T10. Transformaciones geométricas

Índice

- Motivación
- Clasificación de transformaciones
- Representación matricial; coordenadas homogéneas
- Transformación directa e inversa
- Métodos de interpolación
- Otras transformaciones
- *Warping*
- *Morphing*
- Aplicación: registrado

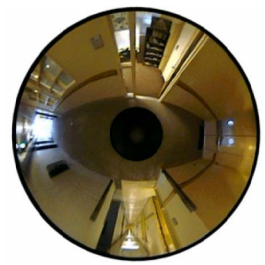
Motivación

- *Eliminar* distorsiones debidas a…
  - la óptica (e.g. *fish-eye lenses*);
  - el tipo de sensor (e.g. imagen omnidireccional);
  - el punto de vista camara-escena;
  - etc.
- *Introducir* distorsiones para…
  - registrar imágenes;
  - estimar movimiento;
  - crear imágenes panorámicas;
  - etc.
- Reconocimiento de formas *invariante* a ciertas transformaciones

Ejemplos de distorsiones

Lente “Ojo de pez”

Omnidireccional



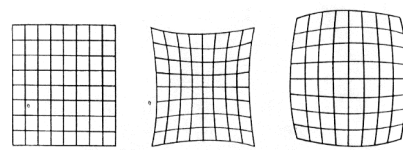
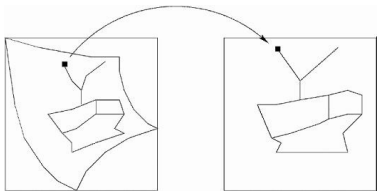
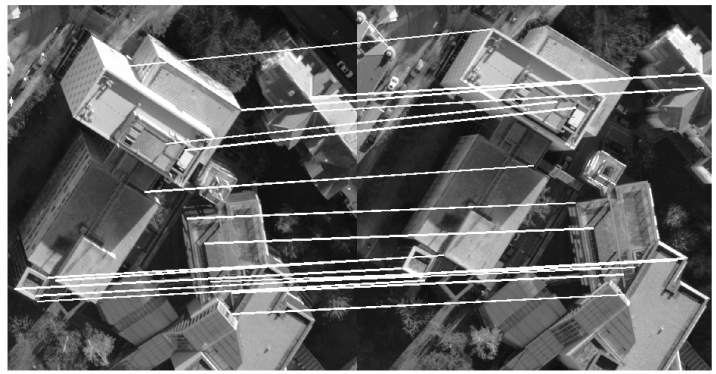


FIGURE 14.2-1. Example of geometric distortion.

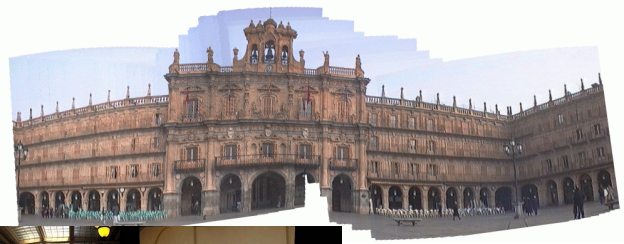
Corrección distorsiones

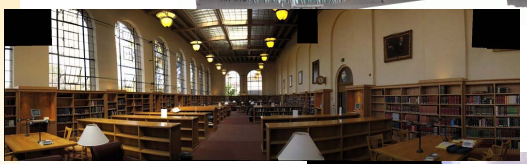


Registrado

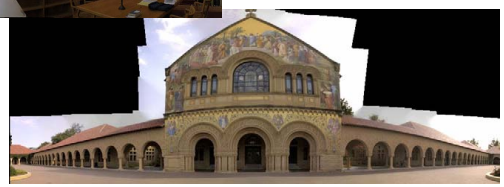


Creación de mosaicos





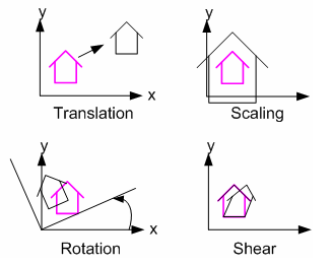
A partir de 21 imágenes



A partir de 33 imágenes

Clasificación de transformaciones

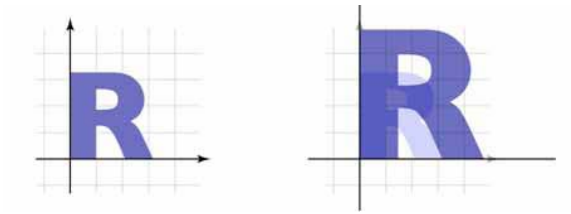
- Lineales
  - Traslación (T)
  - Rotación (R)
  - Escalado (isotrópico) (S)
  - Euclidea: T+R
  - Similitud: T+R+S
  - Afín
    - Similitud + S anisotrópico + Deformación (*shear)*
  - Proyectiva
- Polinómicas
- Generales



Escalado uniforme

$$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} sx \\ sy \end{bmatrix}$$

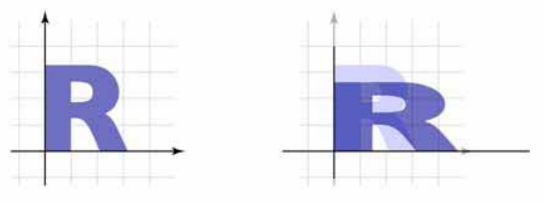
$$\begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$



Escalado no uniforme (anisotrópico)

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

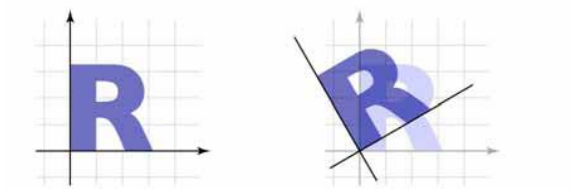
$$\begin{bmatrix} 1.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$



Rotación

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$

$$\begin{bmatrix} 0.866 & -.05 \\ 0.5 & 0.866 \end{bmatrix}$$



Reflejo vertical

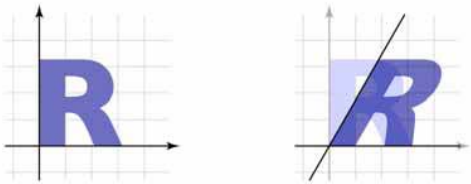
Caso particular de escalado no uniforme

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Inclinación (*Shear*)

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ay \\ y \end{bmatrix}$$

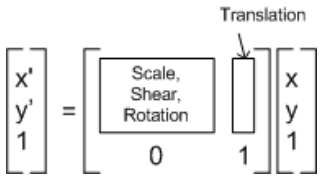
$$\begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$$



Transformación afín

- Combinación de las anteriores (traslación, escalado, rotación, inclinación)
- Se conservan:
  - Líneas rectas
  - Líneas paralelas
  - *Ratios* de longitudes a lo largo de una recta

Representación matricial

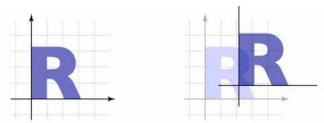


$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$

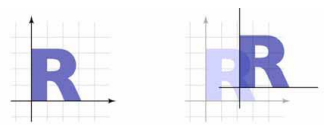
$$\begin{bmatrix} 1 & 0 & t \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t \\ y + s \\ 1 \end{bmatrix}$$

Usando coordenadas homogéneas

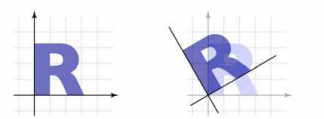
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 2.15 \\ 0 & 1 & 0.85 \\ 0 & 0 & 1 \end{bmatrix}$$



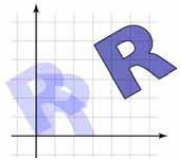
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 2.15 \\ 0 & 1 & 0.85 \\ 0 & 0 & 1 \end{bmatrix}$$



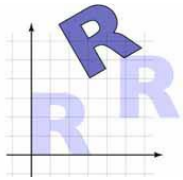
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



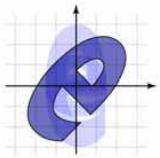
Composición de transformaciones



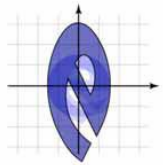
Rotación, traslación



Traslación, rotación



Escalar, rotar



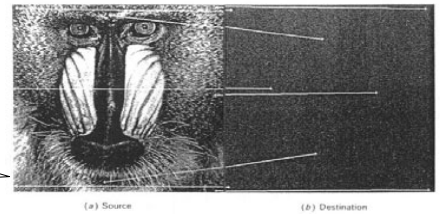
Rotar, escalar

Otras transformaciones

- Proyectiva (u homografía), 8 parámetros
- Cuadrática (o parabólica), 12 parámetros
- Polinómica

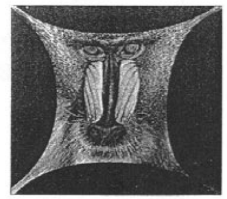
$$u(x,y) = a_1 + a_2x + a_3y + a_4xy$$
$$v(x,y) = a_5 + a_6x + a_7y + a_8xy.$$

Polinomio de 2º orden



(a) Source (b) Destination

$$u(x,y) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} - x$$
$$v(x,y) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} - y.$$



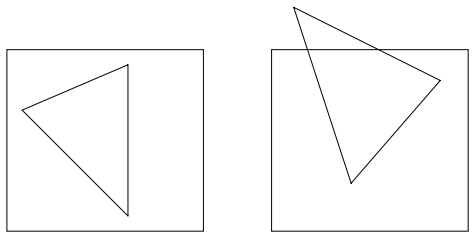
Ejemplos de transformaciones

Cuestiones prácticas

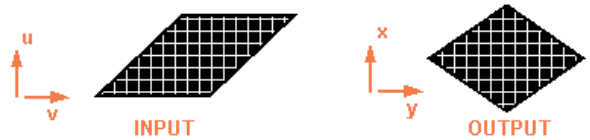
- Pixels *fuera* de la imagen
  - Comprobar límites
- Coordenadas *no enteras*
  - Obtener coordenadas enteras más cercanas

Transformación directa e inversa

- Transformación directa (*forward mapping*)
  - Pérdida de tiempo: se transforman píxels que caen *fuera* de la imagen
  - Hay píxels que se consideran *más de una* vez
  - Hay píxels que no se consideran *nunca*



Transformación directa e inversa



Forward Mapping $[x, y] = [X(u, v), Y(u, v)]$

Can result in 'holes' in output data

Inverse Mapping $[u, v] = [U(x, y), V(x, y)]$

Can result in over-sampling

Transformación general

$$\begin{cases} x' &= T_x(x, y) \\ y' &= T_y(x, y) \end{cases}$$

¿Cómo rellenamos pixels?

original

10	50	67	34
20	30	35	45
29	40	53	42

400% enlargement

10				50				67
20				30				35
29				40				

Copiando del pixel más cercano

original

10	50	67	34
20	30	35	45
29	40	53	42

400% enlargement

10				50				67
20				30				35
29				40				

Variando *gradualmente* el nivel de gris

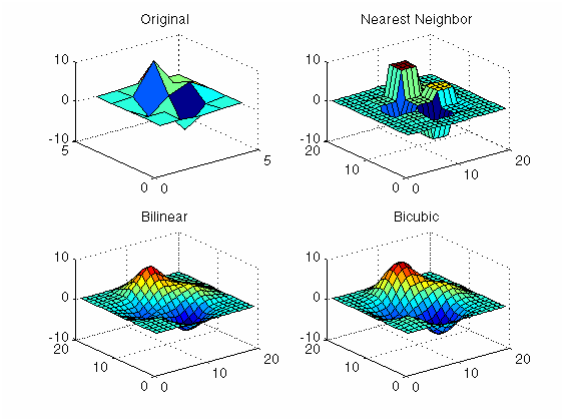
original

10	50	67	34
20	30	35	45
29	40	53	42

400% enlargement

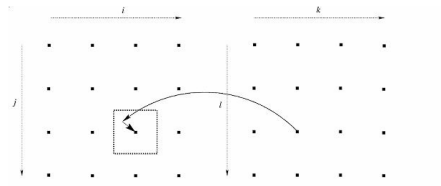
10	20	30	40	50	54	58	63	67
	21			40				
20	22	25	27	30	31	33	34	35
29				40				

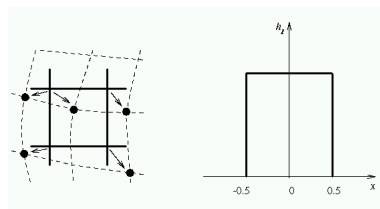
Métodos de interpolación



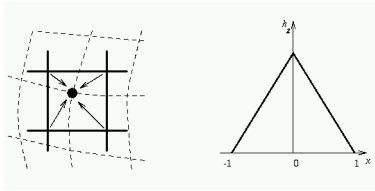
Interpolación: el vecino más próximo

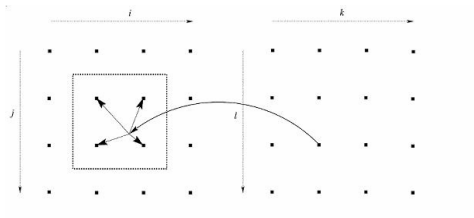
Orden cero



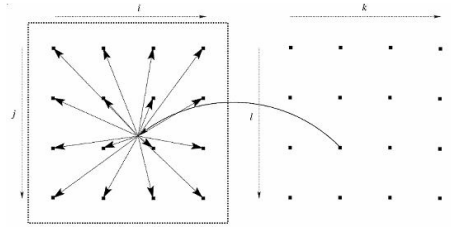


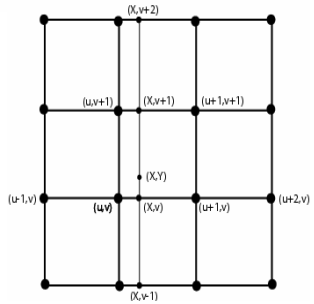
Interpolación bilineal





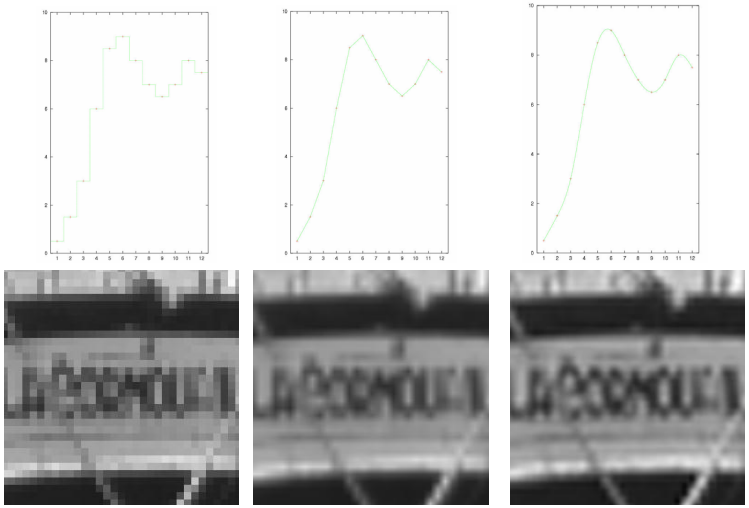
Interpolación bicúbica



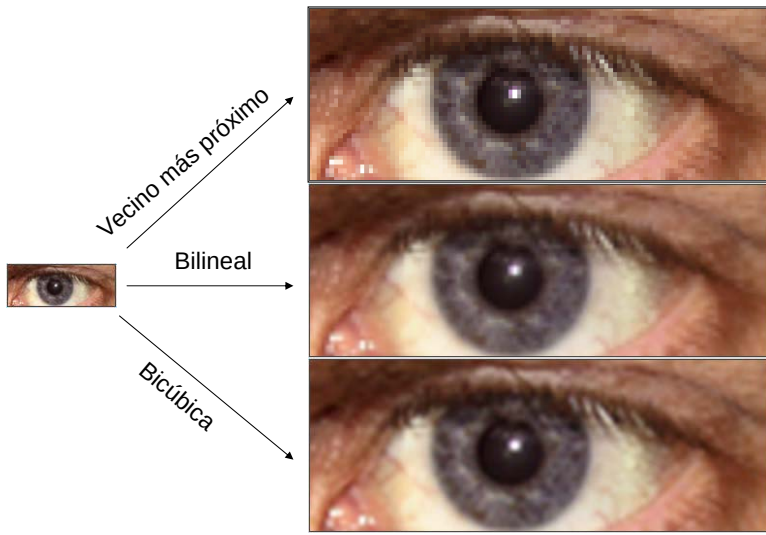


Comparando métodos

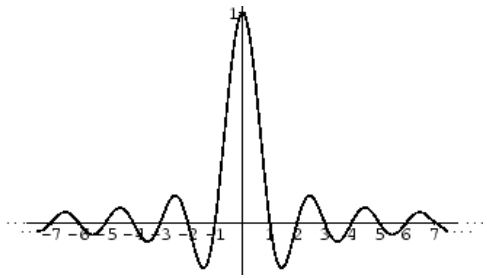
Vecino más próximo (Bi)lineal (Bi)cúbica



Interpolación en un *zoom* digital x450



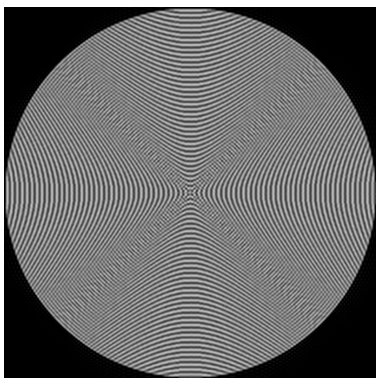
Interpolación *Sinc*



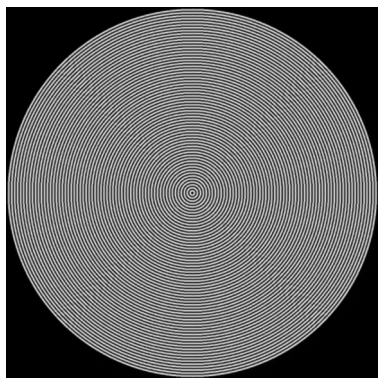
$$x(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin[\pi(t-n)]}{\pi(t-n)} = \frac{\sin(\pi t)}{\pi} \sum_{n=-\infty}^{\infty} x_n \frac{(-1)^n}{t-n}$$

Vecinos más cercanos *vs.* sinc

Bilineal *vs. sinc*

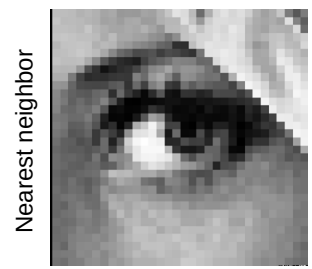


Bilineal



Sinc

Los 4 métodos



Otros métodos de interpolación

- Clásicos: promediado que sólo depende de la *posición*
- **Adaptativos:** también consideran el *nivel de gris*

Sinc no adaptativo

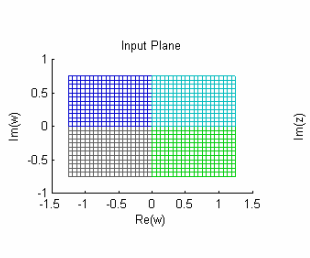
Sinc adaptativo

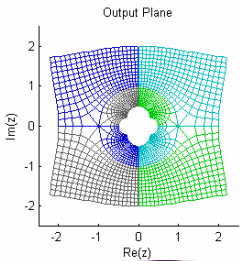
Coste computacional

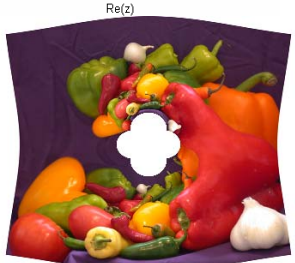
Type of Resampling	Computational Complexi
Nearest-Neighbor	$O(n^2)$
Bilinear Interpolation	$O(n^2)$
Cubic Convolution	$O(n^2)$
Cubic Spline, Direct Computation	$O(n^4)$
Cubic Spline, Using FFT	$O(n^3 \log n)$
Radial Functions with Local Support	$O(n^4)$
Gaussian, Using FFT	$O(n^3 \log n)$

Conformal mappings

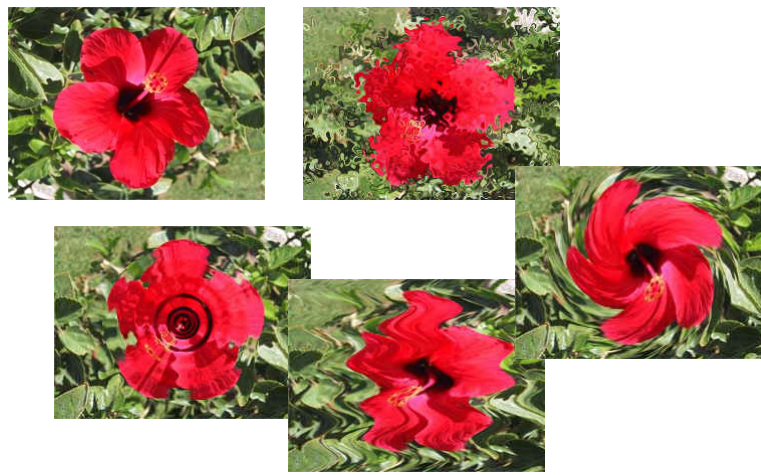
- Una *transformación conforme,* conserva la forma (localmente)







Efectos con transformaciones espaciales



http://www.jhlabs.com/ip/distortion.html

Warping

- Quadratic warp (12 coeficientes)
- Cubic warps (20 coeficientes)
- Puntos de control: 6 (o 10) para resolver sistema
- Con más de 6 (o 10) puntos: sistema sobredeterminado; resolución por mínimos cuadrados
- *Piecewise warping* (“a trozos”), *rejilla de control*

Morphing

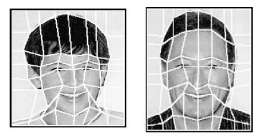
- Transformación *incremental* de una imagen en otra
  - Secuencia de imágenes intermedias
- Se consigue con
  - *Warping*
  - Registrado
  - “Color blending”
- Aplicación en películas, videos, etc.

Morphing (técnica 1): *Cross-dissolve*

$$I_t(x,y) = (1-t)\cdot I_0(x,y) + t\cdot I_1(x,y)$$

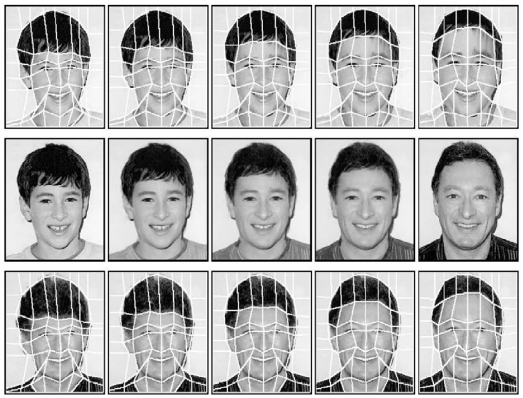
- Muy simple: interpolación *pixel a pixel*
- Resultado
  - Poco realista
  - Transiciones no suaves

Morphing (técnica 2): *cross-dissolve* a trozos

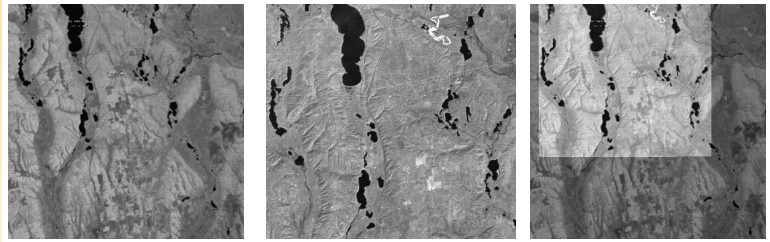


- Puede manejar más situaciones
- W*arps* diferentes a diferente trozos de la imagen
- Elección *manual* de los trozos (o automática!)
- Considera *correspondencias* de características

Mesh warping



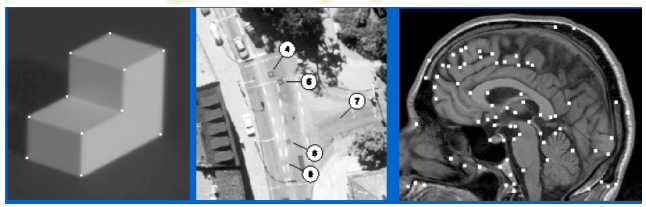
Registrado



- Problema: alinear dos (o más) imágenes
- Encontrar la transformación (función de *warping*)
- Proceso:
  - **Selección** de características (puntos, líneas,…)
  - **Correspondencia** de características

Selección de características

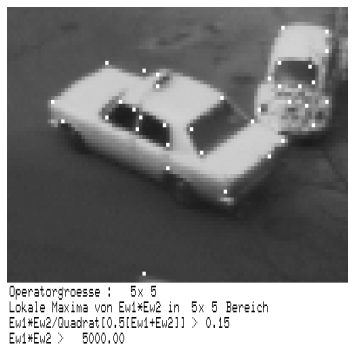
puntos



Picos/valles

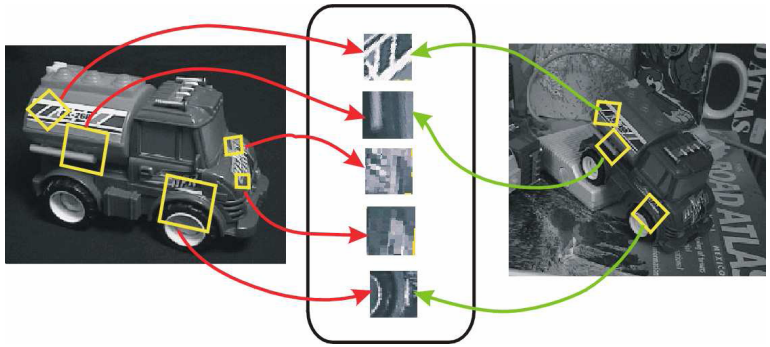
- Características prominentes, distinguibles
- Distribuidas por toda la imagen
- Invariantes a transformaciones, robustas a ruido,…

Detección de esquinas



- Harris & Stephens 88. **A Combined Corner and Edge Detector**, *AlveyVision Conf.'87*, 147-151

Invarianza local



Búsqueda de los parámetros

- Optimización
  - Descenso de gradiente
  - *Simulated annealing*
  - Búsqueda *tabú*
  - Algoritmos genéticos
  - …
- Estrategias
  - Multi-resolución
    - En la imagen
    - En los parámetros

¿Ya están registradas?

Sum of Square Differences (SSD):

$$D(r,c) = \sum_{\substack{(r',c')\in W \\ (r+r',c+c')\in F}} \Big\{ f(r+r',c+c') - w(r',c') \Big\}^2$$

- W: set of pixel positions in template w (template coordinates)
- F: set of pixel positions in image f (image coordinates)

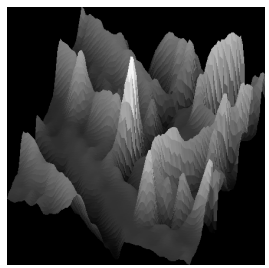
Normalised cross-correlation (NCC), or correlation coefficient:

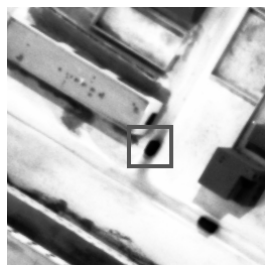
$$C_{nr}(r,c) = \frac{1}{\sqrt{S_f(r,c)\cdot S_w}} \sum \Big[f(r+r',c+c') - \overline{f}(r,c) \Big] \cdot \Big[w(r',c') - \overline{w} \Big]$$

where

$$S_f(r,c) = \sum \Big[f(r+r',c+c') - \overline{f}(r,c) \Big]^2, \qquad S_w = \sum \Big[w(r',c') - \overline{w} \Big]^2$$

Ejemplo de NCC: *template matching*





Enlaces

- **Transformaciones y su estimación**
  http://vision.uji.es/~sicandel
- **Tricks on doing rotation**
  http://www.leptonica.com/rotation.html
- **Geometric Transformation of Digital Images. Interpolation and Image Rotation**
  http://microscopy.fsu.edu/primer/java/digitalimaging/processing/geometricaltransformation/
- **Interpolation and Morphing**
  http://www.biomachina.org/courses/processing/051.pdf
- ***Sinc* interpolation (code)**
  http://slacy.com/upsample/sinc.C
- **Turbo-charged linear interpolation** (demo)
  http://bigwww.epfl.ch/demo/jshiftlinear/start.php
- **JIM - Java Image Manipulator**
  http://www.jhlabs.com/ip/imageeditor.html

Bibliografía

- Básica
  - Nick Efford. ***Digital Image Processing: a practical introduction Using Java***. Addison-Wesley 2000. (Cap. 9)
  - D. Vernon. **Machine Vision. Automatic inspection and Robot vision**. Prentice-Hall, 1991 (Cap. 4.3)
    http://homepages.inf.ed.ac.uk/rbf/BOOKS/VERNON/vernon.htm
  - D. Phillips. **Image processing in C. Analyzing and Enhancing Digital Images**. RanD Publications, 1994. (Cap. 13 y 14) [incluye código]
    http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/
  - G. Pajares, J. M. de la Cruz. **Visión por computador: imágenes digitales y aplicaciones**. Ra-Ma,2001. (Cap. 3.4)
- Avanzada
  - William K. Pratt. ***Digital Image Processing*** (3rd. edition). Joh Wiley & Sons, 2001 (Cap. 13)
  - Bernd Jähne. ***Image processing for Scientific Applications*** CRC Press, 1997 (Cap. 8)

Propuestas de artículos

- J. Shi, C. Tomasi**. Good features to track.** *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, June 1994
  http://www.cs.duke.edu/~tomasi/papers/shi/shiCvpr94.pdf
- Thomas M. Lehmann, Claudia Gönner, Klaus Spitzer. **Survey: Interpolation Methods in Medical Image Processing,** IEEE Transactions on Medical Imaging, 18(11), Nov.1999
  http://www.cvgpr.uni-mannheim.de/hornegger/MEDBV/handouts/lehmann.pdf
- Bojan Vrcelj, P. P. Vaidyanathan. **Efficient Implementation of All-Digital Interpolation.** IEEE Transactions on Image Processing, (10)11, Nov. 2001
  http://www.systems.caltech.edu/dsp/ee112b-spring04/PPVsSplinePaperForClass.pdf

Problema

¿Cómo eliminar las rectas horizontales?

input

output

http://www.leptonica.com/line-removal.html