dirección, pero no contra los que vayan en la dirección contraria. Esto se puede observar mejor en la figura 2.3 y 2.4 donde se presenta el caso en el que el residual se toma en dirección de  $y$ .

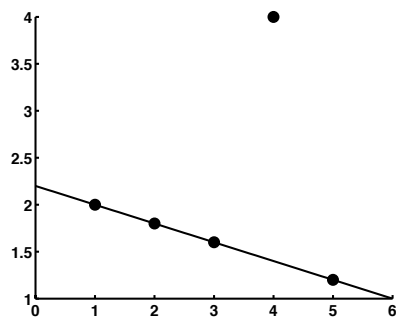


Figura 2.3: Resultado entregado  $L_1$  en presencia de un punto atípico en dirección de  $y$ .

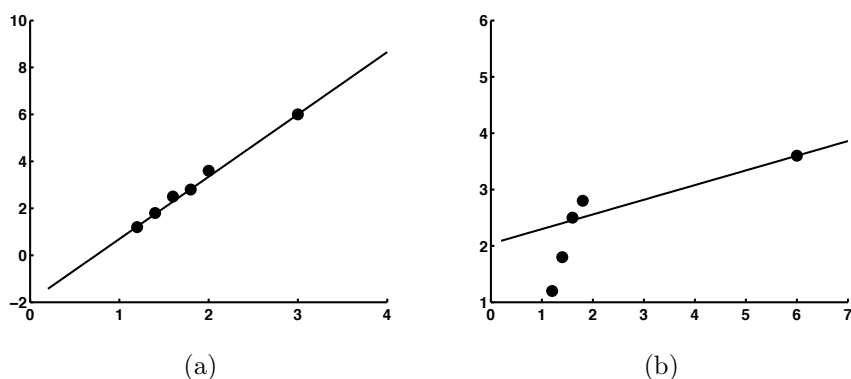


Figura 2.4: (a) presenta un conjunto de cinco puntos que definen una línea. (b) Mismo conjunto de puntos presentado en (a) pero con un punto atípico en la dirección de  $x$ .

## 2.3. Estimadores M

Se han desarrollado muchos métodos para lidiar con el problema de ajuste robusto, pero de entre ellos, los estimadores M se encuentran entre los más populares.

La teoría de los estimadores M fue propuesta por Huber en 1964 y años más tarde generalizó su idea en un método de ajuste robusto (Huber 1973 y Huber 1981) [1, 9]. La idea básica de los estimadores M reside en reemplazar el residual cuadrático  $r_i^2$  de (2.4) por una función simétrica  $\rho$  de los residuales.

$$\min_{\hat{\theta}} \sum_{i=1}^n \rho(r_i) \quad (2.12)$$

donde  $\rho(r_i)$  es una función simétrica con un sólo mínimo cuando  $r_i$  es cero. El propósito de introducir una función  $\rho(r_i)$ , es reducir el efecto de los puntos atípicos. La función

$\rho(r_i)$  seleccionada debe ser diferenciable respecto a los coeficientes de ajuste tal que

$$\sum_{i=1}^n \Psi(r_i) x_i = 0 \quad (2.13)$$

Donde  $\Psi$  es la derivada de  $\rho$  y  $x_i$  es el vector columna de las variables exploratorias del  $i$ -ésimo caso. La solución del sistema de ecuaciones no es equivariante con respecto a los cambios en el eje de las variables de respuesta, por lo cual es necesario estandarizar los residuales con un factor de escala  $\sigma$  que debe estimarse al mismo tiempo, obteniéndose

$$\sum_{i=1}^n \Psi\left(\frac{r_i}{\sigma}\right) x_i = 0 \quad (2.14)$$

donde  $\sigma$  representa la varianza de los residuales.

Los estimadores M son robustos en presencia de puntos atípicos en dirección de las variables de respuesta ( $y$ ), pero susceptibles a los puntos contaminados sobre las variables exploratorias ( $x$ ). Para contrarrestar este efecto. Mallows (1975) propuso la introducción de un peso:

$$\sum_{i=1}^n w(x_i) \Psi\left(\frac{r_i}{\sigma}\right) x_i = 0 \quad (2.15)$$

Se han realizado pruebas utilizando este método de ajuste y se ha llegado a la conclusión de que éste presenta un punto de ruptura dado por  $\frac{1}{1+p}$ , donde  $p$  representa el número de variables exploratorias, v.g., que si  $p = 2$ , el punto de ruptura más alto será a lo más el 33 % de los datos, lo cual lo hace inutilizable en ambientes donde los datos corruptos se encuentran por encima de dicho porcentaje [1, 5, 3, 2, 9].

## 2.4. Método de la mínima mediana de los residuales cuadráticos (MMC)

El método de MMC fue propuesto por Rousseeuw en [1] basándose en la idea de Hampel (1975) [10] de reemplazar la suma de los residuales al cuadrado, por la mediana de ellos, asumiendo que los puntos que pertenecen al ajuste representan más del 50 % del conjunto conocido. Para obtener los parámetros desconocidos del ajuste  $\hat{\theta}$  se debe obtener la mínima mediana de los residuales al cuadrado.

$$\hat{\theta} = \underset{\hat{\theta}}{\text{mín}} \{ \text{mediana}(r_i^2) \} \quad (2.16)$$

Cabe mencionar que (2.16) no tiene una solución cerrada, como en el caso de MC. Para hallar la solución se debe realizar una búsqueda en todo el espacio de posibles

ajustes. Tomando como referencia el caso de la línea, se puede realizar el ajuste de ésta si se conocen al menos dos puntos que pertenecen a ésta, v.g., que si se desea explorar el espacio de los posibles ajustes, se deben considerar  $C_2^n = \frac{n!}{2!(n-2)!}$  pares de puntos donde  $n$  representa el número total de puntos conocidos.

Es fácil observar que el espacio de búsqueda es muy grande, por lo que no se puede esperar que el método proporcione resultados en un tiempo comprensible. En la práctica se utiliza sólo una aproximación de MMC con la finalidad de reducir el tiempo de procesamiento de los datos. Para ello se utiliza una técnica de muestreo aleatorio (RANSAC), con la cual sólo se tomarán  $m$  muestras del conjunto de datos conocido a partir de una probabilidad  $P$  que determina la probabilidad de que una muestra de  $k$  elementos pertenezca al ajuste deseado. En [11] se demuestra que la probabilidad  $P$  se puede obtener a partir de la ecuación (2.17), donde  $m$  representa el mínimo de muestras de  $k$  elementos que deben tomarse del conjunto de puntos conocido para garantizar una probabilidad  $P$  dada y  $\epsilon$  representa la probabilidad de que al menos uno de los  $k$  elementos de la muestra sea un punto atípico.

$$P = 1 - (1 - (1 - \epsilon)^k)^m \quad (2.17)$$

Para obtener un buen ajuste, la probabilidad  $P$  de que un subconjunto pertenezca al ajuste deseado debe ser alta (v.g. mayor al 95 %), por lo tanto el mínimo de muestras  $m$  que deben tomarse para garantizar  $P$  se puede obtener a partir de (2.18).

$$m = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^k)} \quad (2.18)$$

Como era de esperarse, en un principio MMC no se podía generalizar y aplicar a todos los casos de ajuste debido a que presentaba una gran debilidad cuando se enfrentaba a conjuntos de datos que presentaban ruido gaussiano. Para atacar esta deficiencia, Rousseeuw le agregó a MMC un procedimiento de mínimos cuadrados ponderados (o con peso) que actúa después del ajuste inicial realizado con la muestra de  $k$  puntos. De esta forma se definió un factor de escala  $\hat{\sigma}$  para normalizar los residuales y disminuir los efectos del ruido gaussiano sobre el ajuste.

$$\hat{\sigma} = 1.4826 \left( 1 + \frac{5}{n - k} \right) \text{med}_i \sqrt{r_i^2} \quad (2.19)$$

Una vez calculado dicho factor se asigna un peso de cero o uno al residual que se está analizando utilizando el siguiente criterio

$$w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}} > 2.5 \end{cases} \quad (2.20)$$

Después de asignar el peso respectivo a cada uno de los residuales se puede realizar el ajuste final y extraer los parámetros desconocidos resolviendo el siguiente problema:

$$\min_{\hat{\theta}} \sum_{i=1}^n w_i r_i^2 \quad (2.21)$$

El problema presentado en (2.21) se puede ver como el problema de ajuste con MC, pero con la característica de que éste se encontrará libre de puntos atípicos. Sin embargo Rousseeuw observó que MMC era susceptible a los cambios locales que pudieran darse en los puntos, v.g., que el ajuste podía verse afectado si se tenían pequeñas perturbaciones en los puntos que corresponden al ajuste (v.g., con  $w_i = 1$ ). Para remediar este problema introdujo un factor de escala final  $\hat{\sigma}_f$ , el cual garantiza que los puntos que corresponden al ajuste se encuentran libres de puntos contaminados y a la vez de perturbaciones locales:

$$\hat{\sigma}_f = \sqrt{\frac{\sum_{i=1}^n w_i r_i^2}{\sum_{i=1}^n w_i - k}} \quad (2.22)$$

Al obtener el factor de escala final se reasignan los pesos ( $w_i$ ) a cada uno de los residuales utilizando (2.20) pero con  $\hat{\sigma} = \hat{\sigma}_f$  y se obtienen los parámetros del ajuste resolviendo (2.21) con los nuevos pesos asignados. En la figura 2.5 se puede observar que MMC brinda protección contra puntos atípicos en cualquiera de las direcciones.

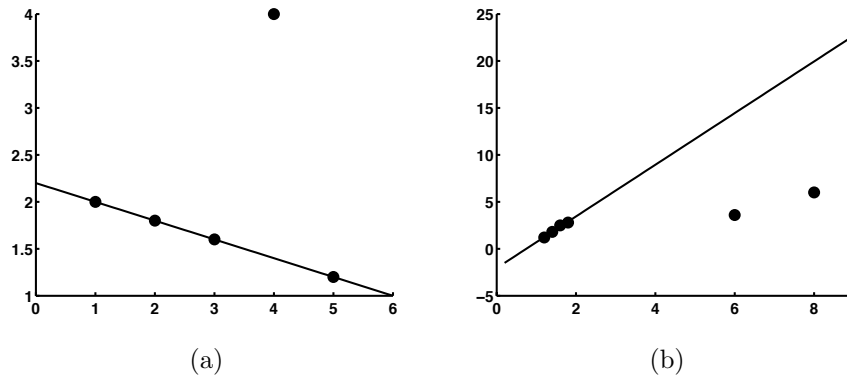


Figura 2.5: (a) Presenta un conjunto de cinco puntos con un punto contaminado en la dirección de  $y$ . (b) Presenta un conjunto de seis puntos con dos puntos atípicos en la dirección de  $x$ .

El método de MMC tiene un excelente desempeño y fortaleza, ya que es capaz de proporcionar un ajuste adecuado en condiciones donde el porcentaje de puntos atípicos es de hasta un 50% [1, 2], lo que ha hecho a MMC un método muy popular para realizar ajustes. Algunas de sus aplicaciones se pueden encontrar en [12], [2] y [4].

## 2.5. Estimador de muestreo aleatorio (RANSAC)

RANSAC es un paradigma que fue propuesto y validado por Fischler y Rolles en [11]. El método consiste en seleccionar aleatoriamente un subconjunto de  $k$  elementos del conjunto de puntos conocido. El subconjunto debe contener los elementos necesarios para realizar un ajuste inicial, a partir del cual se podrán descartar o aceptar puntos dentro del ajuste, siempre y cuando los residuales de dichos puntos sean menores a un valor de umbral  $U$  definido por el usuario. A continuación se muestran los pasos que deben seguirse para realizar un ajuste utilizando RANSAC.

1. Seleccionar aleatoriamente un subconjunto de  $k$  puntos a partir del conjunto de puntos dado. Los  $k$  puntos seleccionados deben ser suficientes para realizar un ajuste inicial, v.g., a partir de ellos se deben extraer los parámetros del ajuste.
2. Realizar el ajuste inicial con los  $k$  puntos del subconjunto seleccionado en el paso anterior.
3. Calcular el residual de los  $n - k$  puntos restantes hacia el ajuste proporcionado por el paso anterior.
4. Contabilizar los puntos que pertenecen al ajuste, v.g., considerar aquellos puntos cuyo residual  $r_i < U$ .
5. Si el número de puntos contabilizados en el paso anterior es mayor al obtenido en la iteración anterior, almacenar dicho conjunto de puntos.
6. Repetir desde el paso 1  $m$  veces.
7. Realizar el ajuste final al conjunto de puntos entregado por el método.

RANSAC resulta un método de ajuste muy poderoso, ya que experimentalmente puede tolerar más del 50% de puntos atípicos. Los parámetros que debe proporcionar el usuario al método son: valor de umbral  $U$ , número de elementos  $k$  del subconjunto de prueba y el número de intentos que se deben realizar. Esto hace que RANSAC dependa en gran medida de dichos parámetros, ya que si no se proporcionan los adecuados, puede no entregar buenos resultados.

## 2.6. Ajuste según la suma de las distancias ortogonales

El ajuste según la suma de las distancias ortogonales difiere de todas las otras técnicas debido a que éste utiliza la distancia geométrica, la cual representa la distancia más corta que existe entre un dato del conjunto conocido y el ajuste estimado. Como se mencionó al inicio del capítulo los métodos que se basan en MC utilizan la llamada distancia algebraica.

La distancia algebraica utiliza la ecuación implícita del ajuste, p.ej. la ecuación implícita para realizar el ajuste del modelo lineal se describió en (2.3). Se puede observar que (2.3) está formada por el vector de las variables de respuesta  $Y$ , la matriz de las variables exploratorias  $X$  y el vector de parámetros desconocidos del ajuste  $\hat{\theta}$ , como se describe a continuación.

$$F(Y; X, \hat{\theta}) = 0 \quad (2.23)$$

Por lo tanto el ajuste algebraico realizado por MC se basa en las desviaciones que sufre (2.23) respecto a su valor esperado (v.g, cero) para cada uno de los puntos dados, ya que cualquier variación indica que el punto no se encuentra sobre el ajuste (v.g., que existe un error de ajuste) [8].

Una de las ventajas mas sobresalientes del ajuste basado en la distancia algebraica, reside en que la mayoría de las veces se tiene una única solución, la cual se puede obtener de forma directa resolviendo un sistema de ecuaciones. Esto hace que el ajuste se realice más rápido y sea fácil de implementar. Sin embargo algunas de las desventajas que trae consigo el utilizar este tipo de ajuste se mencionan a continuación:

- Las distancias (o residuales) entre los puntos y el ajuste propuesto, no se encuentran en las mismas unidades de medida con las que se esta trabajando.
- Las distancias resultan invariantes a las transformaciones geométricas que pueda sufrir el ajuste,(p.ej. al realizar el ajuste de una elipse utilizando la distancia algebraica, se pueden tener dos elipses con parámetros distintos que generan el mismo valor para función de costo).
- Para realizar un ajuste adecuado, los valores de los residuales deben estar ponderados.
- El ajuste puede terminar en una forma geométrica que no corresponde a lo que se busca, p.ej. al extraer secciones cónicas el ajuste algebraico puede confundir una elipse con una hipérbola.

Al utilizar el ajuste geométrico, el residual se define como la distancia ortogonal, o la distancia más corta entre los puntos del conjunto conocido y el ajuste propuesto.

Para realizar el ajuste geométrico se necesitan resolver problemas de optimización no lineal debido a que las funciones de costo que involucran este tipo de residuales no tiene una derivada definida en su mínimo, por lo que es necesario utilizar técnicas iterativas de optimización [8].

## 2.7. Sumario

En este capítulo se presentaron las diferentes técnicas de ajuste robusto que existen en el estado del arte, se mencionaron sus ventajas y desventajas y se dio una descripción de su funcionamiento. En la sección dedicada al ajuste robusto según las distancias ortogonales, se dio una breve descripción de la forma en la que funciona dicho ajuste, ya que en los capítulos posteriores se hablará de forma detallada de este tipo de ajuste para cada una de las formas geométricas que se ajustaron en este trabajo de tesis.



# Capítulo 3

## Ajuste de líneas

---

El método tradicional para encontrar la correlación que existe entre el conjunto de datos conocido, consiste en estimar una línea hacia éstos utilizando la suma de las distancias al cuadrado. Dicho método minimiza la distancia al cuadrado que existe entre el punto y la línea estimada. Este método funciona bien cuando no se tienen puntos contaminados sobre la línea que se desea ajustar, v.g. que los datos no contienen valores que se encuentren fuera de los rangos permisibles. Este tipo de datos, conocidos como puntos atípicos, contribuyen de forma negativa al ajuste, ya que al encontrarse alejados su distancia cuadrática es aún más grande, por lo que la estimación obtenida tratará de reducir dicha distancia, sesgando la línea en dirección de dicho punto [13].

En este capítulo se ajustan de forma robusta líneas a un conjunto de puntos conocidos bajo condiciones de mucho ruido (v.g. más del 50 % de los puntos conocidos se consideran atípicos), utilizando la suma de las distancias ortogonales de los puntos conocidos a la línea ajustada. Este es un problema no lineal que se resuelve utilizando la evolución diferencial. El método propuesto es comparado contra la mínima mediana de los cuadrados (MMC) con RANSAC, demostrando que éste obtiene mejores resultados del ajuste de la línea incluso cuando el número de puntos atípicos representa del 50 % al 100 % de los puntos conocidos.

### 3.1. Descripción del problema

Dado un conjunto de puntos  $\mathbf{p}_i \in \mathbb{R}^2$ , para  $1 \leq i \leq m$ , a un sistema de coordenadas globales  $xy$ , se buscará la línea que represente mejor conjunto de puntos dado a partir del modelo lineal clásico.

$$y_i = ax_i + b \tag{3.1}$$

Como se mencionó al inicio del capítulo, se utilizará la suma de las distancias ortogonales en lugar de sus cuadrados, ya que ésta resulta más estable aún en presencia de datos contaminados. El uso de este tipo de residual tiene sus desventajas, las cuales

se mencionarán con mayor detalle en la sección que habla sobre la distancia ortogonal.

Para realizar un ajuste adecuado se debe realizar una clara distinción entre los dos casos de la línea que se presentan en el modelo de la figura 3.1 donde (3.2) y (3.3) representan la ecuación de las rectas  $l_1$  y  $l_2$ , respectivamente.

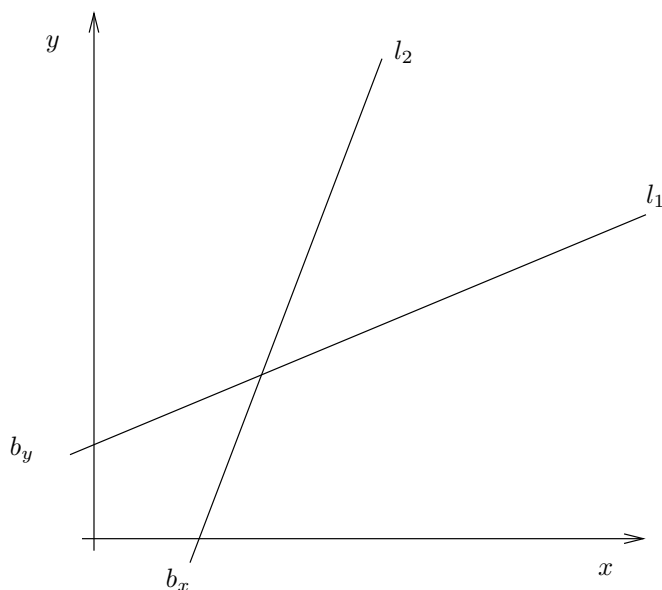


Figura 3.1: Modelo utilizado de la línea.

$$y = a_y x + b_y \text{ para } |a_y| < 1 \quad (3.2)$$

$$x = a_x y + b_x \text{ para } |a_x| < 1 \quad (3.3)$$

Donde  $a_y$  y  $a_x$  representan el valor de la tangente del ángulo formado entre la recta y el eje  $x$  y  $y$ , respectivamente. Si se realiza el despeje de  $y$  de (3.3) se tiene

$$y = \frac{x - b_x}{a_x}$$

Al realizar la división entre  $a_x$  se obtiene

$$y = \frac{x}{a_x} - \frac{b_x}{a_x},$$

indicando que existe una relación entre los factores de (3.3) y (3.2) de la forma

$$a_y = \frac{1}{a_x}, \quad b_y = -\frac{b_x}{a_x}.$$

Esta relación presenta discontinuidades, p.ej. en el caso presentado en la figura 3.2. Donde el valor de  $a_x = 0$ , por lo tanto  $a_y = \frac{1}{0}$  y  $b_y = -\frac{b_x}{0}$  quedando ambos valores indeterminados. Por esta razón es necesario tratar ambos casos de manera independiente como se planteó en (3.2) y (3.3). A continuación se explica de forma detallada el ajuste algebraico y el ajuste utilizando la suma de las distancias ortogonales.

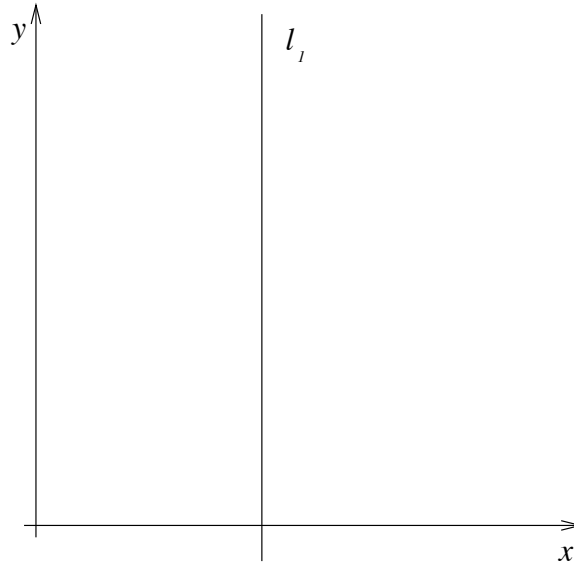


Figura 3.2: Caso de ejemplo.

### 3.1.1. Ajuste algebraico

Como se mencionó anteriormente la recta puede representarse mediante  $y_i = a_y x_i + b_y$  ó  $x_i = a_x y_i + b_x$  tal que se genere el siguiente problema de minimización:

$$\min_{a_y, b_y} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (3.4)$$

y para el segundo caso

$$\min_{a_x, b_x} \sum_{i=1}^m (x_i - \hat{x}_i)^2 \quad (3.5)$$

como se demuestra en [7] los parámetros desconocidos de la recta se pueden obtener como se menciona a continuación.

$$a_y = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^m (x_i - \bar{x})^2} \quad (3.6)$$

$$b_y = \bar{y} - a_y \bar{x} \quad (3.7)$$

$$a_x = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^m (y_i - \bar{y})^2} \quad (3.8)$$

$$b_x = \bar{x} - a_x \bar{y} \quad (3.9)$$

Donde  $\bar{x}$  y  $\bar{y}$  representan la media de los valores de  $x$  y  $y$  del conjunto de puntos conocido.

### 3.1.2. Ajuste según la suma de las distancias ortogonales

El mejor ajuste se obtiene al utilizar la suma de las distancias ortogonales, ya que éste representa la distancia geométrica real entre los puntos y la recta estimada. Al resolver este problema se tiene que la función objetivo no tiene una derivada continua, por lo que un método numérico convencional no podría resolverlo.

Para determinar la distancia ortogonal entre el punto conocido y la recta estimada se tienen los modelos presentados en la figura 3.1.

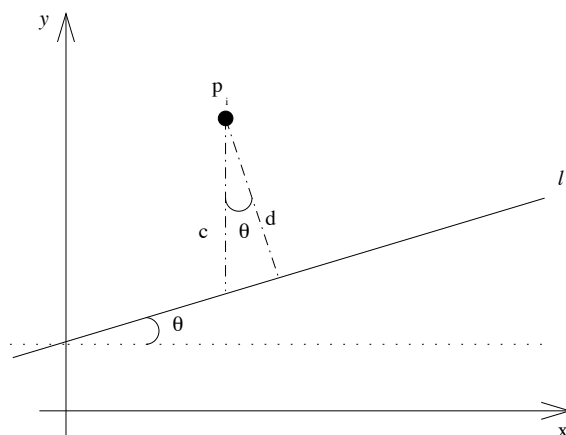


Figura 3.3: Modelo utilizado de la línea para el caso 1.

Para hallar la distancia  $d$  que va del punto conocido  $\mathbf{p}_i$  a la recta estimada  $l$ , es necesario realizar la proyección del vector  $c$  sobre el vector  $d$ :

$$c = |y_i - \hat{y}_i| \quad (3.10)$$

Donde  $\hat{y}_i$  representa la ecuación de la recta estimada  $l$ . Por lo tanto, al realizar la sustitución de  $\hat{y}_i$  en (3.10) se obtiene:

$$c = |y_i - (a_y x_i + b_y)| \quad (3.11)$$

Se sabe que el ángulo  $\theta$  formado entre los vectores  $c$  y  $d$  es el mismo que forma la línea estimada con el eje de las ordenadas por lo que se puede deducir lo siguiente:

$$\cos \theta = \frac{d}{c} \quad (3.12)$$

$$d = |y_i - (a_y x_i + b_y)| |\cos \theta| \quad (3.13)$$

Planteando entonces el siguiente problema de optimización

$$\min_{a_y, b_y} \frac{1}{m} \sum_{i=1}^m d_i \quad (3.14)$$

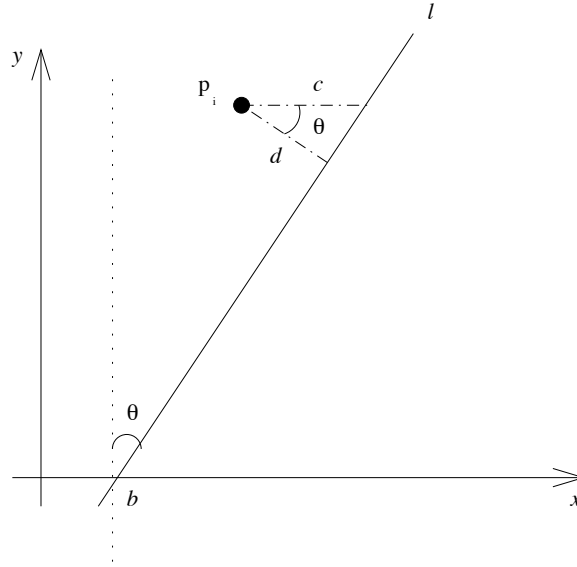


Figura 3.4: Modelo utilizado de la línea para el caso 2.

Para el caso presentado en (3.3) se toma el modelo de la figura 3.4.

Se realiza el mismo procedimiento que en el caso anterior para conocer la distancia  $d$  que va del punto conocido  $\mathbf{p}_i$  a la recta estimada  $l$ , v.g. se realiza la proyección del vector  $c$  sobre el vector  $d$ .

$$c = |x_i - \hat{x}_i| \quad (3.15)$$

Donde  $\hat{x}_i$  representa la ecuación de la recta estimada  $l$ . Por lo tanto, al realizarse la sustitución de  $\hat{x}_i$  en (3.15) se obtiene:

$$c = |x_i - (a_x y_i + b_x)| \quad (3.16)$$

Se sabe que el ángulo  $\theta$  formado entre los vectores  $c$  y  $d$  es el mismo que forma la línea estimada con el eje de las abscisas por lo que se tiene lo siguiente:

$$\cos \theta = \frac{d}{c} \quad (3.17)$$

$$d = |x_i - (a_x y_i + b_x)| |\cos \theta| \quad (3.18)$$

Planteando entonces el siguiente problema de optimización

$$\min_{a_x, b_x} \frac{1}{m} \sum_{i=1}^m d_i \quad (3.19)$$

El problema de optimización básico presentado en (3.14) y (3.19) ha sufrido dos modificaciones: sólo aquellas distancias que sean menores a un valor de umbral  $U$  sugerido serán tomadas en cuenta y al mismo tiempo se maximizará el número de

puntos que forman parte del ajuste. De tal forma que el nuevo problema consistirá en minimizar la siguiente función:

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

Donde  $n$  es el número total de puntos y  $m$  el número de puntos cuya distancia ortogonal es menor a  $U$ .

Para generalizar el problema se puede realizar el cálculo de la distancia ortogonal del punto a la recta estimada en su representación vectorial, v.g. se obtendrá la distancia euclidiana entre el punto conocido  $c$  y el punto de contacto  $d$ , como se muestra en la figura 3.5.

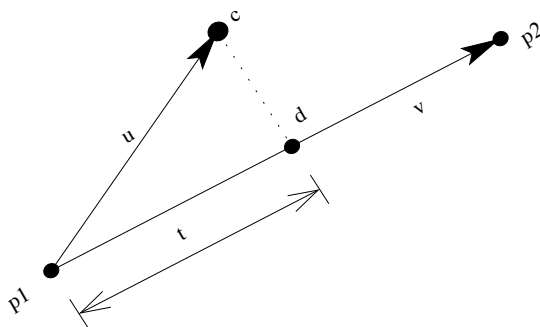


Figura 3.5: Modelo utilizado de la línea para representación vectorial.

Se da por hecho que se conoce el punto final e inicial de la recta estimada ( $p_1$  y  $p_2$ ), por lo que se puede obtener el vector  $\vec{v}$  que va de  $p_1$  a  $p_2$ . Se conoce de igual forma el punto  $c$ , el cual es un elemento del conjunto de puntos conocido, entonces

$$\vec{v} = p_2 - p_1 \quad (3.21)$$

$$\vec{u} = c - p_1 \quad (3.22)$$

Una vez que se han calculado  $\vec{u}$  y  $\vec{v}$  se puede obtener la distancia  $t$  a la que se encuentra el punto de contacto  $d$  del punto inicial  $p_1$  de la siguiente forma:

$$t = \vec{u} \cdot \frac{\vec{v}}{\|\vec{v}\|} \quad (3.23)$$

y conocer el punto  $d$  a partir de

$$d = p_1 + t \frac{\vec{v}}{\|\vec{v}\|} \quad (3.24)$$

al substituir (3.23) en (3.24) se obtiene:

$$d = p_1 + \left( \vec{u} \cdot \frac{\vec{v}}{\|\vec{v}\|} \right) \frac{\vec{v}}{\|\vec{v}\|} \quad (3.25)$$

La distancia ortogonal está dada por la distancia euclidiana entre  $d$  y  $c$

$$D = \|d - c\|, \quad (3.26)$$

generando el siguiente problema de optimización.

$$\min_{p_1, p_2} \frac{1}{m} \sum_{i=1}^m D_i \quad (3.27)$$

La distancia Euclidiana utilizada involucra como función característica  $f(x) = \sqrt{x}$  cuya derivada  $\frac{df(x)}{dx} = \frac{1}{2\sqrt{x}}$  presenta una discontinuidad en su mínimo v.g.,  $\frac{df(x)}{dx} \rightarrow \infty$  cuando  $x \rightarrow 0$  y su valor no está definido para valores negativos de su argumento. El problema de optimización que se presenta no puede resolverse de forma analítica con un método tradicional (p.ej. Levenberg-Marquart), por lo que es necesario utilizar una heurística (p.ej. la evolución diferencial).

El problema de optimización propuesto en (3.27) fue sometido a las mismas modificaciones sugeridas anteriormente, por lo que el nuevo problema de optimización minimizará la siguiente función:

$$g_1(x) = \frac{1}{m} \sum_{\text{si } D_i < U} D_i + \frac{2U(n-m)}{n} \quad (3.28)$$

## 3.2. Ajuste MMC y RANSAC

El método de la mínima mediana de los cuadrados (MMC) es un estimador robusto que proporciona buenos resultados aún cuando los datos se encuentran contaminados con el 50 % de puntos atípicos [1]. En el problema de ajuste de la línea, se definió de tal forma que se encontrara el vector  $\mathbf{x} = [a_y, b_y, a_x, b_x]^T$  tal que la siguiente función sea minimizada:

$$g(x) = \text{mediana} \sum_{i=1}^n d^2(\mathbf{x}) \quad (3.29)$$

donde  $d(\mathbf{x})$  representa la distancia ortogonal entre el punto conocido  $\mathbf{p}_i$  y la recta estimada de acuerdo a (3.2) ó (3.4) según sea el caso. Para resolver la ecuación (3.29) se deben calcular todas las combinaciones de dos puntos de un total de  $n$  puntos v.g.  $C(n, 2) = n!/((n-2)!)2!$  el cual representa un número muy grande de combinaciones. En lugar de ello se seleccionan aleatoriamente dos puntos, de tal forma que se garantice, bajo alguna probabilidad, que ambos puntos pertenecen a la línea que se busca. El algoritmo 1 que resuelve (3.29) se llamará MMC + RANSAC.

**Algoritmo 1** MMC + RANSAC**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ **Entrada:** Probabilidad de que los puntos a ajustar sean *inliers*  $\epsilon_0$  (entre 0.95 y 0.99)**Entrada:** Número de puntos de prueba  $k$  (mínimo 2 para la línea)**Entrada:** Probabilidad de encontrar puntos atípicos en el conjunto de datos  $\epsilon_1$  (usualmente 0.5)**Salida:** Parámetros desconocidos de la recta  $(a_y, b_y, a_x, b_x)$ 

- 1: Determinar el número de iteraciones a realizar a partir de  $m = \frac{\log(1-\epsilon_0)}{\log(1-(1-\epsilon_1)^k)}$
- 2:  $i = 0$
- 3: **while**  $i \neq m$  **do**
- 4:   Seleccionar aleatoriamente un subconjunto de  $k$  puntos  $Q$  del conjunto de datos  $epsilon_0$
- 5:   Con los dos puntos seleccionados obtener la pendiente inicial  $m_0 = \frac{y_2 - y_1}{x_2 - x_1}$
- 6:   **if**  $|m_0| \leq 1$  **then**
- 7:     Realizar un ajuste inicial utilizando  $Q$
- 8:     Calcular los parámetros desconocidos  $a_y$  y  $b_y$  a partir de (3.6) y (3.7)
- 9:     Obtener las distancias ortogonales  $R = \{r_1, \dots, r_n\}$  de cada punto del conjunto  $P$  a la recta ajustada inicialmente a partir de (3.26).
- 10:   **else**
- 11:     Realizar un ajuste inicial utilizando  $Q$
- 12:     Calcular los parámetros desconocidos  $a_x$  y  $b_x$  a partir de (3.8) y (3.9)
- 13:     Obtener las distancias ortogonales  $R = \{r_1, \dots, r_n\}$  de cada punto del conjunto  $P$  a la recta ajustada inicialmente a partir de (3.26).
- 14:   **end if**
- 15:   **if** el valor de umbral no ha sido definido **then**
- 16:     Ordenar el conjunto de residuales  $R$
- 17:     Obtener la mediana del conjunto de residuales  $R$
- 18:     Calcular el factor de escala inicial a partir de  $\hat{\sigma}_0 = 1.4826 \left(1 + \frac{5}{n-k}\right) \text{med.}$
- 19:     Calcular el conjunto de pesos inicial  $W_0$  a partir de  $w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_0} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_0} > 2.5 \end{cases}$
- 20:     **if** mediana actual es menor a la obtenida en la iteración anterior **then**
- 21:       Calcular el factor de escala final a partir de  $\hat{\sigma}_f = \sqrt{\frac{\sum_{i=1}^n w_i r_i^2}{\sum_{i=1}^n w_i - k}}$
- 22:       Calcular el conjunto pesos finales  $W_f$  para garantizar que el conjuntos de datos se encuentra libre de *outliers*  $w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_f} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_f} > 2.5 \end{cases}$
- 23:       Realizar el ajuste final de parámetros con los puntos de  $P$  cuyo peso final  $w_i = 1$  para  $i = 1, \dots, n$ , siguiendo el caso seleccionado anteriormente.
- 24:       Almacenar el conjunto de parámetros obtenidos en el ajuste final
- 25:     **end if**



---

```
26:  else
27:    for  $j = 1$  to  $n$  do
28:      if  $r_j < Umbral$  then
29:         $w_j = 1$ 
30:      else
31:         $w_j = 0$ 
32:      end if
33:    end for
34:    Realizar el ajuste final con los pesos obtenidos siguiendo el caso seleccionado
    anteriormente.
35:    if Error generado del ajuste final es menor al obtenido en la iteración anterior
    then
36:      Almacenar el error generado y el conjunto de parámetros de la recta
37:    end if
38:  end if
39:   $i \leftarrow i + 1$ 
40: end while
```

---

### 3.3. Suma de las distancias ortogonales y evolución diferencial

A continuación se presenta el algoritmo de SD + ED que resolverá (3.28). Para representar adecuadamente el problema planteado por el modelo presentado en la figura 3.5, la evolución diferencial utilizará dos variables  $(var_1, \theta)$ , las cuales representarán el punto inicial y la dirección de la recta, por lo que es necesario realizar una decodificación de las variables antes de utilizarse. La variable que contendrá la dirección de la recta proporcionará el ángulo de inclinación de ésta, el cual se encontrará entre  $0^\circ$  y  $360^\circ$ , mientras que  $var_1$  se encontrará entre 0 y  $2(H + W)$ , donde  $H$  y  $W$  son el largo y ancho de la imagen.

Para realizar la decodificación de las variables se tomó el modelo presentado en la figura 3.6, donde se representa la ventana de la imagen que contiene al conjunto de puntos conocido. Dicha decodificación entregará como producto final las coordenadas del punto inicial  $\mathbf{p}_0$  de la recta propuesta y el vector de dirección  $(\vec{v})$  de ésta.

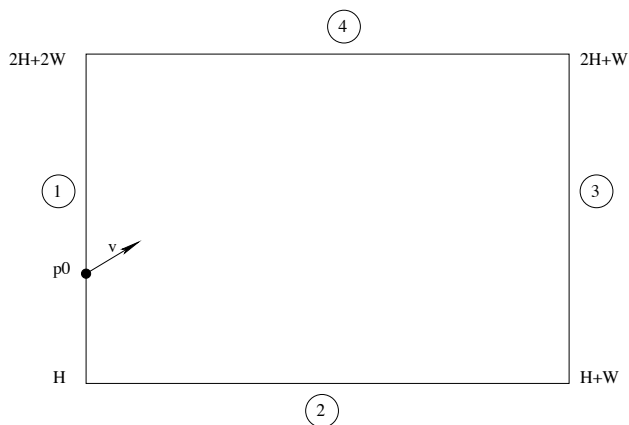


Figura 3.6: Modelo utilizado para la representación del problema en la Evolución diferencial.

Se puede observar que la ventana se ha dividido en cuatro regiones limitadas por los valores de  $H$  y  $W$  que representan la altura y anchura de la imagen donde se realizará la búsqueda de la recta. Como se mencionó anteriormente el ángulo de dirección de la recta que determinara al vector  $\vec{v}$  va de  $0^\circ$  y  $360^\circ$ .

Para que la dirección de la recta concuerde con el modelo presentado, es necesario reparar dicho valor con la finalidad de que los valores que éste proporcione sean válidos p.ej. supongamos que el punto inicial de la recta  $\mathbf{p}_0$  se encuentra en la región 3, v.g. que el valor de la variable  $var_1 \leq 2H + W$  y que el ángulo de dirección  $(\theta)$  presenta un valor de  $45^\circ$  como se observa en la figura 3.7.

Este valor se encuentra fuera del marco de la imagen definida, por lo que se debe obtener el vector de dirección en sentido contrario, v.g. sumarle  $180^\circ$  al ángulo proporcionado para reparar el valor de  $\theta$ .

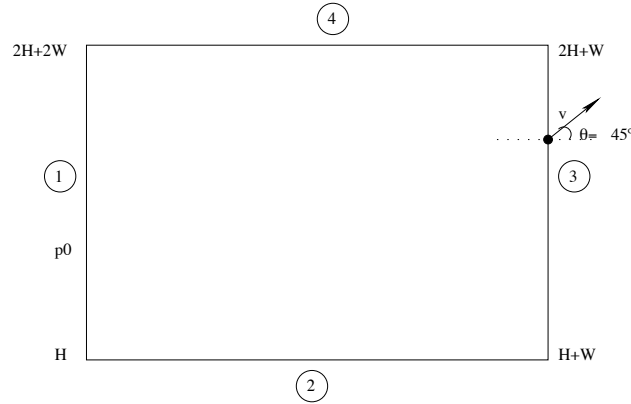


Figura 3.7: Caso de ejemplo para explicar la decodificación de las variables en la Evolución diferencial.

A continuación se proporciona el pseudocódigo de la función *decodificar()* que proporcionara el punto inicial  $p_0$  de la recta, el vector de dirección  $\vec{v}$  y realiza la reparación del individuo a partir de  $var_1$  y  $\theta$ .

---

**Algoritmo 2** *decodificar*( $\mathbf{x}$ )

---

**Entrada:** Vector  $\mathbf{x}$  que representara los parámetros de la recta ( $var_1, \theta$ )

**Salida:** Punto inicial  $\mathbf{p}_0$  y vector de dirección  $\vec{v}$

```

1: region  $\leftarrow 0$ 
2:  $u \leftarrow 0$ 
3: ang  $\leftarrow 0$ 
4:  $x \leftarrow 0$ 
5:  $y \leftarrow 0$ 
6: bandera  $\leftarrow 0$ 
7: if  $var_1 < H$  then
8:   region  $\leftarrow 1$ 
9:    $u \leftarrow H - var_1$ 
10: else
11:   if  $var_1 < H + W$  then
12:     region  $\leftarrow 2$ 
13:      $u \leftarrow var_1 - H$ 
14:   else
15:     if  $var_1 < 2H + W$  then
16:       region  $\leftarrow 3$ 
17:        $u \leftarrow var_1 - H - W$ 

```

---

```
18:   else
19:     region ← 4
20:     u ← 2(H + W) - var1
21:   end if
22: end if
23: end if
24: if region == 1 then
25:   p0 ← [0, u]
26: else
27:   if region == 2 then
28:     p0 ← [u, 0]
29:   else
30:     if region == 3 then
31:       p0 ← [W, u]
32:     else
33:       p0 ← [u, H]
34:     end if
35:   end if
36: end if
37: ang ←  $\frac{\theta\pi}{180}$ 
38: x ← cos(ang)
39: y ← sin(ang)
40: if region == 1 then
41:   if x < 0 then
42:     bandera ← 1
43:   end if
44: else
45:   if region == 2 then
46:     if y < 0 then
47:       bandera ← 1
48:     end if
49:   else
50:     if region == 3 then
51:       if x > 0 then
52:         bandera ← 1
53:       end if
54:     else
55:       if y > 0 then
56:         bandera ← 1
57:       end if
58:     end if
59:   end if
60: end if
```

---

---

```
61: if bandera == 1 then
62:    $x \leftarrow -x$ 
63:    $y \leftarrow -y$ 
64:    $\theta \leftarrow \theta + 180$ 
65:   if  $\theta > 360$  then
66:      $\theta \leftarrow \theta - 360$ 
67:   end if
68: end if
69:  $\vec{v} \leftarrow [x, y]$ 
```

---

Una vez que se ha definido la función *decodificar()*, se puede realizar el cálculo de la distancia ortogonal de la recta estimada a un elemento de conjunto conocido de puntos como se planteó en el modelo propuesto en la figura 3.5. Se presenta el pseudocódigo que permite calcular la distancia ortogonal entre un punto dado y la recta propuesta.

---

**Algoritmo 3** *evaluar(x)*

---

**Entrada:** Vector  $\mathbf{x}$  que representará los parámetros de la línea ( $var_1, \theta$ )

**Salida:** Valor de aptitud del individuo

```
1: Error  $e = 0$ 
2: decodificar(x)
3: Número de puntos  $m = 0$ 
4: for  $j = 0$  to  $Np$  do
5:    $r \leftarrow distanciaOrtogonal(p_0, \vec{v})$  a partir de (3.26)
6:   if  $r < U$  then
7:      $e \leftarrow e + r$ 
8:      $m \leftarrow m + 1$ 
9:   end if
10: end for
11: aptitud  $\leftarrow \frac{te}{m} + \frac{2U(Np-m)}{Np}$ 
```

---

El algoritmo 4 resuelve (3.28) utilizando la evolución diferencial y la suma de las distancias ortogonales como función de error.

---

**Algoritmo 4** SD +ED
 

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Alto y ancho de la imagen  $(W, H)$

**Entrada:** Tamaño de la población inicial  $\mu$ ; número máximo de generaciones  $G_{max}$ ; factor de recombinación  $CR$ ; factor de diferencia  $F$ ; valor de umbral  $U$

**Salida:** Parámetros desconocidos  $(a_y, b_y, a_x, b_x)$

- 1: Generar la población inicial  $X = \{x_1, \dots, x_\mu\}$  (se utilizó la distribución de Halton)
  - 2: Evaluar cada individuo en X:  $evaluar(x_i)$  para  $i = 1, \dots, \mu$
  - 3:  $k = 0$
  - 4: **repeat**
  - 5:   **for**  $j = 1$  to  $\mu$  **do**
  - 6:     Sean  $I_1, I_2$  e  $I_3$  tres enteros aleatorios en  $[1, \mu]$ , tal que  $I_1 \neq I_2 \neq I_3$
  - 7:     Sea  $i_{rand}$  un entero aleatorio en  $[1, numvar]$
  - 8:     **for**  $i = 1$  to  $numvar$  **do**
  - 9:       
$$x'_{i,j} = \begin{cases} x_{i,I_3} + F(x_{i,I_1} - x_{i,I_2}) & : \text{rndreal}(0, 1) < R \text{ ó } i = i_{rand} \\ x_{i,j} & : \text{de otro modo} \end{cases}$$
  - 10:     **end for**
  - 11:     **if**  $evaluar(x'_{i,j}) < evaluar(x_{i,j})$  **then**
  - 12:        $X_j = X'_j$
  - 13:     **end if**
  - 14:   **end for**
  - 15:    $min =$  valor de aptitud más pequeño de la población X
  - 16:    $max =$  valor de aptitud más grande de la población X
  - 17:    $k = k + 1$
  - 18: **until**  $max - min < s$  ó  $k > G_{max}$
- 

### 3.4. Extracción de varias líneas

Los algoritmos presentados (MMC + RANSAC y SD + ED), pueden ser utilizados para realizar la extracción de varias líneas de un conjunto de puntos dado, con la única diferencia de que es necesario definir un umbral  $U$  para MMC + RANSAC ya que el umbral automático que maneja por defecto no funciona para dicha causa. Una vez que una línea es detectada, los puntos que pertenecen a dicho ajuste son eliminados del conjunto de puntos y se reanuda la búsqueda. El pseudocódigo para la extracción de varias líneas se muestra en el algoritmo 5.

---

**Algoritmo 5** Algoritmo para la extracción de varias líneas

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Parámetros para la ejecución de MMC+ANSAC ó SD + ED

**Entrada:** Definir un valor de umbral  $U$  para ambos métodos

**Entrada:** Número de líneas que se desean extraer  $N_l$

**Salida:** Parámetros desconocidos de las  $N_l$  líneas  $(a_y, b_y, a_x, b_x)$

```

1: Sea  $N_p$  el número de total de puntos conocidos
2: for  $i = 1$  to  $N_l$  do
3:   Llamar MMC + RANSAC ó SD + ED
4:   Almacenar los parámetros entregados por el método elegido
5:   for  $j = 1$  to  $n$  do
6:     Calcular el residual  $r$  (distancia ortogonal) del punto  $p_j$  a la línea con los
       parámetros entregados anteriormente
7:     if  $r < U$  then
8:       Eliminar el punto  $p_j$  del conjunto de puntos conocidos
9:        $n \leftarrow n - 1$ 
10:    end if
11:  end for
12: end for

```

---

### 3.5. Pruebas

Para determinar el desempeño de los algoritmos presentados (MMC + RANSAC y SD + ED) se generaron seis pruebas: dos pruebas de extracción de una línea bajo dos condiciones distintas de ruido (Gaussiano y Laplaciano) para una recta con inclinación respecto al eje de las ordenadas mayor y menor a  $45^\circ$ , y la quinta prueba consiste en la extracción de varias líneas con el parámetro  $a_y$  fijo. La sexta prueba consistió en la extracción de varias líneas pero con el parámetro  $b_y$  fijo. En cada uno de los experimentos se realizaron 100 ejecuciones bajo as distintas condiciones propuestas. En todos los experimentos se utilizaron los siguientes parámetros para la evolución diferencial:

- Ancho de la imagen: 640
- Altura de la imagen: 480
- Población: 30 individuos
- Constante de diferencia  $F$ : 0.7
- Constante de recombinación  $R_c$ : 0.9
- Condición de paro  $s$ : 0.01
- Número máximo de generaciones: 10,000

- Valor de umbral  $U$ : 10

Para la primera y segunda prueba se definió una recta con una elevación sobre el eje de las abscisas de -250, v.g  $b_y = -250$  y con un ángulo de inclinación respecto al eje de las ordenadas de  $70^\circ$ . La línea está conformada por 100 puntos sobre los cuales se aplicó ruido Gaussiano con media cero y desviación estándar de 3 ( $\sigma = 3$ ) para la primera prueba. En la segunda prueba se utilizó la misma línea, pero esta vez fue sometida a condiciones de ruido Laplaciano con un factor de escala de 2 ( $b = 2$ ). En ambas pruebas se adicionaron puntos atípicos que van del 0% al 100% del total de los puntos de la recta. Los resultados de dichos experimentos se muestran en las figuras 3.8 y 3.9 donde se observa la media y desviación estándar de cada parámetro bajo los diferentes porcentaje de puntos atípicos.

En el tercer y cuarto experimento se definió una recta con una elevación sobre el eje de las abscisas de 50, v.g  $b_y = 50$  y con un ángulo de inclinación respecto al eje de las ordenadas de  $20^\circ$ . La línea está conformada por 100 puntos sobre los cuales se aplicó ruido Gaussiano con media cero y desviación estándar de 3 ( $\sigma = 3$ ) para la tercera prueba. En la cuarta prueba se utilizó la misma línea, pero esta vez fue sometida a condiciones de ruido Laplaciano con un factor de escala de 2 ( $b = 2$ ). Se adicionaron de igual forma puntos atípicos que van del 0% al 100% del total de los puntos de la recta. Los resultados se pueden observar en las figuras 3.10 y 3.11.

La quinta prueba consistió en generar de 1 a 10 rectas con 100 puntos adicionando ruido Gaussiano sobre éstos con media cero y desviación estándar de 3. Dichas rectas mantuvieron fijo el factor de elevación respecto al eje de las abscisas en 250, v.g.  $b_y = 250$ , mientras que el ángulo de inclinación se varió para cada una de las líneas de  $0^\circ$  a  $180^\circ$ , el número de puntos atípicos se mantuvo en el 90%. Los resultados de la prueba se presentan en la figura 3.12.

La última prueba consistió en generar de 1 a 10 bajo las mismas condiciones de ruido y de puntos atípicos del experimento anterior, dichas rectas mantuvieron fijo el ángulo de inclinación respecto al eje de las ordenadas en  $30^\circ$ , mientras que el factor de elevación respecto a ambos ejes se varió aleatoriamente procurando que las rectas no se interceptaran. Los resultados del experimento se pueden ver en la figura 3.13.

En la figura 3.14 se muestra el ajuste resultante del primer y tercer experimento donde se proponen dos líneas con inclinación respecto al eje  $x$  menor y mayor a  $45^\circ$  además de adicionarle a los puntos ruido Gaussiano con media 0 y desviación estándar de 3. El ajuste que se muestra, contiene el 100% de puntos atípicos y los resultados de SD + ED y MMC + RANSAC.

En la figura 3.15 se muestra el ajuste resultante del segundo y cuarto experimento donde se proponen dos líneas con inclinación respecto al eje  $x$  menor y mayor a  $45^\circ$  además de adicionarle a los puntos ruido Laplaciano con un factor de escala de



2. El ajuste que se muestra, contiene el 100 % de puntos atípicos y los resultados de SD + ED y MMC + RANSAC.

Se muestra también en la figura 3.16 el ajuste resultante de los experimentos cinco y seis, los cuales consisten en realizar la extracción de varias líneas bajo las condiciones descritas anteriormente. Los resultados que se muestran contienen ocho líneas con sus respectivos ajustes utilizando SD + ED y MMC + RANSAC.

### 3.6. Sumario

De los experimentos realizados se puede deducir que SD + ED es superior a MMC + RANSAC, ya que los resultados proporcionados por SD + ED presentan una variabilidad mucho menor que los entregados por MMC + RANSAC al extraer una línea cuando el porcentaje de puntos contaminados es mayor al 70 %. Al utilizar la suma de las distancias ortogonales como criterio para el problema de optimización se consiguen mejores resultados al realizar la estimación de los parámetros desconocidos de la recta, ya que ésta resulta más resistente a la presencia de puntos contaminados. Se puede observar que SD + ED logró extraer exitosamente las líneas propuestas en el 100 % de los casos. El maximizar el número de puntos que forman parte del ajuste permite que SD + ED tenga un mejor desempeño cuando las condiciones de ruido superan el 70 % de puntos atípicos.

La función de agregación propuesta que resuelve ambos problemas de optimización (minimización del error y maximización del número de puntos del ajuste) funcionó adecuadamente, ya que al utilizar el factor  $2U(n - m)/n$ , donde  $n$  es el número de puntos totales y  $m$  el total de puntos que pertenecen al ajuste. El método utilizado propondrá líneas que utilicen todos o la mayoría de los puntos que se conocen, ya que si se usan todos los puntos dicho factor tenderá a cero, mientras que si se usan sólo algunos tiende al error máximo que es  $2U$ .

El utilizar la SD + ED tiene la desventaja de incrementar enormemente el número de evaluaciones de la función objetivo mientras que MMC + RANSAC mantiene un número de evaluaciones mucho más bajo, haciendo que éste sea más rápido que SD + ED.

Se propone un algoritmo para el ajuste de una o varias líneas bajo condiciones donde la cantidad de ruido supera el 70 % de puntos atípicos. Este algoritmo minimiza la suma de las distancias ortogonales de un conjunto de puntos dado a la recta estimada, y al mismo tiempo maximiza el número de puntos que pertenecen al ajuste. Esto genera un problema de optimización no lineal, el cual fue resuelto utilizando la evolución diferencial. El algoritmo propuesto muestra un buen desempeño, ya que es capaz de detectar y extraer líneas el 100 % de las veces en condiciones de ruido que van del 0 % al 100 % de puntos contaminados.

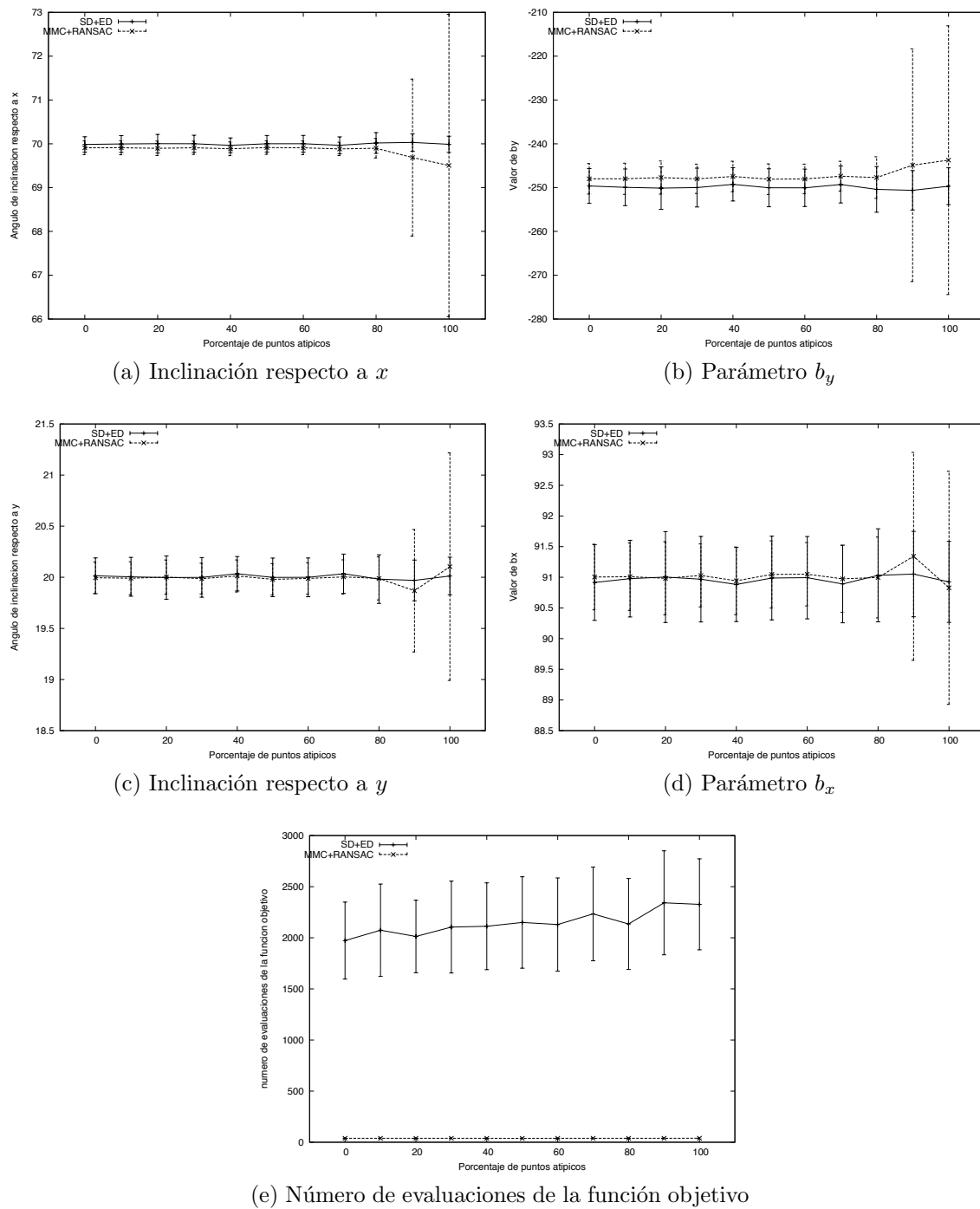
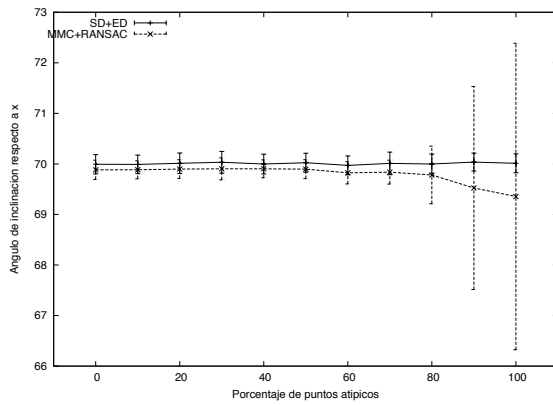
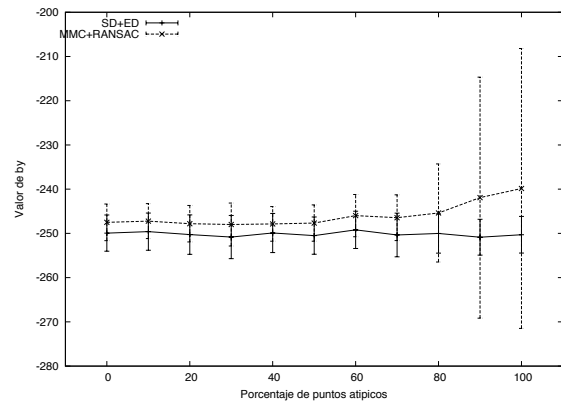
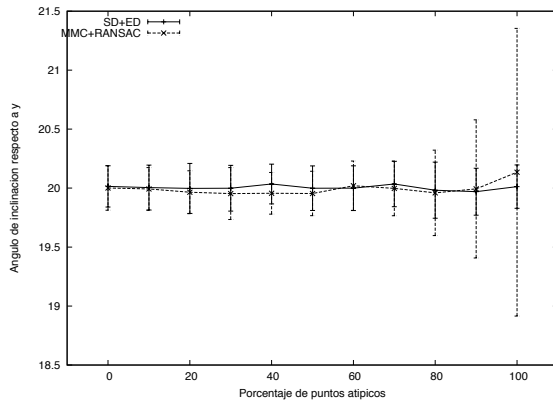
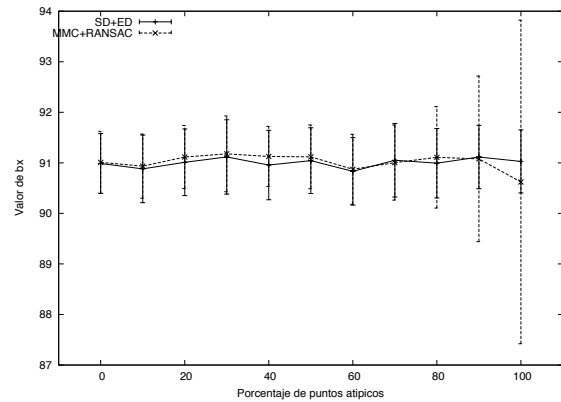
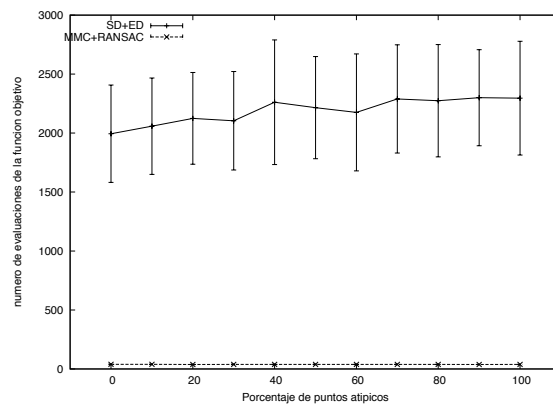


Figura 3.8: Resultados de la primera prueba con ruido Gaussiano sobre los puntos de la recta.

(a) Inclinación respecto a  $x$ (b) Parámetro  $b_y$ (c) Inclinación respecto a  $y$ (d) Parámetro  $b_x$ 

(e) Número de evaluaciones de la función objetivo

Figura 3.9: Resultados de la segunda prueba con ruido Laplaciano sobre los puntos de la recta.

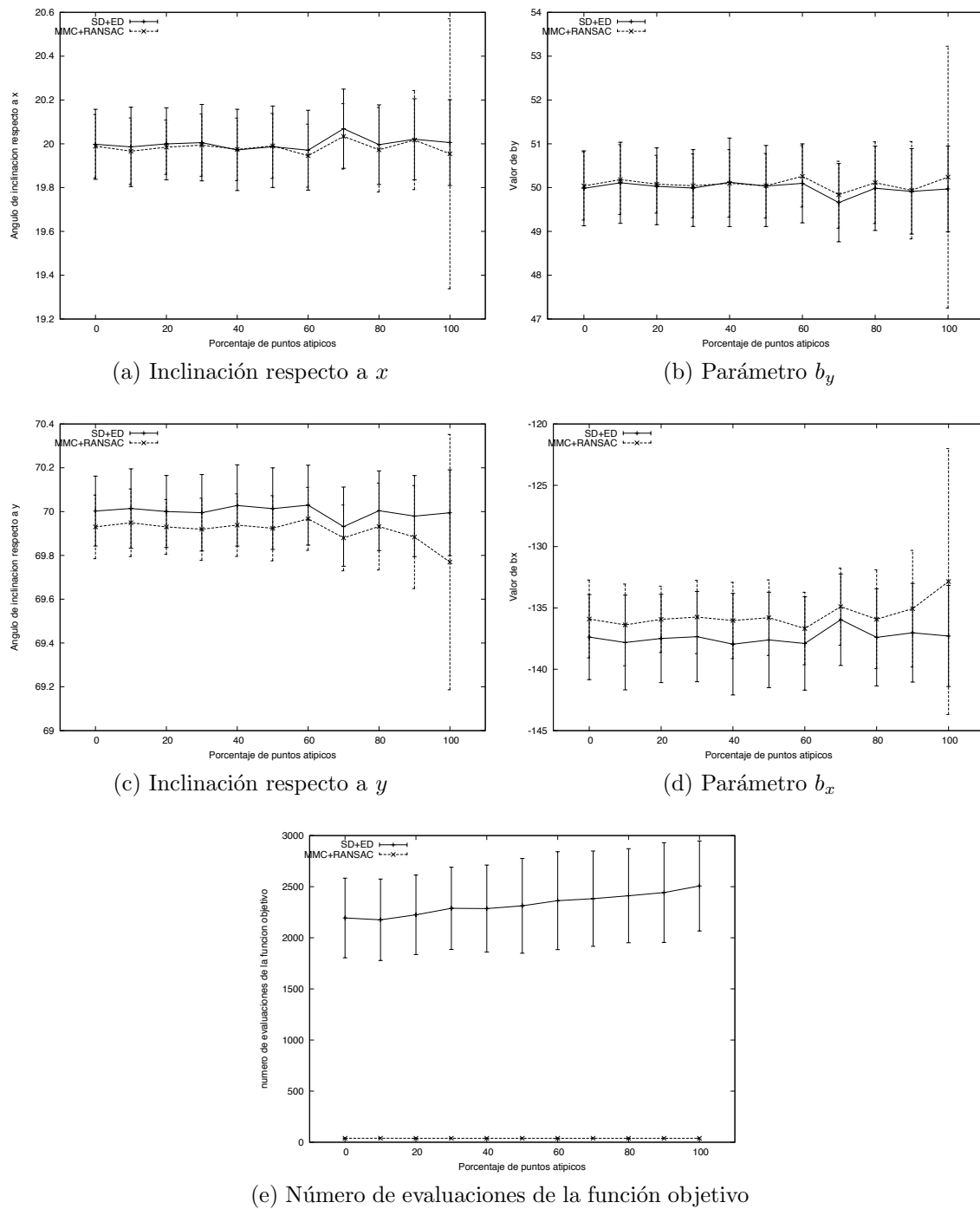
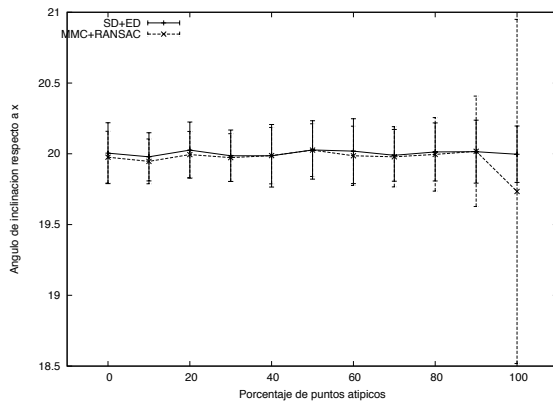
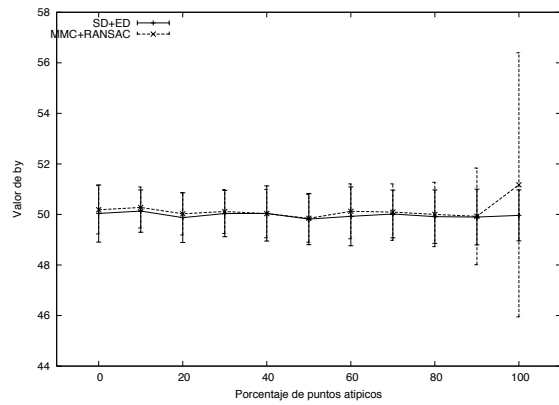
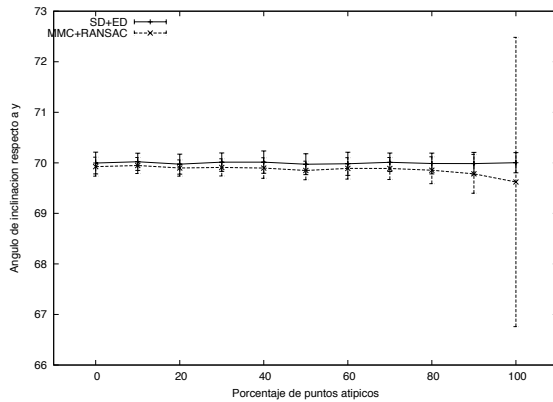
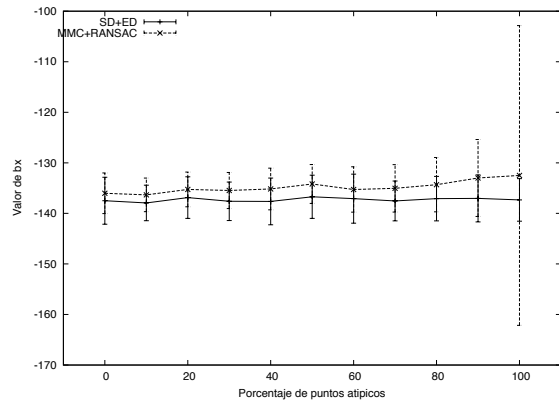
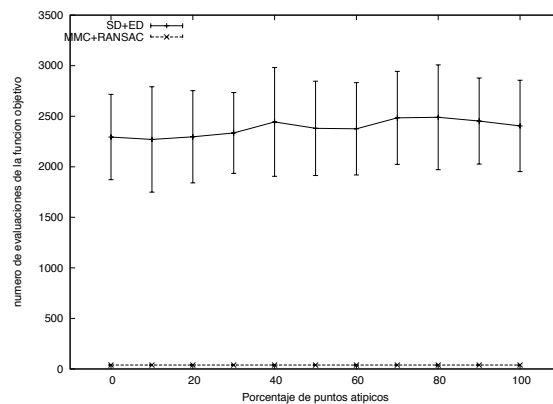
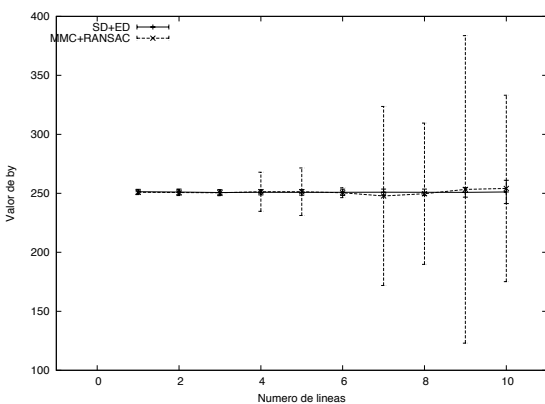
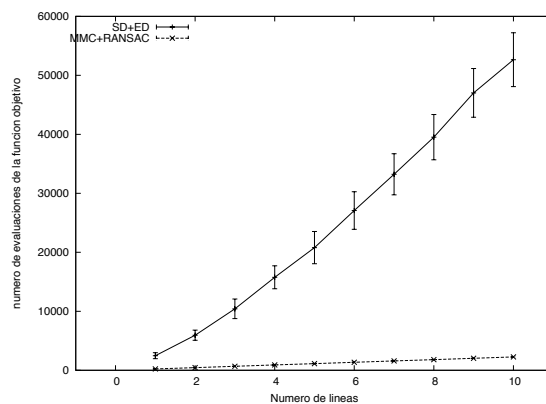


Figura 3.10: Resultados de la tercera prueba con ruido Gaussiano sobre los puntos de la recta.

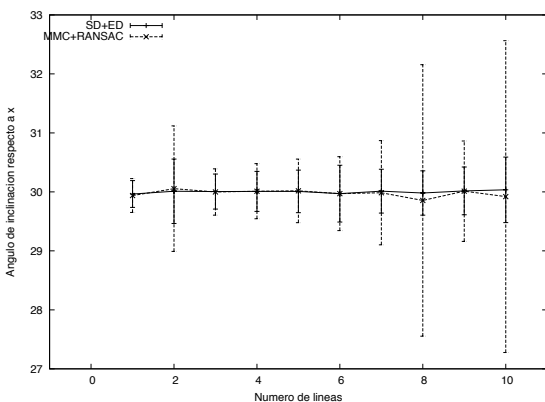
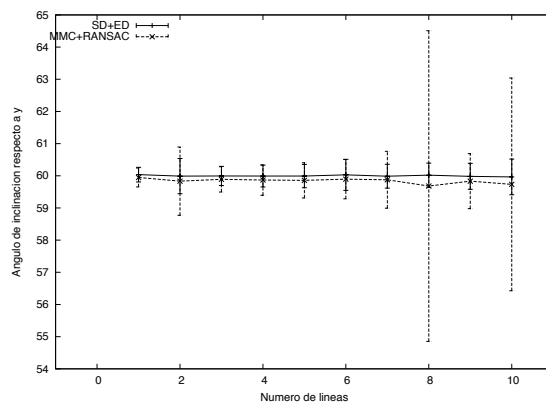
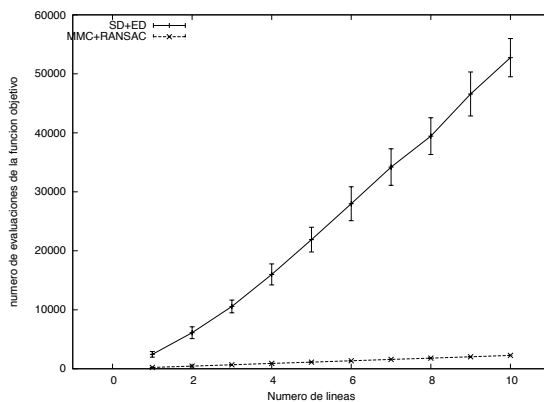
(a) Inclinación respecto a  $x$ (b) Parámetro  $b_y$ (c) Inclinación respecto a  $y$ (d) Parámetro  $b_x$ 

(e) Número de evaluaciones de la función objetivo

Figura 3.11: Resultados de la cuarta prueba con ruido Laplaciano sobre los puntos de la recta.

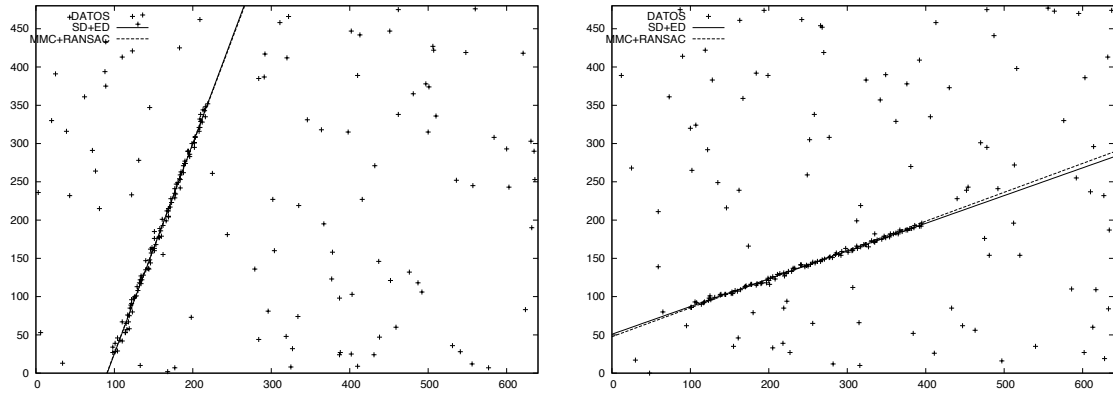
(a) Parámetro  $b_y$ 

(b) Número de evaluaciones de la función objetivo

Figura 3.12: Extracción de varias líneas con  $b_y$  fijo.(a) Inclinación respecto a  $x$ (b) Inclinación respecto a  $y$ 

(c) Número de evaluaciones de la función objetivo

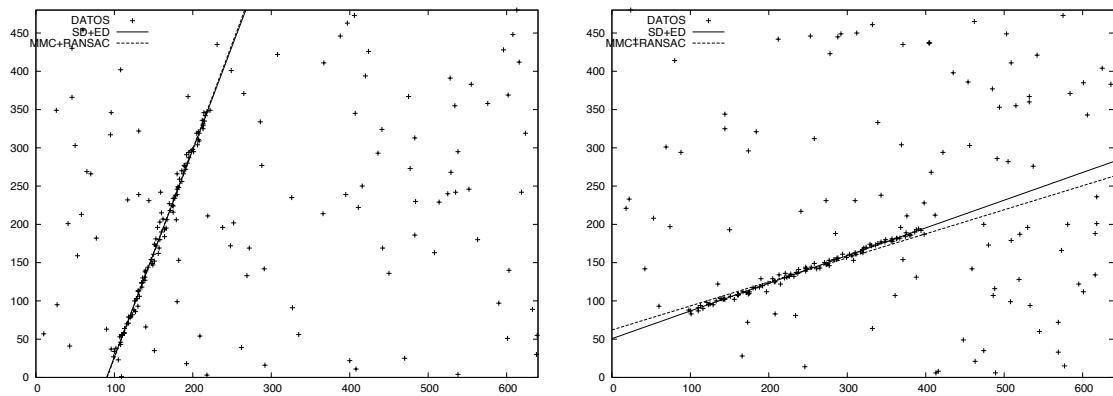
Figura 3.13: Extracción de varias líneas con el ángulo de inclinación respecto a  $x$  y  $y$  fijo.



(a) Ajuste resultante del primer experimento con 100 % de puntos atípicos

(b) Ajuste resultante del tercer experimento con 100 % de puntos atípicos

Figura 3.14: Ajuste obtenido de los experimentos 1 y 3.



(a) Ajuste resultante del segundo experimento con 100 % de puntos atípicos

(b) Ajuste resultante del cuarto experimento con 100 % de puntos atípicos

Figura 3.15: Ajuste obtenido de los experimentos 2 y 4.

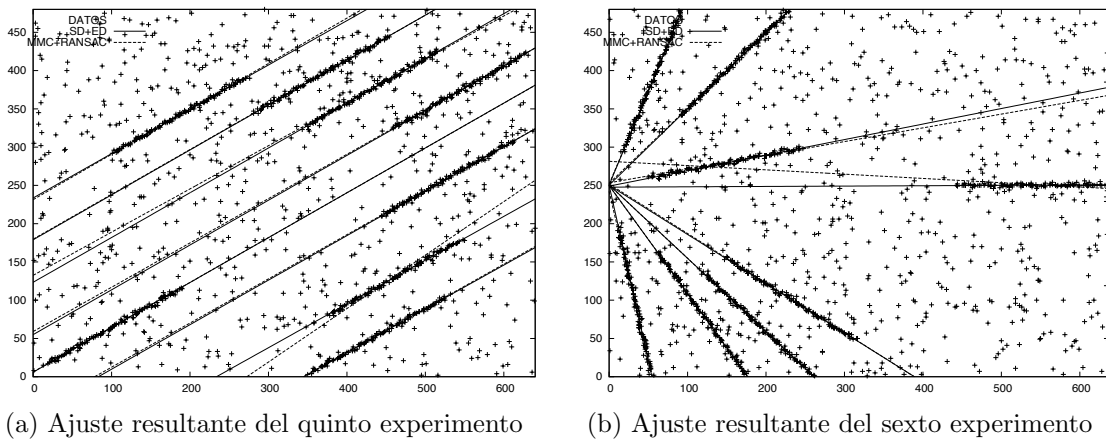


Figura 3.16: Ajuste obtenido de los experimentos 5 y 6.



# Capítulo 4

## Ajuste de elipses

---

El ajuste de formas geométricas es un problema importante en muchos campos de la ciencia y la ingeniería (p.ej. detección de rostros [14], control de calidad [15], astronomía, etc.). En particular, las secciones cónicas son las formas geométricas más comunes en aplicaciones de procesamiento de imágenes. El ajuste de elipses es una tarea importante en Visión por Computadora debido a que la proyección de un círculo puede ser vista como una elipse. Las principales alternativas que existen para la detección y análisis de formas geométricas son la transformada de Hough y el método de momentos [16, 17].

En este capítulo se resuelve el ajuste de elipses a un conjunto de puntos conocidos bajo condiciones de mucho ruido (v.g. más del 50% de los puntos conocidos se consideran atípicos), utilizando la suma de las distancias ortogonales de los puntos conocidos a la elipse ajustada. El problema de ajuste es ahora un problema no lineal y es resuelto utilizando la evolución diferencial. El método propuesto es comparado contra la mínima mediana de los cuadrados (MMC) con RANSAC, demostrando que éste obtiene mejores resultados del ajuste de la elipse incluso cuando el número de puntos atípicos representa del 50% al 100% de los puntos conocidos.

### 4.1. Descripción del problema

Dado un conjunto de puntos  $\mathbf{p}_i \in \mathbb{R}^2$ , para  $1 \leq i \leq m$ , en un sistema de coordenadas globales  $xy$ , se buscará la mejor elipse que represente el conjunto de puntos dado. Dichos puntos deben ser trasladados al nuevo origen  $\mathbf{p}_c = [x_c, y_c]^T$  y rotados un ángulo  $\phi$  en el nuevo sistema de coordenadas  $x'y'$  (ver figura 4.1), de la forma

$$\mathbf{p}'_i = R(\mathbf{p}_i - \mathbf{p}_c) \quad (4.1)$$

donde

$$R = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}$$

En el sistema de coordenadas locales los puntos  $\mathbf{p}'_i$  son parte de la elipse en su forma

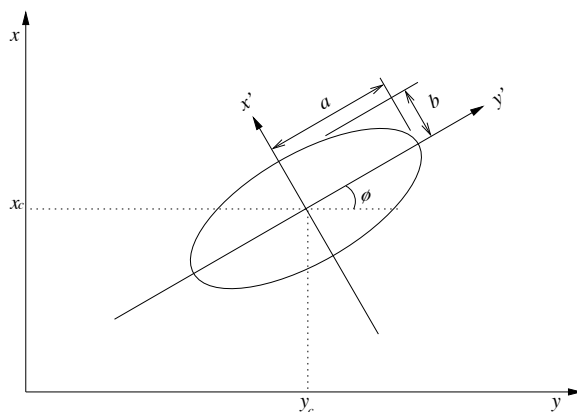


Figura 4.1: Modelo utilizado de la elipse.

canónica

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

donde  $a$  y  $b$  representan el semieje mayor y menor, respectivamente, como se puede observar en la figura 4.1. La elipse se representa entonces por sólo cinco parámetros  $(a, b, x_c, y_c, \phi)$ . Para determinar cuál es la elipse que mejor se ajusta al conjunto de puntos conocidos se utilizará como función de costo o error la suma de las distancias de cada punto conocido  $\mathbf{p}_i$  (en el nuevo sistema de coordenadas local  $x'y'$ ) a la elipse propuesta. Dicha distancia comúnmente es algebraica o geométrica. En este caso se utilizará la suma de las distancias ortogonales (no los cuadrados), ya que ésta no se ve afectada por la presencia de puntos atípicos. Un ejemplo de lo mencionado anteriormente se puede observar en la figura 4.2, donde se presenta un conjunto de 90 puntos que forman una elipse y 5 puntos atípicos. Se realiza el ajuste de la elipse al conjunto de puntos mencionado mediante la suma de las distancias algebraicas y la suma de las distancias ortogonales.

El uso de la suma de las distancias en lugar del cuadrado de éstas tiene una clara desventaja, la cual se explica a continuación. La distancia que se está utilizando es la distancia Euclidiana, la cual involucra como función característica  $f(x) = \sqrt{x}$  cuya derivada  $\frac{df(x)}{dx} = \frac{1}{2\sqrt{x}}$  presenta una discontinuidad en su mínimo v.g.,  $\frac{df(x)}{dx} \rightarrow \infty$  cuando  $x \rightarrow 0$  y su valor no está definido para valores negativos de su argumento. El problema de optimización que se presenta, no puede resolverse de forma analítica con un método tradicional (p.ej. Levenberg-Marquart), por lo que es necesario utilizar una heurística (p.ej. la evolución diferencial). A continuación se realizará la explicación de los métodos que se utilizaron para el ajuste de la elipse mostrada en la figura 4.2.

#### 4.1.1. Ajuste algebraico

Una elipse puede ser representada implícitamente por la ecuación general de las cónicas,  $F(\mathbf{a}, \mathbf{v}) = \mathbf{a}^T \mathbf{v} = a'x^2 + b'xy + c'y^2 + d'x + f' + g' = 0$ , donde  $\mathbf{a} =$

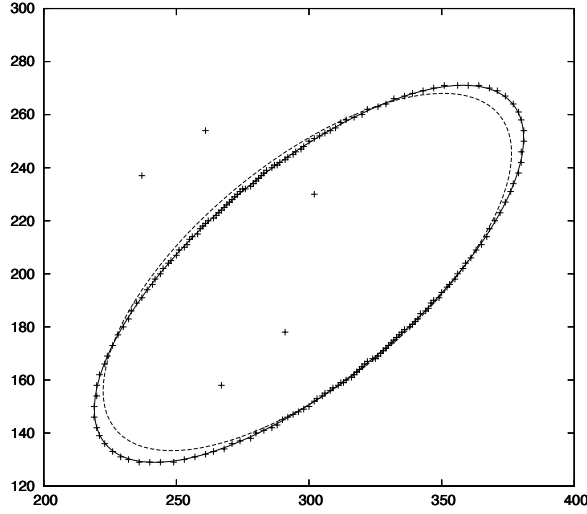


Figura 4.2: Conjunto de 90 puntos que forman una elipse con parámetros  $(100,40,300,200,40^{\circ})$  con cinco puntos atípicos, Ajuste según la suma de las distancias ortogonales (línea continua) y ajuste según la suma de las distancias algebraicas (línea punteada).

$[a', b', c', d', f', g']^T$  y  $\mathbf{v} = [x^2, xy, y^2, x, y, 1]^T$ , dicha ecuación representará una elipse si  $b'^2 - 4a'c' < 0$ . Al substituir el valor de un punto  $\mathbf{p} = [x, y]^T$  en la ecuación implícita de la elipse, el valor obtenido es llamado “distancia algebraica”. El problema de optimización obtenido consiste en minimizar la función

$$g_a(\mathbf{a}) = \sum_{i=1}^n (\mathbf{a}^T \mathbf{v}_i)^2 \quad (4.2)$$

sujeta a  $4a'c' - b'^2 = 1$ . Esta restricción de igualdad es de segundo orden y puede representarse en su forma matricial como  $\mathbf{a}^T C \mathbf{a} = 1$ , donde  $C$  representa la matriz de restricción cuya dimensión es de  $6 \times 6$

$$\mathbf{a}^T \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{a} = 1 \quad (4.3)$$

Brookstein demostró en [18] que si una restricción cuadrática se encuentra sobre los parámetros, el problema de minimización presentado en (4.2) se puede resolver mediante un sistema generalizado de eigenvectores de la forma:

$$\mathbf{v}^T \mathbf{v} \mathbf{a} = \lambda C \mathbf{a} \quad (4.4)$$

Al formular el problema de esta forma se obtiene un algoritmo de ajuste mucho más eficiente, ya que éste puede ser resuelto de forma determinística con un programa que resuelva un sistema generalizado de eigenvectores [17] representado en la ecuación (4.4). Como se tienen cinco incógnitas de los cinco parámetros de la elipse, se necesitan al menos cinco puntos conocidos para ajustar la elipse.

Una vez que se ha resuelto el eigensistema y se ha obtenido el vector de parámetros  $\mathbf{a}$  se debe realizar la transformación de éste en el conjunto de parámetros  $(a, b, x_c, y_c, \phi)$  definido inicialmente. Para realizar dicha transformación se ocuparon las fórmulas que se presentan a continuación [19]:

$$x_c = \frac{c'd' - b'f'}{b'^2 - a'c'} \quad (4.5)$$

$$y_c = \frac{a'f' - b'd'}{b'^2 - a'c'} \quad (4.6)$$

$$a = \sqrt{\frac{2(a'f'^2 + c'd'^2 + g'b'^2 - 2b'd'f' - a'c'g')}{(b'^2 - a'c') \left[ \sqrt{(a' - c')^2 + 4b'^2} - a' - c' \right]}} \quad (4.7)$$

$$b = \sqrt{\frac{2(a'f'^2 + c'd'^2 + g'b'^2 - 2b'd'f' - a'c'g')}{(b'^2 - a'c') \left[ -\sqrt{(a' - c')^2 + 4b'^2} - a' - c' \right]}} \quad (4.8)$$

$$\phi = \tan^{-1} \left( \frac{a' - c'}{2b} \right) \left( \frac{90}{\pi} \right) \quad (4.9)$$

El ángulo  $\phi$  que se obtiene a partir de (4.9) define su valor en sentido horario respecto al semieje mayor de la elipse.

La principal desventaja de utilizar el ajuste algebraico es que el ajuste se verá afectado si los valores conocidos no contienen la suficiente precisión o si algún punto atípico se encuentra presente en el conjunto de puntos conocido.

### 4.1.2. Ajuste según la suma de las distancias ortogonales

El mejor ajuste se obtiene al utilizar la suma de las distancias ortogonales, ya que éste representa la distancia geométrica real entre los puntos y la elipse estimada. Al resolver este problema se tiene que la función objetivo no tiene una derivada continua, por lo que un método numérico convencional no podría resolverlo.

El ajuste de la elipse a partir de la suma de las distancias ortogonales se define con la minimización de la función:

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

$$g_1(x) = \sum_{i=1}^n d_i \quad (4.10)$$

donde  $d_i$  es la distancia euclidiana entre el punto transformado  $\mathbf{p}'_i$  como se definió en (4.1) y el punto de contacto ortogonal  $\mathbf{p}''_i$ . Dicho punto de contacto se obtiene al minimizar la siguiente función objetivo descrita en [6].

$$g_e(t) = yb \cos t - xa \sin t + (a^2 - b^2) \cos t \sin t = 0 \quad (4.11)$$

La función (4.11) se resuelve utilizando el método de Newton con un punto el valor inicial  $t_0 = \tan^{-1}((y/a)/(x/b))$ . La función expresada en (4.11) es muy suave, por lo que el método de Newton converge en sólo tres o cuatro iteraciones [6]. Para el ajuste robusto de la elipse que depende de la suma de las distancias ortogonales, se han sugerido dos modificaciones al problema básico presentado en (4.10): sólo aquellas distancias cuyo valor sean menores a un valor de umbral  $U$  definido serán tomadas en cuenta, y al mismo tiempo se maximizara el número de puntos que pertenecen al ajuste. De tal forma que el nuevo problema consistirá en minimizar la siguiente función:

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

$$g_2(x) = \frac{2U(n-m)}{n} + \frac{1}{m} \sum_{\text{si } d_i < U} d_i \quad (4.12)$$

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

El problema expresado en (4.12) se resolvió utilizando la evolución diferencial con la condición de paro *diff* [20]: las iteraciones se detienen cuando la diferencia entre el peor y el mejor individuo es menor a un valor definido o cuando el número de iteraciones excede el número máximo de iteraciones establecidas. Este algoritmo será llamado SD + ED (suma de las distancias más evolución diferencial)

## 4.2. Ajuste MMC y RANSAC

El método de la mínima mediana de los cuadrados (MMC) es un estimador robusto que proporciona buenos resultados aún cuando los datos se encuentran contaminados con el 50 % de puntos atípicos [1]. El problema de ajuste de la elipse, se definió de tal forma que se encontrara el vector  $\mathbf{x} = [a, b, x_c, y_c, \phi]^T$  tal que la siguiente función sea minimizada.

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

$$g_3(x) = \text{mediana} \sum_{i=1}^n d^2(\mathbf{p}'_i, \mathbf{p}''_i) \quad (4.13)$$

donde  $d()$  representa la distancia euclidiana entre el punto transformado  $\mathbf{p}'_i$  de acuerdo a (4.1) y el punto de contacto ortogonal  $\mathbf{p}''_i$  el cual se obtiene al resolver (4.11).

Para resolver la ecuación (4.13) se deben calcular todas las combinaciones de 5 puntos de un total de  $n$  puntos v.g,  $C(n, 5) = n!/((n - 5)!5!)$  el cual representa un número muy grande de combinaciones. En lugar de ello se seleccionan aleatoriamente cinco puntos, de forma que se garantice bajo alguna probabilidad que los cinco puntos seleccionados pertenecerán a la elipse que se busca. El algoritmo 6 que resuelve (4.13) se llamará MMC+RANSAC.

---

**Algoritmo 6** Método MMC + RANSAC
 

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Probabilidad de que los puntos a ajustar sean *inliers*  $\epsilon_0$  (entre 0.95 y 0.99)

**Entrada:** Número de puntos de prueba  $k$  (mínimo 5 para la elipse)

**Entrada:** Probabilidad de encontrar puntos atípicos en el conjunto de datos  $\epsilon_1$  (usualmente 0.5)

**Salida:** Parámetros desconocidos de la elipse  $(a, b, x_c, y_c, \phi)$

- 1: Determinar el número de iteraciones a realizar a partir de  $m = \frac{\log(1-\epsilon_0)}{\log(1-(1-\epsilon_1)^{\epsilon_1})}$
  - 2:  $i = 0$
  - 3: **while**  $i \neq m$  **do**
  - 4:   Seleccionar aleatoriamente un subconjunto de  $k$  puntos  $Q$  del conjunto de datos  $\epsilonpsilon_0$
  - 5:   Realizar un ajuste inicial utilizando  $Q$
  - 6:   Obtener la distancias ortogonales  $R = \{r_1, \dots, r_n\}$  de cada punto del conjunto  $P$  a la elipse ajustada inicialmente
  - 7:   **if** el valor de umbral no ha sido definido **then**
  - 8:     Ordenar el conjunto de residuales  $R$
  - 9:     Obtener la mediana del conjunto de residuales  $R$
  - 10:    Calcular el factor de escala inicial a partir de  $\hat{\sigma}_0 = 1.4826 \left(1 + \frac{5}{n-p}\right)$  med.
  - 11:    Calcular el conjunto de pesos inicial  $W_0$  a partir de  $w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_0} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_0} > 2.5 \end{cases}$
  - 12:    **if** mediana actual es menor a la obtenida en la iteración anterior **then**
  - 13:     Calcular el factor de escala final a partir de  $\hat{\sigma}_f = \sqrt{\frac{\sum_{i=1}^n w_i r_i^2}{\sum_{i=1}^n w_i - p}}$
  - 14:     Calcular el conjunto pesos finales  $W_f$  para garantizar que el conjuntos de datos se encuentra libre de *outliers*  $w_i = \begin{cases} 1 & : \frac{|r_i|}{\hat{\sigma}_f} \leq 2.5 \\ 0 & : \frac{|r_i|}{\hat{\sigma}_f} > 2.5 \end{cases}$
  - 15:     Realizar el ajuste final de parámetros con los puntos de  $P$  cuyo peso final  $w_i = 1$  para  $i = 1, \dots, n$
  - 16:     Almacenar el conjunto de parámetros obtenidos en el ajuste final
  - 17:    **end if**
-

---

```

18:  else
19:    for  $j = 1$  to  $n$  do
20:      if  $r_j < Umbral$  then
21:         $w_j = 1$ 
22:      else
23:         $w_j = 0$ 
24:      end if
25:    end for
26:    Realizar el ajuste final con los pesos obtenidos
27:    if Error generado del ajuste final es menor al obtenido en la iteración anterior
    then
28:      Almacenar el error generado y el conjunto de parámetros de la elipse
29:    end if
30:  end if
31:   $i \leftarrow i + 1$ 
32: end while

```

---

### 4.3. Suma de las distancias ortogonales y evolución diferencial

A continuación se presenta el algoritmo 7 que resolverá (4.12).

---

#### Algoritmo 7 SD +ED

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Límite de superior e inferior de cada uno de los parámetros de la elipse  $(a, b, x_c, y_c, \phi)$

**Entrada:** Tamaño de la población inicial  $\mu$ ; número máximo de generaciones  $G_{max}$ ; factor de recombinación  $CR$ ; factor de diferencia  $F$ ; valor de umbral  $U$

**Salida:** Parámetros desconocidos  $(a, b, x_c, y_c, \phi)$

- 1: Generar la población inicial  $X = \{x_1, \dots, x_\mu\}$  (se utilizó la distribución de Halton)
  - 2: Evaluar cada individuo en X:  $evaluar(x_i)$  para  $i = 1, \dots, \mu$
  - 3:  $k = 0$
  - 4: **repeat**
  - 5: **for**  $j = 1$  to  $\mu$  **do**
  - 6: Sean  $I_1, I_2$  e  $I_3$  tres enteros aleatorios en  $[1, \mu]$ , tal que  $I_1 \neq I_2 \neq I_3$
  - 7: Sea  $i_{rand}$  un entero aleatorio en  $[1, numvar]$
-

---

```

8:   for  $i = 1$  to  $numvar$  do
9:      $x'_{i,j} = \begin{cases} x_{i,I_3} + F(x_{i,I_1} - x_{i,I_2}) & : \text{rndreal}(0,1) < R \text{ ó } i = i_{rand} \\ x_{i,j} & : \text{de otro modo} \end{cases}$ 
10:  end for
11:  if  $evaluar(x'_{i,j}) < evaluar(x_{i,j})$  then
12:     $X_j = X'_j$ 
13:  end if
14: end for
15:  $min =$  valor de aptitud más pequeño de la población X
16:  $max =$  valor de aptitud más grande de la población X
17:  $k = k + 1$ 
18: until  $max - min < s$  ó  $k > G_{max}$ 

```

---



---

### Algoritmo 8 $evaluar(\mathbf{x})$

---

**Entrada:** Vector  $\mathbf{x}$  que representara los parámetros de la elipse  $(a, b, x_c, y_c, \phi)$

**Salida:** Valor de aptitud del individuo

```

1: Error  $e = 0$ 
2: Número de puntos  $m = 0$ 
3: for  $j = 0$  to  $Np$  do
4:   Calcular  $\mathbf{p}'_j = R(\phi)(\mathbf{p}_j - \mathbf{p}_c)$ 
5:    $r \leftarrow distanciaOrtogonal(\mathbf{p}'_j, a, b)$ 
6:   if  $r < U$  then
7:      $e \leftarrow e + |r|$ 
8:      $m \leftarrow m + 1$ 
9:   end if
10: end for
11: aptitud  $\leftarrow \frac{te}{m} + \frac{2U(Np-m)}{Np}$ 

```

---



## 4.4. Extracción de varias elipses

Los algoritmos presentados (MMC+RANSAC y SD+ED), pueden ser utilizados para detectar y extraer varias elipses de un conjunto de puntos dado, con la única diferencia de que es necesario definir un umbral para MMC+RANSAC ya que el umbral automático que maneja por defecto no funciona para dicha causa. La extracción de varias elipses puede verse como el problema de extraer varias elipses de una imagen capturada. Una vez que una elipse es detectada, los puntos que pertenecen a dicho ajuste son eliminados del conjunto de puntos y se reanuda la búsqueda de otra elipse. El pseudocódigo para la extracción de varias elipses se muestra en el algoritmo 9.

---

**Algoritmo 9** Algoritmo para la extracción de varias elipses

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Parámetros para la ejecución de MMC+ANSAC ó SD+ED

**Entrada:** Definir un valor de umbral  $U$  para ambos métodos

**Entrada:** Número de elipses que se desean extraer  $N_e$

**Salida:** Parámetros desconocidos de las  $N_e$  elipses  $(a, b, x_c, y_c, \phi)$

```
1: Sea  $N_p$  el número de total de puntos conocidos
2: for  $i = 1$  to  $N_e$  do
3:   Llamar MMC+RANSAC o SD+ED
4:   Almacenar los parámetros entregados por el método elegido
5:   for  $j = 1$  to  $n$  do
6:     Calcular el residual  $r$  del punto  $p_j$  a la elipse con los parámetros entregados
       anteriormente
7:     if  $r < U$  then
8:       Eliminar el punto  $p_j$  del conjunto de puntos conocidos
9:        $n \leftarrow n - 1$ 
10:    end if
11:  end for
12: end for
```

---

## 4.5. Pruebas

Para determinar el desempeño de los algoritmos presentados (MMC+RANSAC y SD+ED) se generaron tres pruebas: dos pruebas de extracción de una sola elipse bajo dos condiciones distintas de ruido, y la tercera prueba consiste en la extracción de de varias elipses. En todos los experimentos se utilizaron los siguientes parámetros para la evolución diferencial:

- Población: 30 individuos
- Constante de diferencia  $F$ : 0.7
- Constante de recombinación  $R_c$ : 0.9
- Condición de paro  $s$ : 0.01
- Número máximo de generaciones: 10,000
- Valor de umbral  $U$ : 10

Para el primera y segunda prueba la elipse se definió con los siguientes valores (100, 40, 300, 200, 40), se generaron 90 puntos para determinar la elipse y el valor de los puntos fueron redondeados a su entero inmediato. Los puntos atípicos fueron generados aleatoriamente con  $x \in [0, 400]$  y  $y \in [0, 300]$  con porcentajes que van del 0% al 100%. Para el primer experimento se generó ruido Gaussiano (con media 0 y desviación estándar de 3 ) sobre la posición de los puntos de la elipse. Los límites de búsqueda para SD+ED fueron  $a, b \in [1, 200]$  ,  $x_c, y_c \in [0 : 400]$ ,  $\phi \in [0 : 180]$  realizando 100 ejecuciones de cada prueba. Los resultados de la media y la desviación estándar para cada uno de los cinco parámetros de la elipse se muestran en la figura 4.3 y un ejemplo del ajuste en la figura 4.4.

Para el segundo experimento, las mismas condiciones fueron utilizadas, excepto el ruido agregado a la posición de cada punto (con respecto al sistema de coordenadas local), de la siguiente manera:

```

$d = sqrt($xp*$xp + $yp*$yp); #(xp,yp) representan un punto
$v = rand();
if ( $v < 0.5 ) {
    $v = -3;
}
else {
    $v = 3;
}
$xp = ($d+$v)*cos( $angulo );
$yp = ($d+$v)*sin( $angulo );
$x = $xp*cos($angulo) - $yp*sin($angulo);

```

---

```
$y = $xp*sin($angulo) + $yp*cos($angulo);
```

---

La función `rand()` (utilizada también el algoritmo 7) devuelve un número aleatorio en el intervalo de  $[0, 1]$ . Los resultados de las 100 ejecuciones se muestran en la figura 4.5 y un ejemplo de dicho ajuste en la figura 4.6.

La tercera prueba consistió en extraer tres elipses con parámetros  $a = 80, b = 40$  y valores aleatorios (con distribución uniforme) para  $x_c \in [0, 640], y_c \in [0, 480]$ , y  $\phi \in [0, 180]$  y estos mismos intervalos fueron utilizados para generar los datos contaminados. Las estadísticas de 100 ejecuciones con respecto al porcentaje de valores atípicos se muestran en la figura 4.7 y un ejemplo de la extracción en la figura 4.8.

## 4.6. Sumario

De las simulaciones realizadas se puede deducir que SD + ED es superior a MMC + RANSAC, al extraer una elipse cuando el porcentaje de puntos contaminados es mayor al 50%. Usando la suma de las distancias ortogonales como criterio para el problema de optimización, la presencia de puntos contaminados no afecta la estimación de los parámetros de la elipse. El maximizar el número de los puntos que forman parte del ajuste permite que SD+ED tenga un mejor desempeño cuando las condiciones de ruido superan el 50% de puntos atípicos.

SD+ED logró extraer exitosamente las tres elipses propuestas en el 100% de las ejecuciones, ante cualquier condición de ruido, mientras que MMC+RANSAC no lo logró. Cabe mencionar que MMC+RANSAC fue utilizado de la misma forma en la que se implementó SD+ED para la extracción de varias elipses.

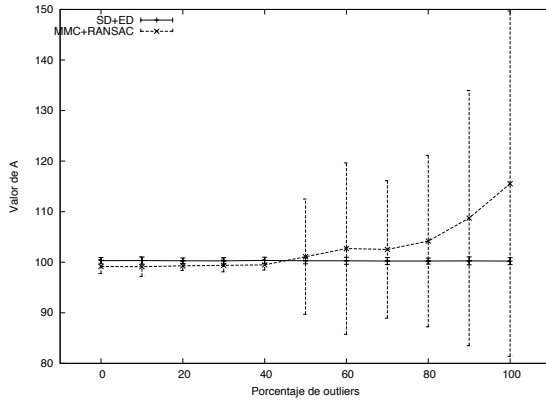
La función de agregación propuesta que resuelve ambos problemas de optimización (minimización del error y maximización del número de puntos del ajuste) funcionó adecuadamente, ya que al utilizar  $2U(n-m)/n$ , donde  $n$  es el número de puntos totales y  $m$  el total de puntos que pertenecen al ajuste, se tiene que si se usan todos los puntos dicha función tiende a cero, mientras que si se usan sólo algunos tiende al error máximo que es  $2U$ .

La diferencia que existe entre MMC+RANSAC y SD+ED es la forma en que operan, ya que MMC+RANSAC realiza la estimación de los parámetros en función de los puntos seleccionados previamente, mientras que SD+ED realiza la estimación de los parámetros y posteriormente discrimina puntos.

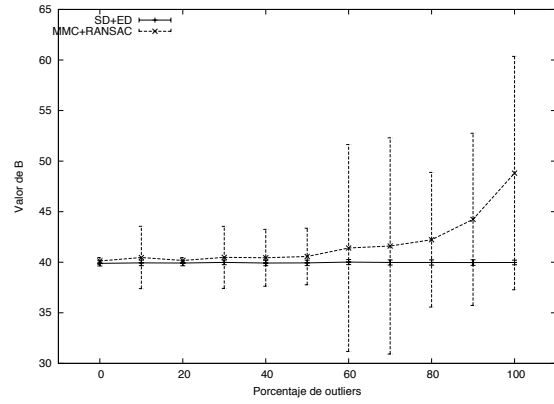
El utilizar la ED tiene la desventaja de incrementar el número de evaluaciones de la función objetivo, pero ésta se ve diluida cuando se está realizando la extracción de varias elipses (como se puede ver en la figura 4.7).

Se propone un algoritmo para el ajuste de una o varias elipses bajo condiciones donde la cantidad de ruido es grande. Este algoritmo minimiza la suma de las distancias ortogonales de un conjunto de puntos dado a la elipse estimada, y al mismo tiempo maximiza el número de puntos que pertenecen al ajuste. Esto genera un problema de optimización no lineal, el cual fue resuelto utilizando la evolución diferencial.

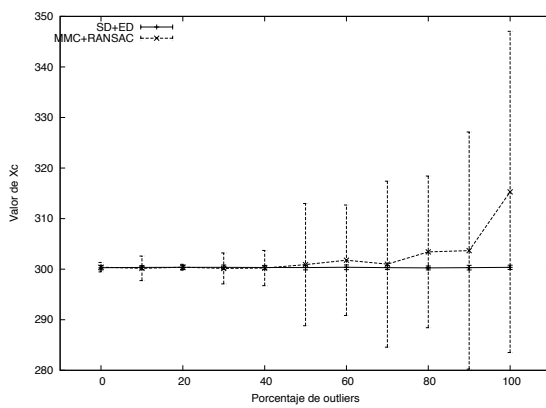
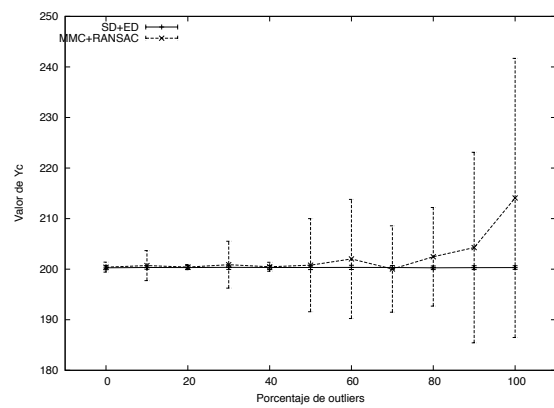
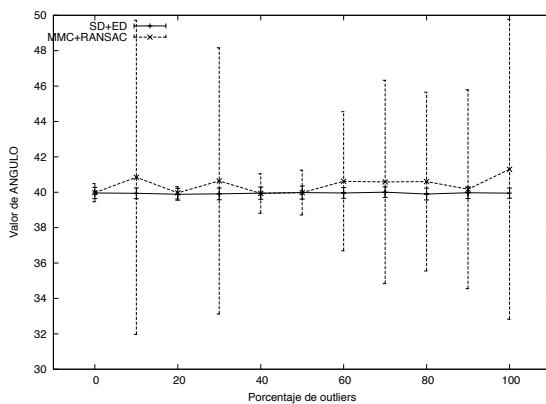
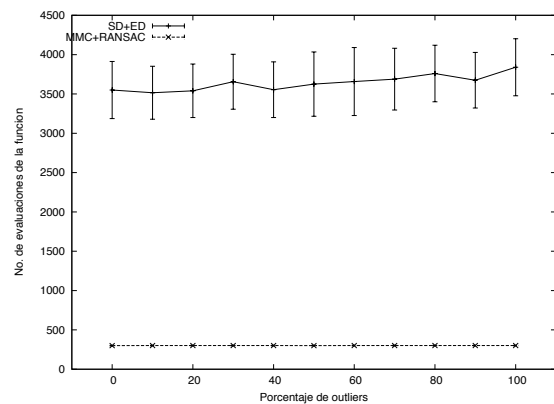
El algoritmo propuesta muestra un buen desempeño, ya que es capaz de detectar y extraer elipses el 100 % de las veces en condiciones de ruido que van del 0 % al 100 % de puntos contaminados.



(a) Parámetro A



(b) Parámetro B

(c) Parámetro  $x_c$ (d) Parámetro  $y_c$ (e) Parámetro  $\phi$ 

(f) Número de evaluaciones de la función objetivo

Figura 4.3: Resultados de la primera prueba.

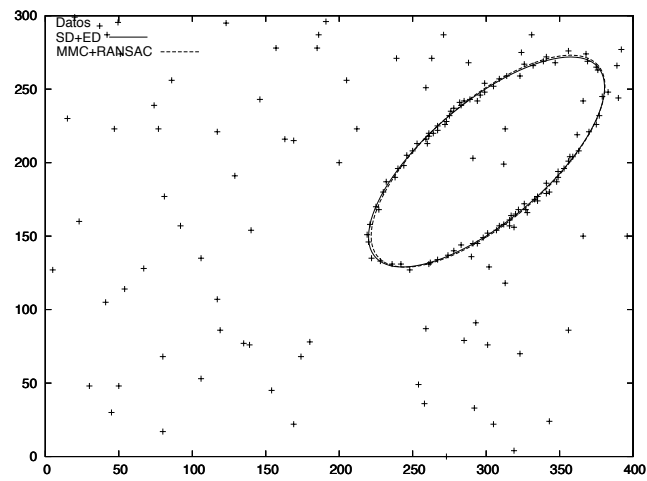
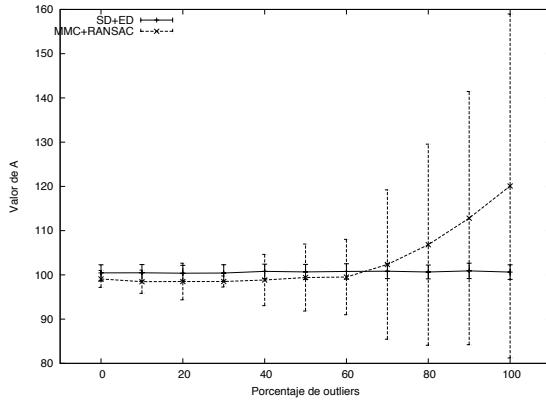
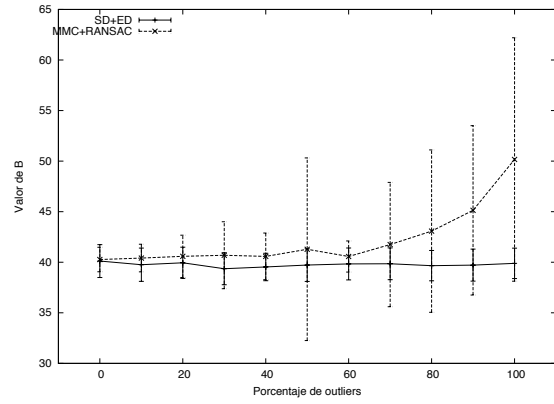


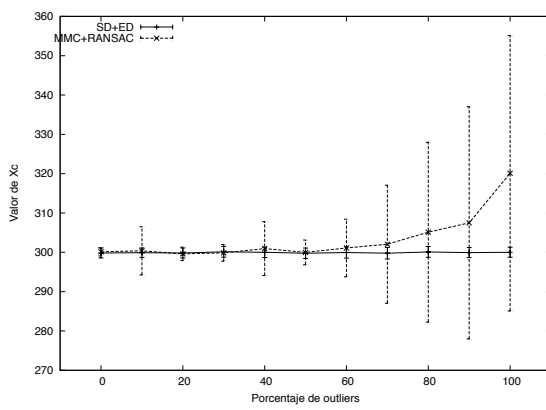
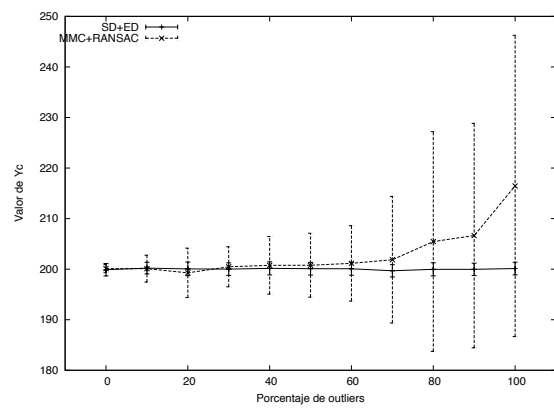
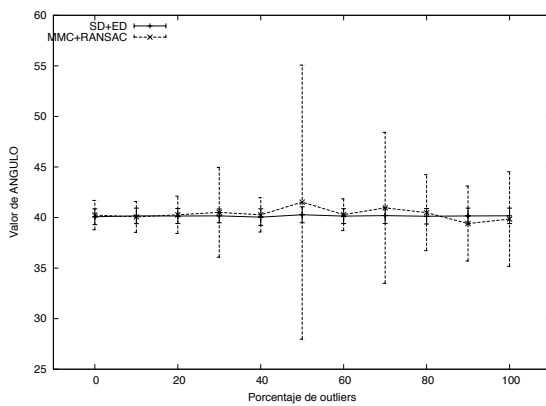
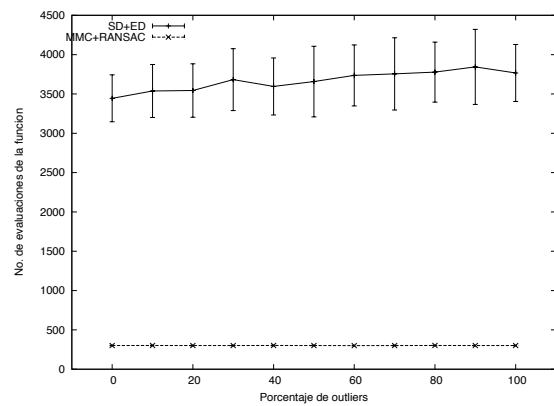
Figura 4.4: Ajuste resultante del primer experimento.



(a) Parámetro A



(b) Parámetro B

(c) Parámetro  $x_c$ (d) Parámetro  $y_c$ (e) Parámetro  $\phi$ 

(f) Número de evaluaciones de la función objetivo

Figura 4.5: Resultados de la segunda prueba.

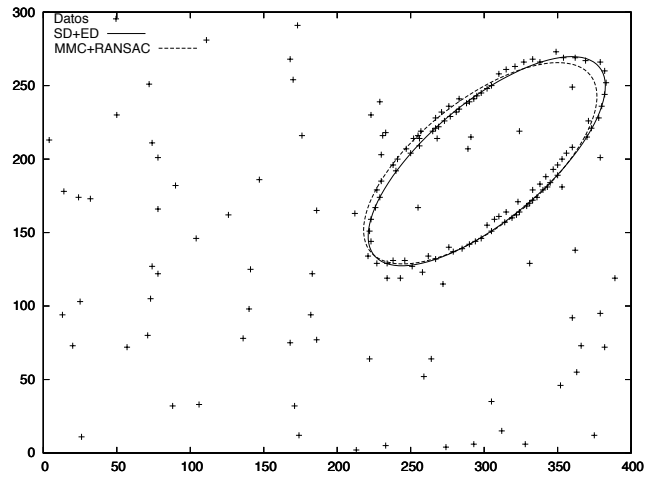
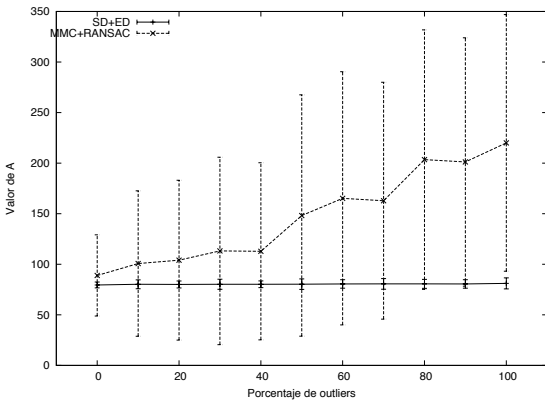
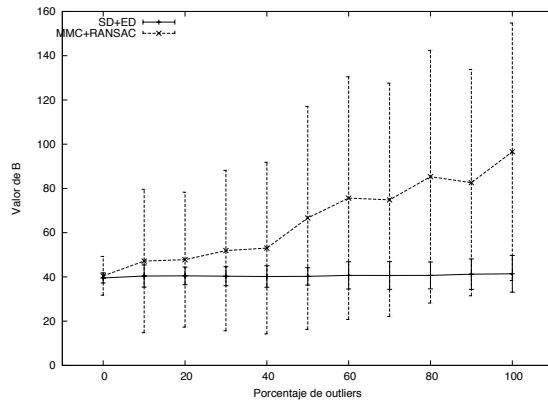


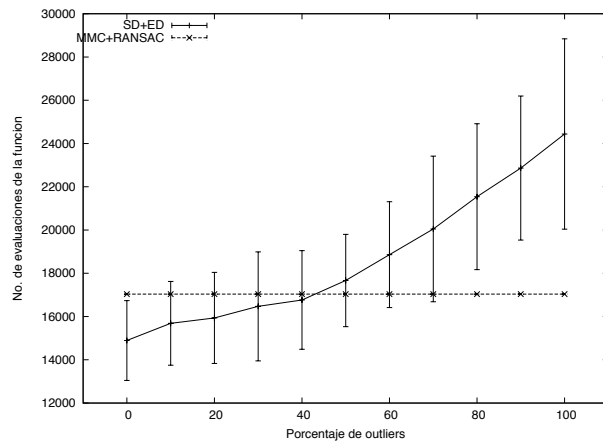
Figura 4.6: Ajuste resultante del segundo experimento.



(a) Parámetro A



(b) Parámetro B



(c) Número de evaluaciones de la función objetivo

Figura 4.7: Resultados de la tercera prueba.



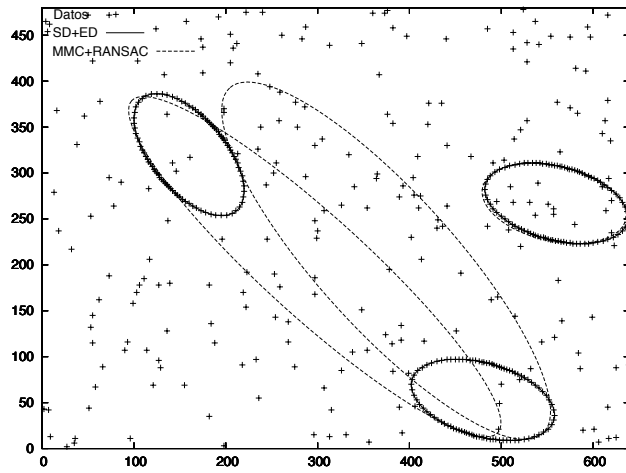


Figura 4.8: Ajuste resultante del tercer experimento.



# Capítulo 5

## Ajuste de triángulos

---

En este capítulo se realiza la extracción de triángulos a partir de un conjunto de puntos que definen el perímetro de uno o más triángulos por medio de un algoritmo de ajuste. Este problema de ajuste se basa en la minimización de la suma de las distancias perpendiculares de los puntos dados al triángulo y es un problema de ajuste no lineal. El problema propuesto se resuelve utilizando la heurística llamada evolución diferencial. Se presentan pruebas y resultados que demuestran el buen desempeño del algoritmo propuesto.

### 5.1. Descripción del problema

Dado un conjunto de puntos  $\mathbf{p}_i \in \mathbb{R}^2$  para  $1 \leq i \leq m$ , en un sistema de coordenadas globales  $xy$ , se buscará el triángulo formado por el conjunto de tres vértices (o puntos)  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  que mejor represente al conjunto de puntos dado.

Para medir la ‘mejor representación’ de los puntos al triángulo hemos utilizado la distancia euclidiana de cada punto a los lados y se plantea minimizar la suma de todas las distancias de los puntos al triángulo. Este problema es no lineal dado que la distancia euclidiana involucra calcular una raíz cuadrada. El procedimiento estándar de mínimos cuadrados, de calcular la derivada del problema e igualarla a cero para encontrar la solución mínima, no puede aplicarse en esta situación ya que la derivada de la raíz cuadrada presenta una discontinuidad (es infinita) en el mínimo de la función raíz cuadrada.

### 5.2. Cálculo de las distancias perpendiculares

Como se ha mencionado en capítulos anteriores, la distancia perpendicular representa la distancia más corta que existe entre el ajuste propuesto y el conjunto de puntos que se están ajustando. Para determinar la distancia perpendicular entre un punto y el triángulo propuesto, se parte de los modelos presentados en las figuras 5.1a y 5.1b.

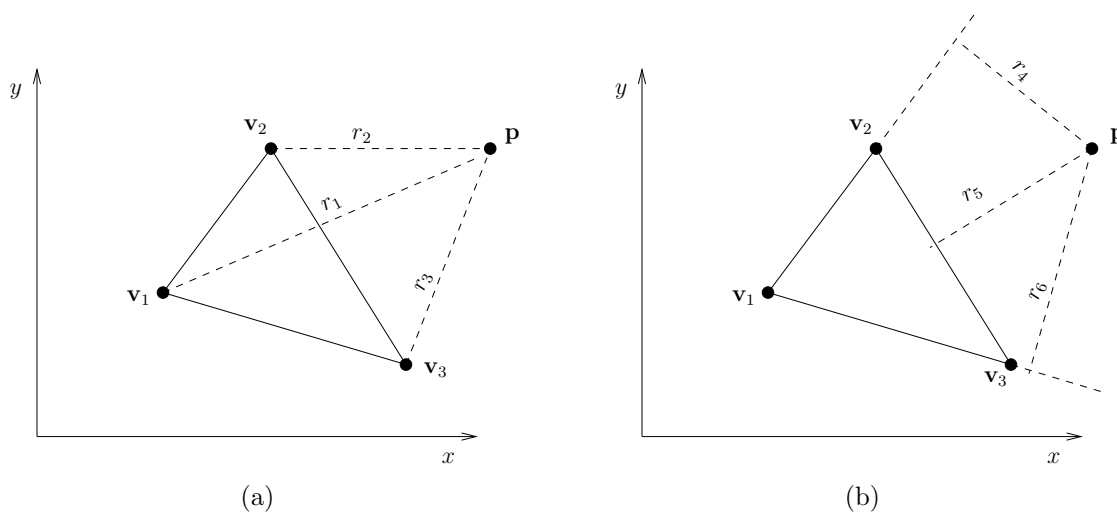


Figura 5.1: (a) Distancias del punto  $\mathbf{p}$  a los vértices de un triángulo. (b) Distancias perpendiculares del punto  $\mathbf{p}$  a los tres lados del triángulo.

El triángulo propuesto se encuentra representado por los puntos  $\mathbf{v}_1, \mathbf{v}_2$  y  $\mathbf{v}_3 \in \mathbb{R}^2$  en un sistema de coordenadas globales  $xy$ , a partir de los cuales se obtendrán las distancias  $(r_1, r_2, r_3)$  del punto conocido  $\mathbf{p}$  a cada uno de los vértices que forman el triángulo propuesto (ver figura 5.1a) y las distancias perpendiculares  $r_4, r_5$  y  $r_6$  del punto  $\mathbf{p}$  a las tres rectas que pasan por los vértices  $\mathbf{v}_1\mathbf{v}_2$ ,  $\mathbf{v}_2\mathbf{v}_3$  y  $\mathbf{v}_3\mathbf{v}_1$  tal y como se muestra en la figura 5.1b.

La distancia del punto  $\mathbf{p}$  al triángulo será entonces:

La menor de las distancias entre el conjunto de distancias válidas a los lados del triángulo y las distancias del punto a los vértices del triángulo.

Una distancia del punto  $\mathbf{p}$  a un lado del triángulo se considera válida si la proyección perpendicular del punto al lado, yace dentro del segmento de recta que forma el lado del triángulo.

En la figura 5.1b las distancias  $r_4$  y  $r_6$  son inválidas, ya que para  $r_4$ , el punto de proyección de  $\mathbf{p}$  está fuera del segmento de recta  $\overline{\mathbf{v}_1\mathbf{v}_2}$ .

$r_5$  es una distancia válida del punto  $\mathbf{p}$  al segmento de recta  $\overline{\mathbf{v}_2\mathbf{v}_3}$ . Para determinar la distancia perpendicular de  $\mathbf{p}$  al triángulo propuesto deben considerarse las distancias a los vértices y a las aristas, ya que si sólo se consideran las distancias perpendiculares a las aristas, la distancia del punto al triángulo no será la correcta por ejemplo, en la figura 5.2a se puede observar que el punto  $\mathbf{p}$  sólo tiene como distancia perpendicular válida a  $r_6$ , si sólo se considera dicha distancia, el ajuste se verá afectado debido al enorme residual que éste genera cuando realmente el punto no se encuentra tan alejado del triángulo propuesto. Pueden presentarse incluso casos en que no existan

distancias perpendiculares válidas a los lados del triángulo. En ese caso, se toma la distancia menor a los vértices, como se puede observar en la figura 5.2b.

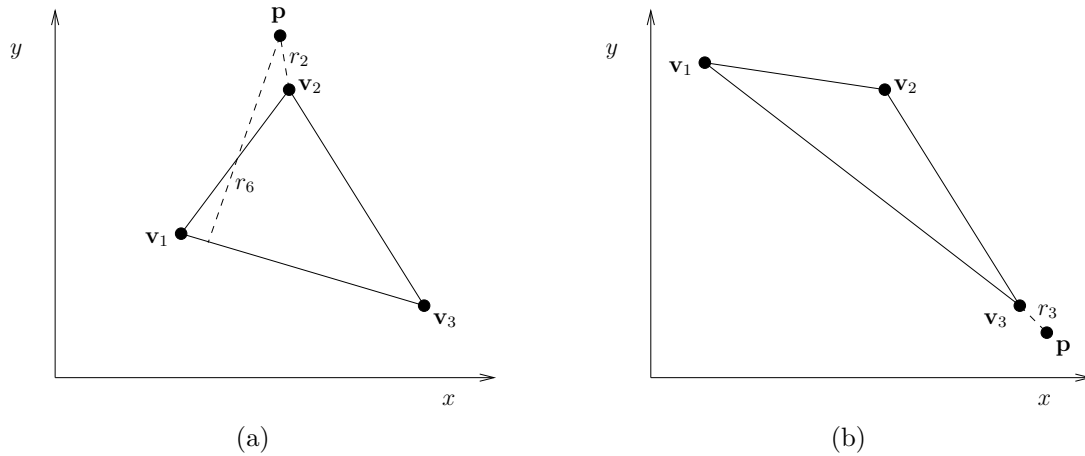


Figura 5.2: (a) Caso de ejemplo en el que se tiene solo una distancia perpendicular válida, la cual es mucho mayor que la distancia al vértice. (b) Ejemplo en el que no existen distancias perpendiculares válidas. La distancia considerada es la menor del punto a los vértices del triángulo.

Las distancias euclidianas del punto  $\mathbf{p}$  a cada vértice del triángulo se obtiene como:

$$r_j = \|\mathbf{v}_j - \mathbf{p}\| = \sqrt{(v_{jx} - p_x)^2 + (v_{jy} - p_y)^2} \quad \text{para } j = 1 \dots 3 \quad (5.1)$$

Para obtener las distancias ortogonales a cada una de las aristas se debe realizar la proyección perpendicular del punto  $\mathbf{p}$  sobre cada una de las tres líneas que pasan por las aristas  $\overline{\mathbf{v}_1\mathbf{v}_2}$ ,  $\overline{\mathbf{v}_2\mathbf{v}_3}$  y  $\overline{\mathbf{v}_3\mathbf{v}_1}$ .

Esta proyección se calcula de la siguiente manera (ver figura 5.3).

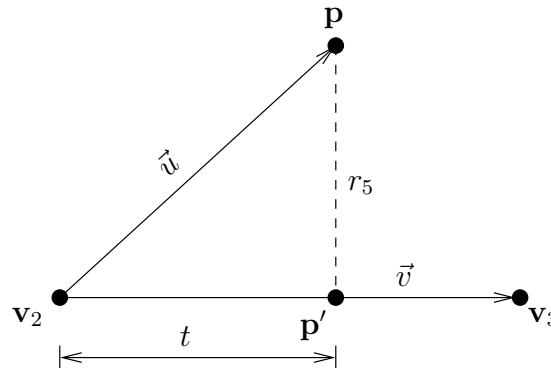


Figura 5.3: Distancia perpendicular de  $\mathbf{p}$  al segmento de recta  $\overline{\mathbf{v}_2\mathbf{v}_3}$ .

$$\vec{u} = \mathbf{p} - \mathbf{v}_2 \quad (5.2)$$

$$\vec{v} = \mathbf{v}_3 - \mathbf{v}_2 \quad (5.3)$$

La distancia  $t$  a la que se encuentra el punto de contacto  $\mathbf{p}'$  tomando como punto inicial del segmento de recta el vértice  $\mathbf{v}_2$  se puede obtener con:

$$t = \vec{u} \cdot \frac{\vec{v}}{\|\vec{v}\|} \quad (5.4)$$

Por lo tanto, el punto de contacto  $\mathbf{p}'$  se puede obtener a partir de (5.5).

$$\mathbf{p}' = \mathbf{v}_2 + t \frac{\vec{v}}{\|\vec{v}\|} \quad (5.5)$$

La distancia perpendicular  $r_5$  del punto  $\mathbf{p}$  al segmento de recta  $\overline{\mathbf{v}_2\mathbf{v}_3}$  será entonces:

$$r_5 = \|\mathbf{p} - \mathbf{p}'\| \quad (5.6)$$

Se puede decir entonces que la distancia perpendicular válida de  $\mathbf{p}$  a un segmento de recta definido es aquella donde el punto de proyección  $\mathbf{p}'$  se encuentra a una distancia  $0 \leq t \leq \|\vec{v}\|$ .

Por lo tanto, el conjunto de distancias se define como:

$$\mathbf{r} = \{r_1, r_2, r_3, \{r_j : t \geq 0 \text{ y } t \leq \|\vec{v}\| \text{ para } j = 4 \cdots 6\}\} \quad (5.7)$$

A partir de (5.7) el conjunto de distancias para los modelos presentados en la figura 5.1a y 5.1b estaría formado por  $\mathbf{r} = \{r_1, r_2, r_3, r_5\}$ .

Una vez que se tiene el conjunto de distancias válidas en el vector  $\mathbf{r}$  como en (5.7), podemos definir una función  $D$ , que toma como argumentos el vector  $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T]$ ,  $\mathbf{v} \in \mathbb{R}^6$  y un punto conocido  $\mathbf{p}$ . Por lo tanto, la distancia entre el punto conocido  $\mathbf{p}$  y el triángulo  $\mathbf{v}$  será el valor que entregue la función:

$$\begin{aligned} D : \mathbb{R}^6 \times \mathbb{R}^2 &\rightarrow \mathbb{R} \\ D(\mathbf{v}, \mathbf{p}) &= \min(r_i) \end{aligned} \quad (5.8)$$

### 5.3. Solución al problema de optimización

Si definimos como  $\mathbf{v}$  el conjunto de tres vértices de un triángulo  $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ , el problema de optimización que debe resolverse es minimizar la siguiente función.

$$\begin{aligned} g_1 : \quad &\mathbb{R}^6 \rightarrow \mathbb{R} \\ g_1(\mathbf{v}) &= \frac{1}{m} \sum_{i=1}^m D(\mathbf{v}, \mathbf{p}_i) \end{aligned} \quad (5.9)$$

Para el problema de optimización planteado en (5.9) se sugirieron inicialmente dos modificaciones: sólo aquellas distancias cuyo valor sean menores a un valor de umbral  $U$  contribuirán al ajuste y al mismo tiempo se maximizará el número de puntos que forman parte del ajuste, de tal forma que el problema de optimización consiste ahora en minimizar (5.10).

$$g_2(\mathbf{v}) = \frac{1}{m} \sum_{\text{si } D < U} D(\mathbf{v}, \mathbf{p}_i) + \frac{2U(n - m)}{n} \quad (5.10)$$

Donde  $n$  representa el número de puntos conocidos y  $m$  el número de puntos que forman parte del ajuste, v.g., aquellos cuya distancia sea menor al valor de umbral  $U$ .

Las modificaciones realizadas al problema de optimización proporcionaron buenos resultados al realizar la extracción de un solo triángulo, ya que éste tendrá el mínimo error y el máximo número de puntos como se muestra en la figura 5.4, donde se puede observar el ajuste entregado (línea continua) sobre el conjunto de puntos conocido (círculos rellenos).

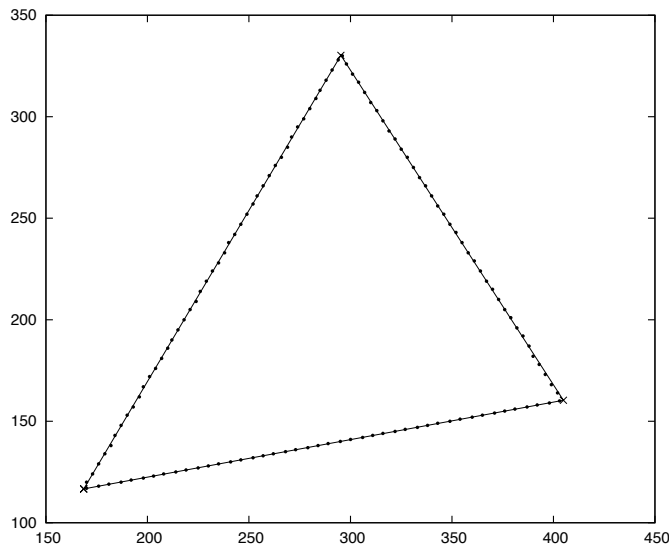


Figura 5.4: Ajuste utilizando el problema de optimización propuesto en (5.10).

El problema se presenta al tratar de realizar el ajuste de un triángulo sobre una imagen que contiene varios de ellos, debido a que se pueden presentar casos en los que el ajuste esté formado por varios triángulos, lo cual incrementa el número de puntos que pertenecen al ajuste y, en consecuencia, el valor de la función objetivo es mucho menor que realizar un ajuste adecuado con un número de puntos de ajuste menor.

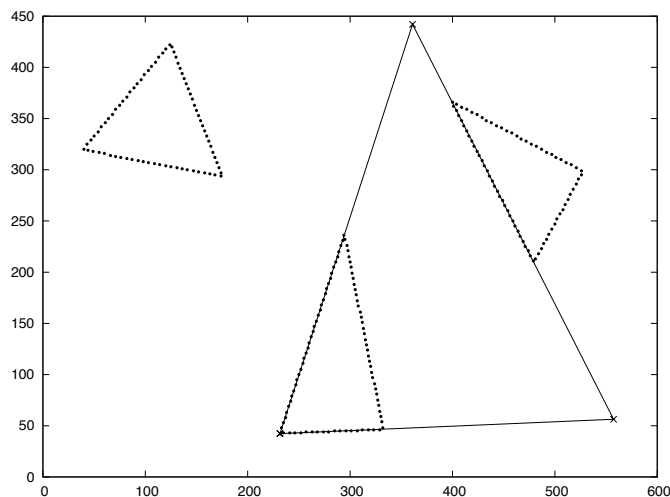


Figura 5.5: Ajuste utilizando el problema de optimización propuesto en (5.10).

En la figura 5.5 se puede observar que el ajuste entregado (línea punteada) contiene un mayor número de puntos que el ajuste correcto, el cual resultaría al extraer el triángulo central de la imagen. En este caso la primera parte de la función objetivo (minimización del error) presentada en (5.10) no tiene gran efecto sobre el ajuste, debido a que la mayoría de los puntos que lo forman se encuentran sobre los segmentos de recta del triángulo, v.g., que su residual es cero o muy cercano a cero, por lo que el resultado final del ajuste recaerá sobre la última parte de la función, la cual maximiza el número de puntos que forman parte del ajuste. En el caso presentado en la figura 5.5 el triángulo esperado (triángulo central) está formado por 91 puntos, mientras que el ajuste final entregado (línea continua) contiene 99 puntos. Por lo tanto, el ajuste entregado se puede considerar correcto, ya que cumple con todas las restricciones establecidas, v.g., tiene el mínimo error y el mayor número de puntos, pero el ajuste es visualmente incorrecto.

Para resolver el problema expuesto anteriormente, se propone utilizar una medida de densidad, la cual proporcionará el número de puntos que existen por unidad de longitud del triángulo ajustado:

$$\rho = \frac{m}{\|\mathbf{v}_1\mathbf{v}_2\| + \|\mathbf{v}_2\mathbf{v}_3\| + \|\mathbf{v}_3\mathbf{v}_1\|} \quad (5.11)$$

Donde  $m$  es el número de puntos que forman parte del ajuste y el divisor de (5.11) representa el perímetro del triángulo ajustado. De esta forma se buscará que se tengan un mayor número de puntos por unidad de longitud y evitar así, resultados como los mostrados anteriormente.



Debe recordarse que el problema de optimización planteado en (5.10) es de minimización, por lo que es necesario utilizar la inversa de  $\rho$ , de tal forma que el problema generado no tenga que lidiar con valores negativos en la función objetivo. La tercera modificación a la que será sometido el problema de optimización planteado en (5.9), consiste en minimizar el inverso de la medida de densidad propuesta en (5.11). Por lo tanto el nuevo problema de optimización consiste en minimizar  $g_3$  para obtener los parámetros desconocidos del ajuste.

$$g_3(\mathbf{v}) = \frac{1}{m} \sum_{\text{si } D < U} D(\mathbf{v}, \mathbf{p}_i) + \frac{2U(n-m)}{n} + \frac{2(\|\mathbf{v}_1\mathbf{v}_2\| + \|\mathbf{v}_2\mathbf{v}_3\| + \|\mathbf{v}_3\mathbf{v}_1\|)}{m} \quad (5.12)$$

Es importante mencionar que el factor de dos que acompaña al tercer término de  $g_2$  fue elegido de forma empírica, ya que presentaba mejores resultados al realizar el ajuste. Las pruebas y resultados obtenidos se mencionarán con mayor detalle en la sección dedicada a las pruebas (ver sección 5.5).

La evolución diferencial es la heurística que resolverá el problema de optimización no lineal planteado en (5.12) y proporcionará los parámetros desconocidos del ajuste, que en este caso serán los vértices del triángulo  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  y  $\mathbf{v}_3$ .

Para representar el problema en la evolución diferencial, el vector de parámetros de cada uno de los individuos contendrá los vértices del triángulo propuesto. Dichos vértices serán ordenados en sentido horario, con la finalidad de que se cumpla con el modelo propuesto en la figura 5.1b.

Para que los individuos contengan triángulos válidos de acuerdo al modelo establecido será necesario reparar a cada uno éstos antes de evaluar la función objetivo (5.12), v.g., que se ordenarán los vértices del triángulo que éste contenga.

El algoritmo 10 muestra el pseudocódigo para el ordenamiento de los puntos. El algoritmo 11 presenta el pseudocódigo para evaluar a cada uno de los individuos y, finalmente, el pseudocódigo que resuelve (5.12) se presenta en el algoritmo 12.

**Algoritmo 10** *ordenar*( $\mathbf{x}$ )**Entrada:** Vector  $\mathbf{x}$  que representará los parámetros del triángulo ( $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ )**Salida:** Vector de parámetros  $\mathbf{x}$  ordenado

- 1: Obtener el vértice con la componente  $x$  más pequeña y con la  $y$  más grande
- 2: **if** El vértice obtenido es diferente a  $\mathbf{v}_1$  **then**
- 3:   Intercambiar dicho vértice de tal forma que el vértice con menor  $x$  y mayor  $y$  se encuentre en  $\mathbf{v}_1$
- 4: **end if**
- 5: Hacer  $\vec{a} = \mathbf{v}_2 - \mathbf{v}_1$
- 6: Hacer  $\vec{b} = \mathbf{v}_3 - \mathbf{v}_1$
- 7: Obtener la componente  $\hat{k}$  del vector resultante de  $\vec{a} \times \vec{b}$ , v.g.,  $\hat{k} = \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}$
- 8: **if**  $\hat{k} > 0$  **then**
- 9:   intercambiar  $\mathbf{v}_2$  y  $\mathbf{v}_3$
- 10: **end if**

**Algoritmo 11** *evaluar*( $\mathbf{x}$ )**Entrada:** Vector  $\mathbf{x}$  que representará los parámetros del triángulo ( $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ )**Salida:** Valor de aptitud del individuo

- 1: Error  $e = 0$
- 2: Inicializar vector de residuales  $\mathbf{r} = \emptyset$
- 3: *ordenar*( $\mathbf{x}$ )
- 4: Número de puntos  $m = 0$
- 5: **for**  $j = 1$  to  $Np$  **do**
- 6:   Obtener la distancia del punto  $\mathbf{p}_j$  a los vértices  $\mathbf{v}_1, \mathbf{v}_2$  y  $\mathbf{v}_3$  a partir de (5.1)
- 7:   Calcular la distancia perpendicular a cada uno de los segmentos de recta definidos por  $\mathbf{v}_1, \mathbf{v}_2$  y  $\mathbf{v}_3$  a partir de (5.6) y determinar las distancias perpendiculares válidas
- 8:   Una vez calculadas las distancias a los vértices y a las aristas llenar el conjunto  $\mathbf{r}$  a partir de (5.7)
- 9:    $d \leftarrow \min(\mathbf{r})$
- 10:   **if**  $d < U$  **then**
- 11:      $e \leftarrow e + d$
- 12:      $m \leftarrow m + 1$
- 13:   **end if**
- 14: **end for**
- 15:  $p \leftarrow \|\mathbf{v}_1\mathbf{v}_2\| + \|\mathbf{v}_2\mathbf{v}_3\| + \|\mathbf{v}_3\mathbf{v}_1\|$
- 16:  $\text{aptitud} \leftarrow \frac{e}{m} + \frac{2U(Np-m)}{Np} + \frac{2m}{p}$

---

**Algoritmo 12** SD +ED

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Alto y ancho de la imagen ( $W, H$ )

**Entrada:** Tamaño de la población inicial  $\mu$ ; número máximo de generaciones  $G_{max}$ ; factor de recombinación  $CR$ ; factor de diferencia  $F$ ; valor de umbral  $U$

**Salida:** Parámetros desconocidos del ajuste ( $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ )

- 1: Generar la población inicial  $X = \{x_1, \dots, x_\mu\}$  (se utilizó la distribución de Halton)
  - 2: Evaluar cada individuo en X:  $evaluar(x_i)$  para  $i = 1, \dots, \mu$
  - 3:  $k = 0$
  - 4: **repeat**
  - 5:   **for**  $j = 1$  to  $\mu$  **do**
  - 6:     Sean  $I_1, I_2$  e  $I_3$  tres enteros aleatorios en  $[1, \mu]$ , tal que  $I_1 \neq I_2 \neq I_3$
  - 7:     Sea  $i_{rand}$  un entero aleatorio en  $[1, numvar]$
  - 8:     **for**  $i = 1$  to  $numvar$  **do**
  - 9:       
$$x'_{i,j} = \begin{cases} x_{i,I_3} + F(x_{i,I_1} - x_{i,I_2}) & : \text{rndreal}(0,1) < R \text{ o } i = i_{rand} \\ x_{i,j} & : \text{de otro modo} \end{cases}$$
  - 10:     **end for**
  - 11:     **if**  $evaluar(x'_{i,j}) < evaluar(x_{i,j})$  **then**
  - 12:        $X_j = X'_j$
  - 13:     **end if**
  - 14:   **end for**
  - 15:    $min =$  valor de aptitud más pequeño de la población X
  - 16:    $max =$  valor de aptitud más grande de la población X
  - 17:    $k = k + 1$
  - 18: **until**  $max - min < s$  ó  $k > G_{max}$
-

## 5.4. Extracción de varios triángulos

SD+ED pueden ser utilizado para realizar la extracción de varios triángulos de un conjunto de puntos dado. Para ello, cada vez que un triángulo es detectado, los puntos que pertenecen a dicho ajuste son eliminados del conjunto de puntos y se reanuda la búsqueda. El pseudocódigo para la extracción de varios triángulos se muestra en el algoritmo 13.

---

**Algoritmo 13** Algoritmo para la extracción de varios triángulos

---

**Entrada:** Conjunto de datos conocidos  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$

**Entrada:** Parámetros para la ejecución de SD+ED

**Entrada:** Número de triángulos que se desean extraer  $N_t$

**Salida:** Parámetros desconocidos de los  $N_t$  triángulos

```

1: Sea  $N_p$  el número de total de puntos conocidos
2: for  $i = 1$  to  $N_t$  do
3:   Llamar SD+ED
4:   Almacenar los parámetros entregados por el método
5:   for  $j = 1$  to  $N_p$  do
6:     Obtener el conjunto de residuales  $\mathbf{r}$  a partir de (5.7)
7:     Obtener el residual  $d$  (distancia perpendicular) del punto  $p_j$  al triángulo con
       los parámetros entregados anteriormente
8:     if  $d < U$  then
9:       Eliminar el punto  $p_j$  del conjunto de puntos conocidos
10:       $N_p \leftarrow N_p - 1$ 
11:     end if
12:   end for
13: end for

```

---

## 5.5. Pruebas

Para determinar el desempeño de SD + ED se generaron cuatro pruebas que contienen el mismo porcentaje de datos atípicos (50%). La primera prueba consiste en realizar la extracción de un triángulo que presenta ruido gaussiano sobre los puntos que lo conforman. Dicho ruido tiene como parámetros media cero y desviación estándar  $\sigma = 2$ .

La segunda prueba consiste en realizar la extracción de dos triángulos que se intersecan. Cada uno de ellos presenta ruido gaussiano sobre los puntos que lo conforman con las mismas características que en la primera prueba.

La tercera prueba consiste en realizar la extracción de tres triángulos que se encuentran distribuidos sobre la imagen. Éstos no presentan ningún tipo de ruido sobre

los puntos que los conforman.

La última prueba consiste en realizar la extracción de tres triángulos que se intersecan en uno o varios puntos. cabe mencionar que se realizaron 100 ejecuciones de cada una de las pruebas y se reportaron los resultados.

Las pruebas se generaron a partir de los siguientes datos:

### Prueba 1

Triángulo	Vértices
1	(169, 116, 296, 330, 405, 160)

### Prueba 2

Triángulo	Vértices
1	(198, 309, 452, 309, 319, 126)
2	(180, 166, 317, 409, 449, 166)

### Prueba 3

Triángulo	Vértices
1	(40, 320, 125, 423, 176, 293)
2	(400, 366, 528, 298, 480, 209)
3	(232, 42, 295, 236, 333, 46)

### Prueba 4

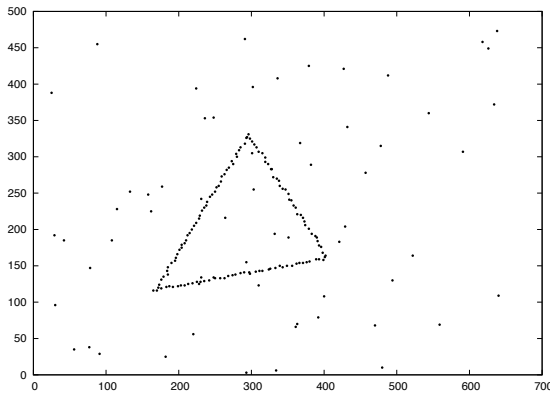
Triángulo	Vértices
1	(75, 237, 217, 315, 295, 87)
2	(131, 91, 387, 367, 407, 223)
3	(273, 393, 539, 352, 361, 256)

Los parámetros que se utilizaron en la evolución diferencial, se muestran a continuación:

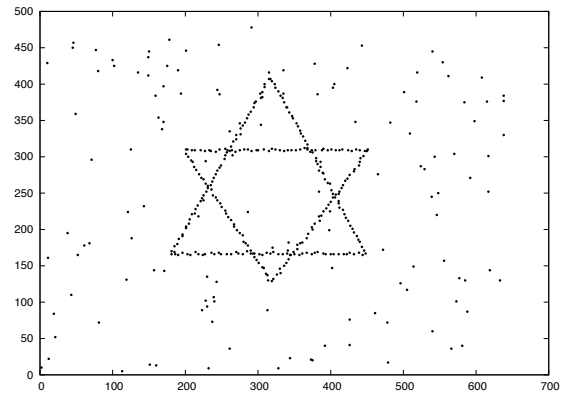
- Límites de la imagen  $X \in [0, 600]$
- Límites de la imagen  $Y \in [0, 480]$
- Número de individuos = 30
- Número máximo de generaciones = 10000
- Coeficiente de recombinación = 0.9
- Factor de diferencia 0.7

- Valor de umbral  $U = 10$

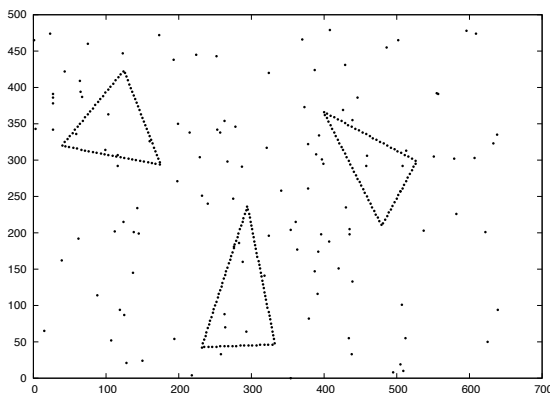
En la figura 5.6 se pueden observar las pruebas gráficamente.



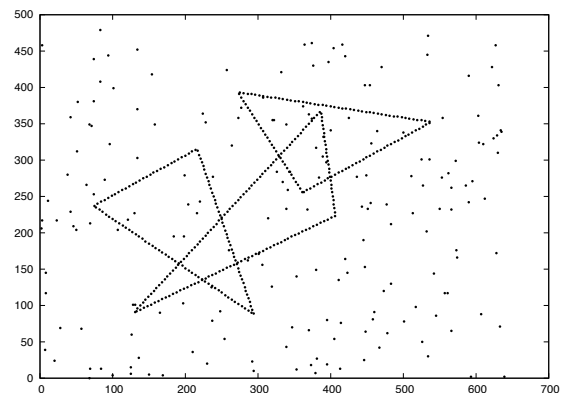
(a)



(b)



(c)



(d)

Figura 5.6: (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4

Los resultados de las ejecuciones se muestran en la tabla 5.1.

	SD+ED			Evaluaciones de la función objetivo	
	Mínimo	Máximo	Media	Media	Desviación estándar
Prueba					
1	1	1	1	6309	587
2	1	2	1.99	15489	1763
3	3	3	3	18066	1565
4	2	3	2.88	23075	2754

Tabla 5.1: Se muestra el mínimo, máximo y la media de los triángulos extraídos exitosamente en cada una de las pruebas presentadas, así como la media y desviación estándar del número de evaluaciones de la función objetivo.

## 5.6. Sumario

Se puede deducir que SD + ED realizó la extracción exitosa en un 99 % de las ejecuciones de las primeras tres pruebas, mientras que sólo acertó el 90 % de las veces al realizar la extracción de los triángulos de la prueba cuatro. Es importante remarcar que realizar la extracción de un triángulo de la forma que se propone en este capítulo es menos costoso que realizar la extracción de tres líneas, v.g., que produce menos evaluaciones de la función objetivo. En el capítulo dedicado al ajuste robusto de líneas, se puede observar que al realizar la extracción de tres líneas se necesitan en promedio 10,000 evaluaciones de la función objetivo, mientras que para realizar la extracción de un triángulo se necesitan en promedio 6,309 evaluaciones de la función objetivo, tal y como se muestra en los resultados obtenidos.

La función de agregación propuesta considera tres problemas que es necesario resolver para obtener un ajuste adecuado (minimiza el error, maximiza el número de puntos y minimiza el inverso de la densidad). El considerar la densidad para realizar el ajuste ha proporcionado mejores resultados que los obtenidos al minimizar el error y maximizar el número de puntos. Al introducir la medida de densidad se ha experimentado con diferentes factores de peso para el agregado de la función objetivo, lo cual deja abierto un tema de discusión para establecer el valor de éste en función del umbral propuesto.

En este capítulo se formuló el ajuste de triángulos como un problema de optimización no lineal que minimiza el error de los puntos conocidos al ajuste, maximiza los puntos que forman parte de éste y minimiza la inversa de la densidad. El problema se resuelve utilizando la heurística llamada evolución diferencial y se puede observar que se logra la extracción exitosa de uno o varios triángulos en al menos 90 % de los

casos.



## Capítulo 6

# Ajuste de cuadriláteros

---

En muchas aplicaciones prácticas de Visión por Computadora es necesario realizar el reconocimiento de objetos a partir de una imagen, los cuales pueden ser descritos por triángulos o cuadriláteros según sea el caso. Rosin menciona en [21] que existen diferentes propiedades de apariencia que pueden ser utilizadas para discriminar objetos a partir de su color, textura o forma, siendo esta última la propiedad más poderosa para describir y diferenciar objetos, por lo que propone las llamadas “medidas de forma” para tres de las formas más comunes: elipses, triángulos y rectángulos. Para determinar el parecido entre un conjunto de puntos y las formas mencionadas propone los factores de elipticidad, triangularidad y rectangularidad pero sin proponer un método de ajuste para alguna de las formas.

En [22] se propone un método de ajuste de rectángulos y cuadrados a las regiones cerradas dadas usando las ideas propuestas en [23], [24] y [25]. La idea reside en la normalización del objeto y sus primitivas simultáneamente, utilizando los momentos de cuarto orden, con lo que se contrarresta el problema de normalización de la rotación. Utiliza los momentos basados en área para obtener un ajuste global, v.g., que es resistente a la presencia de ruido y otro tipo de distorsiones que se puedan presentar. El método recibe como entrada el contorno de la región que se desea ajustar y devuelve las esquinas del rectángulo o cuadrado que encierra dicha región.

El método que se propone en [22] no soporta el ajuste de varios rectángulos o cuadrados. Por la definición del problema es capaz de extraer sólo regiones rectangulares ó cuadradas, pero no irregulares.

En este capítulo se propone un método de ajuste para la extracción de cuadriláteros a partir de un conjunto de puntos conocido utilizando como función de ajuste la suma de las distancias perpendiculares del punto a la forma propuesta. El problema propuesto se resuelve utilizando la heurística llamada evolución diferencial. Se presentan pruebas y resultados que demuestran el buen desempeño del algoritmo propuesto.

## 6.1. Descripción del problema

Dado un conjunto de puntos  $\mathbf{p} \in \mathbb{R}^2$ , para  $1 \leq i \leq m$ , en un sistema de coordenadas globales  $xy$ , se buscará el cuadrilátero formado por el conjunto de cuatro vértices (o puntos)  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$  que mejor represente al conjunto de puntos dado.

Al igual que en el ajuste de triángulos, hemos utilizado el valor de la suma de las distancias perpendiculares de los puntos dados a los lados del cuadrilátero.

## 6.2. Cálculo de las distancias perpendiculares

Para determinar la distancia perpendicular entre un punto y el cuadrilátero propuesto, se toman las distancias a los vértices y a las aristas, tal y como se hizo con el ajuste de triángulos. En este caso es necesario calcular la distancia a un vértice más y, por ende, a un segmento de recta más (ver figuras 6.1a y 6.1b).

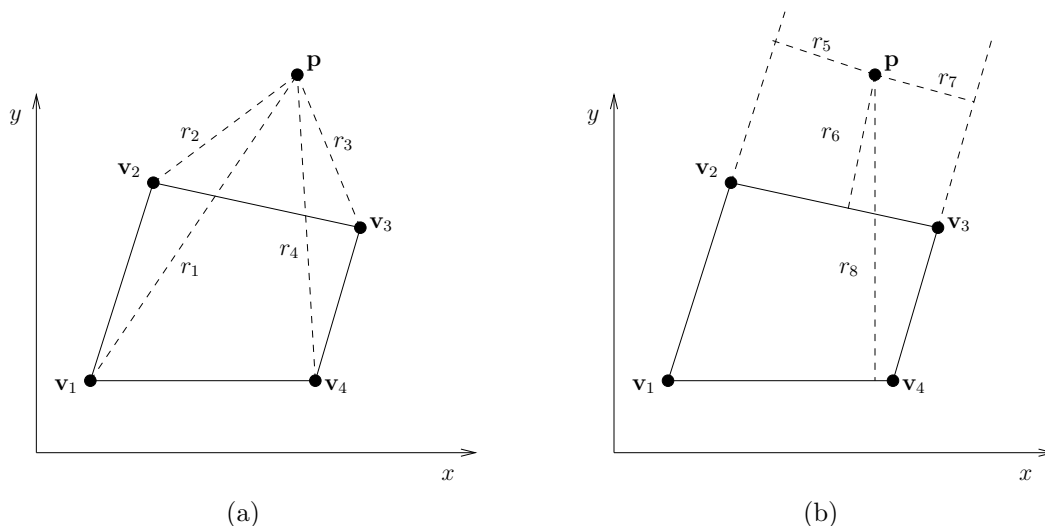


Figura 6.1: (a) Distancias del punto  $\mathbf{p}$  a los vértices de un cuadrilátero. (b) Distancias perpendiculares del punto  $\mathbf{p}$  a los cuatro lados del cuadrilátero.

Para el ajuste de cuadriláteros se deben retomar las consideraciones hechas para el ajuste de triángulos, v.g., la distancia del punto  $\mathbf{p}$  a un lado del cuadrilátero se considera válida si la proyección perpendicular de éste yace dentro del segmento de recta formado por el lado del cuadrilátero. Por lo tanto, el conjunto de distancias  $\mathbf{r}$  para las figuras 6.1a y 6.1b el conjunto de distancias estaría formado por  $\mathbf{r} = \{r_1, r_2, r_3, r_4, r_6, r_8\}$ .

Una vez que se tiene el conjunto de distancias válidas, se puede definir una función  $D$ , que toma como argumentos el vector de vértices  $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{v}_4^T]$  que

forman el cuadrilátero y un punto conocido  $\mathbf{p}$ . Por lo tanto, la distancia entre el punto conocido  $\mathbf{p}$  y el cuadrilátero  $\mathbf{v}$  será el valor que entregue la función:

$$D : \mathbb{R}^8 \times \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$D(\mathbf{v}, \mathbf{p}) = \min(r_i) \quad (6.1)$$

### 6.3. Solución al problema de optimización

La extracción de cuadriláteros, presentó problemas al realizar la extracción de un cuadrilátero sobre una imagen que contiene varios de ellos. Este problema es idéntico al presentado en la extracción de triángulos, ya que se pueden presentar casos en los que el ajuste esté formado por varios cuadriláteros, lo cual incrementa el número de puntos que pertenecen al ajuste y, en consecuencia, el valor de la función objetivo es mucho menor que realizar un ajuste adecuado con un número de puntos mucho menor. Un ejemplo de ello se puede ver en la sección 5.2.

Para resolver el problema, se introdujo una medida de densidad, la cual se modificó para la extracción de cuadriláteros

$$\rho = \frac{m}{\|\mathbf{v}_1\mathbf{v}_2\| + \|\mathbf{v}_2\mathbf{v}_3\| + \|\mathbf{v}_3\mathbf{v}_4\| + \|\mathbf{v}_4\mathbf{v}_1\|} \quad (6.2)$$

donde  $m$  es el número de puntos que forman parte del ajuste y el divisor de (6.2) representa el perímetro del cuadrilátero ajustado. De esta forma se buscará que se tengan un mayor número de puntos por unidad de longitud y evitar así, resultados indeseados.

Para obtener los parámetros desconocidos del ajuste es necesario minimizar la siguiente función:

$$g(\mathbf{v}) = \frac{1}{m} \sum_{\text{si } D < U} D(\mathbf{v}, \mathbf{p}_i) + \frac{2U(n-m)}{n} + \frac{2}{\rho} \quad (6.3)$$

De igual forma que en el caso de los triángulos, el factor de dos que acompaña al tercer término de  $g$  fue elegido de forma empírica, ya que presentaba mejores resultados al realizar el ajuste. La evolución diferencial es la heurística que resolverá el problema de optimización no lineal planteado en (6.3) y proporcionará los parámetros desconocidos del ajuste, que en este caso serán los vértices del cuadrilátero  $\mathbf{v}_1$ ,  $\mathbf{v}_2$ ,  $\mathbf{v}_3$  y  $\mathbf{v}_4$ .

Para representar el problema en la evolución diferencial, el vector de parámetros de cada uno de los individuos contendrá los vértices del cuadrilátero propuesto. Dichos vértices serán ordenados en sentido horario, con la finalidad de que se cumpla con el modelo presentado en la figura 6.1b.

Para que los individuos contengan cuadriláteros válidos de acuerdo al modelo establecido, será necesario reparar a cada uno éstos antes de realizar la evaluación de la función objetivo (6.3), v.g., se ordenarán los vértices del cuadrilátero que éste contenga.

El algoritmo 14 muestra el pseudocódigo para el ordenamiento de los puntos. El pseudocódigo para evaluar a cada uno de los individuos y el que resuelve (6.3) son idénticos a los utilizados para realizar el ajuste de triángulos (descritos en la sección 5.4), pero tomando en cuenta un vértice más.

---

**Algoritmo 14** *ordenar*( $\mathbf{x}$ )

---

**Entrada:** Vector  $\mathbf{x}$  que representará los parámetros del triángulo ( $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ )

**Salida:** Vector de parámetros  $\mathbf{x}$  ordenado

- 1: Obtener el vértice con la componente  $x$  más pequeña y con la  $y$  más grande
  - 2: **if** El vértice obtenido es diferente a  $\mathbf{v}_1$  **then**
  - 3:   Intercambiar dicho vértice de tal forma que el vértice con menor  $x$  y mayor  $y$  se encuentre en  $\mathbf{v}_1$
  - 4: **end if**
  - 5: **for**  $i = 1$  to 2 **do**
  - 6:    $\vec{a} \leftarrow v_{i+1} - v_i$
  - 7:   **for**  $j = i + 2$  to 4 **do**
  - 8:      $\vec{b} \leftarrow v_j - v_i$
  - 9:     Obtener la componente  $\hat{k}$  del vector resultante de  $\vec{a} \times \vec{b}$ , v.g.,  $\hat{k} = \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}$
  - 10:     **if**  $\hat{k} > 0$  **then**
  - 11:       Intercambiar  $v_{i+1}$  y  $v_j$
  - 12:        $\vec{a} \leftarrow v_{i+1} - v_i$
  - 13:     **end if**
  - 14:   **end for**
  - 15: **end for**
- 

## 6.4. Extracción de varios cuadriláteros

SD+ED puede ser utilizado para realizar la extracción de varios cuadriláteros de un conjunto de puntos dado. Para ello, cada vez que un cuadrilátero es detectado, los puntos que pertenecen a éste son eliminados del conjunto conocido y se reanuda la búsqueda, tal y como se muestra en la sección 5.4, considerando que se tiene un vértice más.

## 6.5. Pruebas

Para determinar el desempeño de SD + ED se generaron cuatro pruebas que contienen el mismo porcentaje de datos atípicos (50 %). La primera prueba consiste en realizar la extracción de dos cuadriláteros de diferentes dimensiones, pero con la característica de que el cuadrilátero más pequeño se encuentra dentro del más grande.

La segunda prueba consiste en realizar la extracción de tres cuadriláteros que se intersectan en diferentes puntos, formando incluso nuevos cuadriláteros a partir de los puntos de intersección de sus aristas, lo cual hace a ésta una de las pruebas más difíciles a las que se somete el método propuesto.

La tercera prueba consiste en realizar la extracción de tres cuadrilátero que se encuentran distribuidos sobre la imagen. La última prueba consiste en realizar la extracción de un cuadrilátero. Éste se encuentra acompañado de puntos atípicos que representan el 50 % de los puntos conocidos.

Se realizaron 100 ejecuciones de cada una de las pruebas y se reportaron los resultados obtenidos.

Las pruebas se generaron a partir de los siguientes datos:

### Prueba 1

Cuadrilátero	Vértices
1	(184, 267, 393, 391, 416, 133, 277, 127)
2	(259, 269, 335, 296, 369, 180, 301, 165)

### Prueba 2

Cuadrilátero	Vértices
1	(233, 219, 393, 224, 394, 117, 240, 115)
2	(138, 56, 161, 316, 297, 257, 302, 150)
3	(337, 83, 415, 177, 473, 122, 388, 42)

### Prueba 3

Cuadrilátero	Vértices
1	(100, 422, 210, 423, 210, 320, 100, 322)
2	(395, 86, 490, 154, 517, 68, 461, 45)
3	(314, 217, 318, 273, 561, 277, 562, 221)

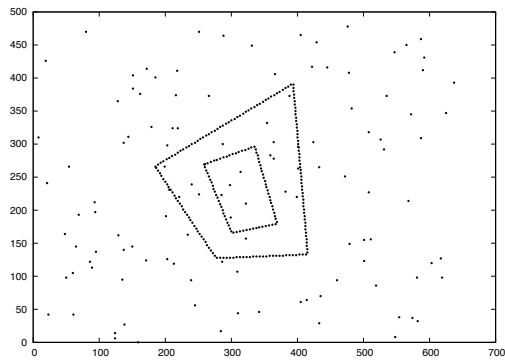
### Prueba 4

Cuadrilátero	Vértices
1	(126, 372, 388, 372, 442, 242, 174, 242)

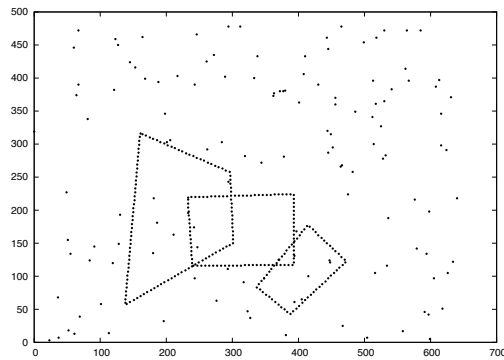
Los parámetros que se utilizaron en la evolución diferencial, se muestran a continuación:

- Límites de la imagen  $X \in [0, 600]$  y  $Y \in [0, 480]$
- Número de individuos = 30
- Número máximo de generaciones = 10000
- Coeficiente de recombinación = 0.9
- Factor de diferencia 0.7
- Valor de umbral  $U = 10$

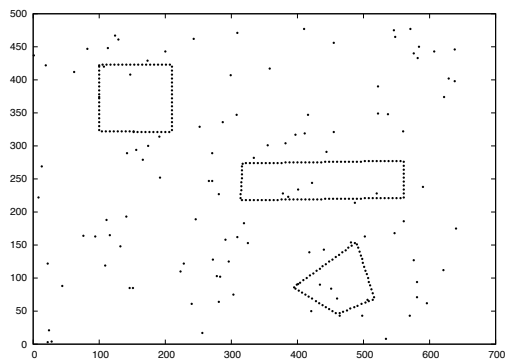
En la figura 6.2 se pueden observar las pruebas gráficamente.



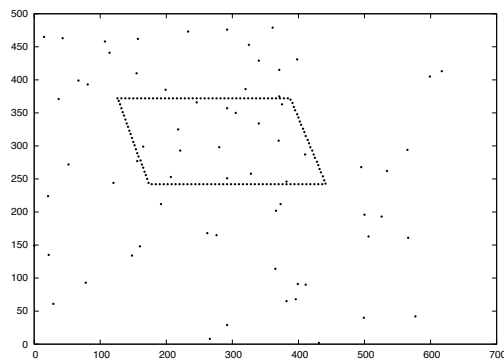
(a)



(b)



(c)



(d)

Figura 6.2: (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4

Los resultados de las ejecuciones se muestran en la tabla 6.1.

	SD+ED			Evaluaciones de la función objetivo	
	Mínimo	Maximo	Media	Media	Desviación estándar
Prueba					
1	0	2	1.96	20052	1735
2	0	3	2.79	34855	3943
3	1	3	2.98	39623	3450
4	1	1	1	10773	1168

Tabla 6.1: Resultados de la extracción de cuadriláteros con SD + ED, se muestra el mínimo, máximo y media de cuadriláteros extraídos exitosamente, así como la media y desviación estándar del número de evaluaciones de la función objetivo.

## 6.6. Sumario

Se puede deducir que SD + ED realizó la extracción exitosa en un 98 % de las ejecuciones de la primera prueba, mientras que sólo realizó el 88 % de extracciones correctas en la prueba dos, la cual se mencionó que era la más complicada de realizar debido a que se forman nuevos cuadriláteros que pueden ocasionar que el algoritmo se quede atrapado en un óptimo local, ya que el uso de una heurística como la evolución diferencial no garantiza que se llegue el óptimo global.

Es importante remarcar que realizar la extracción de un cuadrilátero de la forma que se propone en este capítulo es menos costoso que realizar la extracción de cuatro líneas, v.g., que produce menos evaluaciones de la función objetivo. En el capítulo dedicado al ajuste robusto de líneas, se puede observar que realizar la extracción de cuatro líneas necesitaría en promedio 16,000 evaluaciones de la función objetivo, mientras que realizar la extracción de un triángulo realizaría en promedio 10,773 según los resultados que se obtuvieron en este capítulo.

La función de agregación propuesta considera tres problemas que es necesario resolver para obtener un ajuste adecuado (minimiza el error, maximiza el número de puntos y minimiza el inverso de la densidad). El considerar la densidad para realizar el ajuste ha proporcionado buenos resultados, pero se ha tenido que experimentar con diferentes valores para el peso de éste agregado de la función, lo cual deja abierto un tema de discusión para establecer el valor adecuado que deberá llevar el término en función del residual propuesto, tal y como se hizo con el término que maximiza el número de puntos del ajuste.

En este capítulo se formuló el ajuste de cuadriláteros como un problema de opti-

mización no lineal que minimiza el error de los puntos conocidos al ajuste, maximiza los puntos que forman parte de éste y minimiza la inversa de la densidad. El problema se resuelve utilizando la heurística llamada evolución diferencial y se puede observar que se logra la extracción exitosa de uno o varios cuadriláteros bajo condiciones de ruido de un 50% de datos atípicos.



## Capítulo 7

# Detección de círculos y elipses utilizando la evolución diferencial

---

La detección de elipses/círculos es una tarea importante en reconocimiento de patrones y visión por computadora. Muchas aproximaciones han sido desarrolladas en los últimos 20 años para realizar la detección de elipses. Los métodos desarrollados pueden dividirse en dos grupos [26, 27]: (1) métodos basados en la optimización de una función de costo o error y (2) métodos basados en agrupamiento o votación.

Dentro del grupo (1) tenemos los métodos que minimizan la suma de las distancias algebraicas [28, 29, 30], el error geométrico [8] y los que minimizan la suma de las distancias ortogonales [6]. Entre los métodos del grupo (2) tenemos principalmente la transformada de Hough [31, 32], RANSAC [33] y clasificadores difusos [34]. Estos métodos se consideran robustos porque pueden extraer varias elipses aún en presencia de puntos atípicos.

En el capítulo 2 se demostró que el ajuste de elipses según la suma de las distancias ortogonales, es un procedimiento robusto, que puede resistir grandes cantidades de puntos atípicos. Se probó que el algoritmo llamado SD + ED puede ajustar elipses aún cuando el nivel de puntos atípicos va del 70 % al 100 %.

En este capítulo se extiende la idea presentada previamente para detectar, esto es, para extraer todos los puntos que pertenecen a un cúmulo elipsoidal o circular, a partir de los círculos o elipses ajustadas por SD + ED.

### 7.1. Descripción del problema

El problema y su solución se describe de lo general a lo particular. Se supone que se cuenta con un algoritmo de ajuste de elipses/círculos a partir de un conjunto de puntos conocido. Ahora el problema reside en separar los puntos que corresponden a una elipse/círculo de los puntos atípicos que se encuentran en el conjunto dado.

Para esto, la distancia ortogonal de los puntos conocidos a la forma geométrica,  $d_i$  para cada punto  $i$ , puede modelarse como una variable aleatoria con media cero y una cierta variabilidad. Como el conjunto de puntos conocido puede contener datos atípicos, es necesario utilizar estadística robusta. La medida de variabilidad escogida es la *mediana de las desviaciones absolutas* (MDA), que se define como:

$$\text{MDA} = \text{mediana}|d_i| \quad (7.1)$$

MDA tiene la característica de ser mucho más robusta que la varianza. La relación que existe entre la desviación estándar,  $\sigma$ , y la MDA es  $\sigma \approx 1.4826 \text{ MDA}$ .

Una vez estimada la desviación estándar se pueden separar los puntos que pertenecen al ajuste de los atípicos de la siguiente forma: para cada punto se calcula  $d_i$  y el punto forma parte de la distribución si

$$\frac{d_i}{\hat{\sigma}} \leq 2.5 \quad (7.2)$$

y el punto es considerado atípico si

$$\frac{d_i}{\hat{\sigma}} > 2.5 \quad (7.3)$$

De esta forma se pueden detectar los puntos que pertenecen a la forma geométrica y los que son atípicos. Es importante mencionar que se está suponiendo que el ruido que podrían tener los puntos en su posición está modelado con una distribución gaussiana. Este es el mejor procedimiento que puede realizarse para separar los puntos de la distribución de los atípicos [9].

### 7.1.1. Ajuste de círculos

Ajustar un círculo es un problema más sencillo que ajustar una elipse, ya que sólo se debe obtener el centro del círculo y su radio. Se propone minimizar la suma de las distancias euclidianas, en vez de la suma del cuadrado de las distancias. El problema se define como

$$g_c(\mathbf{x}) : \quad \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$g_c(\mathbf{x}) = \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p}'_i\| \quad (7.4)$$

donde  $\mathbf{p}_i = [x_i, y_i]$ ,  $i \in [1 : n]$ , es el conjunto de puntos conocidos;  $\mathbf{p}'_i$  es el punto de contacto ortogonal del punto  $\mathbf{p}_i$  sobre el círculo. El círculo está representado por  $\mathbf{x} = [x_c, y_c, R]^T$ ; donde  $[x_c, y_c]^T = \mathbf{c}$  es el vector que representa el centro del círculo y  $R$  es su radio.

El punto de contacto  $\mathbf{p}'_i$  se calcula como

$$\mathbf{p}'_i = \mathbf{c} + R \frac{\mathbf{p}_i - \mathbf{c}}{\|\mathbf{p}_i - \mathbf{c}\|}. \quad (7.5)$$

La función  $g_c$ , en (7.4) no puede resolverse por métodos numéricos convencionales dado que involucra la función raíz cuadrada y esta función no tiene derivadas continuas en el espacio de búsqueda. Para resolver el problema presentado en (7.4) se utiliza la heurística llamada evolución diferencial.

La definición simple de la función  $g_c$  puede usarse para ajustar un solo círculo al conjunto de puntos dado, pero si varios círculos están presentes dentro de dicho conjunto, la función  $g_c$  se modifica a:

$$g_c(x) = \frac{1}{n} \sum_{\substack{n \\ \text{si } d_i < U}} d_i(\mathbf{x}) + 2U \frac{m-n}{m} + 2U \exp\left(\frac{-1.3826n}{P}\right), \quad (7.6)$$

donde  $d_i(\mathbf{x}) = \|\mathbf{p}_i - \mathbf{p}'_i(\mathbf{x})\|$ ,  $P = 2\pi R$  es el perímetro del círculo, y sólo se consideran parte del ajuste aquellos puntos cuya distancia al círculo sea menor a un umbral  $U$ . El segundo término en (7.6) maximiza el número de puntos que se utilizan en el ajuste y el tercer término indica que el número de puntos por unidad de perímetro debe ser maximizado.

Un problema de maximización puede cambiarse a otro de minimización simplemente cambiando el signo de la expresión que representa al problema. De esta manera en el segundo término en (7.6) ( $2U(m-n/m)$ ), está cambiado el signo de  $n$  pero para mantener el valor positivo, se normaliza usando el total de puntos conocidos  $m$  y se pondera con el valor de  $2U$ . El tercer término en (7.6) también tiene el signo cambiado para la expresión del número de puntos por unidad de perímetro ( $n/P$ ), pero para que su valor sea positivo se toma su exponencial. El máximo valor que puede tomar el tercer término es  $2U$  si ningún punto es usado en el ajuste y el mínimo valor que puede tener se aproxima a cero en tanto más puntos se usen en el ajuste. La constante de 1.3826 se calculó para hacer que el valor de la exponencial sea igual a 0.5 cuando el valor  $n/P$  es también 0.5.

### 7.1.2. Ajuste de elipses

El problema de ajustar una o más elipses en un conjunto de puntos dado, resulta más complicado que el ajuste de un círculo, ya que se deben estimar cinco parámetros por cada elipse. La misma ecuación (7.6) es minimizada para el ajuste de varias elipses. El problema ahora es calcular la distancia  $d_i$  de un punto  $\mathbf{p}$  a la elipse representada por el vector  $\mathbf{x} = [a, b, x_c, y_c, \phi]^T$  donde  $a$  y  $b$  son el semieje mayor y menor de la elipse,  $(x_c, y_c)$  son las coordenadas del centro de la elipse y  $\phi$  es el ángulo de inclinación de la elipse respecto al eje de las  $x$ .

El cálculo de la distancia ortogonal  $d_i$  de un punto  $\mathbf{p}$  a la elipse dada, es un problema no lineal que se resuelve con el método de Newton. Los detalles de su implementación son descritos en [6].

El perímetro de la elipse utilizado en (7.6), fue calculado con la aproximación de Ramanujan:

$$P \approx \pi \left[ 3(a + b) - \sqrt{(a + 3b)(3a + b)} \right] \quad (7.7)$$

El ajuste de elipses presentado en el capítulo 4 sólo utilizaba imágenes que contenían los bordes de las elipses, los cuales se pueden obtener con un detector de bordes (como Sobel). En este caso el problema se enfoca más hacia el agrupamiento de datos: existen sólo algunos puntos por elipse, dichos datos pueden no cubrir completamente el perímetro de la elipse y es posible que las elipses se encuentren anidadas, v.g., una dentro de la otra.

## 7.2. Algoritmos

Tomando en cuenta que la minimización de la suma de las distancias ortogonales es un problema no lineal que no puede ser resuelto utilizando métodos numéricos convencionales, se utilizaron dos diferentes heurísticas para resolver el problema. La primera heurística fue la evolución diferencial y la segunda RANSAC.

Ambos algoritmos extraen una elipse a la vez, y se ejecutan varias veces hasta que el error, expresado por la ecuación (7.6), sea muy grande, mayor que un umbral dado, indicando que no existe ningún círculo o elipse más en el conjunto de puntos.

Una vez que una elipse es ajustada, los puntos que pertenecen a ella son extraídos con el proceso explicado al principio de la sección 7.1, y el número de puntos totales  $m$  se disminuye acorde al número de puntos del círculo/elipse detectado.

Es importante mencionar que la versión de RANSAC que se utilizó en este capítulo, para resolver el problema de optimización, es diferente a la utilizada para los ajustes explicados en los capítulos 3 y 4 (ajuste robusto de líneas y elipses). La técnica que aquí se utiliza es una versión “acelerada” de RANSAC llamada KRANSAC [33]. Ésta proporciona mejores resultados que el RANSAC tradicional debido a que el número de iteraciones que calcula aumenta enormemente la probabilidad de hallar el ajuste deseado, considerando que se conoce de antemano el número de puntos que forman parte del ajuste buscado, lo cual resulta poco práctico. Los detalles de KRANSAC se discutirán en la siguiente sección.

### 7.2.1. KRANSAC

La forma en la que KRANSAC resuelve los problemas de optimización es muy diferente a como lo hace la ED. Del conjunto de puntos conocido toma aleatoriamente un subconjunto de ellos para estimar el modelo. Si el error del modelo, en este caso la evaluación de (7.6), es mínimo, el modelo se acepta y a los puntos que son tomados se

les denomina conjunto consenso. Ahora el problema radica en cuantas veces se debe repetir el proceso mencionado.

El número de veces que se itera fue calculado acorde a [33]: donde definen la *pureza de los datos*,  $\mu$ , como  $\mu = \frac{n_0}{n}$ , donde  $n_0$  es el número de puntos que pertenecen a las curvas, y  $n$  es el total de puntos del conjunto conocido. La probabilidad de que seleccionando aleatoriamente  $d$  puntos de cada curva se obtenga la curva  $k$  es igual a:

$$P = \sum_{i=1}^k P_i = \sum_{i=1}^k \frac{\binom{n_i}{d}}{\binom{n}{d}} \quad (7.8)$$

donde  $n_i$  es el número de puntos de la curva  $i$ ,  $1 \leq i \leq K$ . Bajo la suposición que el ruido está uniformemente distribuido y que  $\mu n/k \gg d$ , (7.8) puede aproximarse como

$$P \approx P' = \frac{\mu^d}{k^{d-1}} \quad (7.9)$$

Para calcular el número de veces que debe repetirse el proceso, se considera la probabilidad  $P_c$  de que al menos una de las  $K$  hipótesis generadas corresponda a la curva correcta

$$P_c = 1 - (1 - P')^K \quad (7.10)$$

de donde puede despejarse el número de hipótesis  $K$  que es necesario generar

$$K \geq \frac{\log(1 - P_c)}{\log(1 - P')} \quad (7.11)$$

para garantizar que  $P_c \geq 0.95$ . El número de iteraciones  $K$  se repite  $k$  veces para extraer las  $k$  curvas presentes en el conjunto de puntos. Nótese que sólo puede calcularse (7.9) si se conoce el número de puntos que pertenecen a cada curva de antemano.

### 7.3. Pruebas y resultados

Para analizar el comportamiento de los algoritmos propuestos SD + ED y KRAN-SAC, se generó un conjunto de cuatro imágenes con círculos y otras cuatro imágenes con elipses. Ambos conjuntos de imágenes se pueden observar en las figuras 7.1 y 7.2. Las imágenes que contiene círculos fueron inspiradas en [34]. Los algoritmos propuestos se compararon con la transformada de Hough aleatoria (RHT) [32] para el caso de los círculos, y con la implementación de la transformada de Hough (HT) [31] para las elipses.

Las imágenes con círculos son de tamaño  $200 \times 200$  píxeles y las imágenes del conjunto de elipses son de tamaño  $640 \times 480$  píxeles. Éstas fueron generadas a partir de los siguientes datos.

**Prueba 1**

Círculo	Centro	Radio	Número de puntos
1	(100, 120)	55	30
2	(50, 110)	40	30
3	(150, 110)	40	25

**Prueba 2**

Círculo	Centro	Radio	Número de puntos
1	(100, 100)	30	40
2	(100, 100)	80	90

**Prueba 3**

Círculo	Centro	Radio	Número de puntos
1	(125, 80)	40	20
2	(75, 80)	40	20
3	(50, 120)	40	20
4	(100, 120)	40	20
5	(150, 120)	40	20

**Prueba 4**

Círculo	Centro	Radio	Número de puntos
1	(100, 60)	30	25
2	(100, 100)	20	30
3	(100, 100)	95	70
4	(50, 150)	10	10
5	(150, 150)	10	15

**Prueba 5**

Elipse	Centro	$a$	$b$	$\phi$	Número de puntos
1	(320, 160)	80	40	61	90
2	(80, 213)	80	40	128	90
3	(400, 373)	80	40	178	90

**Prueba 6**

Elipse	Centro	$a$	$b$	$\phi$	Número de puntos
1	(449, 231)	80	40	121	90
2	(600, 124)	80	40	93	87
3	(180, 36)	80	40	13	71

**Prueba 7**

Elipse	Centro	$a$	$b$	$\phi$	Número de puntos
1	(200, 200)	100	50	45	90
2	(280, 280)	100	50	45	90

**Prueba 8**

---

Elipse	Centro	$a$	$b$	$\phi$	Número de puntos
1	(150, 350)	100	70	30	30
2	(150, 350)	100	70	30	25
3	(220, 100)	60	30	0	30
4	(360, 230)	80	40	80	30

Los límites de búsqueda para el caso de los círculos fue:  $x_c, y_c \in [0, 200]$ ,  $R \in [5, 100]$ . Los parámetros para la ED fueron: 40 individuos, máximo número de iteraciones igual a 10,000, constante de diferencia igual a 0.7, constante de recombinación igual a 0.9 y el valor de umbral 4.

Para el caso de las elipses los límites fueron:  $x_c \in [0 : 640]$ ,  $y_c \in [0 : 480]$ ,  $a, b \in [10 : 200]$  y  $\phi \in [0, 180]$ . Los parámetros de recombinación y diferencia son idénticos a los utilizados en el caso de los círculos, a diferencia del umbral que se le dio un valor de 5.

Se realizaron 100 ejecuciones de cada una de las técnicas y las estadísticas para el conjunto de imágenes de círculos se pueden ver en la tabla 7.1. Como el número de puntos de cada imagen es constante, el número de iteraciones para KRANSAC es modificado en cada ciclo. La técnica con mejor desempeño fue KRANSAC, con el menor número de iteraciones, pero se debe recordar que el número de iteraciones fue calculado tomando en cuenta que se conoce el número de puntos por círculo de antemano.

Las estadísticas de los resultados para el caso de las elipses se puede ver en la tabla 7.2. HT se ejecutó una sola vez debido a que su implementación utiliza una búsqueda exhaustiva, y esta es la razón del número tan elevado de iteraciones que realiza esta técnica. La técnica con mejores resultados es la que utiliza la ED, ya que presenta el menor número de iteraciones y puede realizar la extracción de todas las elipses presentes en la imagen.

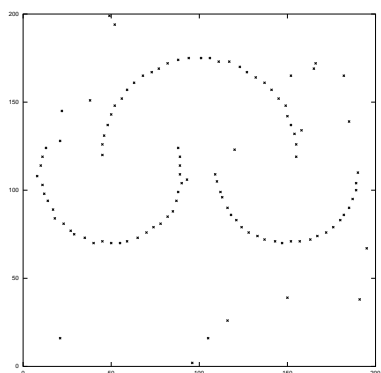
	KRANSAC			Evaluaciones de la función objetivo	
	Mínimo	Máximo	Media	Media	Desviación estandar
Prueba					
1	2	3	2.97	495	0
2	2	2	2	138	0
3	5	5	5	1305	0
4	5	5	5	1755	0
<hr/> <hr/>					
	SD+ED			Evaluaciones de la función objetivo	
	Mínimo	Máximo	Media	Media	Desviación estandar
Prueba					
1	3	3	3	18143	1858.3
2	2	2	2	9701	1141
3	5	5	5	47514	5398
4	4	5	4.98	34599	3101
<hr/> <hr/>					
	RHT			Iteraciones	
	Mínimo	Máximo	Media	Media	Desviación estandar
Prueba					
1	3	3	3	7760	274
2	2	2	2	1914	270
3	5	5	5	5288	769
4	2	2	2	9441	1707

Tabla 7.1: Estadísticas de los resultados obtenidos por KRANSAC, SD + ED y RHT para la extracción de círculos. Muestra el mínimo, máximo y media de los círculos extraídos exitosamente. También se puede observar la media y desviación estándar del número de evaluaciones de la función objetivo.

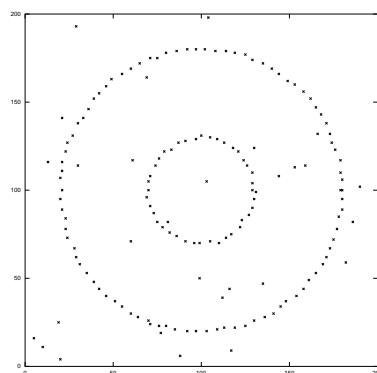


		KRANSAC			Evaluaciones de la función objetivo	
		Mínimo	Máximo	Media	Media	Desviación estándar
Prueba						
1	2	3	2.93	11745	0	
2	0	3	2.70	11709	0	
3	0	2	1.66	5436	0	
4	0	4	1.07	86460	0	
		SD+ED			Evaluaciones de la función objetivo	
		Mínimo	Máximo	Media	Media	Desviación estándar
Prueba						
1	2	3	2.99	40572	5351	
2	2	3	2.99	37588	4656	
3	0	2	1.64	23196	2859	
4	1	4	3.41	23196	2859	
		HT				
		Extracciones	Iteraciones			
Prueba						
1	3	11751531				
2	2	9675569				
3	1	5221934				
4	2	1147649				

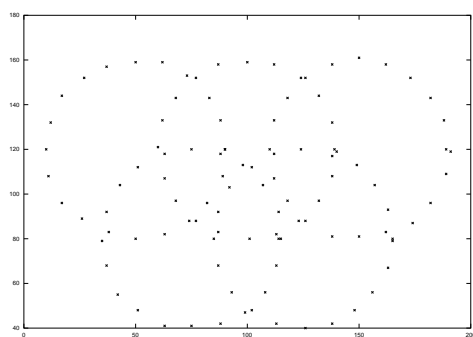
Tabla 7.2: Estadísticas de los resultados obtenidos por KRANSAC y SD + ED para la extracción de elipses. Muestra el mínimo, máximo y media de las elipses extraídas exitosamente. También se puede observar la media y desviación estándar del número de evaluaciones de la función objetivo. Para el caso de HT sólo se muestra el número de extracciones exitosas y las iteraciones, ya que se ejecuta sólo una vez.



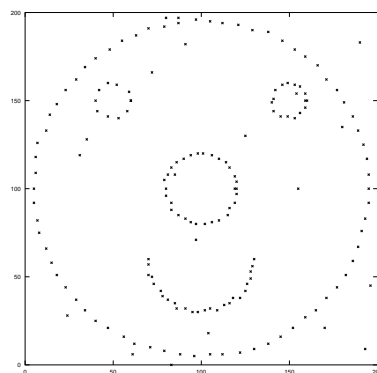
(a)



(b)



(c)



(d)

Figura 7.1: (a) Prueba 1. (b) Prueba 2. (c) Prueba 3. (d) Prueba 4.

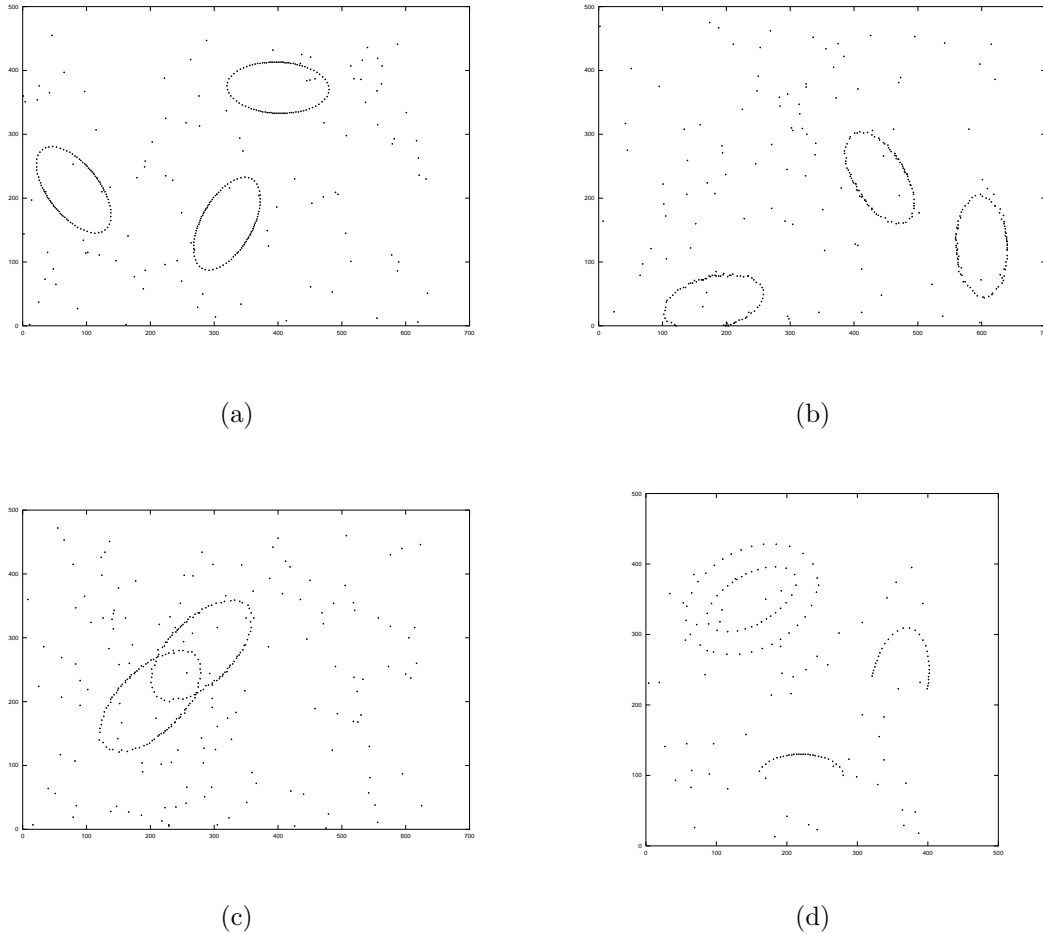


Figura 7.2: (a) Prueba 5 con 50 % de puntos atípicos. (b) Prueba 6 con 50 % de puntos atípicos y ruido Gaussiano sobre los puntos de la elipse con  $\sigma = 2$ . (c) Prueba 7 con 70 % de puntos atípicos. (d) Prueba 8 con 50 % de puntos atípicos

## 7.4. Sumario

El algoritmo KRANSAC no puede realizar la extracción de varias formas si sólo se minimiza el error de los puntos al círculo/elipse. Es necesario maximizar el número de puntos que forman parte del ajuste y al mismo tiempo el número de puntos por unidad de perímetro.

Los pesos dados a cada uno de los términos de la función a minimizar fueron ajustados de forma experimental. Se consideró que el cálculo del valor óptimo de estos pesos o incluso el cambiar de un problema mono objetivo a uno multi objetivo es un tema que podría investigarse a futuro.

La implementación utilizada del algoritmo HT para elipses, no es capaz de detectar elipses parciales debido a que realiza el ajuste a partir de los puntos extremos del semieje mayor. Este algoritmo utiliza un solo acumulador lo que lo hace eficiente en el uso de memoria, pero no en el tiempo de ejecución porque éste utiliza una búsqueda exhaustiva proporcional a  $O(n^3)$  donde  $n$  es el número de puntos conocidos.

¿Cómo saber el número de círculos/elipses a extraer? En el caso de los círculos se puede utilizar el mínimo número de puntos que se necesitan para ajustar un círculo. En el caso de las elipses la extracción se puede detener si el valor de la función de aptitud tiene un valor muy grande (p.ej. un valor de 30). Por lo tanto, esto no es problema para las técnicas descritas en este capítulo.

En este capítulo se propuso un nuevo método para la detección de círculos o elipses a partir de un conjunto de puntos dado con una porción grande de puntos atípicos. El método se puede ver como una combinación de un método de ajuste y uno de agrupamiento, ya que puede extraer varias formas geométricas ajustando primero las formas.

El ajuste robusto está basado en la minimización de la suma de las distancias, en lugar del cuadrado de éstas. Esto generó un problema con derivadas discontinuas en el espacio de búsqueda, el cual fue resuelto utilizando la heurística llamada evolución diferencial.

El método tiene la ventaja de que sólo se requiere especificar el espacio de búsqueda de una forma muy cruda (p.ej. el centro de las formas se buscan en toda la imagen), y encuentra más formas en promedio que KRANSAC y la HT aunque representa mayor tiempo de ejecución.

Como trabajo futuro se puede realizar un algoritmo que disminuya enormemente el número de iteraciones realizadas, ya que aunque no exista la derivada en el mínimo de la suma de las distancias, la derivada existe en otros puntos y podría usarse para acelerar la convergencia del método.

# Capítulo 8

## Conclusiones

---

En este trabajo de tesis se propuso un método de ajuste robusto que realiza la extracciones de una o varias de las formas geométricas analizadas (líneas, elipses, círculos, triángulos y cuadriláteros) presentes en un conjunto de puntos dado. El método propuesto es capaz de realizar la extracción exitosa de las formas aún en presencia de más del 50 % de puntos atípicos en el conjunto de puntos conocido.

El método obtiene los parámetros desconocidos del ajuste resolviendo un problema de optimización, el cual debe minimizar la suma de las distancias perpendiculares que van de cada uno de los puntos conocidos al ajuste propuesto. Éste resultó ser no lineal, ya que presentaba una discontinuidad de su derivada en el punto óptimo, por lo que fue necesario resolverlo utilizando la heurística llamada evolución diferencial. En un inicio el problema de optimización sólo realizaba la extracción de una forma geométrica, por lo que se le realizaron modificaciones al problema original, las cuales consistieron en:

- Se consideran parte del ajuste sólo aquellos puntos cuya distancia perpendicular a la forma propuesta sea menor a un valor de umbral dado.
- Para realizar la extracción de varias de las formas se debe maximizar el número de puntos que forma parte del ajuste.

Las modificaciones anteriores permitieron que se realizaran varias extracciones en el mismo conjunto de puntos. El nuevo método propuesto, llamado SD + ED, se comparó contra MMC + RANSAC y presentó mejores resultados en la extracción de líneas y elipses, ya que podía realizar la extracción de varias de las formas incluso con el 100 % de puntos atípicos y aún así tener una variabilidad baja en los parámetros de ajuste.

El nuevo método se utilizó para realizar la extracción de triángulos sin entregar buenos resultados en un inicio debido a los problemas que se presentaron para la extracción de varios de éstos. Dichos problemas se describen a detalle en la sección 5.3, por lo que se tuvo que agregar una nueva modificación al problema ya propuesto.

Dicha modificación consistió en introducir una medida que nos permitió determinar el número de puntos por unidad de longitud del triángulo ajustado. La introducción de dicha modificación proporcionó muy buenos resultados, los cuales no pudieron compararse contra otro método de ajuste de triángulos, ya que no se encontró un método que fuese capaz de realizar la extracción de cualquier tipo de triángulo.

El método ahora compuesto de las tres modificaciones, se utilizó para extraer cuadriláteros, lo cual se tomó como una extensión del problema de extracción de triángulos, ya que para generar un cuadrilátero sólo es necesario agregar un vértice más. Se investigó si existían métodos de ajuste para cuadriláteros y se encontró sólo uno, pero éste sólo es capaz de extraer rectángulos y no cualquier tipo de cuadrilátero. Por lo tanto se puede decir que:

El método de extracción de triángulos y cuadriláteros propuesto en esta tesis es capaz de realizar el ajuste de una o varias de las diferentes variaciones de las formas mencionadas, v.g., puede extraer triángulos escalenos, isósceles, rectángulos y cualquier cuadrilátero con una gran eficiencia (mayor al 95 %).

Con los resultados obtenidos decidimos retomar el ajuste de elipses, ya que éste presentaba problemas con la extracción de elipses concéntricas, los cuales son similares a los observados en los triángulos. Al realizar el ajuste de círculos se observaron las mismas complicaciones tal y como se puede ver en la sección 7.1.1, pero en este caso se trató de ajustar el multiplicador que acompaña a la nueva “restricción”. Cabe mencionar que las modificaciones se aplicaron también al método de MMC + RANSAC, con la finalidad de que se encontrara al mismo nivel de SD + ED. El nuevo método de ajuste se comparó contra otro método representativo de la literatura para ajustar círculos y elipses basado en la transformada de Hough.

En general podemos decir que en este trabajo de tesis se propone un método de ajuste robusto para líneas, elipses, círculos, triángulos y cuadriláteros muy superior a los considerados representativos en la literatura especializada (MMC + RANSAC y HT).

La principal desventaja del método de ajuste que aquí se propone, es el tiempo de ejecución que éste necesita para realizar las extracciones, el cual se encuentra muy por encima del presentado por los métodos elegidos para la comparación.

Este trabajo de tesis produjo los siguientes artículos:

- Luis Gerardo de la Fraga and Gustavo M. López Domínguez. Robust Fitting of Ellipses with Heuristics. *WCCI 2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, July 10 - 23, 2010, pp. 3990 - 3997.
- Gustavo M. López Domínguez y Luis Gerardo de la Fraga. Ajuste robusto de triángulos y cuadriláteros usando la evolución diferencial. Aceptado en el VII

Taller-Escuela de Procesamiento de Imágenes (PI10) en el El Centro de Investigación en Matemáticas A.C. (CIMAT), Guanajuato, Gto, 20 - 22 octubre, 2010.

## 8.1. Trabajo futuro

Como trabajo futuro se puede:

- Realizar un análisis más exhaustivo para determinar los pesos óptimos de cada uno de los términos de la función de costo propuesta, ya que éstos fueron elegidos de forma empírica.
- Realizar un algoritmo que disminuya considerablemente el número de evaluaciones de la función objetivo, ya que aunque no exista la derivada en el mínimo de dicha función, existe en otros puntos y podría usarse para acelerar la convergencia del método.
- Realizar pruebas con datos reales bajo diferentes circunstancias.
- Comparar SD + ED con otros métodos de ajuste que han reportado buenos resultados, como la transformada de Hough aleatoria (Randomized Hough Transform).





# Bibliografía

---

- [1] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.
- [2] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.
- [3] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 04 1991/04/01/.
- [4] EP On and M Spann. Robust optical flow computation based on least-median-of-squares regression. *International Journal of Computer Vision*, 31(1):51–82, Feb 1999.
- [5] J.Jurečková and J. Pícek. *Robust Statistical Method with R*. Chapman & Hall, 1st edition, 2006.
- [6] 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.
- [7] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.
- [8] S. J. Ahn, W. Rauh, and Hans-Jürgen. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283 – 2303, 2001.
- [9] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. Wiley, 2 edition, 2009.
- [10] F. R. Hampel, P. J. Rousseeuw, and E. Ronchetti. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, 1986.
- [11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis an automated cartography. *Communications of the ACM*, 24(6), June 1981.

- [12] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59 – 76, 1997.
- [13] H. Qjidaa and L. Radouane. Robust line fitting in a noisy image by the method of moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 21(11):1216 – 1223, November 1999.
- [14] R.Y.D Xu and J.S. Jin. Camera control and multimedia interaction using individual object recognition. *Journal of Multimedia*, 2(3):77–85, June 2008.
- [15] B.V. Sankar R.L. Karkkainen and J.T. Tzeng. A direct micromechanical approach toward the development of quadratic stress gradient failure criteria for textile composites. *Journal of Composite Materials*, 41(16):1917–1937, Jan 2007.
- [16] R. Safaee-Rad, K. C. Smith, B. Benhabib, and I. Tchoukanov. Application of moment and fourier descriptors to the accurate estimation of elliptical-shape parameters. *Pattern Recognition Letters*, 13(7):497–508, 1992.
- [17] M. Stojmenovic and A. Nayak. *Advances in Image and Video Technology*, volume 4872 of *Lecture Notes in Computer Science*, chapter Direct Ellipse Fitting and Measuring Based on Shape Boundaries, pages 221–235. Springer Berlin / Heidelberg, 2007.
- [18] F.L. Brookstein. Fitting conic sections to scattered data. *Computer Graphics and image Processing*, 9(1):56–71, 1979.
- [19] E. W.Weisstein. Ellipse. From MathWorld–A Wolfram Web Resource. <http://mathworld.wolfram.com/Ellipse.html>.
- [20] K. Zielinsky and R. Laur. *Advances in Differential Evolution*, volume 143, chapter Stopping criteria for differential evolution in constrained single-objective optimization, pages 111–138. Springer, 2008.
- [21] P. L. Rosin. Measuring shape: ellipticity, rectangularity, and triangularity. *Machine Vision and Applications*, 14:172–184, 2003.
- [22] H. Suesse and K. Voss. A new efficient algorithm for fitting of rectangles and squares. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages 1134 – 1137, 2005.
- [23] I. Rothe, H. Suesse, and K. Voss. The method of normalization to determine invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:366–376, 1996.
- [24] K. Voss and H. Suesse. Invariant fitting of planar objects by primitives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:80–84, 1997.

- [25] K. Voss and H. Suesse. A new one-parametric fitting method for planar objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:646–651, 1999.
- [26] Si-Cheng Zhang and Zhi-Qiang Liu. A robust, real-time ellipse detector. *Pattern Recognition*, 38(2):273 – 287, 2005.
- [27] R. Halir and J. Flusser. Numerically stable direct least squares fitting of ellipses, 1998.
- [28] P. O’Leary and P. Z. Murray. Direct and specific least-square fitting of hyperbolæ and ellipses. *Journal of Electronic Imaging*, 13(3):492–503, 2004.
- [29] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:476–480, 1999.
- [30] D. Umbach and K. N. Jones. A few methods for fitting circles to data. *IEEE Transactions on Instrumentation and Measurement*, 20(5):1 – 5, 2000.
- [31] Y. Xie and Q. Ji. A new efficient ellipse detection method. *Pattern Recognition, International Conference on*, 2:1051–4651, 2002.
- [32] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized hough transform (rht). *Pattern Recognition Letters*, 11(5):331 – 338, 1990.
- [33] Y. C. Cheng and S. C. Lee. A new method for quadratic curve detection using k-ransac with acceleration techniques. *Pattern Recognition*, 28(5):663 – 682, 1995.
- [34] R. N. Dave and T. Fu. Robust shape detection using fuzzy clustering: practical applications. *Fuzzy Sets Syst.*, 65(2-3):161–185, 1994.