

VISIÓN POR COMPUTADORA EVOLUTIVA

Dr. Luis Gerardo de la Fraga

E-mail: fraga@cs.cinvestav.mx
Cinvestav
Departamento de Computación

29 de agosto, 2008

1. Introducción a visión
2. Reconstrucción usando cuboides
3. Reconstrucción usando planos (homografías)
4. Reconstrucción usando dos imágenes y el algoritmo de la Evolución Diferencial
5. Reconstrucción usando varias imágenes
6. Triangulación usando algoritmos evolutivos
7. Conclusiones

La obtención de una reconstrucción tridimensional a partir de imágenes bidimensionales es un problema importante en diferentes campos, tales como: biología, medicina, microscopía electrónica, topografía, diseño asistido por computadora, simulación por computadora, visualización científica, robótica, etc.

Las técnicas para obtener una reconstrucción 3D pueden dividirse en:

1. Reconstrucción del volumen (tomografía), se usa en medicina y microscopía electrónica, por ejemplo.
2. Reconstrucción de la superficie, se aplica en visualización de terrenos, robótica, arqueología y realidad aumentada.

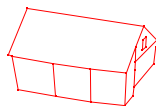
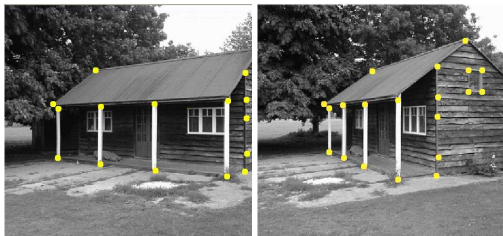
EL ÁREA DE VISIÓN POR COMPUTADORA ^a

^aFoley, 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.

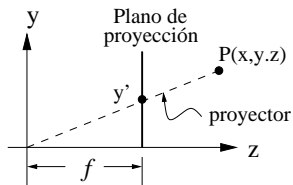
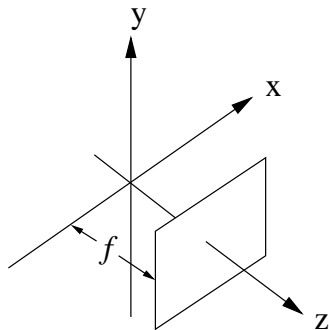
- ▶ Conocimiento de la escena (paralelismo, ortogonalidad, escenas con planos)
- ▶ Conocimiento del movimiento de la escena
- ▶ Correspondencias de puntos

RECONSTRUCCIÓN A PARTIR DE CORRESPONDENCIAS DE PUNTOS



- PASO 1 Procesamiento de imagen.** Involucra la adquisición de las imágenes, el filtrado de ellas para mejorar sus condiciones, y encontrar la correspondencia de puntos en las imágenes.
- PASO 2 Visión.** Consiste en la *calibración de la cámara y la obtención de los parámetros de orientación y posición* para la vista correspondiente en cada imagen, y en la *triangulación*, esto es, la obtención de los puntos tridimensionales.
- PASO 3 Visualización.** Se extraen las texturas en las imágenes para mapearlas en las caras del poliedro reconstruido.

PROYECCIÓN EN PERSPECTIVA



Por triángulos semejantes:

$$\frac{y'}{f} = \frac{y}{z}$$
$$y' = \frac{fy}{z}$$

Las ecuaciones:

$$x' = \frac{fx}{z}, \quad y \quad y' = \frac{fy}{z},$$

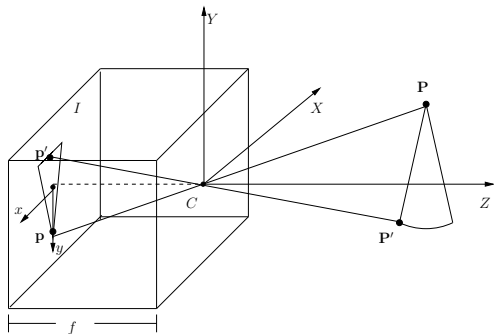
pueden escribirse en forma matricial como:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Usando solo coordenadas homogneas:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

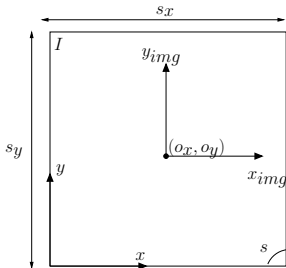
MODELO DE LA CÁMARA OSCURA (1/2)



MODELO DE LA CÁMARA OSCURA (2/2)

- ▶ Un modelo más general de la cámara involucra otros parámetros internos,

$$K = \begin{bmatrix} s_x & s & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & s & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$



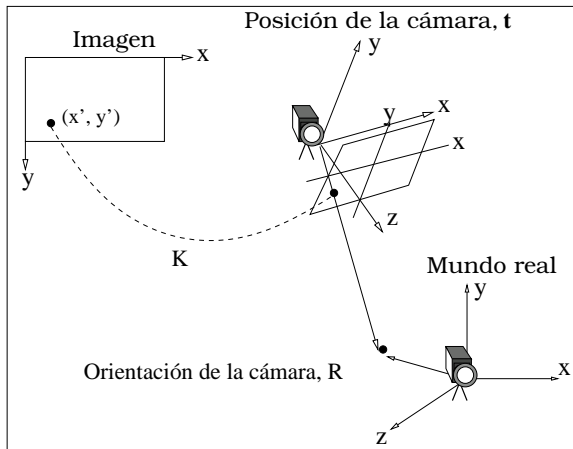
- ▶ El punto principal (o_x, o_y)
- ▶ El parámetro de oblicuidad s
- ▶ Los factores de escala por píxel de la imagen (s_x, s_y)

Usando la matriz de rotación R y el vector de translación \mathbf{t} de la cámara con respecto al sistema de coordenadas del mundo:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \alpha & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\lambda \mathbf{p} = K[R|\mathbf{t}]\mathbf{P}$$

PARÁMETROS DE LA CÁMARA



R tiene tres grados de libertad, que puede verse como tres ángulos de rotación alrededor de los ejes:

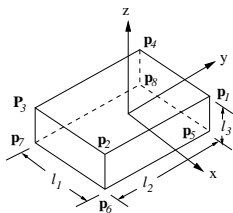
$$R = R_z(\theta_3) \cdot R_y(\theta_2) \cdot R_z(\theta_1)$$

Checar

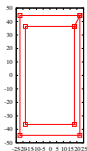
http://en.wikipedia.org/wiki/Euler_angles

EJEMPLO CON CUBOIDES

Con el siguiente modelo parametrizado de un cuboide:



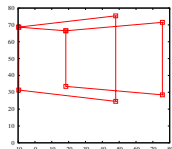
Para todas las imágenes: $f = 200$, $l_1 = l_3 = 1$, $l_2 = 2$



$$(u_0, v_0) = (0, 0),$$

$$\theta_1 = 0, \theta_2 = 0, \theta_3 = 0,$$

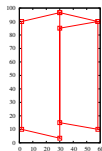
$$\mathbf{t} = [0, 0, 5]^T.$$



$$(u_0, v_0) = (0, 0),$$

$$\theta_1 = 90, \theta_2 = 45, \theta_3 = 0,$$

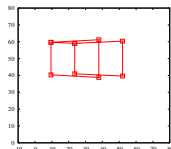
$$\mathbf{t} = [0, 0, 5]^T.$$



$$(u_0, v_0) = (30, 50),$$

$$\theta_1 = 0, \theta_2 = 45, \theta_3 = 0,$$

$$\mathbf{t} = [0, 0, 5]^T.$$



$$(u_0, v_0) = (0, 0),$$

$$\theta_1 = 90, \theta_2 = 45, \theta_3 = 0,$$

$$\mathbf{t} = [0, 0, 10]^T.$$

- ▶ Se basa en el trabajo de Wilczkowiak (2001) ¹
- ▶ Es posible reconstruir un cuboide a partir de una sola imagen.

¹M. Wilczkowiak and E. Boyer and P. Sturm, Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds, Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, Jul. 2001, p. 142–148. <http://perception.inrialpes.fr/Publications/2001/WBS01a>

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

$$\begin{bmatrix} \alpha_1 u_1 & \alpha_2 u_2 & \cdots & \alpha_8 u_8 \\ \alpha_1 v_1 & \alpha_2 v_2 & \cdots & \alpha_8 v_8 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_8 \end{bmatrix} = \tilde{X} \cdot \begin{bmatrix} 1 & 1 & \cdots & -1 \\ 1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

La matrix \tilde{X} puede estimarse a partir de seis puntos en la imagen

Una estimación por mínimos cuadrados de \tilde{X} para el sistema homogéneo de ecuaciones resultante se obtiene usando la descomposición en valores singulares (DVS).

La matriz de proyección \tilde{X} captura toda la información geométrica contenida en la imagen de proyección del cuboide.

\tilde{X} es la matriz de proyección de tamaño 3×4 la cual está definida hasta un factor de escala como:

$$\tilde{X} \sim M \cdot \tilde{\Lambda} \sim K \cdot [R|\mathbf{t}] \cdot \tilde{\Lambda}$$

Considerando la relación $X \sim K \cdot R \cdot \Lambda$, entonces $K^{-1}X \sim R \cdot \Lambda$, elevando esta última al cuadrado y considerando que $R^T R = I$, se obtiene:

$$X^T \cdot K^{-T} \cdot K^{-1} \cdot X \sim \Lambda^T \cdot \Lambda$$

ESTIMACIÓN DE LOS PARÁMETROS INTRÍNSECOS

Tenemos tres incógnitas en K , que son los valores para f , u_0 y v_0 . De $\omega = K^{-T} \cdot K^{-1}$ resulta:

$$\omega = \begin{bmatrix} \omega_{11} & 0 & \omega_{13} \\ 0 & \omega_{11} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix} = \frac{1}{\omega_{11}} \begin{bmatrix} 1 & 0 & \omega_{13}/\omega_{11} \\ 0 & 1 & \omega_{23}/\omega_{11} \\ \omega_{13}/\omega_{11} & \omega_{23}/\omega_{11} & \omega_{33}/\omega_{11} \end{bmatrix}$$

que se resuelve aplicando las restricciones lineales siguientes, donde X_i es la i -ésima columna de X :

► $X_i^T \cdot \omega \cdot X_j = 0.$

y puede establecerse un sistema de ecuaciones lineales no-homogéneo.

ESTIMACIÓN DE LOS PARÁMETROS DE ORIENTACIÓN Y POSICIÓN

$$R = \frac{1}{\lambda} K^{-1} X \Lambda^{-1}$$

R no cumple las propiedades de una matriz ortogonal ($R^{-1}R = R^T R = I$), por lo tanto para cumplir estas propiedades se fuerzan las condiciones de ortogonalidad usando de nuevo DVS.

\mathbf{t} se determina de:

$$\begin{aligned}\tilde{X} &= \lambda K \cdot [R | \mathbf{t}] \tilde{\Lambda}, \text{ y} \\ \lambda [R | \mathbf{t}] &= K^{-1} \tilde{X} \tilde{\Lambda}^{-1}\end{aligned}$$

$\lambda = 1/\|\mathbf{r}_1\| = 1/\|\mathbf{r}_2\| = 1/\|\mathbf{r}_3\|$, donde \mathbf{r}_i es la i -ésima columna de la matriz R .

SOLUCIÓN USANDO LA EVOLUCIÓN DIFERENCIAL

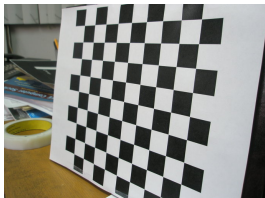
El problema puede establecerse como un problema de optimización:
Dados \mathbf{p}_i puntos de las esquinas de un cuboide en una imagen,
obtener un vector \mathbf{x} , el cual contiene:

- ▶ l_2 y l_3 (l_1 se supone igual a 1),
- ▶ R (tres ángulos),
- ▶ \mathbf{t} (tres valores para el vector de translación), y
- ▶ f, u_0, v_0

de tal forma que \mathbf{x} minimice

$$g : Q \subset \mathbb{R}^{11} \rightarrow \mathbb{R}$$
$$g(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{p}_i - \hat{\mathbf{p}}_i(l_2, l_3, K, R, \mathbf{t})\|^2$$

donde la función g se conoce como error de reproyección.



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\lambda \mathbf{p} = \mathbf{H} \mathbf{P}$$

Z. Zhang, A Flexible New Technique for Camera Calibration, IEEE Trans on Patt Anal & Mach Intel (22) 11, 2000, p. 1330–1334.

RESTRICCIONES SOBRE LOS PARÁMETROS INTRÍNSECOS

Si $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$, tenemos

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

Sabiendo que \mathbf{r}_1 y \mathbf{r}_2 son ortogonales, tenemos:

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

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

$$\begin{aligned} \omega &= K^{-T} K^{-1} = \\ &= \begin{bmatrix} \frac{1}{f} & 0 & 0 \\ 0 & \frac{1}{f} & 0 \\ \frac{-u_0}{f} & \frac{-v_0}{f} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{f} & 0 & \frac{-u_0}{f} \\ 0 & \frac{1}{f} & \frac{-v_0}{f} \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{f^2} & 0 & \frac{-u_0}{f^2} \\ 0 & \frac{1}{f^2} & \frac{-v_0}{f^2} \\ \frac{-u_0}{f^2} & \frac{-v_0}{f^2} & \frac{u_0^2}{f^2} + \frac{v_0^2}{f^2} + 1 \end{bmatrix} = \\ &= \begin{bmatrix} \omega_{11} & 0 & \omega_{13} \\ 0 & \omega_{11} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix} \end{aligned}$$

Se puede establecer un sistema de ecuaciones homogéneo de la forma:

$$V\mathbf{b} = 0,$$

donde $\mathbf{b} = [\omega_{11}, \omega_{13}, \omega_{23}, \omega_{33}]^T$.

Con una sola imagen, se puede asumir que (u_0, v_0) está en el centro de la imagen, y estimar una solución al sistema de ecuaciones usando DVS.

$$\lambda \mathbf{p} = H \mathbf{P}$$

Los puntos \mathbf{p} y \mathbf{P} los conocemos y podemos establecer un sistema de ecuaciones homogéneo de la forma:

$$A \mathbf{h} = 0$$

Los valores de los puntos en la imágenes deben normalizarse:

$$\begin{aligned} T_1 \mathbf{p} &= H' T_2 \mathbf{P} \\ \mathbf{p}' &= H' \mathbf{P}' \end{aligned}$$

Para recuperar H hacemos:

$$\begin{aligned} T_1^{-1} T_1 \mathbf{p} &= T_1^{-1} H' T_2 \mathbf{P}, \\ \mathbf{p} &= T_1^{-1} H' T_2 \mathbf{P}', \end{aligned}$$

$$\text{y } H = T_1^{-1} H' T_2.$$

La normalización de los datos es a media cero y d.e. 1.

El método lineal para calcular f , u_0 , v_0 , \mathbf{r}_1 , \mathbf{r}_2 y \mathbf{t} se puede mejorar aplicando un estimador de máxima verosimilitud, minimizando:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}_{ij} - \hat{\mathbf{p}}(K, R_i, \mathbf{t}_i, \mathbf{P}_j)\|^2$$

para n imágenes y m puntos por imagen.

LA EVOLUCIÓN DIFERENCIAL (1/2)

- ▶ Para resolver el problema de optimización se empleó esta heurística.
- ▶ Utiliza representación real.
- ▶ Ha mostrado un mejor desempeño que la mayoría de los algoritmos evolutivos, en problemas del mundo real
- ▶ Su desempeño depende del control de pocos parámetros, lo cual facilita su uso.
- ▶ Posee una gran capacidad de exploración, evitando que obtenga óptimos locales.
- ▶ Basada en un operador de diferencia (simula el gradiente en esa zona).
- ▶ La variante más popular y robusta es la *rand/1/bin*.

LA EVOLUCIÓN DIFERENCIAL (2/2)

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

Ensure: Una población g evolucionada.

```
1: Iniciar aleatoriamente a cada individuo  $i_k^0 \forall k = 1, 2, \dots, p$ 
2: Evaluar cada individuo  $g(i_k^g) \forall k = 1, 2, \dots, p.$ 
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  $g(r_k^g) < g(i_k^g)$  then
15:         $i_k^{g+1} = r_k^g$ 
16:      else
17:         $i_k^{g+1} = i_k^g$ 
18:      end if
19:    end for
20:  end for
```

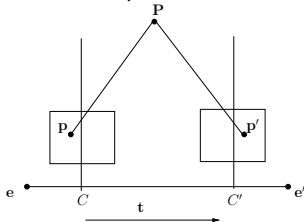
Cuando no conocemos el modelo tridimensional, se realiza:

1. Calcular la matriz fundamental a partir de las correspondencias de puntos.
2. Calibrar la cámara (K) usando las ecuaciones de Kruppa o sus derivaciones [KNS06, SCCN05].
3. Estimar los parámetros de orientación y posición (R y \mathbf{t})
4. Calcular los puntos tridimensionales

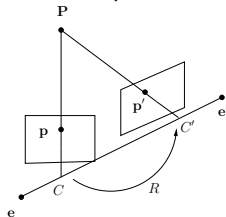
- ▶ Encontrar las matrices de proyección y los puntos tridimensionales a partir de las correspondencias de puntos extraídas de dos imágenes.
- ▶ Las matrices de proyección se generan a partir de los parámetros de la cámara, de tal forma que es necesario calibrar la cámara y estimar los parámetros de orientación y posición por cada imagen.

- ▶ Existen configuraciones (posiciones) de la cámara donde, al tomar las imágenes, no es posible estimar los parámetros de la cámara empleando las ecuaciones de Kruppa.

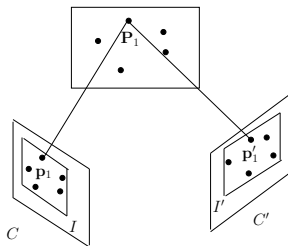
- ▶ Traslación pura.



- ▶ Rotación pura.



- ▶ Los puntos tridimensionales yacen sobre un mismo plano.



- ▶ P. Sturm y Z.L. Cheng en 2005 [SCCN05].
- ▶ K. Kanatani y A. Nakatsuji en 2006 [KNS06].
- ▶ Métodos que calibran la cámara a partir de dos imágenes.
- ▶ Parten de la matriz fundamental y derivaciones de las ecuaciones de Kruppa.
- ▶ Obtienen soluciones aceptables cerca de las configuraciones críticas pero no en ellas.

- ▶ No emplea la matriz fundamental.
- ▶ Estima directamente los parámetros de la cámara a través del modelo de la cámara oscura ($\mathbf{p} = M\mathbf{P}$).
- ▶ Se planteó como un problema de optimización de la siguiente forma: encontrar $R(\alpha, \beta, \gamma)$, $R'(\alpha', \beta', \gamma')$, $\mathbf{t}(t_x, t_y, t_z)$, $\mathbf{t}'(t'_x, t'_y, t'_z)$, f y los n puntos tridimensionales \mathbf{P}_i tal que minimice la distancia euclidiana entre los puntos de la imagen ($\mathbf{p}_i, \mathbf{p}'_i$) y los puntos estimados ($\hat{\mathbf{p}}_i = M\mathbf{P}_i, \hat{\mathbf{p}}'_i = M'\mathbf{P}_i$),

$$g(\mathbf{P}_i, M, M') = \frac{1}{n} \sum_{i=1}^n (d(\mathbf{p}_i, \hat{\mathbf{p}}_i) + d(\mathbf{p}'_i, \hat{\mathbf{p}}'_i)).$$

EVALUACIÓN DE UN INDIVIDUO

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

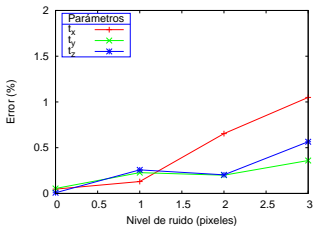
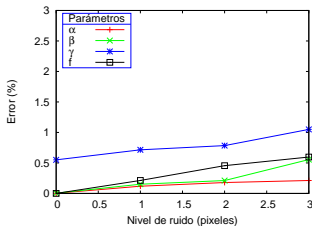
Ensure: La evaluación de \mathbf{i}_k^g mediante la función objetivo.

- 1: Extraer el parámetro intrínseco f de la cámara de \mathbf{i}_k^g .
- 2: Extraer los parámetros de orientación (θ, β, γ) y posición (t_x, t_y, t_z) de \mathbf{i}_k^g (primera imagen).
- 3: Extraer los parámetros de orientación $(\theta', \beta', \gamma')$ y posición (t'_x, t'_y, t'_z) de \mathbf{i}_k^g (segunda imagen).
- 4: Generar las dos matrices de proyección $M = K[R | \mathbf{t}]$, $M' = K[R' | \mathbf{t}']$.
- 5: **for** $i = 1$ hasta n **do**
- 6: Estimar el punto tridimensional $\hat{\mathbf{P}}_i$ mediante el método lineal de triangulación y ecuaciones normales.
- 7: Calcular los puntos estimados $(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}'_i)$, a partir de las matrices de proyección y el punto tridimensional estimado $\hat{\mathbf{P}}_i$.
- 8: Calcular la distancia euclidiana entre los puntos de cada imagen y los puntos estimados $(d(\mathbf{p}, \hat{\mathbf{p}}_i) + d(\mathbf{p}, \hat{\mathbf{p}}'_i))$.
- 9: **end for**
- 10: Regresar la media de la distancia euclidiana de las n correspondencias de puntos como el valor de la función objetivo.

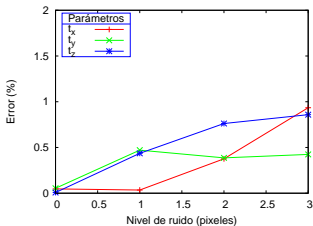
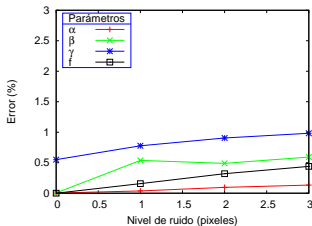
- ▶ Se generó un par de imágenes a partir de 74 puntos que pertenecen la superficie de una esfera.
- ▶ Se tomaron 20 y 24 correspondencias de puntos aleatoriamente.
- ▶ A cada punto se le añadió ruido gaussiano con media cero y desviación estándar de 1 a 3 píxeles.
- ▶ Parámetros ED: 30 individuos, 5,000 generaciones, 0.9 para la constante de recombinación y 0.85 para la constante de la diferencia.

RESULTADOS - 40 EJECUCIONES

20 correspondencias de puntos



24 correspondencias de puntos



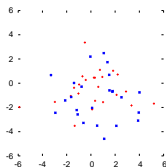
EXPERIMENTO - CONFIGURACIONES CRÍTICAS

- ▶ Traslación pura.
- ▶ Se generó un par de imágenes de un objeto.
- ▶ A cada punto se le añadió ruido gaussiano con media cero y desviación estándar de 1 a 3 píxeles.
- ▶ Parámetros ED: 30 individuos, 5000 generaciones, 0.9 en la constante de recombinación y 0.85 en la constante de diferencia.

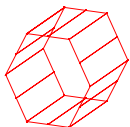
Imagen TP	α	β	γ	t_x	t_y	t_z	f
1	0	0	0	5	7	20	500
2	0	0	0	25	7	20	500

Estadísticas del error de 40 ejecuciones

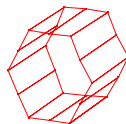
Puntos	Ruido	Mejor	Media	Peor	d.e.	i.c. de la media
24	0	0.00	0.00	0.00	0.00	[0.00,0.00]
	1	0.75	0.75	0.75	0.00	[0.75,0.75]
	2	1.26	1.26	1.26	0.00	[1.26,1.26]
	3	2.51	2.51	2.51	0.00	[2.51,2.51]



Error entre puntos sin ruido y puntos estimados (ruido 3 píxeles).



Objeto real



Objeto reconstruido

EXPERIMENTO - SUPERFICIE CRÍTICA

- ▶ Superficie crítica empleando imágenes reales.
- ▶ Par de imágenes reales de una cuadrícula.
- ▶ Parámetros ED: 30 individuos, 10000 generaciones, 0.9 en la constante de recombinación y 0.85 en constante de diferencia.

SUPERFICIE CRÍTICA (IMÁGENES REALES)

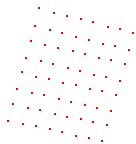
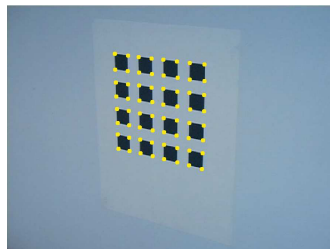
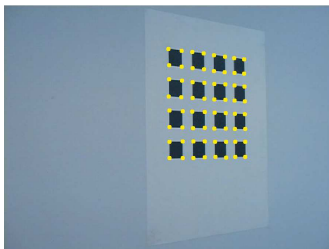
Estadísticas del error de 40 ejecuciones

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

Estadísticas de la longitud del foco

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

SUPERFICIE CRÍTICA (IMÁGENES REALES)



EXPERIMENTO - TRASLACIÓN PURA

- ▶ Traslación pura con imágenes reales.
- ▶ Par de imágenes reales de un juguete.
- ▶ Parámetros ED: 30 individuos, 7000 generaciones, 0.9 en la constante de recombinación y 0.85 en constante de diferencia.

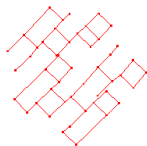
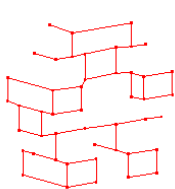
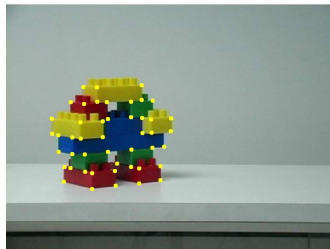
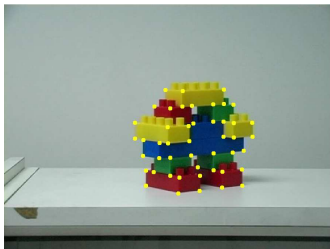
Estadísticas del error de 40 ejecuciones

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

Estadísticas de los parámetros de la cámara

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

TRASLACIÓN PURA (IMÁGENES REALES)



RECONSTRUCCIÓN 3D A PARTIR DE TRES O MÁS IMÁGENES

Extensión de la solución propuesta de tres o más imágenes

- ▶ Se divide en dos fases.
- ▶ Primera fase: estima la matriz de calibración, los parámetros de orientación y posición, y los puntos tridimensionales a partir de las primeras dos imágenes empleando la solución anterior.
- ▶ Segunda fase: estima los parámetros de orientación y posición, y los puntos tridimensionales de las imágenes restantes por cada imagen añadida, de forma iterativa.
- ▶ En la segunda fase también se emplea la evolución diferencial.

EVALUACIÓN DE UN INDIVIDUO (SEGUNDA FASE)

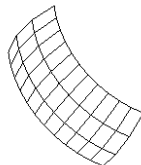
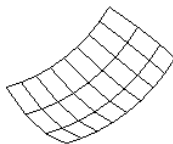
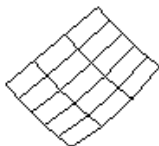
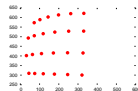
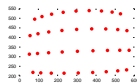
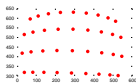
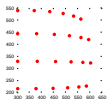
Require: Un individuo \mathbf{i}_k^g , K , R_{j-1} y \mathbf{t}_{j-1} .

Ensure: La evaluación de \mathbf{i}_k^g mediante la función objetivo.

- 1: Extraer de \mathbf{i}_k^g los parámetros de orientación $(\theta_j, \beta_j, \gamma_j)$ y posición (t_{xj}, t_{yj}, t_{zj}) pertenecientes a la j -ésima vista.
- 2: Extraer de \mathbf{i}_k^g el factor de escala s .
- 3: Generar las dos matrices de proyección $M_{j-1} = K[R_{j-1}|\mathbf{t}_{j-1}]$ y $M_j = K[R_j|\mathbf{st}_j]$.
- 4: **for** $i = 1$ hasta q **do**
- 5: Calcular los puntos estimados a partir de las matrices de proyección y el i -ésimo punto tridimensional, el cual es estimado en la fase o iteración anterior.
- 6: Calcular la distancia euclidiana entre los puntos de cada imagen y los puntos estimados $(d(\mathbf{p}, \hat{\mathbf{p}}_i) + d(\mathbf{p}, \hat{\mathbf{p}}_i))$.
- 7: **end for**
- 8: Regresar la media de la distancia euclidiana de las q correspondencias de puntos como el valor de la función objetivo.

RESULTADOS DEL EXPERIMENTO

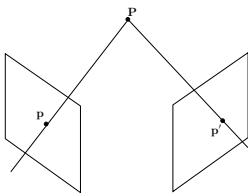
Fase	Puntos	Sin ruido	1 píxel	2 píxeles	3 píxeles
1 (imagen 1 y 2)	24	0.00	0.76	1.39	2.10
2 (imagen 2 y 3)	12	0.06	0.92	2.20	3.34
2 (imagen 3 y 4)	12	0.05	0.82	1.65	2.73



EL PROBLEMA DE TRIANGULACIÓN (1/2)

En la triangulación se calcula los puntos 3D a partir de las correspondencias de puntos sobre imágenes 2D.

Cuando los puntos no tienen ruido, la triangulación es la intersección de dos líneas; con la presencia de ruido esta intersección no existe y se debe estimar la mejor solución.



EL PROBLEMA DE TRIANGULACIÓN (2/2)

Si se tiene un punto \mathbf{P} en el espacio 3D y dos matrices de proyección, M y M' , entonces se pueden obtener los puntos bidimensionales $\hat{\mathbf{p}}$ y $\hat{\mathbf{p}}'$:

$$\hat{\mathbf{p}} = M\mathbf{P}, \text{ y}$$

$$\hat{\mathbf{p}}' = M'\mathbf{P}$$

Cuando solamente se cuenta con los puntos 2D \mathbf{p} y \mathbf{p}' , 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 M y M' .

Encontrar el punto 3D \mathbf{P} en coordenadas del mundo (X_w, Y_w, Z_w) tal que:

Minimice:

$$g(\mathbf{P}) = d(\mathbf{p}, \hat{\mathbf{p}})^2 + d(\mathbf{p}', \hat{\mathbf{p}}')^2,$$

donde $d(*, *)$ son distancias Euclidianas, \mathbf{p} y \mathbf{p}' representan puntos 2D sobre la primera y segunda imágenes, respectivamente, y $\hat{\mathbf{p}}$ es un punto 2D reconstruido (estimado) con coordenadas (\hat{x}, \hat{y}) .

Encontrar el punto 3D \mathbf{P} en coordenadas del mundo (X_w, Y_w, Z_w) tal que:

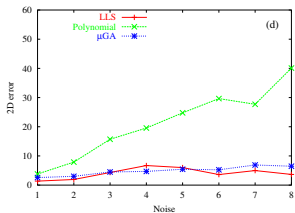
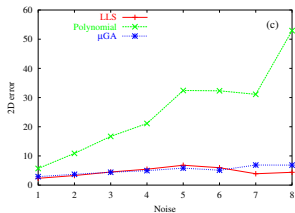
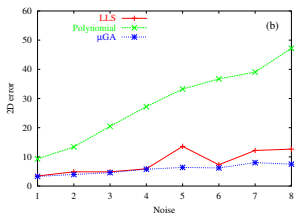
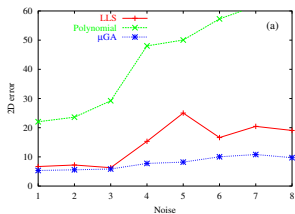
Minimice:

$$g_1(\mathbf{P}) = d(\mathbf{p}, \hat{\mathbf{p}})^2, \quad y \quad g_2(\mathbf{P}) = d(\mathbf{p}', \hat{\mathbf{p}}')^2,$$

Este es el tercer trabajo donde la multi-objetivización se pudo aplicar exitosamente.

I. Vite Silva, N. Cruz Cortés, G. Toscano Pulido, and L.G. de la Fraga, Optimal Triangulation in 3D Computer Vision Using a Multi-Objective Evolutionary Algorithm, EvoWorkshops 2007, Springer, LNCS 4448, 2007, pp. 330–339

RESULTADOS



Results of the second experiment. Graphic (a) was calculated for 8 point pairs, (b) for 12 pairs, (c) for 16, and (d) for 20 point pairs.

1. El objetivo de visión es la reconstrucción tridimensional de la superficie de objetos, a partir de imágenes bidimensionales.
2. Para visión, es necesario tener conocimiento de algebra lineal y análisis numérico.
3. Los algoritmos evolutivos permiten resolver directamente un problema de visión.
4. El resultado de aplicar la ED puede usarse como punto inicial para un ajuste no lineal.
5. Aplicar la ED permite calibrar la cámara y encontrar los parámetros de orientación y translación para una vista aún en configuraciones críticas.



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

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

International Journal of Computer Vision, 66(2):109–122, 2006.



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

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

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

Ofrecemos la **maestría** y **doctorado** en
Ciencias de la Computación.

- ▶ Página del Departamento: <http://www.cs.cinvestav.mx>
- ▶ Página personal: <http://delta.cs.cinvestav.mx/~fraga>
- ▶ Correo-e: fraga@cs.cinvestav.mx

- ▶ Luis Gerardo de la Fraga and Israel Vite Silva, Direct 3D Metric Reconstruction From Two Views Using Differential Evolution. *WCCI '08: Proceedings of the IEEE World Congress on Computational Intelligence*. IEEE Press, 2008.
- ▶ Luis Gerardo de la Fraga and Israel Vite Silva, Direct 3D Metric Reconstruction From Multiple Views Using Differential Evolution, M. Giacobini et al. (Eds.): *EvoWorkshops 2008*, LNCS 4974, pp. 341-346, 2008. Springer-Verlag.
- ▶ Ricardo Landa Becerra and Luis Gerardo de la Fraga, Triangulation using Differential Evolution, M. Giacobini et al. (Eds.): *EvoWorkshops 2008*, LNCS 4974, pp. 341-346, 2008. Springer-Verlag.
- ▶ Israel Vite Silva, Luis Gerardo de la Fraga and Nareli Cruz Cortés, Optimal Triangulation in 3D Computer Vision Using a Multi-Objective Evolutionary Algorithm, *EvoWorkshops 2007*, LNCS 4448. pp. 330-339. Springer-Verlag.
- ▶ Luis Gerardo de la Fraga, Israel Vite Silva and Nareli Cruz Cortés, Euclidean Distance Fit of Ellipses with a Genetic Algorithm, *EvoWorkshops 2007*, LNCS 4448. pp. 359-366. Springer-Verlag.