



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

**Unidad Zacatenco**

**Departamento de Computación**

**Algoritmos para la detección robusta y automática de  
homografías**

Tesis que presenta

**Luis Alberto Camacho Vázquez**

para obtener el Grado de

**Maestro en Ciencias**

**en Computación**

Director de la Tesis

**Dr. Luis Gerardo de la Fraga**

México, D.F.

Septiembre 2012



# Resumen

---

En visión por computadora, una homografía es la matriz que transforma los puntos de un plano a otro. En este trabajo estamos interesados en la detección automática de una o más homografías, esto es, detectar automáticamente los planos presentados en una, dos o varias imágenes, pero en un ambiente con puntos típicos, atípicos y con ruido. En nuestro problema, los puntos atípicos se pueden considerar como los puntos en otros planos, porque representan diferentes distribuciones de datos (u otros conjuntos de puntos de los diversos planos) y el ruido se puede observar, como puntos sobre las imágenes que no forman parte de los planos en la escena tomada (son datos que aparecen a menudo por error de medición).

Para una detección robusta (esto es, a partir de un conjunto de datos que contienen puntos atípicos y con ruido), en el estado del arte, el algoritmo estándar más conocido para resolver este problema es el *algoritmo de consenso de muestra aleatoria* (RANSAC, por sus siglas en inglés). En dicho algoritmo, de todo el conjunto de datos de entrada, se toma un subconjunto de ellos al azar con el fin de estimar el modelo (en nuestro problema, una homografía). Si el error del modelo es mínimo, éste es aceptado y el subconjunto de los puntos implicados se denomina conjunto de consenso. El modelo se estima con algoritmos lineales estándares (que también son los más rápidos). Aquí el problema es cuántas veces el proceso descrito debe repetirse con el fin de obtener la mejor estimación del modelo. RANSAC tiene el inconveniente de establecer de antemano el número de iteraciones en el algoritmo.

En este trabajo se probaron otros dos algoritmos: uno utiliza un algoritmo genético simple aplicado en la tarea de la detección automática de homografías para elegir el conjunto de consenso, en lugar de utilizar muestras al azar. El segundo algoritmo realizado utiliza la metaheurística llamada evolución diferencial, para minimizar el error entre las correspondencias de puntos del conjunto de consenso, al mismo tiempo que se maximiza el número de dichas correspondencias. Se demuestra con experimentos de simulación e imágenes reales, la ventaja de utilizar los algoritmos propuestos en esta tesis, con los que se reduce casi 6 veces, en promedio, el número de iteraciones requeridas por el algoritmo original RANSAC.



# Abstract

---

In computer vision, a homography is the matrix that transforms the points on a plane to another plane. In this work, we are interested in the automatic detection of one or more homographies, that is, we aim to automatically detect the planes submitted into one, two or several images, but in an environment with inliers, outliers and noise. In our problem, the inliers can be considered to be the points in other planes, because they represent different distributions of information (or other point sets of the diverse planes) and the noise can be observed as points on the images that do not form part of the planes in the taken scene (it is information that frequently appears due to measurement errors).

For a robust detection (i. e., from a set points that contains outliers and noise), the state in the art algorithm to solve this problem is called *RANdom SAmple Consensus* (RANSAC). In such algorithm, the entire set of input data, takes a random subset of them to estimate the model (in our problem, a homography). If the error in the model is minimal, this model is accepted and the subset of the implied points is called the consensus set. The model is estimated with standard linear algorithms (which are also the fastest). The problem is to know how many times the described process must be repeated in order to obtain the best estimation of the model. RANSAC has the disadvantage of establishing in advance the number of iterations to be performed in the algorithm.

In this work, two different algorithms were tested: first, a simple genetic algorithm was applied to obtain the sample points of the consensus set. The second approach adopts the metaheuristic called differential evolution to minimize the error between the corresponding points, while also maximizing the number of points used to calculate the homography. It is empirically shown through a series of experiments and real images, that it is advantageous to use the algorithms proposed in this thesis. Using the designed algorithms we were able to obtain a reduction of about six times, on average, the number of iterations performed, with respect to the original RANSAC algorithm.



# Agradecimientos

---

*A mis padres, Alberto Camacho Sánchez y Maria Minerva Vázquez Pérez por amarme y apoyarme incondicionalmente en cada una de mis decisiones.*

*A mi novia, Jazmin por ser siempre para mi 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.*

*Al Dr. Carlos Artemio Coello Coello y al Dr. Guillermo Morales Luna por todo el tiempo dedicado a la corrección de esta tesis, además de sus invaluable consejos para mejorarla.*

*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.*





# Índice general

---

<b>Índice de figuras</b>	<b>x</b>
<b>Índice de tablas</b>	<b>xii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Homografía . . . . .	4
1.3. Cálculo de una homografía mediante el algoritmo lineal . . . . .	4
1.4. Descripción del problema . . . . .	7
1.5. Objetivo de la tesis y resultados esperados . . . . .	8
1.6. Organización de la tesis . . . . .	8
<b>2. Descripción de la propuesta</b>	<b>11</b>
2.1. Algoritmo RANSAC . . . . .	11
2.2. Algoritmo GASAC . . . . .	12
2.3. Algoritmo basado en evolución diferencial . . . . .	15
2.4. Extracción de varias homografías . . . . .	17
<b>3. Resultados y discusión</b>	<b>25</b>
3.1. Imágenes de prueba . . . . .	25
3.1.1. Criterios por los cuales se escogieron los experimentos . . . . .	26
3.2. Resultados . . . . .	28
3.2.1. Ángulo de separación entre dos planos . . . . .	36
3.3. Análisis de los resultados obtenidos . . . . .	42
3.4. URL del trabajo . . . . .	42
3.4.1. Programas . . . . .	42
<b>4. Conclusiones</b>	<b>45</b>
4.1. Trabajo a futuro . . . . .	48

<b>Bibliografía</b>	<b>49</b>
<b>Páginas Web</b>	<b>53</b>

# Índice de figuras

---

1.1. Puntos típicos, atípicos y con ruido . . . . .	3
1.2. Relación entre dos imágenes determinada por una homografía $H$ . . . . .	4
1.3. Imágenes en las que el punto al infinito se sitúa en medio de las mismas . . . . .	7
2.1. Un ejemplo de una cadena binaria . . . . .	13
2.2. Uso de dos puntos de cruce entre dos individuos, en este caso se mantienen los genes de los extremos, y se intercambian los del centro . . . . .	14
2.3. Diagrama que muestra la secuencia general de operaciones para el algoritmo genético usado en este trabajo . . . . .	14
2.4. Diagrama general de la ED . . . . .	18
3.1. Las imágenes sintéticas con dos planos se muestran en (a) y en (b) se muestran las correspondencias de puntos de cada plano . . . . .	28
3.2. Las dos imágenes reales con tableros se muestran en (a) y en (b) se muestran las correspondencias de puntos seleccionados sobre ellos. En (c) se muestran los planos asociados a las dos homografías en las imágenes de los datos en (b) . . . . .	29
3.3. Las dos imágenes reales con triángulos se muestran en (a) y en (b) se muestran las correspondencias de puntos seleccionados sobre ellos. En (c) se muestran los planos asociados a las dos homografías en las imágenes de los datos en (b) . . . . .	30
3.4. Las imágenes sintéticas con dos planos y un ángulo de separación de $170^\circ$ , $160^\circ$ y $150^\circ$ grados entre ellos se muestran en (a), (b) y (c) . . . . .	31
3.5. Las imágenes sintéticas con dos planos y un ángulo de separación de $140^\circ$ , $130^\circ$ y $120^\circ$ grados entre ellos se muestran en (a), (b) y (c) . . . . .	32
3.6. Las imágenes sintéticas con dos planos y un ángulo de separación de $110^\circ$ , $100^\circ$ y $90^\circ$ grados entre ellos se muestran en (a), (b) y (c) . . . . .	33
3.7. Las imágenes sintéticas con dos planos y un ángulo de separación de $80^\circ$ y $70^\circ$ grados entre ellos se muestran en (a) y (b) . . . . .	34

- 3.8. Los resultados estadísticos promedio de 100 ejecuciones con los algoritmos propuestos RANSAC, GASAC y ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes en (a), (b) y (c) . . . . . 39
- 3.9. Los resultados estadísticos promedio de 100 ejecuciones con los algoritmos propuestos RANSAC, GASAC y ED, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes en (a), (b) y (c) 40

# Índice de tablas

---

3.1. Parámetros de entrada del algoritmo GASAC . . . . .	36
3.2. Parámetros de entrada de la ED . . . . .	36
3.3. Los resultados estadísticos de 100 ejecuciones con RANSAC, sobre el número de iteraciones en las tres series de imágenes . . . . .	36
3.4. Los resultados estadísticos de 100 ejecuciones con RANSAC, sobre la función de error en las tres series de imágenes . . . . .	37
3.5. Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre el número de iteraciones en las tres series de imágenes . . . . .	37
3.6. Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes	37
3.7. Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes . . . . .	38
3.8. Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de iteraciones en las tres series de imágenes . . . . .	38
3.9. Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes	38
3.10. Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes . . . . .	41
3.11. Los resultados estadísticos de 100 ejecuciones con RANSAC, GASAC y ED sobre el número de evaluaciones de la función objetivo en las tres series de imágenes, para tamaños de población de 10 y 5 en GASAC y ED, respectivamente	41

3.12. Los resultados estadísticos de 100 ejecuciones para determinar el ángulo de separación entre dos planos con RANSAC, GASAC y ED, sobre el número de evaluaciones de la función objetivo y la función de error, para tamaños de población de 10 y 5 en GASAC y ED, respectivamente . . . . . 44

# Capítulo 1

## Introducción

---

Este trabajo de tesis puede englobarse dentro del área de *reconocimiento de patrones*, específicamente en el reconocimiento de planos presentes en una o varias imágenes. Después de algunos antecedentes del tema de tesis, se describirá qué es una homografía y cómo calcularla, para posteriormente definir los objetivos de nuestro trabajo.

### 1.1. Antecedentes

Los algoritmos y técnicas de visión por computadora [1] cada vez tienen más auge en muchas de las etapas productivas de la industria, reconstrucción y reconocimiento de objetos o piezas, seguimiento y vigilancia, etc. Aquí destacan las aplicaciones que tratan de reconocer objetos, caras [2], caracteres caligráficos o huellas digitales.

Los algoritmos y técnicas de reconocimiento han llevado consigo avances en materia de extracción de características invariantes [3], segmentación de regiones, identificación de formas y mecanismos de clasificación.

Actualmente, y desde la aparición de los primeros sistemas de reconocimiento de objetos mediante visión por computadora, se busca identificar una entidad geométrica tratando de encontrar un conjunto de características correspondientes a ésta en la imagen capturada. El problema en este caso, es que el punto de vista de la cámara con el que se ha capturado la imagen delimita el éxito del proceso de reconocimiento, puesto que la visibilidad del objeto viene determinada por la perspectiva con la que se observa al mismo.

También es importante mencionar que otro de los problemas en los que los sistemas de reconocimiento han hecho especial hincapié es la búsqueda de características invariantes que no dependan de la perspectiva con la que se observe el objeto. De hecho, algunas de las mejoras en los procesos de reconocimiento se deben a la incorporación de características invariantes

a la escala, orientación, iluminación y pequeños cambios de perspectiva.

Dada la naturaleza compleja del problema, la detección robusta de entidades geométricas se puede llevara cabo mediante un enfoque basado en la corrección del movimiento de los objetos, el cual necesita partir de al menos dos imágenes capturadas por una cámara digital (en el caso del seguimiento visual) o generadas por computadora (en el caso de simulaciones), una de las cuales es obtenida desde una posición inicial y la otra se encuentra en una posición deseada o esperada. Se necesita extraer puntos de interés [4] en las dos o más imágenes obtenidas y aplicar un proceso de correlación entre éstos. Debido a la necesidad de comparar imágenes, debe existir una cierta dependencia entre éstas, la cual está dada por una secuencia de las mismas, lo que genera vistas similares.

A partir de las correspondencias obtenidas en las imágenes, se debe obtener una relación de éstas, la cual viene determinada por una homografía [5]. La homografía permite mapear los puntos de interés en un plano de la imagen o como se observa ese plano en la otra imagen. Aunque se puede calcular la homografía con al menos cuatro correspondencias de puntos, normalmente en las tareas de seguimiento visual [4] se cuenta con un conjunto de correspondencias más amplio.

Cuando se desea diseñar algoritmos en los que todos los pasos, desde el proceso de correspondencia hasta el cálculo de homografías, se obtengan de modo automático, es posible que aparezcan valores espurios o no deseados, conocidos como puntos atípicos [6]. Dichos puntos añaden el problema, de que al intentar solucionar un sistemas de ecuaciones que los incluyan, se obtendrán soluciones erróneas.

Un método de estimación robusto de homografías debe tener en cuenta la presencia de datos atípicos, así como un mecanismo para identificarlos y eliminarlos a fin de que no se consideren en el cálculo o estimación.

Una estimación consiste en encontrar una solución aproximada de un modelo matemático a partir de un conjunto de datos que contienen puntos típicos, atípicos y con ruido, y además obtener una forma de establecer cuál de todas las posibles soluciones es la mejor. Generalmente, cuando se extrae un conjunto de puntos de interés correspondientes a dos o más imágenes y se desea estimar homografías de forma automática, la única medida para establecer cuál de las posibles soluciones obtenidas es la mejor, es el error obtenido en la posición al extraer los puntos de interés en cada imagen.

Sin duda el algoritmol más conocido para resolver este problema es RANSAC (Random Sample Consensus) [7], una heurística propuesta y validada por Fischler y Rolles en 1981, para estimar los parámetros de un modelo matemático a partir de un conjunto de datos que contienen puntos típicos, atípicos y con ruido.



La principal desventaja de este algoritmo radica en la búsqueda de los parámetros del modelo matemático mediante el uso de un procedimiento aleatorio; el tiempo de ejecución que requiere para encontrar resultados aceptables puede ser considerable.

Para comprender mejor el objeto de la tesis, es importante realizar una clara distinción entre los puntos típicos, atípicos y con ruido. Por ejemplo si tomamos los puntos:

$$P_1 = (x_1, y_1) \text{ y } P_2 = (x_2, y_2),$$

del conjunto de datos de la figura 1.1 y realizamos un ajuste mediante una recta:

$$y - y_1 = \left( \frac{y_2 - y_1}{x_2 - x_1} \right) (x - x_1),$$

los puntos típicos son datos cuya distribución se explican por el conjunto de parámetros del modelo. Los puntos atípicos son datos que no encajan en el modelo e implican un patrón que es inconsistente con el conjunto de puntos típicos. Finalmente, el ruido son datos que pueden aparecer a menudo como error de medición y pueden pasar desapercibidos.

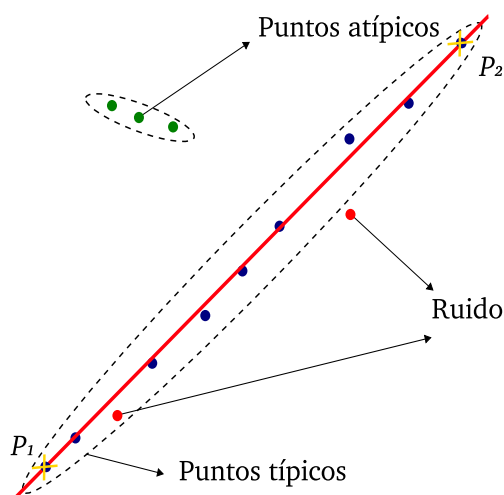


Figura 1.1: Puntos típicos, atípicos y con ruido

Entendiendo los conceptos de puntos típicos, atípicos y con ruido, es como podemos determinar que la estimación automática de homografías implica un problema de optimización debido a la presencia de puntos atípicos y con ruido en el conjunto de datos de entrada. Muchas de las soluciones implementadas actualmente optan por utilizar el algoritmo RAN-SAC, pero como ya se mencionó anteriormente, encontrar resultados aceptables mediante este algoritmo puede requerir tiempos de ejecución enormes.

En virtud de lo anterior, el objetivo final de este trabajo de tesis consiste en demostrar que se puede realizar la estimación automática de homografías, de forma fiable y eficiente

aún con la presencia de puntos atípicos y con ruido, utilizando un algoritmo genético [8] (al cual nos referiremos como AG) y la metaheurística llamada evolución diferencial [9] (a la cual nos referiremos como ED). Lo que se busca es, que al reemplazar una búsqueda meramente aleatoria por una metaheurística, se tengan métodos más robustos que RANSAC.

## 1.2. Homografía

La relación entre planos, en una o dos imágenes, viene determinada por una homografía  $H$  [5]. Ésta permite mapear características de un plano a otro, de modo que almacena el movimiento proyectivo de éstos como se muestra en la figura 1.2.

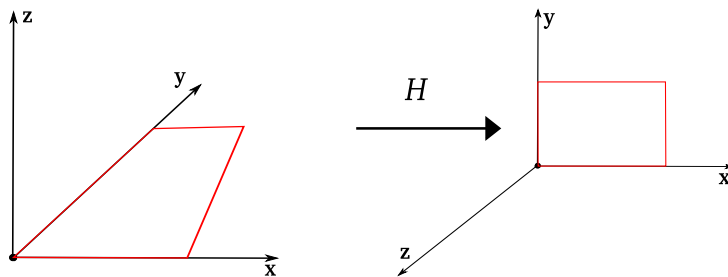


Figura 1.2: Relación entre dos imágenes determinada por una homografía  $H$

El enunciado del problema es el siguiente: Dadas  $n \geq 4$  correspondencias de puntos en cada imagen  $\{p_i \leftrightarrow p'_i\}$ , se busca determinar la homografía  $H \in \mathbb{R}^{3 \times 3}$  tal que  $\lambda p'_i = H p_i$ , donde  $\lambda$  es un escalar cualquiera diferente de cero, con  $i \in \{1, 2, \dots, n\}$ .

Son necesarias, al menos, cuatro correspondencias de puntos para calcular la matriz  $H$ , debido a que la matriz  $H$  tiene nueve elementos, pero, a falta de un factor de proporcionalidad, tiene  $9 - 1$  grados de libertad. Por otro lado, cada pareja de puntos proporciona dos restricciones sobre cómo se transforman dos de sus coordenadas.

La solución  $H$  es única si sólo se proporcionan cuatro correspondencias de puntos. Con más de cuatro puntos se tiene un sistema sobredeterminado de ecuaciones que se resuelve por mínimos cuadrados.

## 1.3. Cálculo de una homografía mediante el algoritmo lineal

Una vez que se obtienen las correspondencias de puntos que definen el área de un plano en la primera imagen  $p_i = [u_i, v_i, 1]^T$  y en la segunda imagen  $p'_i = [x_i, y_i, 1]^T$ , para  $i \in$

$\{1, 2, \dots, n\}$ , se les debe aplicar una normalización con media cero y desviación estándar uno, con lo que se obtiene:

$$T_1 = \begin{bmatrix} \frac{1}{\sigma_u} & 0 & -\frac{\bar{u}}{\sigma_u} \\ 0 & \frac{1}{\sigma_v} & -\frac{\bar{v}}{\sigma_v} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad T_2 = \begin{bmatrix} \frac{1}{\sigma_x} & 0 & -\frac{\bar{x}}{\sigma_x} \\ 0 & \frac{1}{\sigma_y} & -\frac{\bar{y}}{\sigma_y} \\ 0 & 0 & 1 \end{bmatrix},$$

donde  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{u}$  y  $\bar{v}$ , son los promedios y  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_u$  y  $\sigma_v$ , las desviaciones estándares de todas las coordenadas  $x_i$ ,  $y_i$ ,  $u_i$  y  $v_i$  de los puntos. Entonces, dada la expresión:

$$\lambda p_i = H p'_i,$$

donde  $H$  es una matriz en  $\mathbb{R}^{3 \times 3}$  y  $\lambda$  es un escalar cualquiera diferente de cero, se aplica la normalización como transformación mediante la matriz  $T \in \mathbb{R}^{3 \times 3}$ :

$$\lambda T_1 p_i = H' T_2 p'_i. \quad (1.1)$$

Si  $q_i = T_1 p_i$  y  $q'_i = T_2 p'_i$ , esto es,  $q_i$  y  $q'_i$  son los puntos transformados correspondientes a los puntos  $p_i$  y  $p'_i$ , tenemos la relación:

$$\lambda q_i = H' q'_i, \quad (1.2)$$

si expandimos la ecuación anterior de la forma:

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & h'_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

podemos escribir las tres ecuaciones que resultan como:

$$\begin{aligned} \lambda u_i &= h'_{11} x_i + h'_{12} y_i + h'_{13} \\ \lambda v_i &= h'_{21} x_i + h'_{22} y_i + h'_{23} \\ \lambda &= h'_{31} x_i + h'_{32} y_i + h'_{33}, \end{aligned}$$

donde:

$$u_i = \frac{h'_{11} x_i + h'_{12} y_i + h'_{13}}{h'_{31} x_i + h'_{32} y_i + h'_{33}} \quad \text{y} \quad v_i = \frac{h'_{21} x_i + h'_{22} y_i + h'_{23}}{h'_{31} x_i + h'_{32} y_i + h'_{33}},$$

y si definimos  $h$  como un vector formado por los primeros ocho elementos de  $H'$ , podemos establecer un sistema de ecuaciones de la forma:

$$Ah = b,$$

imponiendo  $h'_{33} = 1$  (la imposición de la condición se justifica ya que el vector solución  $h$  está definido, salvo un factor de escala),  $A \in \mathbb{R}^{2n \times 8}$  y  $b$  es un vector de  $2n$  elementos. Tal ecuación puede resolverse utilizando técnicas estándar para resolver ecuaciones lineales (tales como la eliminación de Gauss) en el caso en el que  $A$  contiene sólo 8 filas (el caso mínimo), o mediante técnicas de mínimos cuadrados en el caso de un sistema de ecuaciones sobredeterminado, de tal manera que:

$$Ah = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2x_2 & -v_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -u_nx_n & -u_ny_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -v_nx_n & -v_ny_n \end{bmatrix} \begin{bmatrix} h'_{11} \\ h'_{12} \\ h'_{13} \\ h'_{21} \\ h'_{22} \\ h'_{23} \\ h'_{31} \\ h'_{32} \end{bmatrix} = b = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{bmatrix},$$

con:

$$h = A^{-1}b,$$

donde para el cálculo de  $A^{-1}$ , se utiliza la descomposición en valores singulares (DVS).

La DVS es una de las descomposiciones más útiles de una matriz, la cual, al aplicarse en  $A$  nos da:

$$A = UDV^T.$$

Dado que  $U$  y  $V$  son matrices ortogonales y  $D$  es una matriz diagonal con entradas no negativas, entonces:

$$A^{-1} = (UDV^T)^{-1} = (V^T)^{-1} D^{-1} U^{-1},$$

pero como  $U$  y  $V$  son ortogonales, con lo que  $U^{-1} = U^T$ , se tiene:

$$A^{-1} = VD^{-1}U^T,$$

donde  $D^{-1}$  se calcula como:

$$D^{-1} = \begin{bmatrix} \frac{1}{d_{11}} & 0 & 0 \\ 0 & \frac{1}{d_{22}} & 0 \\ 0 & 0 & \frac{1}{d_{33}} \end{bmatrix}.$$

Por último se obtiene  $H$  como se muestra a continuación:

$$H = T_1^{-1}H'T_2 = T_1^{-1} \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & 1 \end{bmatrix} T_2 \quad (1.3)$$

Debido a la imposición de  $h'_{33} = 1$ , si la verdadera solución tiene  $h'_{33} = 0$ , entonces no hay un  $\lambda$ , tal que  $\lambda h'_{33} = 1$ . Esto significa que la verdadera solución no se podrá encontrar. Por esta razón, se producirán resultados inestables en el caso en el que el verdadero valor de  $h'_{33}$  esté muy cercano a cero.

$h'_{33}$  puede ser cero si el origen de coordenadas se asigna a un punto en el infinito de  $H$ . Puesto que  $[0, 0, 1]^T$ , representa el origen de coordenadas  $x_0$  y también la recta al infinito  $l$ , esta condición se puede escribir como:

$$l^T H x_0 = [0, 0, 1] H [0, 0, 1]^T = 0,$$

por lo tanto  $h'_{33} = 0$ .

Es posible que en algunas imágenes, la recta al infinito se sitúa en medio de las mismas y por tanto se tiene una alta probabilidad de que el origen de coordenadas caiga sobre ella.

Este tipo de imágenes no se tratan en esta tesis. Un ejemplo de este tipo de imágenes se muestra en la figura 1.3 (tomadas de la página web [2], en la pág. 53). Nosotros vamos a tratar imágenes de planos y un punto del plano no puede yacer en el punto al infinito.



Figura 1.3: Imágenes en las que el punto al infinito se sitúa en medio de las mismas

Este es el algoritmo lineal para el cálculo de homografías [1].

## 1.4. Descripción del problema

Dado un conjunto de correspondencias de puntos en un sistema de coordenadas globales  $xy$ :

$$P_1 = \{p_{11}, p_{12}, \dots, p_{1n}\} \text{ y } P_2 = \{p_{21}, p_{22}, \dots, p_{2n}\},$$

que contienen puntos típicos, atípicos y con ruido, se buscarán las homografías  $H$ , que mejor representen al conjunto de puntos típicos.

Se desea realizar el ajuste de las homografías  $H$ , para los siguientes casos:

- Dos imágenes, con correspondencias de un plano.
- Dos imágenes, con correspondencias de varios planos.
- Muchas imágenes, con correspondencias de varios planos.

La única forma de determinar cuáles de las homografías  $H$  representan mejor al conjunto de puntos típicos, es usando la función de costo que minimiza la suma de los errores obtenidos  $e_i$  mediante la distancia euclidiana  $d(\cdot, \cdot)$  entre las correspondencias  $p_{1i}$  y  $p_{2i}$  con las estimadas mediante  $H$  y  $H^{-1}$ ,  $\hat{p}_{1i}$  y  $\hat{p}_{2i}$  respectivamente, para  $i \in \{1, 2, \dots, n\}$  como se describe en la ecuación (1.4).

$$e_i^2 = d(p_{1i}, \hat{p}_{1i})^2 + d(p_{2i}, \hat{p}_{2i})^2, \quad (1.4)$$

donde  $\hat{p}_{1i} = Hp_{1i}$  y  $\hat{p}_{2i} = H^{-1}p_{2i}$ .

## 1.5. Objetivo de la tesis y resultados esperados

El objetivo de la tesis es realizar un método de estimación robusto por medio de la minimización de la suma de las distancias euclidianas entre las correspondencias de puntos como se muestra en la ecuación (1.4).

El método debe ser capaz de extraer varias homografías presentes en un mismo conjunto de correspondencias de puntos.

El problema de estimación será no lineal y se plantea resolverlo utilizando un AG simple para elegir el conjunto de correspondencias de consenso, en lugar de utilizar muestras al azar como se realiza en RANSAC. Además, se propone utilizar ED para resolver el problema, buscando minimizar la distancia entre las correspondencias de puntos como se mencionó anteriormente.

Se compararán los resultados obtenidos contra el algoritmo RANSAC con la finalidad de demostrar mediante experimentos con datos reales y simulados, la ventaja de utilizar un AG y ED en este problema.

## 1.6. Organización de la tesis

El capítulo 2 comprende los algoritmos de estimación robusta de homografías implementados en esta tesis. En dicho capítulo también se proporcionan detalles sobre dichas implementaciones.

También se detalla la propuesta del problema de optimización que se resuelve para obtener los parámetros de cada algoritmo, así como la representación que se le dio al problema de estimación robusta de homografías para ocupar un AG y ED.

Finalmente, se describe la implementación del AG y de la ED para la estimación de homografías y se definen las condiciones necesarias para comparar los nuevos métodos propuestos con RANSAC.

En el capítulo 3 se encuentran los experimentos y los resultados de la detección automática de homografías de los algoritmos que se presentan en el capítulo 2, así como las ventajas y desventajas de cada uno de ellos, con lo que se incita al lector a discutir acerca de los resultados obtenidos.

El capítulo 4 contiene las conclusiones generales de este trabajo de tesis y el trabajo futuro.





# Capítulo 2

## Descripción de la propuesta

---

En este capítulo, a partir de un conjunto de correspondencias  $P_1$  y  $P_2$  que contienen puntos típicos, atípicos y con ruido, se estima de forma robusta la matriz de homografía  $H$  consistente con los puntos típicos. Se utiliza el error obtenido entre las correspondencias de puntos para determinar cuál de las posibles soluciones es la mejor.

Se presentan las soluciones propuestas utilizando los algoritmos RANSAC, un algoritmo genético (AG) simple y evolución diferencial (ED), así como los algoritmos necesarios para extraer varias homografías utilizando dichas soluciones.

### 2.1. Algoritmo RANSAC

En la presente sección utilizamos la heurística RANSAC descrita en [7], la cual fue propuesta y validada por Fischler y Rolles en 1981. Dicha heurística ha sido utilizada en los trabajos [8, 10, 11] para la detección automática de varios círculos o elipses presentes en un conjunto de datos que contienen puntos típicos, atípicos y con ruido. En dichos trabajos se ha demostrado la ventaja de utilizar un algoritmo genético simple y la metaheurística llamada evolución diferencial con respecto al algoritmo RANSAC para resolver dicho problema.

En el caso de la detección automática de homografías, sin duda el algoritmo RANSAC más citado para resolver este problema se encuentra en [1]. Utilizando los conceptos teóricos de dicha solución se obtuvo el Algoritmo 1, el cual consiste en la estimación de la homografía  $H$  y un conjunto de correspondencias de puntos consistentes (los puntos típicos), junto con las correspondencias que no cumplen con el modelo (los puntos atípicos).

El umbral  $U$ , que indica si un punto es o no típico, y se suele determinar empíricamente, aunque es posible analizarlo desde el punto de vista estadístico [1]: se desea elegir dicho umbral, tal que con una probabilidad  $\alpha$  un punto es típico (normalmente se elige  $\alpha = 0.95$ ),

entonces:

$$U^2 = F_2^{-1}(\alpha) \sigma^2,$$

siendo  $F$  una distribución chi-cuadrado con dos grados de libertad (esto significa que hay dos valores por cada coordenada) y suponiendo que el error de medición es gaussiano con media cero y desviación estándar  $\sigma$ , entonces:

$$F_2^{-1}(0.95) = 5.9915,$$

por lo tanto, se puede establecer:

$$U = \sqrt{5.9915} \sigma.$$

Los casos en los que no se puede calcular  $H$  en la línea 5 del Algoritmo 1 (quizá porque se obtuvo un conjunto de correspondencias colineales), no deben ser considerados en el procedimiento y por tal motivo se debe regresar a la instrucción indicada en la línea 4. En cuestión de análisis estos casos se deben sumar al número final de iteraciones realizadas.

Los errores de la línea 6 en el Algoritmo 1, se calculan como se describió en la ecuación (1.4) (en la pág. 8) y el número de iteraciones en la línea 1, se calcula utilizando la misma expresión de la línea 13:  $\epsilon_0$  es la probabilidad de que RANSAC en alguna iteración seleccione sólo cuatro puntos típicos;  $\epsilon_0$  se establece a una probabilidad alta de 0.99;  $\epsilon_1$  es la probabilidad de elegir un punto típico cada vez que se selecciona un elemento y se calcula como en la línea 12.  $\epsilon^4$  es la probabilidad de que todos los cuatro puntos seleccionados sean típicos, por lo tanto:

$$1 - \epsilon^4,$$

es la probabilidad de que al menos uno de los cuatro puntos sea un punto atípico. Como:

$$1 - \epsilon_0 = (1 - \epsilon^4)^N,$$

donde  $N$  es el número de iteraciones, se obtiene la expresión de la línea 13.

Por último, en la línea 4 se puede observar la principal desventaja del Algoritmo 1, debido a la búsqueda de las correspondencias de puntos mediante el uso de un procedimiento al azar. Dicha desventaja da pie a utilizar un AG simple y ED para encontrar resultados aceptables en un tiempo de ejecución menor al que utiliza RANSAC para resolver el problema.

## 2.2. Algoritmo GASAC

En el algoritmo RANSAC se tiene un límite superior del número de iteraciones y siempre es una estimación porque no conocemos de antemano el número de puntos típicos, atípicos y con ruido.

En lugar de utilizar el enfoque probabilista para calcular el número de iteraciones, se puede utilizar una técnica heurística estocástica llamada algoritmo genético (AG), la cual se describe en [12, 13, 14].

El primer AG [15] (originalmente llamado plan reproductivo) fue desarrollado por John Holland en 1962, como un medio para estudiar el comportamiento de adaptación y aplicarlo al aprendizaje de máquina. Sin embargo, este tipo de algoritmos han sido ampliamente utilizados como métodos de optimización.

La representación tradicional usada por un AG para codificar un conjunto de soluciones, es el esquema binario, en el cual un cromosoma es una cadena binaria, como se muestra en la figura 2.1, donde los ceros y unos se denominan alelos.

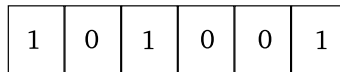


Figura 2.1: Un ejemplo de una cadena binaria

Nosotros decidimos utilizar la representación binaria para codificar los índices de cuatro correspondencias de puntos del conjunto de consenso, porque tomando en cuenta que en el trabajo [4] (donde se comparan diversos algoritmos en tareas de seguimiento visual) el máximo número de correspondencias de puntos es de 1000, entonces para este caso se tendrían los índices del 0 al 999, con los cuales necesitaríamos para su representación binaria al menos 10 bits, con lo que un cromosoma con cuatro índices estaría compuesto por 40 bits.

La codificación del Algoritmo 2 se observa de manera general en la figura 2.3. Primero, se genera aleatoriamente la población inicial, que estará compuesta por un conjunto de cromosomas que contienen los índices en binario a cuatro puntos.

A cada uno de los cromosomas de esta población se le aplicará la función de aptitud *Evaluar al nuevo individuo* del Algoritmo 3, a fin de saber qué tan buena es la solución y reparar el cromosoma si uno de los índices es mayor que el número de correspondencias de puntos: un nuevo índice se recalcula al azar dentro del rango válido.

Sabiendo la aptitud de cada cromosoma, se procede a la selección de los que se cruzarán en la siguiente generación, para lo cual utilizamos la selección mediante el torneo binario [16]: al azar se seleccionan dos soluciones y se toma de ellas la mejor.

Una vez realizada la selección, se procede con la cruce de los individuos seleccionados, utilizando la cruce de dos puntos: se seleccionan dos posiciones del cromosoma al azar y su material cromosómico es intercambiado, como se muestra en la figura 2.2.

Debido a que usamos representación binaria, la mutación cambia de 0 a 1 y viceversa,

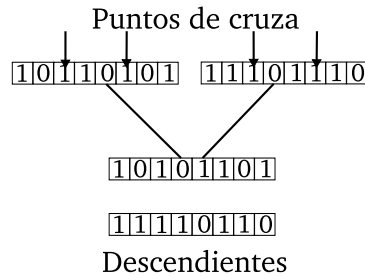


Figura 2.2: Uso de dos puntos de cruce entre dos individuos, en este caso se mantienen los genes de los extremos, y se intercambian los del centro

mientras que el elitismo mantiene intacto al individuo más apto a lo largo de las distintas generaciones (dicho individuo no se cruza con nadie), hasta que surge otro individuo mejor que él, que lo desplazará.

Detener el AG no es trivial (debido a que no conocemos de antemano la respuesta a la que queremos llegar), éste se detiene cuando la población se haya estabilizado (en dos iteraciones sucesivas el mejor resultado es el mismo) o cuando se realiza el número máximo de generaciones.

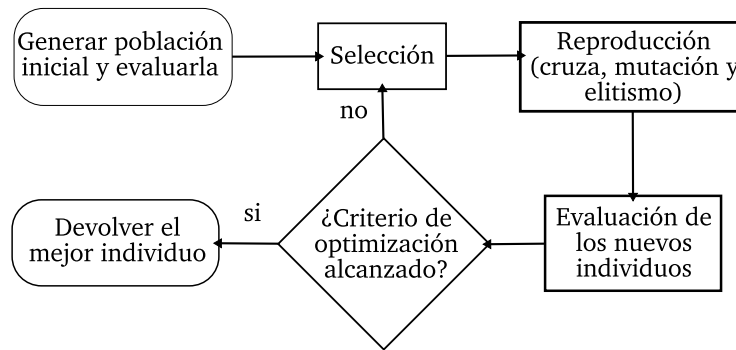


Figura 2.3: Diagrama que muestra la secuencia general de operaciones para el algoritmo genético usado en este trabajo

Como hemos podido observar, la principal ventaja de este AG radica en su sencillez. Se requiere poca información sobre el espacio de búsqueda, ya que se trabaja sobre un conjunto de individuos y se busca una solución por aproximación de la población, en lugar de utilizar un enfoque probabilístico como en RANSAC. Con un control adecuado podemos mejorar la aptitud promedio de la población, obteniendo nuevos y mejores individuos y por lo tanto, mejores soluciones.

Quizás el punto más delicado del Algoritmo 2 se encuentra en la definición de la función *Evaluar al nuevo individuo*, ya que de su eficacia depende el obtener buenos resultados.

Desde luego, en el procedimiento descrito, quedan muchos detalles importantes, como cuál ha de ser el tamaño de la población y cuáles las probabilidades de cruce ( $p_c$ ) y de mutación ( $p_m$ ), ya que de estos detalles dependerá, en gran parte, el éxito o fracaso del algoritmo. Los parámetros utilizados en el AG se detallarán en el capítulo 3.

Por último, decidimos llamar a nuestro algoritmo propuesto GASAC (Genetic Algorithm SAmple Consensus), porque aunque se trata de un AG, éste utiliza un grupo de consenso en la evaluación de la función objetivo.

## 2.3. Algoritmo basado en evolución diferencial

La metaheurística llamada evolución diferencial (ED) fue propuesta por Rainer Storn y Kenneth Price en 1995 [17], con una codificación de punto flotante para la optimización global sobre espacios continuos.

En los trabajos [10, 18, 19, 20], se puede encontrar una descripción detallada de la implementación de la ED para resolver diferentes problemas, como, por ejemplo; la detección automática de varios círculos o elipses presentes en un conjunto de datos que contienen puntos típicos, atípicos y con ruido.

Un diagrama del Algoritmo 4 se observa en la figura 2.4 (muestra de manera general los procesos de mutación, cruce y selección descritos más adelante). Este procedimiento fue diseñado para la optimización de los ocho parámetros de  $H'$  de la ecuación (1.3):  $H = T_1^{-1}H'T_2$ . La principal idea es emplear la diferencia entre los valores de cada individuo para generar nuevos individuos. Se maneja un conjunto de individuos, los cuales contienen una posible solución del problema. Este conjunto de individuos, conocido como población  $G$ , se representa por:

$$h_k^r \in G^r, \text{ para } k \in \{1, 2, \dots, t_G\},$$

donde  $h$  es un individuo de  $G$  compuesto por los ocho elementos de  $H'$ ,  $r$  es la generación actual y  $t_G$  es el tamaño de la población  $G$ .

Para inicializar cada individuo en la generación se utilizan los límites superiores  $l_s$  e inferiores  $l_i$  de cada variable, los cuales se determinan de la siguiente forma: Primero se deben obtener las matrices  $T_1$  y  $T_2$  utilizando todas las correspondencias de  $P_1$  y  $P_2$ , como se explica en la sección 1.3 del capítulo 1 (en la pág. 4). Segundo, se deben calcular mediante el algoritmo lineal (en la misma sección 1.3) las matrices  $H_j$  de cada cuatro correspondencias de puntos:

$$C_1 = \{c_1, c_2, c_3, c_4\}, C_2 = \{c_2, c_3, c_4, c_5\}, \dots \text{ y } C_j = \{c_j, c_{j+1}, c_{j+2}, c_{j+3}\},$$

donde  $c_j = \{p_{1j}, p_{2j}\}$ , para  $0 < j \leq n - 3$ . Las coordenadas de cada subconjunto  $C_j$  se normalizan mediante  $H'_j = T_1 H_j T_2^{-1}$ , con  $T_1$  y  $T_2$  obtenidas del primer paso. Entonces los límites son:

$$l_{s_b} = \bar{h}_{j_b} + 2\sigma_{h_{j_b}} \text{ y } l_{i_b} = \bar{h}_{j_b} - 2\sigma_{h_{j_b}},$$

con  $\bar{h}_{j_b}$  y  $\sigma_{h_{j_b}}$  como la media y desviación estándar de cada uno de los  $b \in \{1, 2, \dots, 8\}$  elementos de  $H'_j$ , normalizados con  $h_{j_{33}} = 1$ .

Ahora podemos inicializar los primeros  $j$  individuos con  $H'_j$  y el resto de la población al azar mediante:

$$h_{kb}^0 = \text{rand}()_{kb} \cdot (l_{s_b} - l_{i_b}) + l_{i_b},$$

donde  $\text{rand}()$  es una función que genera valores al azar con una distribución uniforme en el intervalo de  $[0, 1)$ .

Posteriormente, la población se actualiza por medio de un esquema de reproducción en cada generación, en el cual la mutación y cruza se utilizan para generar nuevos individuos (vectores de prueba). En este procedimiento se toma de la población, un conjunto de individuos extraídos al azar  $\phi = \xi_1, \xi_2$  y  $\xi_3$  (necesarios para efectuar la diferencia) por cada individuo  $h_k^r$  (vector objetivo) y como resultado se obtiene un nuevo individuo  $p_k^r$ , mediante:

$$p_{kb}^r = \begin{cases} \xi_{3b} + F_d (\xi_{1b} - \xi_{2b}), & \text{si } \text{rand}() < C_r \text{ o } b = b_{rand}, \\ h_{kb}^r, & \text{en otro caso,} \end{cases}$$

donde  $F_d \in [0, 2)$  es la constante de diferencia,  $C_r \in [0, 1)$  es la constante de recombinación y  $b_{rand}$  se calcula con la función  $\text{int}(8 * \text{rand}())$  que genera números enteros al azar con una distribución uniforme en el intervalo de  $[0, 8)$ .

Después la selección usa un mecanismo en el cual si la aptitud obtenida del nuevo individuo  $g(p_k^r)$  (descrita en el Algoritmo 5, la cual utiliza las matrices  $T_1$  y  $T_2$  calculadas anteriormente a partir de todas las correspondencias de puntos de  $P_1$  y  $P_2$ ), es mejor a la aptitud del vector objetivo  $g(h_k^r)$ , entonces  $p_k^r$  se agrega en la población de la siguiente generación ( $h_k^{r+1} = p_k^r$ ); en caso contrario,  $h_k^r$  será quien sobreviva a la próxima generación ( $h_k^{r+1} = h_k^r$ ).

Para obtener la aptitud de un individuo  $h_k^r$  como se muestra en la línea 8 del Algoritmo 5, se busca maximizar el número de puntos típicos consistentes con la matriz de homografía  $H$  de la línea 3 y minimizar el error obtenido entre dichas correspondencias de puntos. Ambas restricciones se pueden combinar en un problema de minimización utilizando la función de error:

$$f(x) = n - m + \sum_1^m e_i^2, \quad (2.1)$$

donde  $n$  es el número total de correspondencias de puntos de  $P_1$  y  $P_2$ ,  $m$  es el número de puntos típicos consistentes con la matriz de homografía  $H$  y  $e_i^2$  es el error obtenido de cada una de las  $m$  correspondencias de puntos típicos con índice  $i$ , mediante la ecuación (1.4) (en la pág. 8).

Por último, además del número total de generaciones MAX\_G, para detener la ED se utilizó la condición de paro llamada *diff* en [21] (utilizada de forma exitosa en los trabajos [10, 18, 19]), la cual consta de buscar la aptitud del mejor y peor individuo de la generación actual. Una vez efectuado lo anterior, realiza la diferencia entre la peor menos la mejor aptitud y si ésta es menor que  $C_p$  (la constante para la condición de paro), se detiene la ejecución del algoritmo.

La principal ventaja del presente procedimiento es su sencillez para ser implementado y su baja complejidad en relación con otros algoritmos evolutivos, además de que el número de parámetros empleados es bajo, por lo cual, la puesta a punto es sencilla como se detallará en el capítulo 3.

## 2.4. Extracción de varias homografías

Los algoritmos presentados en las secciones 2.1, 2.2 y 2.3 del presente capítulo (RANSAC, GASAC y ED), pueden ser utilizados para detectar y extraer varias homografías de un conjunto de correspondencias dadas  $P_1$  y  $P_2$ . En la ED se utiliza un procedimiento diferente al resto de las soluciones propuestas, debido a la inicialización de los límites  $l_s$  y  $l_i$ .

La extracción de varias homografías puede verse como el problema de obtener las matrices de homografía  $H$ , presentes en el conjunto de correspondencias  $P_1$  y  $P_2$ , consistentes con los puntos de los planos que se encuentran en las imágenes de entrada.

Una vez que una homografía es detectada, los puntos consistentes con dicho ajuste son eliminados del conjunto de correspondencias de entrada y se reanuda la búsqueda de otra homografía.

El algoritmo propuesto para la extracción de varias homografías utilizando RANSAC y GASAC, se muestra en el Algoritmo 6.

El Algoritmo 7, muestra la solución propuesta utilizando la ED. Para inicializar los límites  $l_s$  y  $l_i$ , se debe obtener un conjunto de correspondencias de puntos de los datos de entrada.

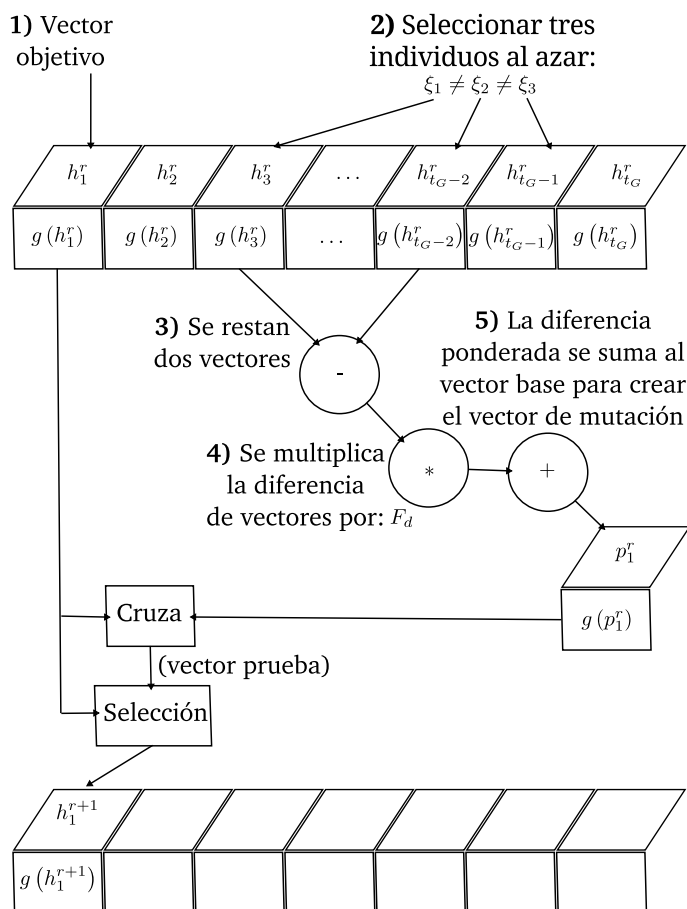


Figura 2.4: Diagrama general de la ED

Para encontrar dichas correspondencias se utiliza la primera imagen, en la cual mediante un punto al azar y un radio  $r$  (el ancho de la misma), se genera un círculo. Se usan los puntos que están dentro del círculo para calcular la homografía.

Para determinar si un punto se encuentra dentro del círculo, basta comprobar si la distancia desde el centro del círculo al punto en cuestión es menor al radio del círculo.

A partir del conjunto de puntos que se encuentran dentro del círculo, se obtienen las correspondencias con las cuales se inicializa la ED.



---

**Algoritmo 1** Algoritmo RANSAC para detectar una homografía.

---

**Entrada:** Conjunto de correspondencias  $P_1$  y  $P_2$ ; umbral  $U$ .

**Salida:** La homografía presente en los datos de entrada.

```

1:  $N :=$  Calcular el número de iteraciones.
2:  $m_0 = 0$ 
3: while  $i < N$  do
4:   Seleccionar una muestra al azar  $Q$ , de 4 parejas de puntos del conjunto de datos  $P_1$  y  $P_2$ .
5:   if  $H$  se puede calcular a partir de  $Q$  then
6:     Calcular el conjunto de errores  $E = \{e_1, e_2, e_3, \dots, e_n\}$  para  $P_1$  y  $P_2$  usando  $H$ .
7:     Obtener el conjunto de pesos  $W$  como  $w_i = \begin{cases} 1, & \text{si } e_i \leq U, \\ 0, & \text{si } e_i > U. \end{cases}$ 
8:      $m :=$  El número de puntos típicos para los cuales  $w_i = 1$ .
9:     if  $m > m_0$  then
10:       $Q_H := W$ 
11:    end if
12:    Calcular  $\epsilon_1 = m/n$ 
13:    Calcular  $N_a = \frac{\log(1-\epsilon_0)}{\log[1-\epsilon_1^4]}$ 
14:    if  $N_a > N$  then
15:       $N := N_a$ 
16:    end if
17:  end if
18:   $i ++$ 
19:  Re-estimar  $H$ , utilizando todos los puntos típicos en  $Q_H$ .
20: end while

```

---

---

**Algoritmo 2** AG para detectar una homografía.

---

**Entrada:**  $N$ , número de iteraciones;  $Pop$ , tamaño de la población.

**Entrada:** Probabilidad de cruce  $p_c$  y probabilidad de mutación  $p_m$ .

**Entrada:** Dos conjuntos de puntos  $P_1$  y  $P_2$ , cada uno de ellos con  $n$  elementos; umbral  $U$ .

**Salida:** La homografía presente en los datos de entrada.

- 1: Número de bits que representan los índices,  $b := \lceil \log_2 n \rceil$ .
- 2: Tamaño de cromosoma :=  $4 * b$ .
- 3:  $i := 0$ ;  $m_0 = 0$
- 4: Inicializar la población al azar.
- 5: **while**  $i < N$  **do**
- 6:   **for**  $0 \leq j < n$  **do**
- 7:     dosindividuos := Seleccione dos individuos por torneo.
- 8:     **if**  $rand() \leq p_c$  **then**
- 9:       Aplicar cruce a los individuos.
- 10:     **end if**
- 11:     **if**  $rand() \leq p_m$  **then**
- 12:       Aplicar la mutación a los dos individuos.
- 13:     **end if**
- 14:      $m_1 :=$  Evaluar al nuevo individuo 1.
- 15:      $m_2 :=$  Evaluar al nuevo individuo 2.
- 16:     **if**  $m_1 > m_0$  **then**
- 17:        $k :=$  El individuo 1 representa la solución tentativa.
- 18:        $m_0 := m_1$
- 19:     **end if**
- 20:     **if**  $m_2 > m_0$  **then**
- 21:        $k :=$  El individuo 2 representa la solución tentativa.
- 22:        $m_0 := m_2$
- 23:     **end if**
- 24:     Aplicar elitismo: se retiene intacto el mejor individuo.
- 25:     **if** En dos iteraciones sucesivas el mejor resultado es el mismo **then**
- 26:       Detener.
- 27:     **end if**
- 28:   **end for**
- 29:    $i := i + 2$
- 30: **end while**
- 31: Re-estimar  $H$  utilizando todos los puntos típicos representados por el individuo  $k$ .
- 32: Re-calcular el número de puntos típicos con  $H$  re-estimada.

---

---

**Algoritmo 3** Evaluar al nuevo individuo.

---

**Entrada:** Conjunto de correspondencias  $P_1$  y  $P_2$ ; umbral  $U$ .

**Entrada:** Nuevo individuo, cromosoma  $C$ .

**Salida:** Número de puntos típicos,  $m$ .

- 1:  $I := \text{decodificar}(C)$ ; //Obtener el índice de las 4 correspondencias de  $C$ .
  - 2: **if** Uno de los índices  $I$  es mayor que  $n$  **then**
  - 3:   Re-calcular el índice dentro del rango válido.
  - 4: **end if**
  - 5: **if**  $H$  se puede calcular a partir de las correspondencias de  $I$  **then**
  - 6:   Calcular el conjunto de errores  $E = \{e_1, e_2, e_3, \dots, e_n\}$  para  $P_1$  y  $P_2$  usando  $H$ .
  - 7:   Obtener el conjunto de pesos  $W$  como  $w_i = \begin{cases} 1, & \text{si } e_i \leq U, \\ 0, & \text{si } e_i > U. \end{cases}$
  - 8:    $m :=$  El número de puntos típicos para los cuales  $w_i = 1$ .
  - 9: **end if**
  - 10: **return**  $m$
-

---

**Algoritmo 4** Algoritmo basado en ED para detectar una homografía.**Entrada:** Dos conjuntos de puntos  $P_1$  y  $P_2$ , cada uno de ellos con  $n$  elementos; umbral  $U$ .**Entrada:** Parámetros de la ED; condición de paro  $C_p$ .**Salida:** La homografía presente en los datos de entrada.

- 1: Inicializar cada individuo de  $h_k^0$ , para  $k \in \{1, 2, \dots, t_G\}$ .
- 2: Evaluar cada individuo  $g(h_k^r)$ , para  $k \in \{1, 2, \dots, t_G\}$ .
- 3: **for**  $1 \leq r \leq \text{MAX\_G}$  **do**
- 4:   **for**  $1 \leq k \leq t_G$  **do**
- 5:     Seleccionar tres individuos al azar:  $\xi_1 \neq \xi_2 \neq \xi_3$ .
- 6:     Obtener una posición al azar,  $b_{rand} = \text{int}(8 * \text{rand}())$ .
- 7:     **for**  $1 \leq b \leq 8$  **do**
- 8:       **if**  $\text{rand}() < C_r$  or  $b = b_{rand}$  **then**
- 9:          $p_{kb}^r = \xi_{3b} + F_d(\xi_{1b} - \xi_{2b})$
- 10:       **else**
- 11:          $p_{kb}^r = h_{kb}^r$
- 12:       **end if**
- 13:     **end for**
- 14:     **if**  $g(p_k^r) < g(h_k^r)$  **then**
- 15:        $h_k^{r+1} = p_k^r$
- 16:     **else**
- 17:        $h_k^{r+1} = h_k^r$
- 18:     **end if**
- 19:   **end for**
- 20:   // Obtener la aptitud del mejor y peor individuo de la generación  $r$ .
- 21:    $m_{Apt} = \text{mejor}(g(h_k^r)); p_{Apt} = \text{peor}(g(h_k^r))$ , para  $k \in \{1, 2, \dots, t_G\}$ .
- 22:   **if**  $(p_{Apt} - m_{Apt}) < C_p$  **then**
- 23:     Detener.
- 24:   **end if**
- 25: **end for**
- 26: Re-estimar  $H$  utilizando todos los puntos típicos representados por  $h_k^{r+1}$ .
- 27: Re-calcular el número de puntos típicos con  $H$  re-estimada.

---

---

**Algoritmo 5** Evaluar individuo.

---

**Entrada:** Conjunto de correspondencias  $P_1$  y  $P_2$ ; umbral  $U$ .

**Entrada:** Nuevo individuo  $h$ ;  $T_1$  y  $T_2$ .

**Salida:** Aptitud del individuo  $h$ .

- 1:  $H' := h$  // Con  $h_{33} = 1$ .
  - 2: **if** Se puede calcular  $H'$  **then**
  - 3:    $H := T_1^{-1}H'T_2$
  - 4:   Calcular el conjunto de errores  $E = \{e_1, e_2, e_3, \dots, e_n\}$  para  $P_1$  y  $P_2$  usando  $H$ .
  - 5:   Obtener el conjunto de pesos  $W$  como  $w_i = \begin{cases} 1, & \text{si } e_i \leq U, \\ 0, & \text{si } e_i > U. \end{cases}$
  - 6:    $m :=$  El número de puntos típicos para los cuales  $w_i = 1$ .
  - 7: **end if**
  - 8: **return**  $n - m + \sum_1^m e_i^2$ , para  $w_i = 1$ .
- 

---

**Algoritmo 6** Extracción de múltiples homografías con RANSAC o GASAC.

---

**Entrada:** Algoritmo para extraer una homografía  $AE_H$ ; RANSAC o GASAC.

**Entrada:** Número de homografías  $N_H$  que se extraerán.

**Entrada:** Conjunto de correspondencias  $P_1$  y  $P_2$ .

**Salida:** Cada homografía  $H$  presente en los datos de entrada;  $P_c$ , sus puntos consistentes.

**Salida:** Puntos inconsistentes del conjunto de datos de entrada.

- 1:  $n := 0$ ; //Número de puntos inconsistentes.
  - 2: **for**  $1 \leq i \leq N_H$  **do**
  - 3:   Ejecutar  $AE_H$  con  $P_1$  y  $P_2$ .
  - 4:   Regresar los parámetros de la homografía  $H$ , así como el conjunto  $P_c$  obtenido.
  - 5:   Eliminar  $P_c$  del conjunto de datos  $P_1$  y  $P_2$ .
  - 6:    $n :=$  Obtener el número de correspondencias del conjunto de datos  $P_1$  y  $P_2$ .
  - 7:   **if**  $n < 4$  **then**
  - 8:     Detener.
  - 9:   **end if**
  - 10:    $i ++$ ;
  - 11: **end for**
  - 12: Regresar los puntos inconsistentes restantes en  $P_1$  y  $P_2$ .
-

---

**Algoritmo 7** Extracción de múltiples homografías con ED.

---

**Entrada:** Número de homografías  $N_H$  que se extraerán.**Entrada:** Conjunto de correspondencias  $P_1$  y  $P_2$ ; Ancho y alto de la imagen  $I_{P_1}$ .**Salida:** Cada homografía  $H$  presente en los datos de entrada;  $P_c$ , sus puntos consistentes.**Salida:** Puntos inconsistentes del conjunto de datos de entrada.

- 1:  $r := \frac{1}{5}$  del ancho de  $I_{P_1}$ ;
  - 2: **for**  $1 \leq i \leq N_H$  **do**
  - 3:   Seleccionar un punto al azar  $p_i$  dentro de  $I_{P_1}$ .
  - 4:   Encontrar en  $I_{P_1}$  todos los puntos  $P_e$  dentro del círculo con centro  $p_i$  y radio  $r$ .
  - 5:   Inicializar los límites de la ED con las correspondencias formadas por los puntos  $P_e$ .
  - 6:   Ejecutar ED con  $P_1$  y  $P_2$ .
  - 7:   Regresar los parámetros de la homografía  $H$ , así como el conjunto  $P_c$  obtenido.
  - 8:   Eliminar  $P_c$  del conjunto de datos  $P_1$  y  $P_2$ .
  - 9:    $n :=$  Obtener el número de correspondencias del conjunto de datos  $P_1$  y  $P_2$ .
  - 10:   **if**  $n < 4$  **then**
  - 11:     Detener.
  - 12:   **end if**
  - 13:    $i ++$ ;
  - 14: **end for**
  - 15: Regresar los puntos inconsistentes restantes en  $P_1$  y  $P_2$ .
-

# Capítulo 3

## Resultados y discusión

---

Para demostrar el desempeño de las soluciones propuestas es necesario ponerlas a prueba bajo diversos escenarios. En principio, para poder observar su desempeño, se describen las imágenes reales y simuladas que se utilizan para los experimentos del problema de la detección automática de homografías. Posteriormente se presentan los experimentos del problema (donde se emplean correspondencias que contienen puntos típicos, atípicos y con ruido), detallando los parámetros que se utilizan en cada uno de ellos. Por último se muestran los resultados obtenidos y la comparación de los algoritmos propuestos.

### 3.1. Imágenes de prueba

Para analizar el comportamiento de los algoritmos propuestos RANSAC, GASAC y ED (en la sección 2.1, 2.2 y 2.3 del capítulo 2 en las págs. 11, 12 y 15) se utilizaron para la primera prueba dos imágenes generadas de tamaño  $600 \times 600$  píxeles. Estas imágenes fueron generadas a partir de un modelo 3D con dos planos y el modelo de cámara oscura. Ambas imágenes se muestran en la figura 3.1. En esta configuración, conocemos de antemano las correspondencias de puntos en las imágenes.

Se utilizó una cámara digital (Webcam USB, sus características pueden observarse en la página web [3], en la pág. 53), con una resolución de dos megapíxeles y un ajuste automático de balance y color, para generar las imágenes reales.

Para la segunda prueba se utilizaron imágenes reales de  $1600 \times 1200$  píxeles. Para éstas se utilizaron tres tableros impresos en hojas de papel montados sobre tablas delgadas de madera, las cuales se colocaron sobre una mesa de uno de los laboratorios del Departamento de Computación del Cinvestav. Estos tableros nos permiten representar mediante las esquinas de sus cuadrados las correspondencias de puntos pertenecientes a tres planos. Las imágenes

de esta prueba se pueden observar en la figura 3.2.

Para la tercera prueba se utilizaron también imágenes reales pero de  $800 \times 600$  píxeles. En dichas imágenes se utilizaron triángulos de cartulina recortados a mano con las mismas proporciones y se colocaron sobre hojas de papel. Las hojas de papel junto con los triángulos se montaron sobre tres planos diferentes, la mesa de la prueba anterior y dos gabinetes de computadora. Las esquinas de los triángulos se utilizaron para representar las correspondencias de puntos de cada uno de los planos. Las imágenes de dicha prueba se pueden observar en la figura 3.3.

Para la segunda y tercera prueba se utilizó el detector de esquinas de Harris [22], para obtener automáticamente las características en las imágenes. Mediante las características determinadas con el detector de Harris se seleccionaron a mano las correspondencias de puntos. Usamos el detector de esquinas de Harris, porque es la mejor selección en las tareas de seguimiento visual [4].

En las figuras 3.2 y 3.3, en el inciso (c) se muestran los planos asociados a las homografías en las imágenes con diferentes marcas en las correspondencias de puntos. Las marcas triangulares negras corresponden a los puntos inconsistentes (obtenidos mediante el detector de esquinas de Harris) con los planos de interés.

Por último, en la cuarta prueba se utilizaron imágenes generadas de tamaño  $600 \times 600$  píxeles. Estas imágenes fueron obtenidas a partir del modelo 3D de la primera prueba, decrecentando el ángulo de separación entre los dos planos de  $170^\circ$  hasta  $70^\circ$ , de  $10^\circ$  en  $10^\circ$  grados. Estas imágenes se muestran en las figuras 3.4 a la 3.7 y cabe mencionar que en esta configuración conocemos de antemano las correspondencias de puntos en las imágenes.

### 3.1.1. Criterios por los cuales se escogieron los experimentos

Las imágenes de prueba de la sección anterior fueron seleccionadas tomando en cuenta los criterios del trabajo [4] (en la sección del diseño del conjunto de imágenes de prueba para tareas de seguimiento visual).

La primera prueba correspondiente a las series de imágenes de la figura 3.1, se utilizó porque:

- Para una secuencia de video, el tamaño de las imágenes generadas de  $600 \times 600$  píxeles es adecuado [4].
- Es una simulación básica planteada en los objetivos de la tesis; dos imágenes con dos planos que carecen de ruido con distribución gaussiana, donde los puntos atípicos del primer plano se representan por el conjunto puntos del segundo y viceversa.



- No necesita aplicar un proceso de correlación porque conocemos de antemano las correspondencias de puntos.

Esta prueba es útil porque nos permite probar los algoritmos propuestos en un ambiente controlado.

La segunda prueba correspondiente a las series de imágenes de la figura 3.2, se utilizó porque:

- Las imágenes generadas tienen el tamaño máximo soportado por la cámara digital utilizada,  $1600 \times 1200$  píxeles (dicha especificación se muestra en la página web [3], en la pág. 53).
- Esta es una prueba con imágenes reales, en la cual se busca trabajar con los tres planos visibles en la escena, los cuales se representan por los tableros.
- Se necesita aplicar un proceso de extracción de puntos de interés, para lo cual se utilizó el detector de esquinas de Harris [22]. Debido a que son imágenes reales, aquí es posible que aparezcan puntos atípicos.

Dicha prueba es útil porque nos permite examinar la robustez de los algoritmos propuestos en un ambiente real [4], con un conjunto de correspondencias que contienen puntos típicos, atípicos y con ruido.

La tercera prueba correspondiente a las series de imágenes de la figura 3.3, se utilizó por las mismas razones de la segunda prueba, pero en este caso:

- Las imágenes generadas tienen un tamaño de  $800 \times 600$  píxeles (la cual puede utilizarse en una secuencia de video [4]).
- El modelo utilizado es un patrón formado por triángulos recortados a mano para representar los tres planos en la escena.

Tanto en la segunda y tercera pruebas, el proceso de correlación para obtener las correspondencias de puntos se realizó manualmente. Dicho proceso en las tareas de seguimiento visual debe realizarse de forma automática, para lo cual existe una comparación exhaustiva de los algoritmos que realizan dicha tarea en el trabajo [4].

La cuarta prueba correspondiente a las series de imágenes de las figuras 3.4 a la 3.7, se utilizó por las mismas razones de la primera prueba, pero en este caso:

- Se disminuye el ángulo de separación entre dos planos de  $170^\circ$  hasta  $70^\circ$ , de  $10^\circ$  en  $10^\circ$  grados.

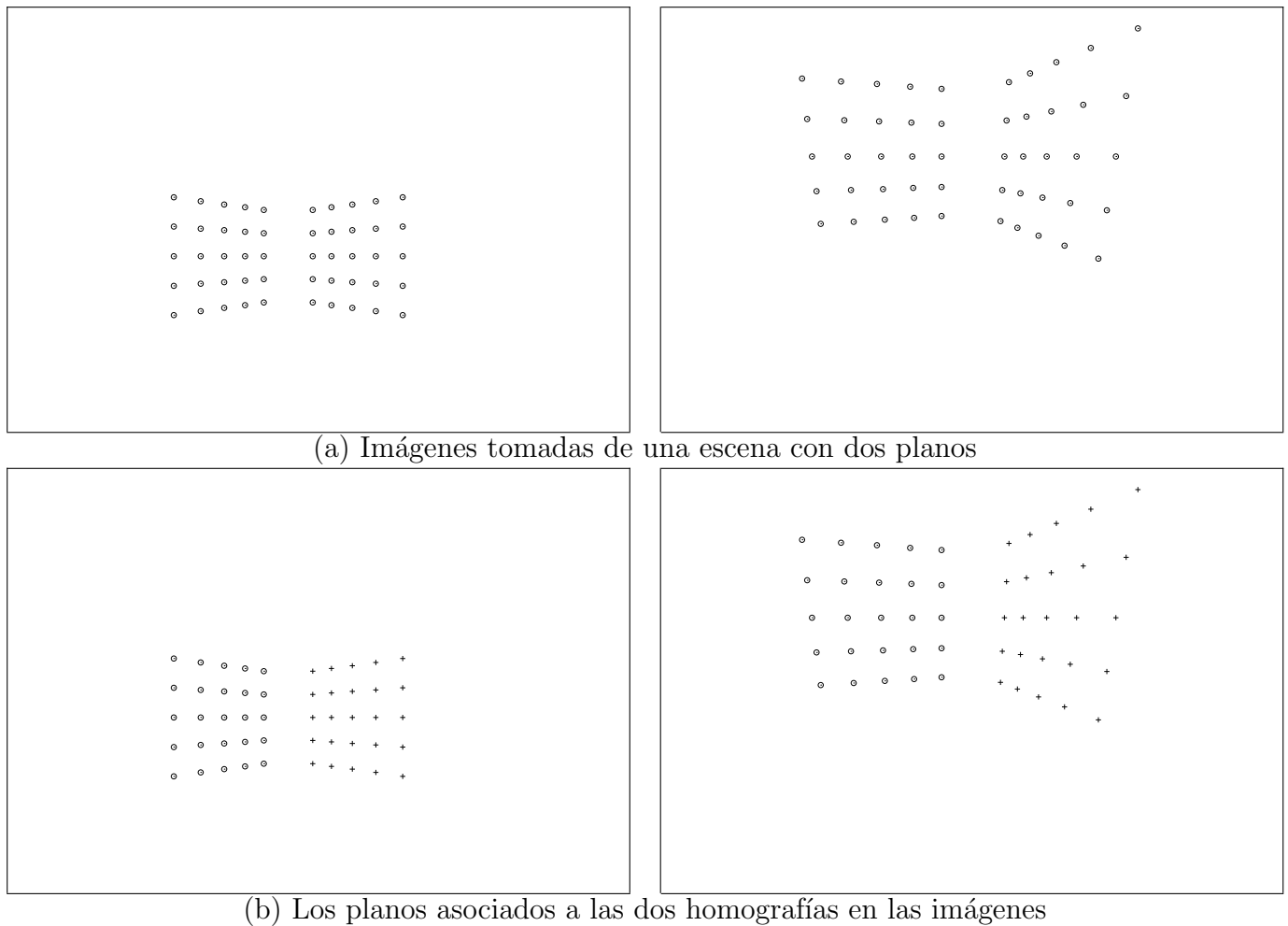


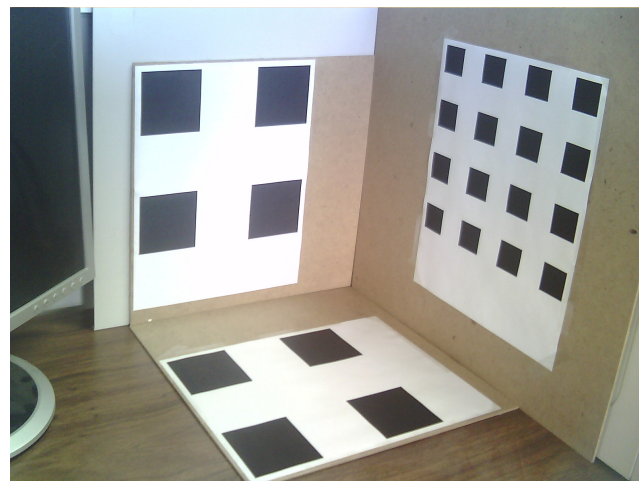
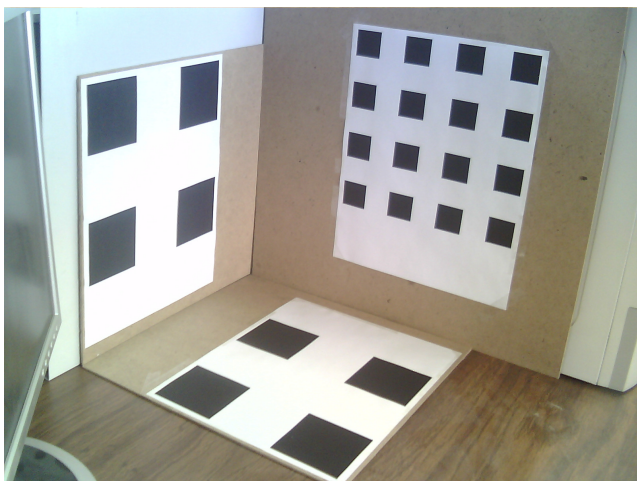
Figura 3.1: Las imágenes sintéticas con dos planos se muestran en (a) y en (b) se muestran las correspondencias de puntos de cada plano

La prueba es útil porque nos permite determinar en una escena el ángulo de separación entre dos planos, con el cual los algoritmos propuestos tienen un funcionamiento adecuado.

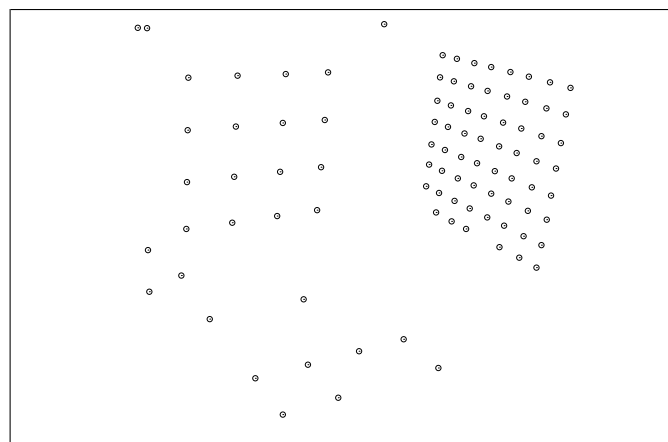
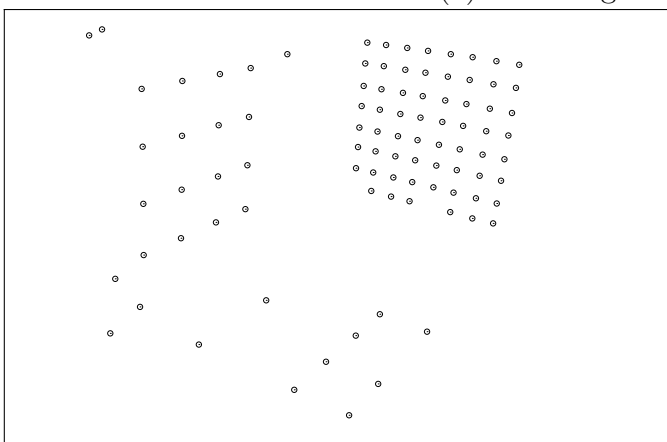
Entonces, incluso antes de ejecutar los algoritmos propuestos, si conocemos el ángulo de separación entre dos planos dentro de una escena, podríamos tener idea del comportamiento que deberían seguir.

## 3.2. Resultados

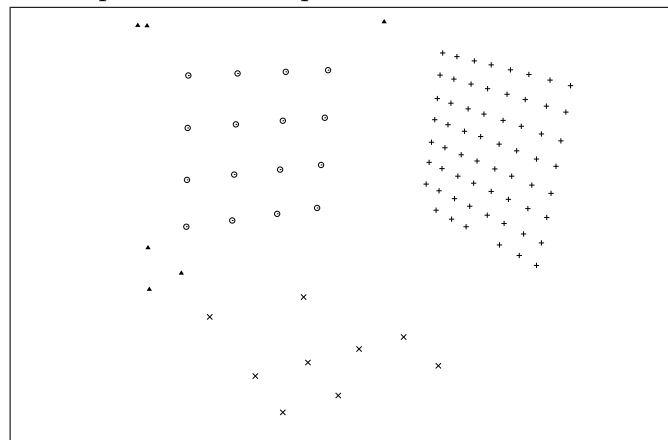
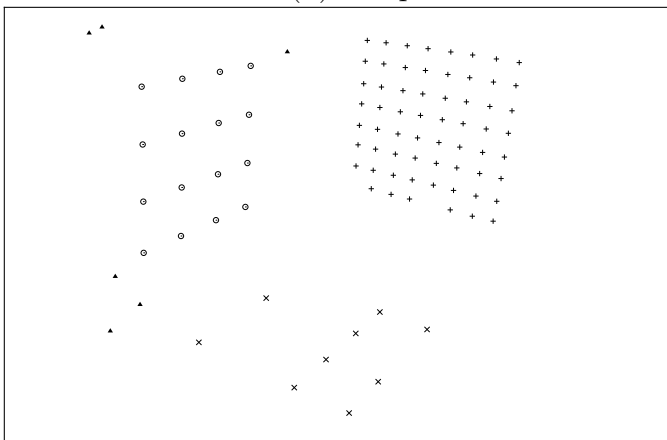
Con el fin de extraer las dos o tres homografías presentadas en las figuras 3.1, 3.2 y 3.3, mediante los algoritmos RANSAC y GASAC (descritos en la sección 2.1 y 2.2 del capítulo 2



(a) Dos imágenes reales con tableros

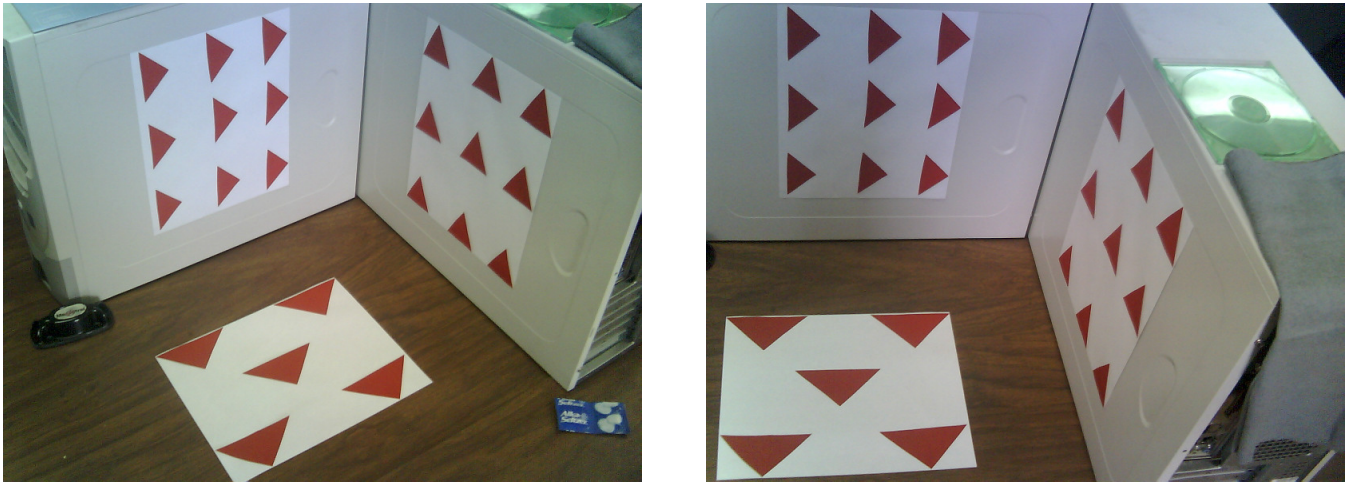


(b) Los pares seleccionados de correspondencias de puntos

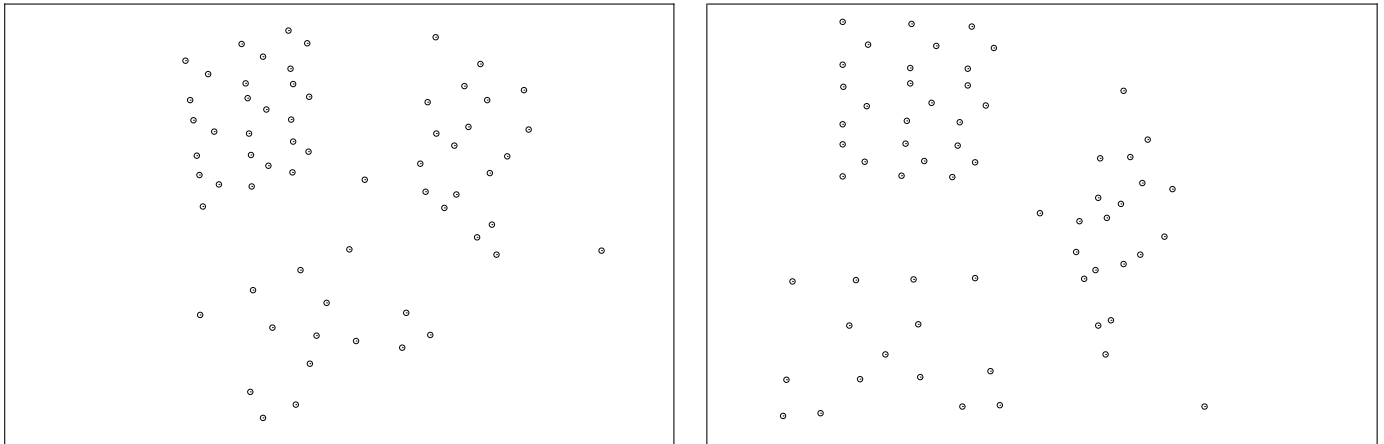


(c) Los planos asociados a las tres homografías en las imágenes

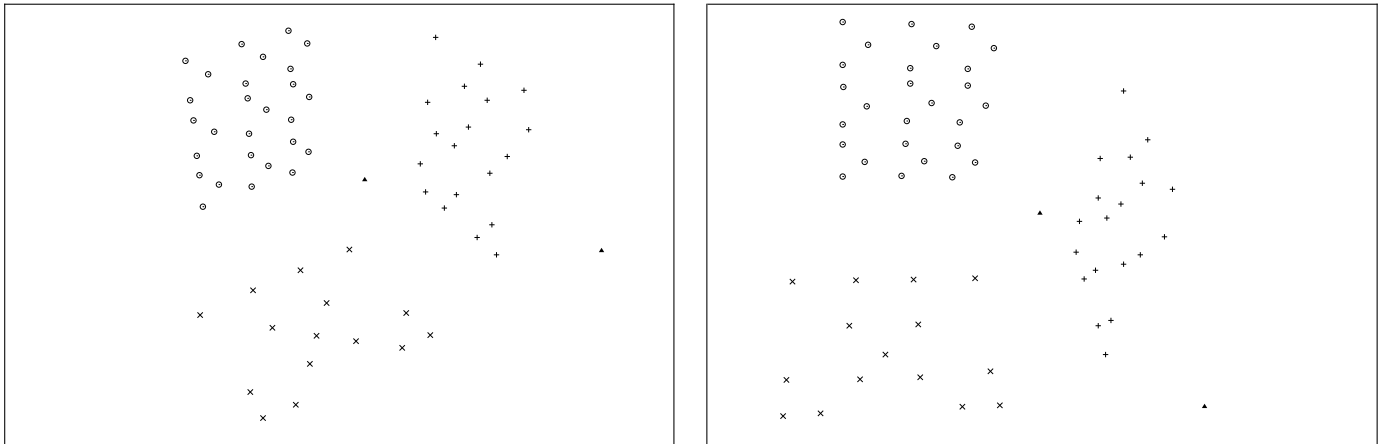
Figura 3.2: Las dos imágenes reales con tableros se muestran en (a) y en (b) se muestran las correspondencias de puntos seleccionados sobre ellos. En (c) se muestran los planos asociados a las dos homografías en las imágenes de los datos en (b)



(a) Dos imágenes reales con triángulos



(b) Los pares seleccionados de correspondencias de puntos



(c) Los planos asociados a las tres homografías en las imágenes

Figura 3.3: Las dos imágenes reales con triángulos se muestran en (a) y en (b) se muestran las correspondencias de puntos seleccionados sobre ellos. En (c) se muestran los planos asociados a las dos homografías en las imágenes de los datos en (b)

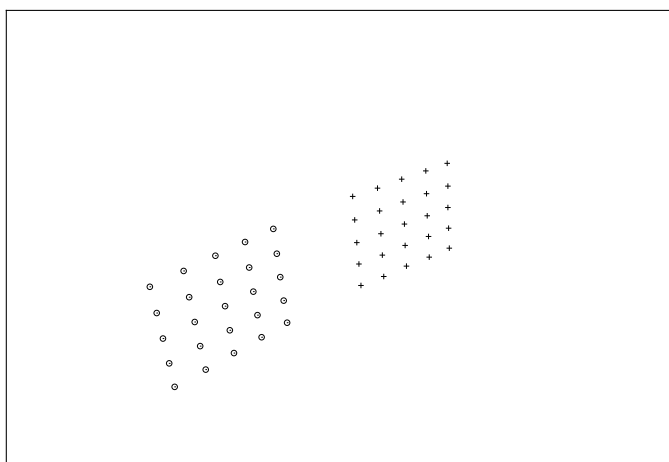
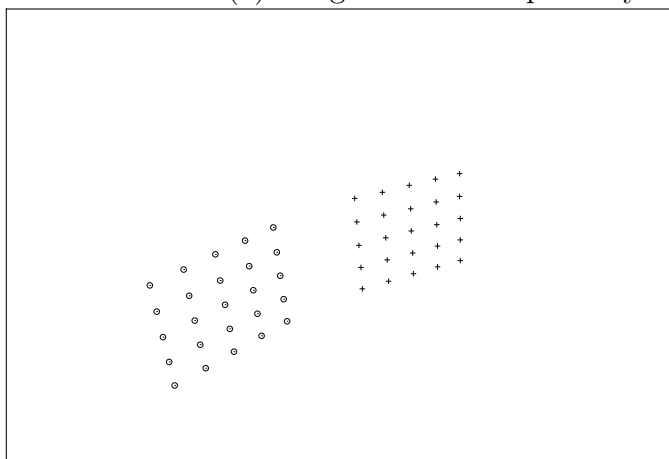
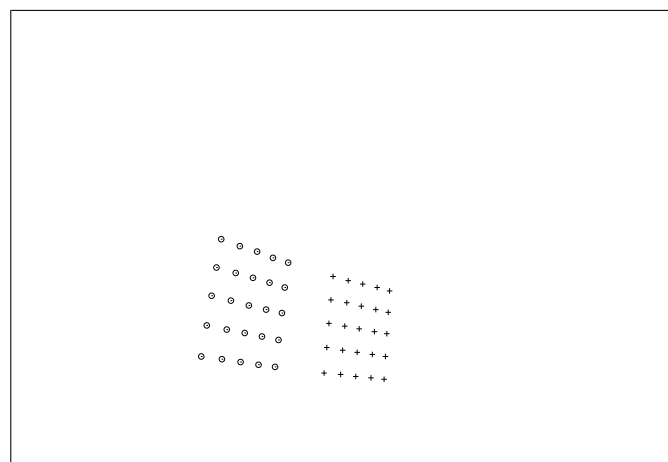
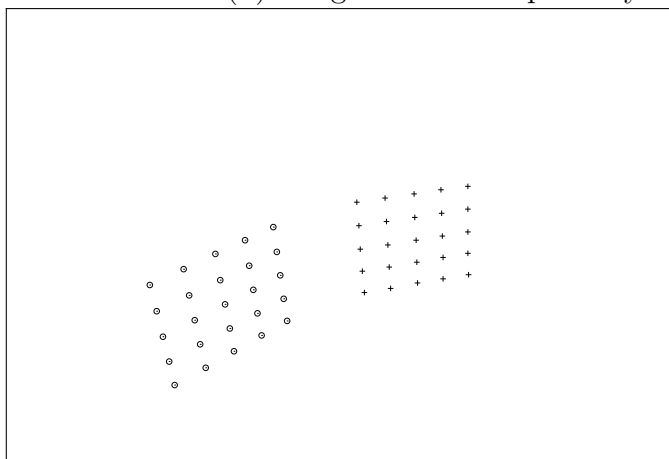
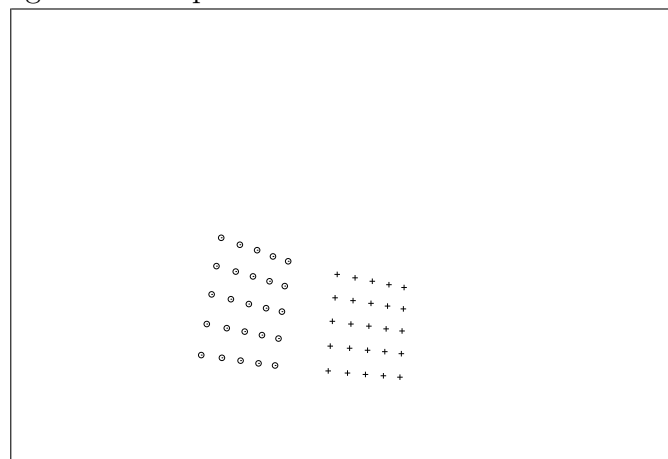
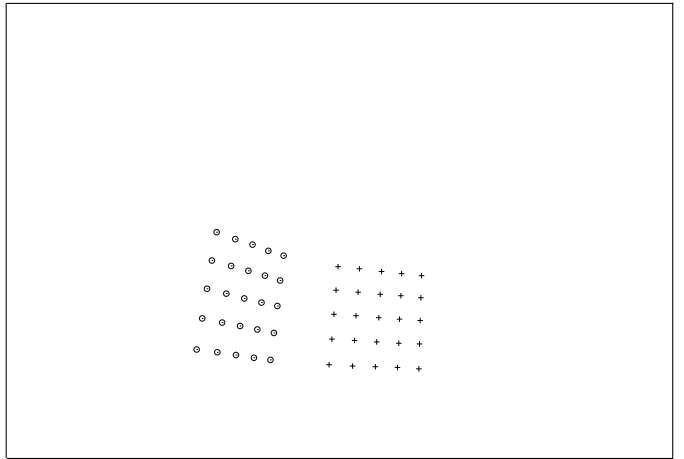
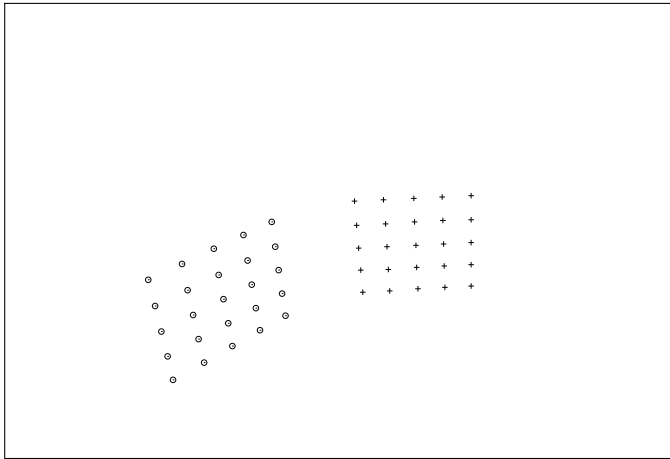
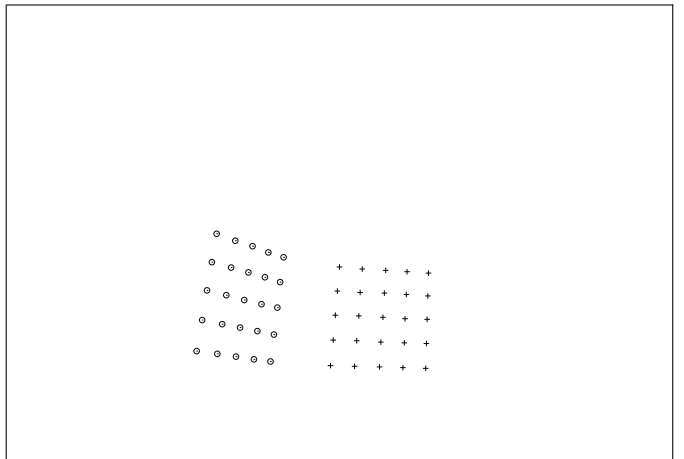
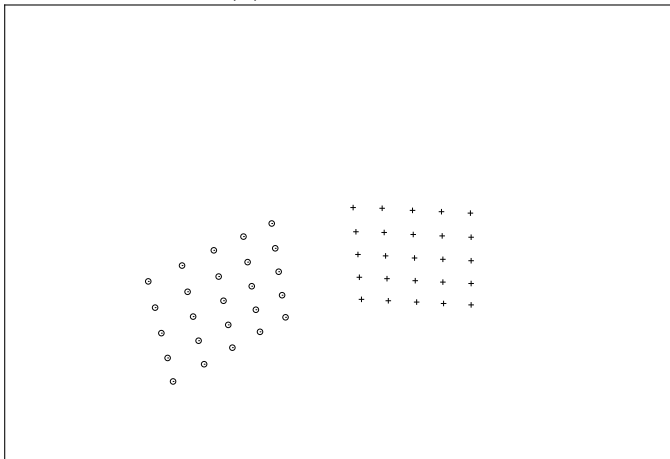
(a) Imágenes con dos planos y  $170^\circ$  grados de separación entre ellos(b) Imágenes con dos planos y  $160^\circ$  grados de separación entre ellos(c) Imágenes con dos planos y  $150^\circ$  grados de separación entre ellos

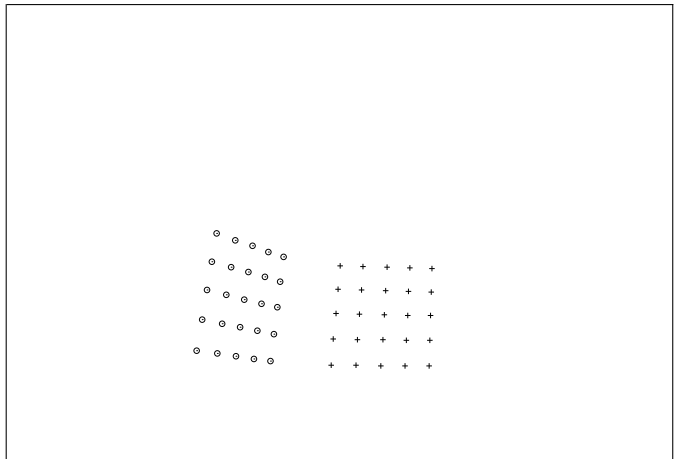
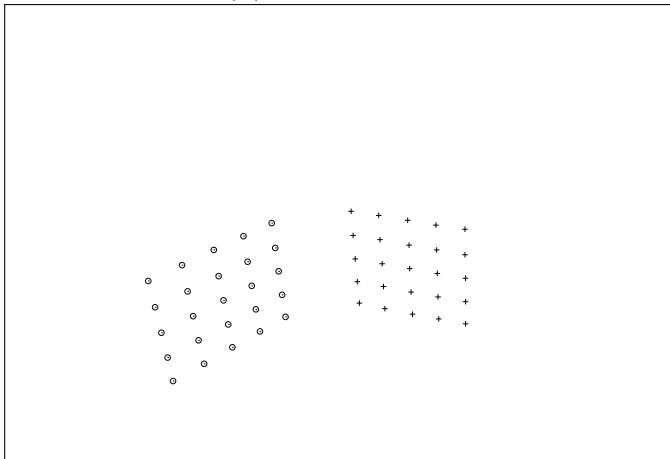
Figura 3.4: Las imágenes sintéticas con dos planos y un ángulo de separación de  $170^\circ$ ,  $160^\circ$  y  $150^\circ$  grados entre ellos se muestran en (a), (b) y (c)



(a) Imágenes con dos planos y  $140^\circ$  grados de separación entre ellos



(b) Imágenes con dos planos y  $130^\circ$  grados de separación entre ellos



(c) Imágenes con dos planos y  $120^\circ$  grados de separación entre ellos

Figura 3.5: Las imágenes sintéticas con dos planos y un ángulo de separación de  $140^\circ$ ,  $130^\circ$  y  $120^\circ$  grados entre ellos se muestran en (a), (b) y (c)

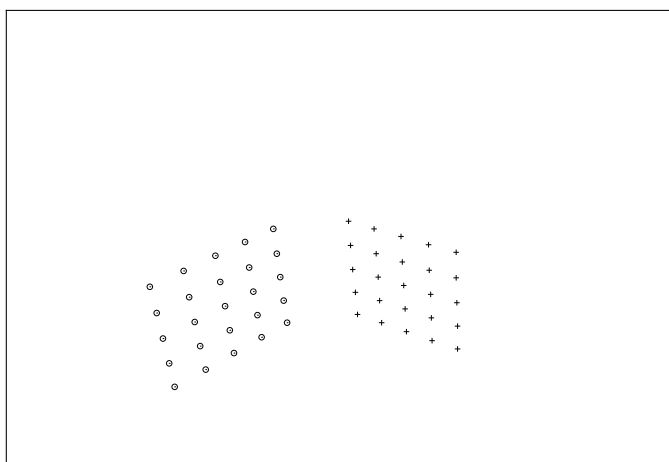
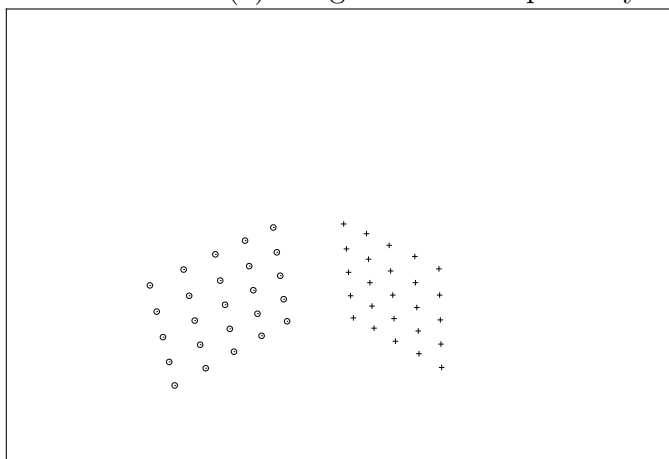
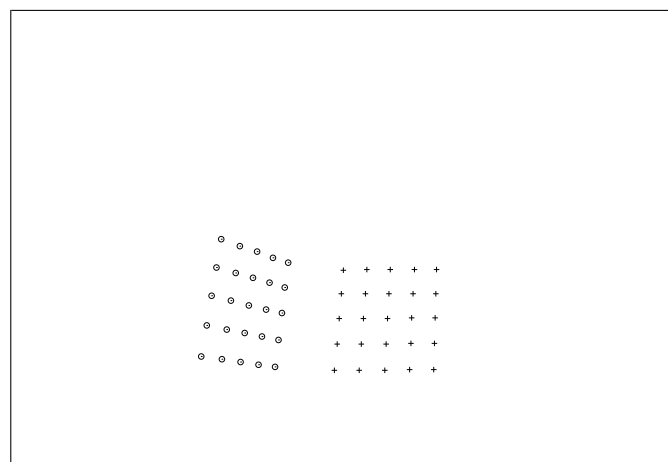
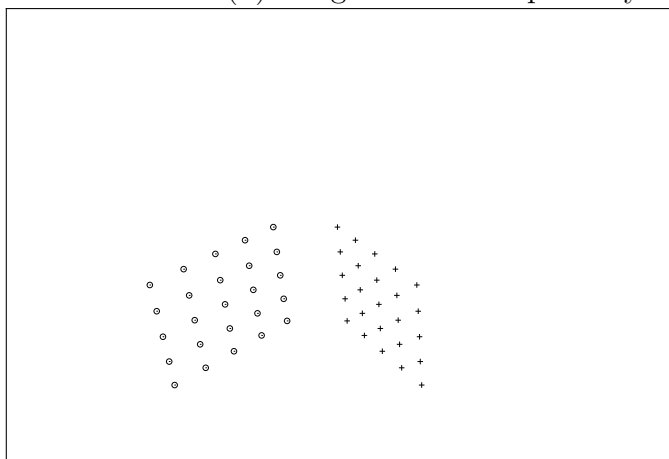
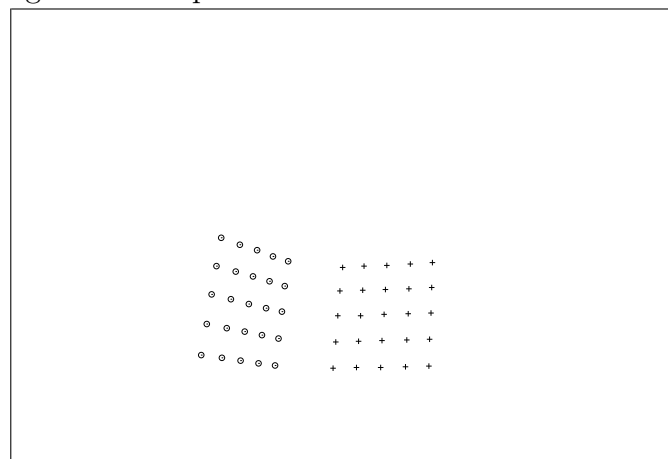
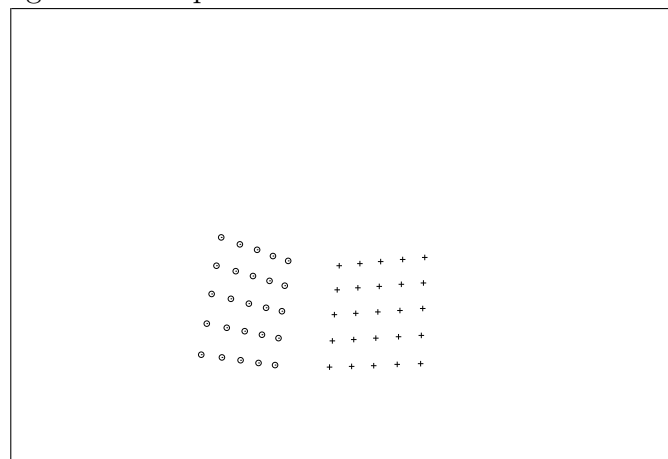
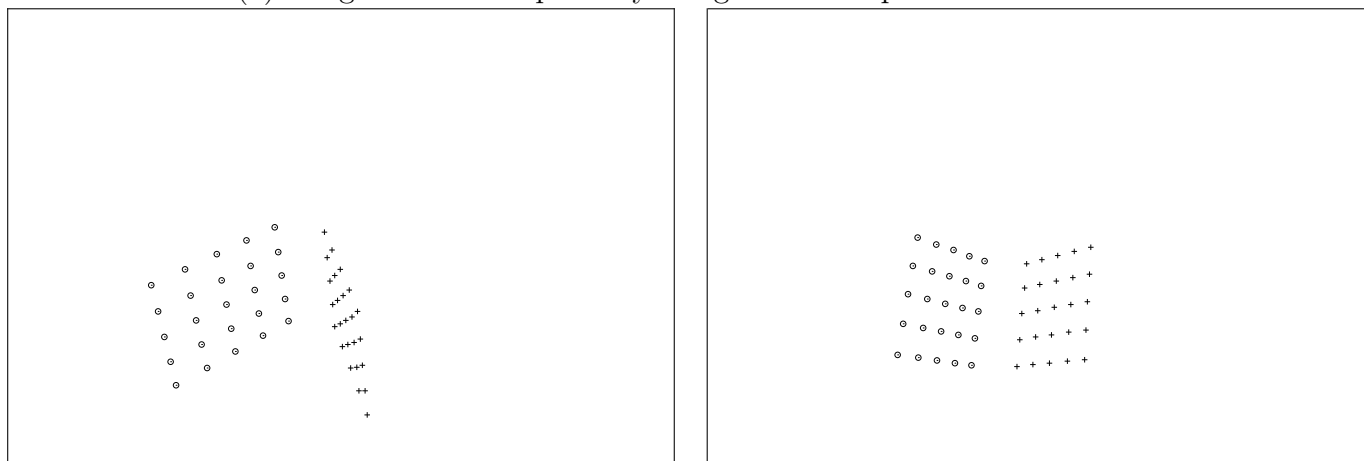
(a) Imágenes con dos planos y  $110^\circ$  grados de separación entre ellos(b) Imágenes con dos planos y  $100^\circ$  grados de separación entre ellos(c) Imágenes con dos planos y  $90^\circ$  grados de separación entre ellos

Figura 3.6: Las imágenes sintéticas con dos planos y un ángulo de separación de  $110^\circ$ ,  $100^\circ$  y  $90^\circ$  grados entre ellos se muestran en (a), (b) y (c)



(a) Imágenes con dos planos y  $80^\circ$  grados de separación entre ellos



(b) Imágenes con dos planos y  $70^\circ$  grados de separación entre ellos

Figura 3.7: Las imágenes sintéticas con dos planos y un ángulo de separación de  $80^\circ$  y  $70^\circ$  grados entre ellos se muestran en (a) y (b)



en las págs. 11 y 12) es necesario ejecutarlos dos y tres veces respectivamente con el Algoritmo 6 (descrito en la sección 2.4 del capítulo 2 en la pág. 23) y con la ED (descrita en la sección 2.3 del capítulo 2 en la pág. 15) se utiliza el Algoritmo 7 (descrito en la sección 2.4 del capítulo 2 en la pág. 24).

Se establece el valor de umbral  $U$  en 10 para RANSAC, GASAC y ED. En RANSAC, el valor inicial para el parámetro  $\epsilon_1$  se establece en 0.615 y esto nos da 207 iteraciones. El modelo probabilístico de RANSAC para estimar el número de iteraciones sólo tiene en cuenta la presencia de un único modelo de datos. Por tal motivo, se fija el número de iteraciones a 200 y las líneas 12-16 del Algoritmo 1 (en la sección 2.1 del capítulo 2 en la pág. 19) no se consideran.

Las estadísticas resultantes de 100 ejecuciones con el algoritmo RANSAC para el número de iteraciones y la función de error (descrita en la ecuación (1.4) de la pág. 8) se muestran en las tablas 3.3 y 3.4, mientras que usando GASAC y ED con los parámetros de las tablas 3.1 y 3.2 (variando el tamaño de la población en 5, 10, 15 y 20) los resultados se dan en las tablas 3.5 a la 3.7 y 3.8 a la 3.10, respectivamente.

Para comparar los resultados que se muestran en las tablas 3.5 y 3.8 es necesario multiplicar el tamaño de la población por la columna de resultado y dicho cálculo corresponde al número de evaluaciones de la función objetivo en las tablas 3.7 y 3.10.

Hemos observado en nuestras ejecuciones que algunas de las iteraciones de RANSAC son inválidas, debido a que mediante las cuatro correspondencias de puntos seleccionadas es imposible calcular una homografía (en la línea 5 del Algoritmo 1 en la sección 2.1 del capítulo 2 en la pág. 19); esto sucede debido a que, por ejemplo, tres o cuatro de las correspondencias son colineales.

Comparando los resultados de las tablas 3.5 a la 3.7 y 3.8 a la 3.10, se obtiene una configuración óptima (marcada en negritas en las tablas) de los parámetros de GASAC y ED con un tamaño de la población de 10 y 5, respectivamente. Dichos resultados se muestran en la tabla 3.11, en la cual es claro que GASAC reduce el número de evaluaciones en seis (6) y la ED realiza hasta un 25 % de las evaluaciones, efectuadas por RANSAC.

La medida de las ejecuciones con éxito en las tablas 3.3, 3.4, 3.5, 3.6, 3.8 y 3.9 nos indica si todos los puntos que pertenecen a las homografías se extraen correctamente. GASAC, al igual que la ED, algunas veces dan menos cantidad de puntos típicos.

Se puede deducir a partir de la tabla 3.5 con un tamaño de población de 10 individuos, que GASAC realizó la extracción exitosa en un 93 % de las ejecuciones de las tres pruebas realizadas, mientras que para la ED en la tabla 3.8 con un tamaño de población de 5 individuos, acertó el 99 % de las veces.

Parámetro	Valor
Número de generaciones	10
Probabilidad de Cruza	0.9
Probabilidad de Mutación	0.3

Tabla 3.1: Parámetros de entrada del algoritmo GASAC

Parámetro	Valor
Máximo número de generaciones	10
Probabilidad de Cruza	0.9
Probabilidad de Mutación	0.7
Condición de paro	1.8

Tabla 3.2: Parámetros de entrada de la ED

Por último se muestra en una única gráfica los resultados estadísticos de los algoritmos RANSAC, GASAC y ED de las tablas 3.3, 3.7 y 3.10 (utilizando la configuración óptima marcada en negritas) en la figura 3.8, donde se observa el número de evaluaciones de la función objetivo (el cual se obtiene multiplicando el promedio de las generaciones realizadas en GASAC y ED por el tamaño de la población), mientras que en la figura 3.9 se utilizan las tablas 3.4, 3.6 y 3.9 para analizar la función de error (como se citó anteriormente, esta describe el error obtenido entre las correspondencias de puntos). Ambas gráficas se obtienen utilizando tamaños de población de 5, 10, 15 y 20 individuos con las tres series de imágenes de la primera, segunda y tercera prueba.

### 3.2.1. Ángulo de separación entre dos planos

Para determinar el ángulo de separación entre dos planos dentro de una escena con el que los algoritmos propuestos funcionan de manera óptima, se utilizaron las figuras 3.4 a la 3.7 y los parámetros de las tablas 3.1 y 3.2 con un tamaño de la población de 10 y 5 para GASAC y ED, respectivamente.

	Imágenes en la figura 3.1	Imágenes en la figura 3.2	Imágenes en la figura 3.3
Ejecuciones con éxito	100	100	99
Mínimo	416	600	600
Máximo	444	601	600
Desviación estándar	5.31	0.20	0.00
Promedio	429.68	600.04	600

Tabla 3.3: Los resultados estadísticos de 100 ejecuciones con RANSAC, sobre el número de iteraciones en las tres series de imágenes

	Imágenes en la figura 3.1	Imágenes en la figura 3.2	Imágenes en la figura 3.3
Ejecuciones con éxito	100	100	99
Mínimo	0.34	1.67	1.30
Máximo	0.34	2.28	4.15
Desviación estándar	0.00	0.25	0.57
Promedio	0.34	1.91	1.50

Tabla 3.4: Los resultados estadísticos de 100 ejecuciones con RANSAC, sobre la función de error en las tres series de imágenes

	GASAC											
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	5	<b>10</b>	15	20	5	<b>10</b>	15	20	5	<b>10</b>	15	20
Ejecuciones con éxito	81	<b>98</b>	100	100	65	<b>92</b>	98	100	49	<b>90</b>	97	100
Mínimo	6	<b>6</b>	6	6	9	<b>9</b>	9	9	9	<b>9</b>	9	9
Máximo	10	<b>9</b>	8	8	17	<b>14</b>	13	13	15	<b>14</b>	13	13
Desviación estándar	1.00	<b>0.65</b>	0.45	0.33	1.82	<b>1.25</b>	0.94	0.81	1.46	<b>1.29</b>	0.99	0.87
Promedio	6.65	<b>6.32</b>	6.17	6.10	11.34	<b>10.2</b>	9.63	9.51	10.98	<b>10.64</b>	10.02	9.76

Tabla 3.5: Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre el número de iteraciones en las tres series de imágenes

	GASAC											
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	5	<b>10</b>	15	20	5	<b>10</b>	15	20	5	<b>10</b>	15	20
Ejecuciones con éxito	81	<b>98</b>	100	100	65	<b>92</b>	98	100	49	<b>90</b>	97	100
Mínimo	0.34	<b>0.34</b>	0.34	0.34	1.26	<b>1.26</b>	1.26	1.26	1.23	<b>1.43</b>	1.28	1.28
Máximo	0.34	<b>0.34</b>	0.34	0.34	2.04	<b>2.01</b>	2.23	2.11	1.98	<b>4.14</b>	2.68	1.34
Desviación estándar	0.00	<b>0.00</b>	0.00	0.00	0.18	<b>0.18</b>	0.18	0.16	0.10	<b>0.56</b>	0.17	0.02
Promedio	0.34	<b>0.34</b>	0.34	0.34	1.58	<b>1.58</b>	1.61	1.62	1.33	<b>2.53</b>	1.34	1.33

Tabla 3.6: Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes

GASAC												
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	5	<b>10</b>	15	20	5	<b>10</b>	15	20	5	<b>10</b>	15	20
Promedio	33.25	<b>63.2</b>	92.55	122	56.7	<b>102</b>	144.45	190.2	54.9	<b>106.4</b>	150.3	195.2

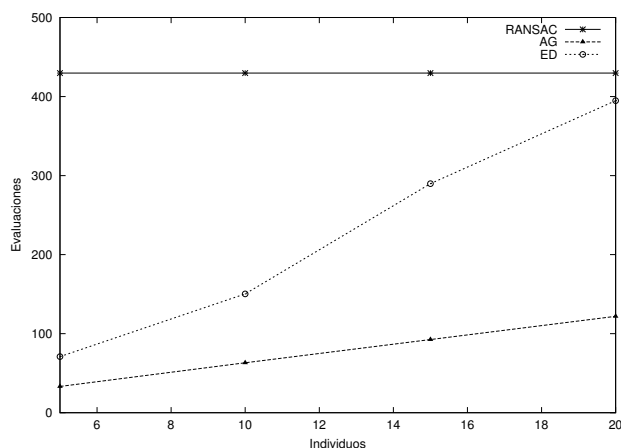
Tabla 3.7: Los resultados estadísticos de 100 ejecuciones con GASAC, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes

ED												
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	<b>5</b>	10	15	20	<b>5</b>	10	15	20	<b>5</b>	10	15	20
Ejecuciones con éxito	<b>100</b>	100	100	100	<b>99</b>	100	100	100	<b>100</b>	100	100	100
Mínimo	<b>8</b>	11	15	15	<b>26</b>	30	30	30	<b>22</b>	22	30	30
Máximo	<b>20</b>	20	20	20	<b>30</b>	30	30	30	<b>30</b>	30	30	30
Desviación estándar	<b>3.50</b>	2.51	1.33	0.93	<b>0.56</b>	0.00	0.00	0.00	<b>2.32</b>	1.63	0.00	0.00
Promedio	<b>14.19</b>	15.03	19.32	19.74	<b>29.92</b>	30	30	30	<b>28.45</b>	29.44	30	30

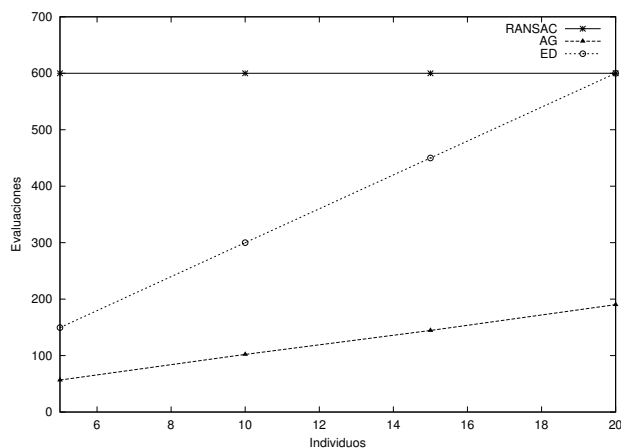
Tabla 3.8: Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de iteraciones en las tres series de imágenes

ED												
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	<b>5</b>	10	15	20	<b>5</b>	10	15	20	<b>5</b>	10	15	20
Ejecuciones con éxito	<b>100</b>	100	100	100	<b>99</b>	100	100	100	<b>100</b>	100	100	100
Mínimo	<b>0.34</b>	0.34	0.34	0.34	<b>1.26</b>	1.26	1.26	1.26	<b>1.30</b>	1.28	1.28	1.28
Máximo	<b>0.34</b>	0.34	0.34	0.34	<b>2.16</b>	2.23	2.23	2.16	<b>1.34</b>	2.40	2.33	2.53
Desviación estándar	<b>0.00</b>	0.00	0.00	0.00	<b>0.12</b>	0.11	0.10	0.08	<b>0.02</b>	0.38	0.37	0.21
Promedio	<b>0.34</b>	0.34	0.34	0.34	<b>1.67</b>	1.69	1.68	1.67	<b>1.32</b>	1.49	1.48	1.35

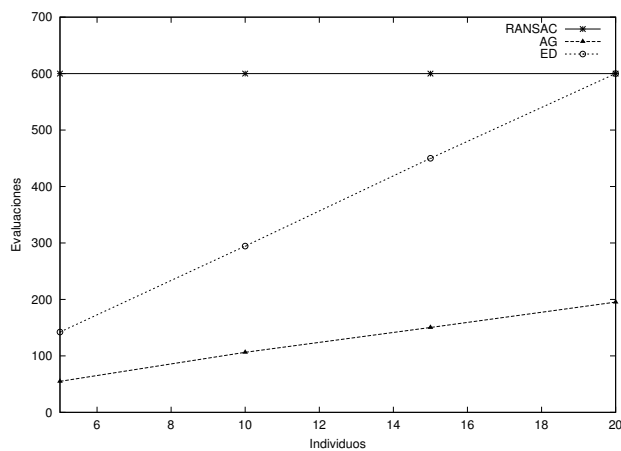
Tabla 3.9: Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes



(a) Resultados estadísticos correspondientes a la figura 3.1

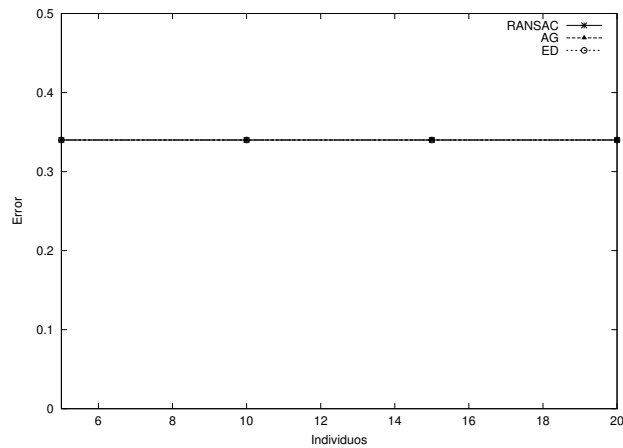


(b) Resultados estadísticos correspondientes a la figura 3.2

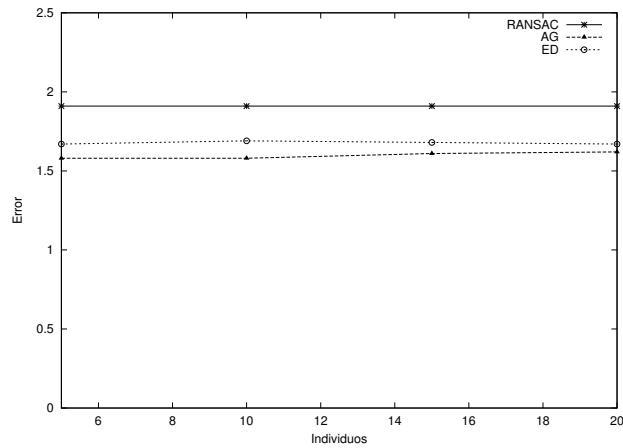


(c) Resultados estadísticos correspondientes a la figura 3.3

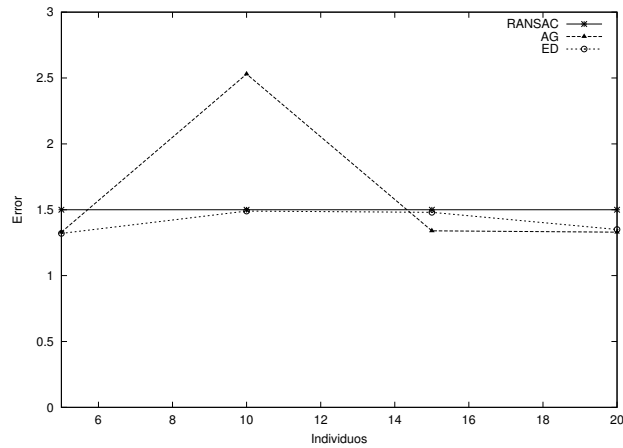
Figura 3.8: Los resultados estadísticos promedio de 100 ejecuciones con los algoritmos propuestos RANSAC, GASAC y ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes en (a), (b) y (c)



(a) Resultados estadísticos correspondientes a la figura 3.1



(b) Resultados estadísticos correspondientes a la figura 3.2



(c) Resultados estadísticos correspondientes a la figura 3.3

Figura 3.9: Los resultados estadísticos promedio de 100 ejecuciones con los algoritmos propuestos RANSAC, GASAC y ED, para tamaños de población de 5, 10, 15 y 20, sobre la función de error en las tres series de imágenes en (a), (b) y (c)

	ED											
	Imágenes en la figura 3.1				Imágenes en la figura 3.2				Imágenes en la figura 3.3			
Tamaño de la población	5	10	15	20	5	10	15	20	5	10	15	20
Promedio	<b>70.95</b>	150.3	289.8	394.8	<b>149.6</b>	300	450	600	<b>142.25</b>	294.4	450	600

Tabla 3.10: Los resultados estadísticos de 100 ejecuciones con ED, para tamaños de población de 5, 10, 15 y 20, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes

	Imágenes en la figura 3.1			Imágenes en la figura 3.2			Imágenes en la figura 3.3		
	RANSAC	GASAC	ED	RANSAC	GASAC	ED	RANSAC	GASAC	ED
Promedio	429.68	63.20	70.95	600.04	102	149.60	600	106.40	142.25

Tabla 3.11: Los resultados estadísticos de 100 ejecuciones con RANSAC, GASAC y ED, sobre el número de evaluaciones de la función objetivo en las tres series de imágenes, para tamaños de población de 10 y 5 en GASAC y ED, respectivamente

Es necesario ejecutar dos veces los algoritmos RANSAC y GASAC (discutidos en la sección 2.1 y 2.2 del capítulo 2 en las págs. 11 y 12) mediante el Algoritmo 6 (discutido en la sección 2.4 del capítulo 2 en la pág. 23) y la ED (descrita en la sección 2.3 del capítulo 2 en la pág. 15) con el Algoritmo 7 (discutido en la sección 2.4 del capítulo 2 en la pág. 24) para extraer las homografías presentadas en las figuras 3.4 a la 3.7.

Las estadísticas resultantes de 100 ejecuciones con los algoritmos RANSAC, GASAC y ED para el número de evaluaciones de la función objetivo (el cual se obtiene multiplicando el promedio de las generaciones realizadas en GASAC y ED por el tamaño de la población 10 y 5, respectivamente) y la función de error (descrita en la ecuación (1.4) de la pág. 8) se muestran en la tabla 3.12. Se decidió agregar la función de error como una columna más en la tabla 3.12, debido a que en todas las ejecuciones el mínimo y máximo error obtenido entre las correspondencias de puntos es el mismo, con una desviación estándar de cero, por lo que el error promedio es siempre igual a éstos.

En la tabla 3.12 se puede observar la extracción exitosa de las homografías en las figuras 3.4 a la 3.7, donde con un ángulo de separación entre los dos planos de  $170^\circ$ , los algoritmos propuestos RANSAC, GASAC y la ED acertaron un 0%, 19% y 37% de las veces, mientras que con  $160^\circ$ , RANSAC se mantiene con 0% y GASAC junto con la ED consiguen un 96%, con lo que a partir de  $150^\circ$  y hasta  $70^\circ$  dichos algoritmos mantienen una tendencia de éxito mayor al 90%, donde GASAC reduce el número de evaluaciones en seis (6) y la ED realiza hasta un 25% de las evaluaciones, efectuadas por RANSAC.

### 3.3. Análisis de los resultados obtenidos

Los resultados obtenidos en esta tesis (para resolver el problema de la detección automática de homografías), son correctos porque se logró extraer de manera exitosa las dos y tres homografías correspondientes a los planos en las imágenes de la primera, segunda, tercera y cuarta pruebas (descritas en la sección 3.1 del presente capítulo en la pág. 25).

Dichos resultados son admisibles porque los algoritmos GASAC y ED (discutidos en la sección 2.2 y 2.3 del capítulo 2 en las págs. 12 y 15), demostraron empíricamente ser igual de robustos que el algoritmo más representativo en la literatura especializada, es decir RANSAC (discutido en la sección 2.1 del capítulo 2 en la pág. 11).

También, los resultados son útiles porque GASAC reduce el número de evaluaciones en seis (6) y la ED realiza hasta un 25 % de las evaluaciones efectuadas por RANSAC. Entonces para obtener resultados aceptables, GASAC y la ED necesitarían tiempos de ejecución menores al realizado por RANSAC. Lo cual en tareas de seguimiento visual [4] representaría una mejora considerable.

### 3.4. URL del trabajo

Es posible disponer de la información aquí expuesta y de los programas desarrollados mediante la página web [1], en la pág. 53.

En dicha página se muestra un resumen de nuestro trabajo, las exposiciones presentadas al largo del desarrollo, un artículo referente a nuestro trabajo de tesis y la exposición hecha para su presentación.

También hacemos disponible los programas realizados, su documentación y las imágenes de prueba utilizadas con sus respectivas correspondencias de puntos.

#### 3.4.1. Programas

Entre los programas tenemos:

1. Los que realizan la detección robusta de una homografía mediante el algoritmo RANSAC, un algoritmo genético simple (AG) y la evolución diferencial (ED).
2. Los que realizan la extracción automática de múltiples homografías (utilizando los algoritmos anteriores).
3. El programa que realiza la extracción automática de puntos de interés.

El código de RANSAC esta realizado en el lenguaje de programación C, al igual que el código del AG, mientras que para la ED se utilizó C++. En estas implementaciones para



el cálculo de la descomposición en valores singulares (SVD) se utilizó la biblioteca descrita en [23].

Debido a la sencillez de los programas que realizan la extracción automática de múltiples homografías, estos fueron desarrollados en el lenguaje de programación C y no necesitan de alguna biblioteca no estándar para realizar sus procesos.

El programa que realiza la extracción automática de puntos de interés, utiliza el algoritmo del detector de esquinas de Harris y se realizó con el lenguaje de programación C++ utilizando las funciones de la biblioteca OpenCV (la implementación de dicho algoritmo se describe en [22]).

Por último los programas realizados son multiplataforma, debido a que su codificación se realizó con lenguajes de programación estándar. Debido a lo anterior, no es necesario profundizar en la plataforma utilizada para la ejecución de dichos algoritmos, porque en este trabajo de tesis sólo se analiza el número de iteraciones realizadas y el error entre las correspondencias de puntos, los cuales son independientes de la plataforma (en cualquier plataforma compatible con los programas realizados, se deben obtener los mismos resultados).

Algoritmo	Grados	Ejecuciones con éxito	Mínimo	Máximo	Desviación estándar	Promedio	Error Promedio
RANSAC	170°	0	-	-	-	-	-
GASAC		19	6	7	0.36	61.6	0.50
ED		37	18	20	0.32	99.75	0.50
RANSAC	160°	0	-	-	-	-	-
GASAC		96	6	11	0.80	64.4	0.48
ED		96	20	20	0.00	100	0.48
RANSAC	150°	92	400	408	1.95	402.89	0.43
GASAC		99	6	10	0.72	64.9	0.43
ED		100	20	20	0.00	100	0.43
RANSAC	140°	100	400	415	3.00	404.51	0.37
GASAC		100	6	9	0.71	63.9	0.37
ED		100	16	20	0.40	99.8	0.37
RANSAC	130°	100	400	408	1.75	402.73	0.48
GASAC		100	6	11	0.87	65.1	0.48
ED		95	13	20	1.62	96.4	0.48
RANSAC	120°	100	400	409	1.76	402.95	0.43
GASAC		98	6	10	0.77	64.3	0.43
ED		100	20	20	0.00	100	0.43
RANSAC	110°	100	400	405	1.40	401.81	0.48
GASAC		99	6	10	0.86	65.2	0.48
ED		100	14	20	0.77	99.45	0.48
RANSAC	100°	100	400	406	1.34	401.38	0.63
GASAC		99	6	8	0.59	64.1	0.63
ED		100	20	20	0.00	100	0.63
RANSAC	90°	100	400	406	1.45	402.04	0.39
GASAC		99	6	9	0.77	64.9	0.39
ED		100	20	20	0.00	100	0.39
RANSAC	80°	100	400	405	1.17	401.31	0.76
GASAC		100	6	8	0.68	64.4	0.76
ED		100	20	20	0.00	100	0.76
RANSAC	70°	100	400	409	1.60	401.35	0.82
GASAC		99	6	9	0.73	64.9	0.82
ED		100	20	20	0.00	100	0.82

Tabla 3.12: Los resultados estadísticos de 100 ejecuciones para determinar el ángulo de separación entre dos planos con RANSAC, GASAC y ED, sobre el número de evaluaciones de la función objetivo y la función de error, para tamaños de población de 10 y 5 en GASAC y ED, respectivamente

# Capítulo 4

## Conclusiones

---

En esta tesis se resolvió el problema de la detección automática de homografías. A partir de un conjunto de correspondencias que contienen puntos típicos, atípicos y con ruido, se buscó extraer las matrices de homografía  $H$  consistentes con los puntos pertenecientes a los planos dentro de la escena en las imágenes analizadas. La única forma de determinar cuáles de las homografías  $H$  representan mejor al conjunto de puntos típicos, es la función de costo que minimiza la suma de los errores obtenidos mediante la distancia euclidiana entre las correspondencias estimadas con  $H$  y  $H^{-1}$ , como se describió en la sección 1.4 del capítulo 1 (en la pág. 7).

Se propusieron tres soluciones para la detección automática de homografías:

1. El algoritmo RANSAC (en la sección 2.1 del capítulo 2 en la pág. 11): recibe como entrada un conjunto de correspondencias que contienen puntos típicos, atípicos y con ruido y selecciona al azar un subconjunto de cuatro correspondencias de puntos con el fin de estimar la homografía  $H$ , mediante el algoritmo lineal (en la sección 1.3 del capítulo 1 en la pág. 4). Si se maximiza el número de correspondencias de puntos que pertenecen a la homografía,  $H$  es aceptada y el subconjunto de los cuatro puntos implicados en el cálculo de  $H$  se denomina conjunto de consenso. Este procedimiento se repite un número determinado de veces, con el fin de obtener la mejor estimación de  $H$ .
2. El algoritmo GASAC (en la sección 2.2 del capítulo 2 en la pág. 12): trabaja sobre un conjunto de individuos y busca una solución por aproximación mejorando la aptitud promedio de la población puesto que se obtienen nuevos y mejores individuos y por lo tanto, mejores soluciones. La aptitud medida es también el número de correspondencias de puntos que pertenecen a una homografía dada. Utiliza como cromosoma los índices en binario a cuatro puntos y repara el cromosoma dentro de la función objetivo, si uno de los índices es mayor que el número de correspondencias de puntos. Aplica un

proceso de selección obteniendo dos soluciones al azar y tomando de ellas la mejor, mientras que la cruce selecciona al azar dos posiciones del cromosoma e intercambia su material cromosómico (debido a la representación binaria, la mutación cambia de 0 a 1 y viceversa).

3. El algoritmo basado en evolución diferencial (en la sección 2.3 del capítulo 2 en la pág. 15): maneja un conjunto de individuos conocido como población, los cuales contienen una solución del problema, que son los ocho parámetros de  $H'$  de la ecuación (1.3):  $H = T_1^{-1}H'T_2$ , donde  $T_1$  y  $T_2$  son matrices en  $\mathbb{R}^{3 \times 3}$  para la normalización del conjunto total de correspondencias de puntos con media cero y desviación estándar uno, obtenidas mediante el algoritmo lineal (en la sección 1.3 del capítulo 1 en la pág. 4). Se inicializa cada individuo de la población mediante los límites de cada parámetro y posteriormente, la población se actualiza por medio de los operadores de la evolución diferencial. La principal idea de la evolución diferencial es emplear la diferencia entre los valores de otros dos individuos para generar nuevos individuos, de los cuales solo aquellos que tienen mejor aptitud pasan de una generación a la siguiente.

En el primer enfoque, usando RANSAC, se calcula el número de iteraciones del algoritmo con un enfoque probabilista, mientras que en los otros dos enfoques, se hace uso de un algoritmo genético (AG) y de la evolución diferencial (ED). Con el AG se usó como condición de paro el que, en dos generaciones consecutivas se tuviera el mismo número de puntos para calcular la homografía. Con la ED se checa la diferencia entre el mejor y el peor individuo; si esta diferencia es menor que 1.8, se detiene la ejecución del algoritmo.

Cabe señalar que decidimos llamar a nuestra segunda solución GASAC (Genetic Algorithm Sample Consensus) porque aunque se trata de un AG, éste utiliza un grupo de consenso en la evaluación de la función objetivo.

A lo largo de la tesis, lo que se hizo fue aplicar GASAC y ED para resolver el problema de la detección robusta de homografías y demostrar que funcionan en comparación con RANSAC. Las dos soluciones propuestas han demostrado ser métodos igual de robustos que RANSAC, pero realizando un número menor de iteraciones.

Lo anterior, se debe a que GASAC mejora la simple búsqueda aleatoria de RANSAC para encontrar los cuatro puntos del conjunto de consenso y con la ED creemos que se tiene la ventaja de usar las condiciones de paro naturales de la misma, que en sí es como cualquier otro algoritmo con números reales.

El método propuesto GASAC es capaz de realizar la extracción exitosa de las homografías en un 93% de las ejecuciones en las pruebas realizadas. Se muestra empíricamente que la incorporación de un AG para la selección de cuatro puntos del conjunto de consenso, en el algoritmo RANSAC para detectar homografías, reduce casi 6 veces en promedio el número de evaluaciones de la función objetivo con respecto a dicho algoritmo.

El éxito del método GASAC radica en lo siguiente:

- En la población inicial, todos los individuos deben ser válidos, esto quiere decir que con los cuatro puntos asociados al individuo, se puede calcular una homografía (no se puede calcular si los cuatro puntos son colineales, por ejemplo).
- Maximizar el número de puntos que forman parte del ajuste al realizar la extracción de varias homografías, y si se tiene igual número de puntos, se escoge la solución con un error menor.

El método propuesto en esta tesis basado en ED para la detección automática de homografías, logra realizar la extracción exitosa en un 99% de las ejecuciones de las pruebas realizadas, demostrando que se pueden obtener los ocho parámetros de  $H'$  de la ecuación (1.3):  $H = T_1^{-1}H'T_2$ , mediante el método descrito en la sección 2.3 del capítulo 2 en la pág. 15, con lo que se reduce casi 6 veces en promedio el número de evaluaciones de la función objetivo del algoritmo original de RANSAC.

El éxito del método que utiliza el algoritmo basado en ED radica en lo siguiente:

- Resolver el problema mediante un enfoque de minimización, combinando las restricciones de maximizar el número de puntos típicos consistentes con la matriz de homografía  $H$  y minimizar el error obtenido entre dichas correspondencias, como se muestra en la ecuación (2.1) (en la sección 2.3 del capítulo 2 en la pág. 17).

El ángulo de separación entre dos planos, para que los algoritmos RANSAC, GASAC y el basado en ED funcionen de manera óptima, es entre  $70^\circ$  y  $150^\circ$ . GASAC y ED también resuelven el problema cuando los dos planos están con un ángulo entre  $160$  y  $170$  grados (los dos planos son casi iguales con este ángulo de separación). RANSAC no resuelve el problema en este último caso.

Podemos decir que la principal desventaja del método RANSAC es que utiliza un enfoque probabilista para calcular el número de iteraciones. No está claro con este enfoque probabilista cómo calcular el número de iteraciones si está presente más de una homografía en el conjunto de datos. Además de que sólo puede estimarse crudamente el número de iteraciones, dado que el número de puntos típicos, atípicos y con ruido sólo pueden conocerse al resolver el problema. El límite dado por RANSAC es un límite superior del número de iteraciones. Con el AG y la ED se puede mejorar la obtención aleatoria del conjunto de consenso y se deduce el número de veces que tiene que calcularse un nuevo grupo de consenso.

Por último, el problema del ajuste robusto de homografías también puede verse como un problema de clasificación de cúmulos (clustering) con una restricción geométrica (la homografía). En 2009 se demostró que la búsqueda de cúmulos es un problema NP-duro [24, 25].

## 4.1. Trabajo a futuro

De las soluciones propuestas en esta tesis se desprenden posibles trabajos futuros con base en el problema que se ataca:

- Creemos que el enfoque del AG, así como el de la ED, puede ser incorporado a cualquier algoritmo RANSAC.
- Para la detección de homografías es necesario primero detectar los puntos de correspondencia presentes en las imágenes. Pensamos que es un reto para atacar en un futuro próximo, encontrar métodos eficaces para la detección automática de homografías y correspondencias de puntos al mismo tiempo. Para la detección de correspondencias podrían usarse los algoritmos [3, 26], de los cuales existe una comparación exhaustiva en [4].
- Se puede elaborar un análisis más exhaustivo para determinar los parámetros de los algoritmos propuestos, ya que éstos fueron elegidos de forma empírica con solo el grupo de imágenes usado en nuestras pruebas.
- Consideramos que es imprescindible realizar pruebas con datos reales bajo tareas de seguimiento visual [4], con la finalidad de evaluar el desempeño de los algoritmos propuestos en esta tesis en dichas tareas.

La entrada de datos a nuestros algoritmos son dos conjuntos de correspondencias de puntos, esto es, este problema de encontrar las correspondencias de puntos ya debe estar resuelto. Sin embargo, para la detección automática de homografías ambos problemas deben resolverse al mismo tiempo. Consideramos que en este problema deberían orientarse los esfuerzos futuros.

Como resultado de este trabajo de tesis se publicó el artículo [27].

# Bibliografía

---

- [1] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, March 2004.
- [2] R. Xu and J. Jin. Camera Control and Multimedia Interaction using Individual Object Recognition. *Int. J. Comput. Multimedia*, 2(3):77–85, June 2007.
- [3] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [4] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. *Int. J. Comput. Vision*, 94(3):335–360, September 2011.
- [5] Z. Zhang. A Flexible New Technique for Camera Calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, November 2000.
- [6] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley; Sons, Inc., New York, NY, USA, 2003.
- [7] 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.
- [8] C. Cruz, L.G. de la Fraga, and O. Schütze. Fitness Function Evaluation for the Detection of Multiple Ellipses Using a Genetic Algorithm. In *Electrical Engineering Computing Science and Automatic Control (CCE), 2011 8th International Conference on*, pages 1–6, October 2011.
- [9] K.V. Price. New ideas in Optimization. chapter An Introduction to Differential Evolution, pages 79–108. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.
- [10] L.G. de la Fraga and G. López. Robust Detection of Several Circles or Ellipses with Heuristics. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 484–490, june 2011.

- [11] L.G. de la Fraga and G. López. Robust Fitting of Ellipses with Heuristics. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, july 2010.
- [12] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [13] C.A. Coello. Introducción a los Algoritmos Genéticos. *Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios*, (17):5–11, 1995.
- [14] L.G. de la Fraga and C.A. Coello. Pattern Recognition, Machine Intelligence and Biometrics. chapter A Review of Applications of Evolutionary Algorithms in Pattern Recognition, pages 3–28. Higher Education Press, Beijing and Springer-Verlag, 2011.
- [15] J.H. Holland. Outline for a Logical Theory of Adaptive Systems. *J. ACM*, 9(3):297–314, July 1962.
- [16] D.E. Goldberg and K. Deb. Foundations of Genetic Algorithms. chapter A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, pages 69–93. Morgan Kaufmann, 1991.
- [17] R. Storn and K. Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Tech. Rep. TR-95-012, International Computer Science Institute, march 1995.
- [18] L.G. de la Fraga and O. Schütze. Direct Calibration by Fitting of Cuboids to a Single Image Using Differential Evolution. *International Journal of Computer Vision*, 81:119–127, 2009. 10.1007/s11263-008-0183-z.
- [19] L.G. de la Fraga. Self-calibration from Planes Using Differential Evolution. In Eduardo Bayro-Corrochano and Jan-Olof Eklundh, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 5856 of *Lecture Notes in Computer Science*, pages 724–731. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-10268-4\_85.
- [20] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997. 10.1023/A:1008202821328.
- [21] K. Zielinski and R. Laur. Stopping criteria for differential evolution in Constrained Single-Objective Optimization. In *Advances in Differential Evolution*. Springer, 2008.
- [22] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly, Cambridge, MA, 2008.
- [23] B. Gough. *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3rd edition, 2009.



- [24] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75:245–249, 2009.
- [25] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The Planar k-Means Problem is NP-Hard. *Lecture Notes in Computer Science*, 5431:274–285, 2009.
- [26] H. Bay, A. Ess, T. Tuytelaars, and L. Van. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.
- [27] L.G. de la Fraga and L.A. Camacho. Robust Detection of Homographies With a Genetic Algorithm. In *EVOLVE 2012*, august 2012. Proceedings published in CDROM.



# Páginas Web

---

Enunciamos aquí los URL's consultados.

- [1] *Disponibilidad de la Información del Trabajo de Tesis*, 2012, Disponible en:  
<https://computacion.cs.cinvestav.mx/~lcamacho/>
- [2] *Ejemplos de Perspectiva de Un Punto*, 2012, Disponible en:  
[http://en.wikipedia.org/wiki/Perspective\\_\(graphical\)](http://en.wikipedia.org/wiki/Perspective_(graphical))
- [3] *Webcam USB*, 2012, Disponible en:  
<http://www.steren.com.mx/catalogo/prod.asp?f=0&sf=0&c=444&p=104406&desc=webcam-usb-con-microfono--de-2-megapixeles>