

# RESOLUCIÓN DE TRES PROBLEMAS DE VISIÓN CON ALGORITMOS EVOLUTIVOS

Dr. Luis Gerardo de la Fraga

E-mail: [fraga@cs.cinvestav.mx](mailto:fraga@cs.cinvestav.mx)  
Departamento Computación  
Cinvestav

8 de marzo, 2007

# CONTENIDO

## ANTECEDENTES

# CONTENIDO

## ANTECEDENTES

## AJUSTE DE UN CONJUNTO DE PUNTOS A UNA ELIPSE

# CONTENIDO

ANTECEDENTES

AJUSTE DE UN CONJUNTO DE PUNTOS A UNA ELIPSE

TRIANGULACIÓN

# CONTENIDO

ANTECEDENTES

AJUSTE DE UN CONJUNTO DE PUNTOS A UNA ELIPSE

TRIANGULACIÓN

CALIBRACIÓN DE LA CÁMARA

# CONTENIDO

ANTECEDENTES

AJUSTE DE UN CONJUNTO DE PUNTOS A UNA ELIPSE

TRIANGULACIÓN

CALIBRACIÓN DE LA CÁMARA

CONCLUSIONES

# ANTECEDENTES

- ▶ Visión por computadora
- ▶ Optimización: ajuste de puntos a una recta
- ▶ Métodos numéricos: Newton
- ▶ Computación evolutiva
- ▶ Ruido

# EL ÁREA DE VISIÓN POR COMPUTADORA <sup>a</sup>

<sup>a</sup>Foley, van Dam, Feiner, Hughes, Computer Graphics: principles and practice, 2000, Addison Wesley.

1. **Graficación:** Trata la síntesis pictórica de objetos reales o imaginarios a partir de sus modelos computacionales.
2. **Procesamiento de Imagen:** Trata el análisis de escenas o la reconstrucción 2D ó 3D de objetos a partir de sus fotografías.
  - 2.1 **Realzado de imagen**
  - 2.2 **Detección y reconocimiento de patrones**
  - 2.3 **Análisis de escenas**
  - 2.4 **Visión por computadora:** Reconstrucción de un modelo 3D de una escena a partir de varias imágenes 2D.



# OPTIMIZACIÓN

Este problema puede representarse de la siguiente forma:

**DADA:** una función  $f : A \rightarrow \mathbf{R}$

**ENCONTRAR:** un elemento  $x_0$  en  $A$  tal que

$f(x_0) \leq f(x)$  para todo  $x$  en  $A$  (minimización), o

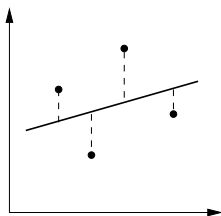
$f(x_0) \geq f(x)$  para todo  $x$  en  $A$  (maximización)

- ▶ Típicamente,  $A$  es un subconjunto del espacio euclidiano  $\mathbf{R}^n$ , especificado por un conjunto de restricciones.
- ▶ Los elementos de  $A$  se les llama *soluciones factibles* o *soluciones candidatas*.
- ▶ La función  $f$  se le llama *función objetivo* o *función costo*.
- ▶ Una solución factible que minimize la función objetivo se le llama *solución óptima*.
- ▶ El dominio de  $A$  es el espacio de búsqueda.

# AJUSTE A UNA RECTA POR MÍNIMOS CUADRADOS

La suma de las distancias cuadráticas verticales,  $R^2$  es

$$R^2 = \sum [y_i - f(x_i, a_1, a_2, \dots, a_n)]^2$$



El uso de las distancias absolutas resulta en derivadas discontinuas que no pueden tratarse analíticamente.

La condición para que  $R^2$  sea un mínimo es

$$\frac{\partial(R^2)}{\partial a_i} = 0$$

para  $i = 1, \dots, n$ . Para un ajuste lineal

$$f(a, b) = a + bx$$

de forma que

$$R^2(a, b) = \sum_{i=1}^n [y_i - (a + bx_i)]^2$$

$$\frac{\partial(R^2)}{\partial a} = -2 \sum_{i=1}^n [y_i - (a + bx_i)] = 0$$

$$\frac{\partial(R^2)}{\partial b} = -2 \sum_{i=1}^n [y_i - (a + bx_i)]x_i = 0$$

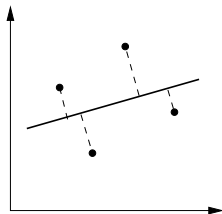
que resulta en el sistema de ecuaciones

$$na + b \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

(Tomado de:  
www.mathworld.com)

# AJUSTE A UNA RECTA SEGÚN LA DISTANCIA PERPENDICULAR



La distancia perpendicular de un punto a una línea  $y = a + bx$  al punto  $(x_i, y_i)$  está dada por:

$$d_i = \frac{|y_i - (a + bx_i)|}{1 - b^2}$$

la función a ser minimizada es:

$$R = \sum_{i=1}^n \frac{|y_i - (a + bx_i)|}{1 - b^2}$$

Sin embargo, como la función valor absoluto no tiene derivadas continuas, minimizar  $R$  no se puede manejar con una solución analítica.

(Tomado de: [www.mathworld.com](http://www.mathworld.com))

# MÉTODOS NUMÉRICOS: NEWTON (1/2)

El método de Newton es un algoritmo para hallar raíces; usa los primeros términos de la expansión en series de Taylor de la función en la vecindad de una raíz supuesta.

La serie de Taylor de  $f$  en el punto  $x = x_0 + \epsilon$  está dado por.

$$f(x_0 + \epsilon) = f(x_0) + f'(x_0)\epsilon + \frac{1}{2}f''(x_0)\epsilon^2 + \dots$$

Tomando el término de primer orden, haciendo  $f(x_0 + \epsilon) = 0$  y resolviendo para  $\epsilon = \epsilon_0$  resulta:

$$\epsilon_0 = -\frac{f(x_0)}{f'(x_0)}$$

la cual es el ajuste de primer order para la posición de la raíz. Haciendo  $x_1 = x_0 + \epsilon_0$ , se calcula una nueva  $\epsilon_1$ , y se repite el proceso hasta que converja hacia una raíz usando

$$\epsilon_n = -\frac{f(x_n)}{f'(x_n)}$$

## MÉTODOS NUMÉRICOS: NEWTON (2/2)

El método de Newton también se puede usar para encontrar el máximo local o mínimo local de funciones: si un número real  $x^*$  es un punto estacionario de una función  $f(x)$ , entonces  $x^*$  es una raíz de la derivada  $f'(x)$ , y por lo tanto se puede resolver para  $x^*$  aplicando el método de Newton a  $f'(x)$ . Entonces, suponiendo que  $f(x)$  es una función doblemente diferenciable y se tiene  $x_0$  lo suficientemente cerca de  $x^*$ , la secuencia definida por

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n), \quad n \geq 0$$

convergerá hacia  $x^*$

# COMPUTACIÓN EVOLUTIVA

- ▶ Son heurísticas inspiradas en los principios de la teoría Neo-darwiniana de la evolución natural.
- ▶ Es utilizada principalmente para la optimización.
- ▶ Sus características principales:
  - ▶ Una población de individuos.
  - ▶ Operadores que actúen sobre la población.
  - ▶ Proceso generacional.
  - ▶ Función de aptitud.
  - ▶ Mecanismos de selección.

# ESTRUCTURAL GENERAL DE UN AG

1. Crear aleatoriamente una población inicial de tamaño  $N$
2. Repetir
  - 2.1 Calcular la aptitud de los individuos
  - 2.2 Seleccionar a los  $N$  padres a reproducirse
  - 2.3 Con probabilidad  $P_c$ , aplicar el operador de cruce a los  $N$  padres
  - 2.4 Aplicar el operador de mutación a los hijos con probabilidad  $P_m$
  - 2.5 Los hijos formarán parte de la siguiente generación de la población
  - 2.6 Aplicar el mecanismo de elitismo para quedarse con el mejor individuo
  - 2.7 Se termina hasta alcanzar un número de generaciones establecido.
3. Presentar el individuo con mejor aptitud



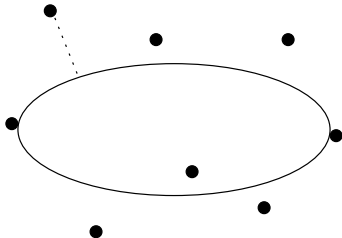
## DIFERENTES TÉCNICAS

- ▶ Gradiente descendente
- ▶ Método amoeba
- ▶ Simplex
- ▶ Método elipsoide
- ▶ Método de Newton
- ▶ Métodos cuasi-Newton
- ▶ Métodos de puntos interiores
- ▶ Métodos de gradiente conjugado
- ▶ Escalando la colina
- ▶ Recocido simulado
- ▶ Búsqueda tabú
- ▶ Algoritmos genéticos
- ▶ Optimización por colonia de hormigas
- ▶ Estrategias evolutivas
- ▶ Evolución diferencial
- ▶ Optimización por cúmulo de partículas

## RUIDO

- ▶ Discrepancia entre la posición de los puntos dados y los os de la imagen o reales.
- ▶ Causas: sensibilidad del dispositivo, variaciones en el ambiente, falta de precisión (error de cuantificación) en la detección y elección los puntos.
- ▶ Para simularlo se añade ruido gaussiano a la posición de los puntos.
- ▶ Punto con ruido  $\mathbf{x}_r = (x_r, y_r) = (x_i + \frac{N(0,k)}{\sqrt{2}}, y_i + \frac{N(0,k)}{\sqrt{2}})$ , donde  $(x_i, y_i)$  es la posición del punto en la imagen y  $N(0, k)$  es el ruido gaussiano con media cero y varianza  $k$ .
- ▶ Permite probar el desempeño y robustez de un algoritmo.

# AJUSTE A UNA CÓNICA (1/3)



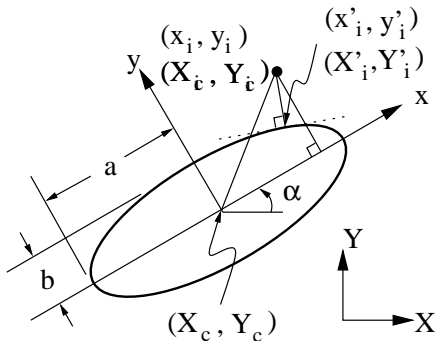
## AJUSTE A UNA CÓNICA (2/3)

El problema de ajuste se transforma a un problema de optimización: el algoritmo genético se ha usado para minimizar

$$e = \sum_{i=0}^N D_i$$
$$D_i = \|\mathbf{X}_i - \mathbf{X}'_i\|,$$

para un conjunto de  $N$  puntos  $\mathbf{X}_i$  dados,  $\mathbf{X}'_i$  representa el punto más cercano a la cónica, ortogonal a ésta.

## AJUSTE A UNA CÓNICA (3/3)



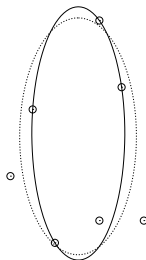
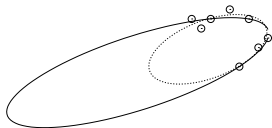
## MÉTODO <sup>1</sup>

1. Se inicializa con una elipse ajustada algebraicamente o con un círculo ajustado a los puntos dados.
2. Se hallan los puntos  $\mathbf{X}'_i$ , los puntos más cercanos a la cónica, ortogonal a ésta, con el método de Newton.
3. Con el método de Gauss-Newton se minimiza la suma de las distancias cuadráticas.

---

<sup>1</sup>S.J. Ahn, W. Rauth and HJ. Warnecke, Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola, *Pattern Recognition* 34, (12) 2001, 2283-2303

# RESULTADOS



## ELIPSE

- ▶ Encontrar el centro de la elipse  $\mathbf{x}_c = [x_c, y_c]^T$ , el eje mayor  $a$ , el eje menor  $b$ , y el grado de inclinación  $\alpha$ .
- ▶ Sin restricciones.
- ▶ Con restricciones  $\alpha = \pi/2$  y  $ab = 20$ .

$a$	$b$	$x_c$	$y_c$	$\alpha$	$\sum d$	$\sum d^2$
4.58061	4.36740	6.17691	4.46768	1.89805	49.79220	358.04831

CUADRO: Resultados obtenidos por el algoritmo MCDO [1].



## ELIPSE

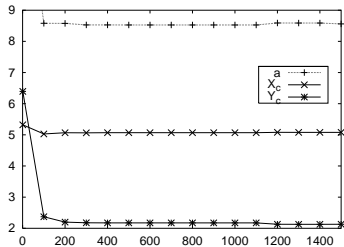
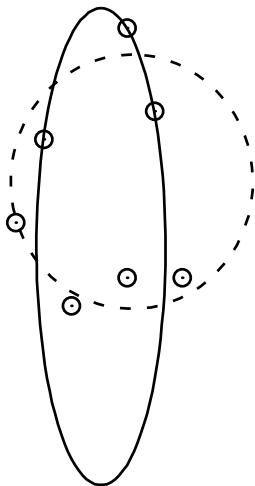
- ▶ Algoritmos genéticos (GA), variantes:
  - ▶ T. selección: torneo binario, ruleta, sobrante estocástico.
  - ▶ Población: 50, 100, 200 individuos.
  - ▶ P.M.: 0.1,  $1/L$ ,  $2/L$ .
  - ▶ T. cruza: un punto, dos puntos.
- ▶ Aplicación sobre todas las cónicas.
- ▶ Se realizó un análisis de varianza (ANOVA).

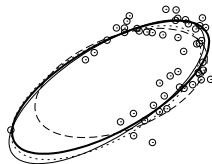
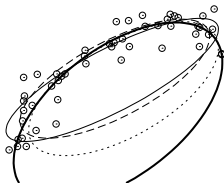
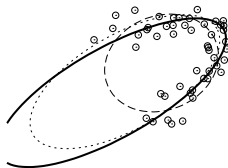
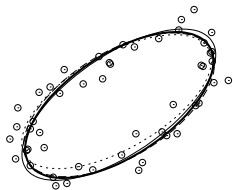
## ELIPSE

- ▶ Algoritmo genético. (50 ejecuciones, 100 individuos, 3000 generaciones, sobrante estocástico, p.c. 0.8, p.m.  $1/L$ , cruce de un punto, binaria).
- ▶ Valor de las restricciones  $\alpha = 1,57079$  y  $ab = 20$ .

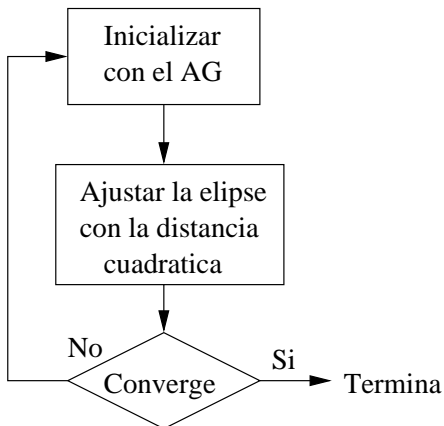
	$a$	$x_c$	$y_c$	$\sum d$
Mejor	8.56997	5.08115	2.12223	3.95637
Promedio	8.57041	5.08132	2.12165	3.95674
Peor	8.56143	5.07802	2.13356	3.96167
d.e.	0.00043	0.00047	0.00168	0.00077
i.c.	[8.56,8.57]	[5.08,5.08]	[2.12,2.12]	[3.95,3.95]
	$a$	$x_c$	$y_c$	$\sum d^2$
Mejor	9.53247	4.98852	2.74635	3.98204
Promedio	9.53247	4.98852	2.74635	3.98204
Peor	9.53246	4.98852	2.74636	3.98204
d.e.	0.0	0.0	0.0	0.0
i.c.	[9.53,9.53]	[4.98,4.98]	[2.74,2.74]	[3.98,3.98]

$a$	$b$	$x_c$	$y_c$	$\alpha$	$\sum d$	$\sum d^2$
4.58061	4.36740	6.17691	4.46768	1.89805	49.79220	358.04831





## APLICACIÓN: INICIALIZAR CON EL AG

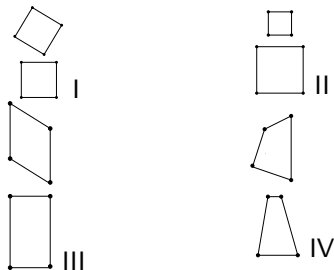


## TRANSFORMACIONES

Transformación proyectiva se representa por una matriz no singular.

$$\mathbf{x}' = H\mathbf{x}$$

- ▶ I. Isometrías.  $H = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$
- ▶ II. Similitud.  $H = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & s \end{bmatrix}$
- ▶ III. Afin.  $H = \begin{bmatrix} A_{(n-1) \times (n-1)} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$
- ▶ IV. Proyectiva.  $H = [A_{n \times n}]$



# PARÁMETROS INTRÍNSECOS Y EXTRÍNSECOS DE LA CÁMARA

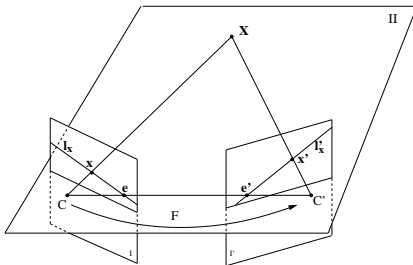
$$\lambda_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

$$P = A[R|T]$$

$$A = \begin{pmatrix} f_x & \alpha & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \quad R|T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

## GEOMETRÍA EPIPOLAR

- La geometría epipolar nos permite relacionar puntos de dos imágenes de una misma escena.





## MATRIZ FUNDAMENTAL

- ▶ Matriz  $F_{3 \times 3}$  que relaciona el punto  $\mathbf{x}$  en la imagen  $I$  con el punto  $\mathbf{x}'$  en la imagen  $I'$ .

$$\mathbf{x}'^T F \mathbf{x} = 0 \quad (1)$$

- ▶ Al aplicar  $F$  a un punto  $\mathbf{x}$ , se convierte en una línea epipolar  $\mathbf{l}'_x$  en la segunda imagen.

$$\mathbf{l}'_x = F \mathbf{x} \quad (2)$$

- ▶ La línea epipolar  $\mathbf{l}'_x$  donde debe yacer el punto  $\mathbf{x}'$ .

$$\mathbf{x}'^T \mathbf{l}'_x = 0 \quad (3)$$

## PROBLEMA DE TRIANGULACIÓN (1/2)

Si se tiene un punto  $\mathbf{X}$  en el espacio 3D y dos matrices de proyección,  $P$  y  $P'$ , entonces se pueden obtener dos puntos diferentes,  $\mathbf{x}$  y  $\mathbf{x}'$ , en dos dimensiones, esto es:

$$\mathbf{x} = P\mathbf{X}, \text{ y}$$

$$\mathbf{x}' = P'\mathbf{X}$$

Cuando solamente se cuenta con los puntos 2D  $\mathbf{x}$  y  $\mathbf{x}'$ , se puede usar la geometría epipolar para recuperar la matrix fundamental  $F$ , A partir de  $F$ , se puede realizar una reconstrucción proyectiva, esto es, es posible obtener  $P$  y  $P'$ .

## APLICACIÓN DE ALGORITMOS EVOLUTIVOS

- ▶ Triangulación como un problema mono-objetivo. Minimizar la distancia Euclidiana:

$$f(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}'), \quad (4)$$

- ▶ Algoritmos genéticos.
- ▶ Cúmulo de partículas.

## PRUEBAS Y RESULTADOS

- ▶ Bajo la reconstrucción proyectiva.
- ▶ Se tomaron 8,12,16,20 puntos de un conjunto de 24 correspondencias de puntos de un poliedro.
- ▶ Se les añadió ruido RMS gaussiano de 1 a 8 píxeles.
- ▶ Las matrices de proyección se calcularon a partir de la matriz fundamental  $F$  estimada de los puntos con ruido.
- ▶ Error calculado mediante la distancia Euclidiana.
- ▶ Algoritmos genéticos (GA), variantes:
  - ▶ T. selección: torneo binario, ruleta, sobrante estocástico.
  - ▶ Población: 50, 100, 200 individuos.
  - ▶ P.M.: 0.1,  $1/L$ ,  $2/L$ .
  - ▶ T. cruza: un punto, dos puntos.



## APLICACIÓN DE ALGORITMOS EVOLUTIVOS

Debido a los resultados desfavorables, se adoptó la propuesta de Knowles: transformar a un problema multi-objetivo.

- ▶ Problema multi-objetivo. Minimizar la distancia Euclidiana:

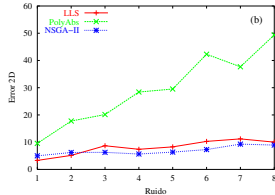
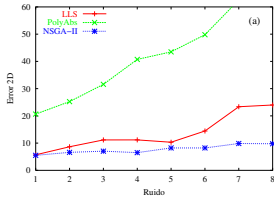
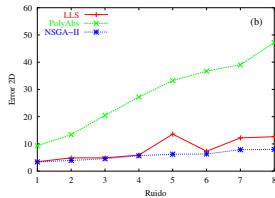
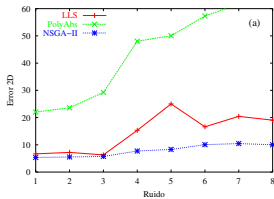
$$f_1(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}}) \quad (5)$$

$$f_2(\mathbf{X}) = d(\mathbf{x}', \hat{\mathbf{x}}') \quad (6)$$

- ▶ Micro GA y el NSGA-II (más robusto).
- ▶ Del conjunto de Pareto se eligió, la solución mejor nivelada en los dos objetivos.

## PRUEBAS Y RESULTADOS

- ▶ Bajo la reconstrucción proyectiva y la reconstrucción afín.
- ▶ Se tomaron 8,12,16,20 puntos de un conjunto de 24 correspondencias de puntos de un poliedro.
- ▶ Se les añadió ruido gaussiano de 1 a 8 píxeles.
- ▶ Las matrices de proyección se calcularon a partir de la matriz fundamental  $F$  estimada de los puntos con ruido.
- ▶ Error calculado mediante la distancia Euclidiana.
- ▶ NSGA-II:
  - ▶ Generaciones: 300, P.C.: 0.9, P.M.: 0.33, población: 100.



Reconstrucción proyectiva (a) 8 y (b) 12 puntos.

Reconstrucción afín (a) 8 y (b) 12 puntos.



## PLANTEAMIENTO DEL PROBLEMA

- ▶ Estimar los parámetros intrínsecos ( $K$ ) y extrínsecos ( $R, \mathbf{t}$ ), para cada una de las cámaras.

$$M = K[R|\mathbf{t}] \quad M' = K'[R'|\mathbf{t}'] \quad (7)$$

## TIPOS DE CALIBRACIÓN DE LA CÁMARA

- ▶ Basados en las ecuaciones de Kruppa y formulaciones derivadas de estas ecuaciones [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].
- ▶ Basados en el conocimiento de la escena (paralelismo, ortogonalidad, escenarios con planos) [12, 13, 14, 15, 16, 17, 18].
- ▶ Basados en el conocimiento de cómo se ha desplazado la cámara a través del tiempo [19, 20].

$$K = \begin{bmatrix} f_x & \alpha & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow K = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}$$

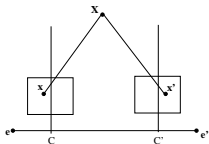
## CALIBRACIÓN DE LA CÁMARA

- ▶ Aplicar las matrices de calibración sobre la matriz fundamental.

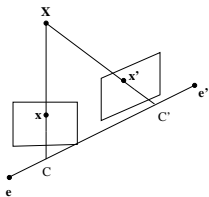
$$E = K'^T FK \quad (8)$$

- ▶ Se genera la matriz esencial.
- ▶ Matriz, mediante la cual se generan las matrices de proyección.
- ▶ Con estas matrices, se obtiene una reconstrucción métrica.

## DEGENERACIÓN I



## DEGENERACIÓN II



## TRABAJO RELACIONADO I

- ▶ P. Sturm y Z.L. Cheng en 2005 [21] presentan un método de calibración donde encuentran el foco a partir de dos imágenes con cámaras no calibradas.
- ▶ Basado en la matriz fundamental (geometría epipolar).
- ▶ Incorpora soluciones posibles al momento de identificar que el foco  $f$  no es correcto.
- ▶ Obtiene soluciones aceptables cerca de la degeneración.
- ▶ Degeneración cuando los ejes ópticos son paralelos.

## TRABAJO RELACIONADO II

- ▶ K. Kanatani y A. Nakatsuji en 2006 [22] presentan un método capaz de estabilizar el foco a partir de dos imágenes con cámaras no calibradas.
- ▶ Basado en la matriz fundamental (geometría epipolar).
- ▶ Utilizan un cambio de variable sobre los focos de ambas cámaras.
- ▶ Itera eliminando puntos y vuelve a calcular la matriz fundamental.
- ▶ Obtiene soluciones aceptables cerca de la degeneración.

## PROPUESTA

- ▶ Evitar el cálculo de la matriz fundamental.
- ▶ Estimar directamente los parámetros intrínsecos ( $K, K'$ ) y extrínsecos ( $R, \mathbf{t}, R', \mathbf{t}'$ ).
- ▶ Asumir la misma cámara en ambas imágenes y,

$$K = K' = \begin{pmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{pmatrix}$$

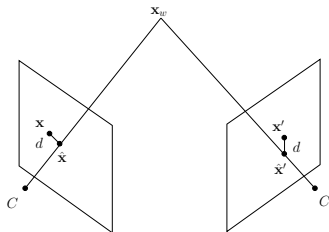
- ▶ 12 variables parámetros extrínsecos (PE) y 1 variable parámetros intrínsecos (PI).

## ERROR DE REPROYECCIÓN

- ▶ Evitar caer en degeneraciones.
- ▶ Usar error de reproyección basado en la distancia Euclidiana.

$$e_{rep} = d(\mathbf{x}, \hat{\mathbf{x}}) + d(\mathbf{x}', \hat{\mathbf{x}}') \quad (9)$$

$$e_{rep} = d(\mathbf{x}, M\hat{\mathbf{X}}) + d(\mathbf{x}', M'\hat{\mathbf{X}})$$





## EVOLUCIÓN DIFERENCIAL

- ▶ Heurística de búsqueda directa. Optimización de variables reales.
- ▶ Se ha demostrado que trabaja mejor que los demás algoritmos evolutivos en espacios continuos.
- ▶ Simple de programar y muy eficiente.
- ▶ Basados en un operador de diferencia (simula el gradiente en esa zona).
- ▶ La mejor variante "*rand/1/bin*".

```

1: Iniciar aleatoriamente a cada individuo  $\mathbf{l}_k^0 \forall k = 1, 2, \dots, p.$ 
2: Evaluar cada individuo  $f(\mathbf{l}_k^g) \forall k = 1, 2, \dots, p.$ 
3: for  $1 \leq g \leq \text{MAX\_GEN}$  do
4:   for  $1 \leq k \leq p$  do
5:     Seleccionar tres individuos aleatoriamente:  $\xi_1^g \neq \xi_2^g \neq \xi_3^g$ 
6:     Seleccionar una posición aleatoria:  $j_{\text{rand}} = \text{randint}(1, d)$ 
7:     for  $1 \leq j \leq d$  do
8:       if  $\text{rand}(0, 1) < C_r$  or  $j = j_{\text{rand}}$  then
9:          $\rho_{k,j}^g = \xi_{3,j}^g + f_d(\xi_{1,j}^g - \xi_{2,j}^g)$ 
10:      else
11:         $\rho_{k,j}^g = i_{k,j}^g$ 
12:      end if
13:    end for
14:    if  $f(\mathbf{r}_k^g) < f(\mathbf{l}_k^g)$  then
15:       $\mathbf{l}_k^{g+1} = \mathbf{r}_k^g$ 
16:    else
17:       $\mathbf{l}_k^{g+1} = \mathbf{l}_k^g$ 
18:    end if
19:  end for
20: end for

```

## EVALUACIÓN DE UN INDIVIDUO

**Require:** Un individuo  $\mathbf{i}_k^g$ .

**Ensure:** El valor de la función objetivo  $f$  del individuo dado.

- 1: Obtener los PE de la primera cámara  $(\theta, \beta, \gamma)$  y  $(t_x, t_y, t_z)$  de  $\mathbf{i}_k^g$ .
- 2: Obtener los PE de la segunda cámara  $(\theta', \beta', \gamma')$  y  $(t'_x, t'_y, t'_z)$  de  $\mathbf{i}_k^g$ .
- 3: Obtener los PI de ambas cámaras  $f$  de  $\mathbf{i}_k^g$ .
- 4: Generar las dos matrices de proyección  $M = K[R|t]$ ,  $M' = K[R'|t']$ .
- 5: **for**  $j = 1$  hasta  $n$  **do**
- 6: Estimar el punto tridimensional  $\mathbf{X}_j$  mediante la triangulación usando el algoritmo LLS.  
 $\mathbf{x} = M\mathbf{X}$   
 $\mathbf{x}' = M'\mathbf{X}$
- 7: Calcular los puntos estimados  $(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}'_j)$ .
- 8: Calcular el error de reproyección.

$$d(\mathbf{x}_j, \hat{\mathbf{x}}_j) + d(\mathbf{x}'_j, \hat{\mathbf{x}}'_j) \quad (10)$$

9: **end for**

10: Regresar la suma del error de reproyección.

## PRUEBAS Y RESULTADOS

- ▶ Primera prueba, observar el desempeño y convergencia del algoritmo propuesto (30 corridas).
- ▶ Imágenes sintéticas generadas a partir de una esfera.
- ▶ Sin ruido y con 1, 2 y 3 píxeles de ruido gaussiano.
- ▶ Parámetros:
  - ▶ Tamaño de Población: 30
  - ▶ Generaciones: 5000
  - ▶ Porcentaje de recombinación: 0.8
  - ▶ Constante de diferencia: 0.8
  - ▶ Número de variables: 13

- ▶ El error reportado es el error residual

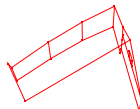
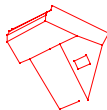
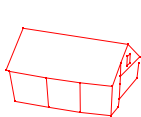
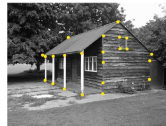
$$e_r = \frac{1}{\sqrt{4n}} \left( \sum_{i=1}^n d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + \sum_{i=1}^n d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right)^{1/2}.$$

## Estadísticas del error residual:

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

## Pruebas sobre imágenes reales.

Mejor	Media	Peor	d.e.	i.c.
0.220	0.224	0.238	0.004	[0.223,0.226]



## PRUEBAS SOBRE LAS DEGENERACIONES

- ▶ Degeneración I.
  - ▶ Primera cámara se tiene:
    - ▶  $\theta_x = 0, \theta_y = 0, \theta_z = 0.$
    - ▶  $t_x = 5, t_y = 7, t_z = 20.$
  - ▶ Para la segunda cámara se tiene:
    - ▶  $\theta_x = 0, \theta_y = 0, \theta_z = 0.$
    - ▶  $t_x = 35, t_y = 7, t_z = 20.$

Puntos	Ruido	Mejor	Media	Peor	d.e.	Media i.c.
20	0	0.0	0.001275	0.020333	0.004042	[0.000169,0.011274]

## PRUEBAS SOBRE LAS DEGENERACIONES

### ► Degeneración II.

#### ► Primera cámara se tiene:

►  $\theta_x = 0, \theta_y = 0, \theta_z = 0.$

►  $t_x = 0, t_y = 0, t_z = 20.$

#### ► Para la segunda cámara se tiene:

►  $\theta_x = 0, \theta_y = 40, \theta_z = 0.$

►  $t_x = 0, t_y = 0, t_z = 20.$

Puntos	Ruido	Mejor	Media	Peor	d.e.	Media i.c.
20	0	0.0	0.010720	0.073994	0.016002	[0.006035,0.022030]



## CONCLUSIONES

1. Usar algoritmos evolutivos para resolver problemas de visión nos permite:
  - 1.1 No se requiere obtener la derivada
  - 1.2 No se requiere un punto inicial
  - 1.3 Se pueden usar otras medidas de distancia diferentes que la distancia cuadrática
  - 1.4 Se puede usar el resultado de una corrida para inicializar un método numérico convencional
2. Sin embargo se tienen las siguientes desventajas:
  - 2.1 El tiempo de ejecución es alto
  - 2.2 Da un resultado de naturaleza estocástica
3. En los experimentos que hemos realizado, usar ED reduce en un orden de magnitud el número de evaluaciones de la función objetivo, en comparación a un algoritmo genético con representación en código grey.

## PUBLICACIONES

- ▶ Israel Vite Silva and Luis Gerardo de la Fraga. Three-dimensional reconstruction of textured parallelepipeds from digital video. In Alexander Gelbukh, Cornelio Yañez, and Oscar Camacho, editors, *Advances in Artificial Intelligence and Computer Science*, pages 107-115. Research on Computing Science, IPN, 2005. ISSN: 1665-9899.
- ▶ Israel Vite Silva, Luis Gerardo de la Fraga y Nareli Cruz Cortés. Optimal Triangulation in 3D Computer Vision Using a Multi-Objective Evolutionary Algorithm. M. Giacobini et al. (Eds.): *EvoWorkshops 2007 (EvoASP)*, LNCS 4448. pp. 330-339. Springer-Verlag, Valencia España.
- ▶ Luis Gerardo de la Fraga, Israel Vite Silva y Nareli Cruz Cortés. Euclidean Distance Fit of Ellipses with a Genetic Algorithm. M. Giacobini et al. (Eds.): *EvoWorkshops 2007 (EvoASP)*, LNCS 4448. pp. 359-366. Springer-Verlag, Valencia España.
- ▶ Israel Vite Silva and Luis Gerardo de la Fraga. Direct 3D Metric Reconstruction from Two Views Using Differential Evolution. Artículo en desarrollo.

# ¡GRACIAS!

- ▶ Página del Departamento de Computación:  
<http://www.cs.cinvestav.mx>
- ▶ Mi página personal:  
<http://delta.cs.cinvestav.mx/~fraga>
- ▶ Mi correo electrónico: [fraga@cs.cinvestav.mx](mailto:fraga@cs.cinvestav.mx)

# BIBLIOGRAFÍA I



S.J. Ahn, W. Rauh, and H.J. Warnecke.

Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola.  
*Pattern Recognition*, 34(12):2283–2303, December 2001.



S. J. Maybank and O. D. Faugeras.

A theory of self-calibration of a moving camera.  
*Int. J. Comput. Vision*, 8(2):123–151, 1992.



Q.-T. Luong and O. D. Faugeras.

The fundamental matrix: theory, algorithms, and stability analysis.  
*Int. J. Comput. Vision*, 17(1):43–75, 1996.



Cyril Zeller and Olivier Faugeras.

Camera self-calibration from video sequences: the kruppa equations revisited.  
Technical Report RR-2793, INRIA, France, 1996.



Q.-T. Luong and O. D. Faugeras.

Self-calibration of a moving camera from point correspondences and fundamental matrices.  
*Int. J. Comput. Vision*, 22(3):261–289, 1997.



Y. Ma, R. Vidal, J. Kosecka, and S. Sastry.

Kruppa equation revisited: its renormalization and degeneracy.  
In *ECCV (2)*, pages 561–577, 2000.

## BIBLIOGRAFÍA II



A. Heyden and K. Astrom.

Euclidean reconstruction from constant intrinsic parameters.

In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, page 339, Washington, DC, USA, 1996. IEEE Computer Society.



S. Bognoux.

From projective to euclidean space under any practical situation, a criticism of self-calibration.

In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 790, Washington, DC, USA, 1998. IEEE Computer Society.



P. Sturm.

A case against kruppa's equations for camera self-calibration.

*IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1199–1204, 2000.



Y. Ma, J. Kořecká, and S. Sastry.

Optimization criteria and geometric algorithms for motion and structure estimation.

*Int. J. Comput. Vision*, 44(3):219–249, 2001.



R. I. Hartley and Silpa-Anan.

Reconstruction from two views using approximate calibration.

In *Proceedings 5th Asian Conf. Computer Vision*, volume 1, pages 338–343, Melbourne, Australia, 2002.



Proc. IEEE Conf. Computer Vision and Pattern Recognition.

*An efficient and accurate camera calibration technique for 3D Machine Vision*, 1986.

## BIBLIOGRAFÍA III



B. Caprile and V. Torre.

Using vanishing points for camera calibration.

*International Journal of Computer Vision*, 4:127–140, 1990.



B. Triggs.

Autocalibration from planar scenes.

*Lecture Notes in Computer Science*, 1406:89, 1998.



R. Cipolla, T. Drummond, and D. Robertson.

Calibration from vanishing points in image of architectural scenes.

In *The 10th British Machine vision conference*, 1999.



Z. Zhang.

A flexible new technique for camera calibration.

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.



D. Svedberg and S. Carlsson.

Calibration, pose and novel views from single images of constrained scenes.

*Pattern Recogn. Lett.*, 21(13-14):1125–1133, 2000.



J. Kosecka and W. Zhang.

Video compass.

In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 476–490, London, UK, 2002. Springer-Verlag.

# BIBLIOGRAFÍA IV



R. Enciso and T. Vieville.

Experimental self-calibration from four views.

*In ICIAP '95: Proceedings of the 8th International Conference on Image Analysis and Processing*, pages 307–312, London, UK, 1995. Springer-Verlag.



M. Pollefeys, R. Koch, and L. V. Gool.

Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters.

*Int. J. Comput. Vision*, 32(1):7–25, 1999.



P. Sturm, Z. Cheng, P. C. Chao, and A.Ñeow Poo.

Focal length calibration from two views: method and analysis of singular cases.

*Computer Vision and Image Understanding*, 99(1):58–95, jul 2005.



K. Kanatani, A.Ñakatsuji, and Y. Sugaya.

Stabilizing the focal length computation for 3-D reconstruction from two uncalibrated views.

*Int. J. Comput. Vision*, 66(2):109–122, 2006.