U10. Motion Estimation

SJK002 Computer Vision

Master in Intelligent Systems

UNIVERSITAT JAUME I

UNIVERSITAT JAUME I

Index

- Application examples.
- Optical flow.
- Global motion: image registration.
  - Gray level-based registration.
  - Criterion functions.
  - Optimization methods.

Video mosaicing

Video compression

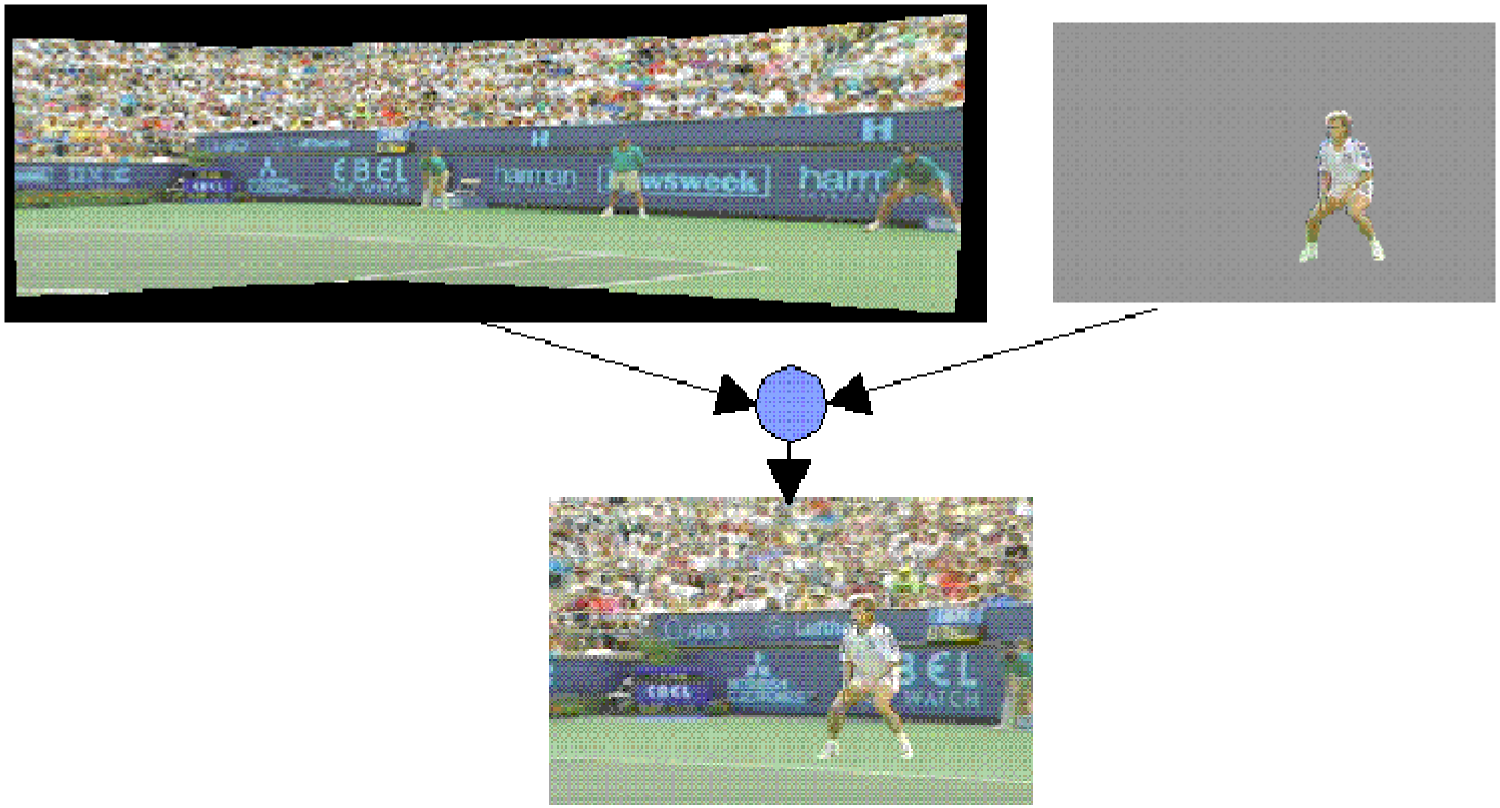
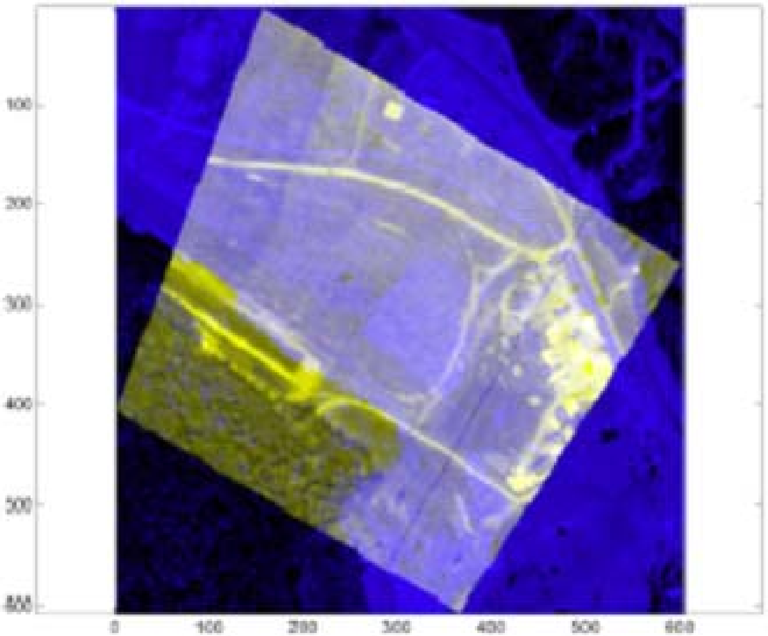


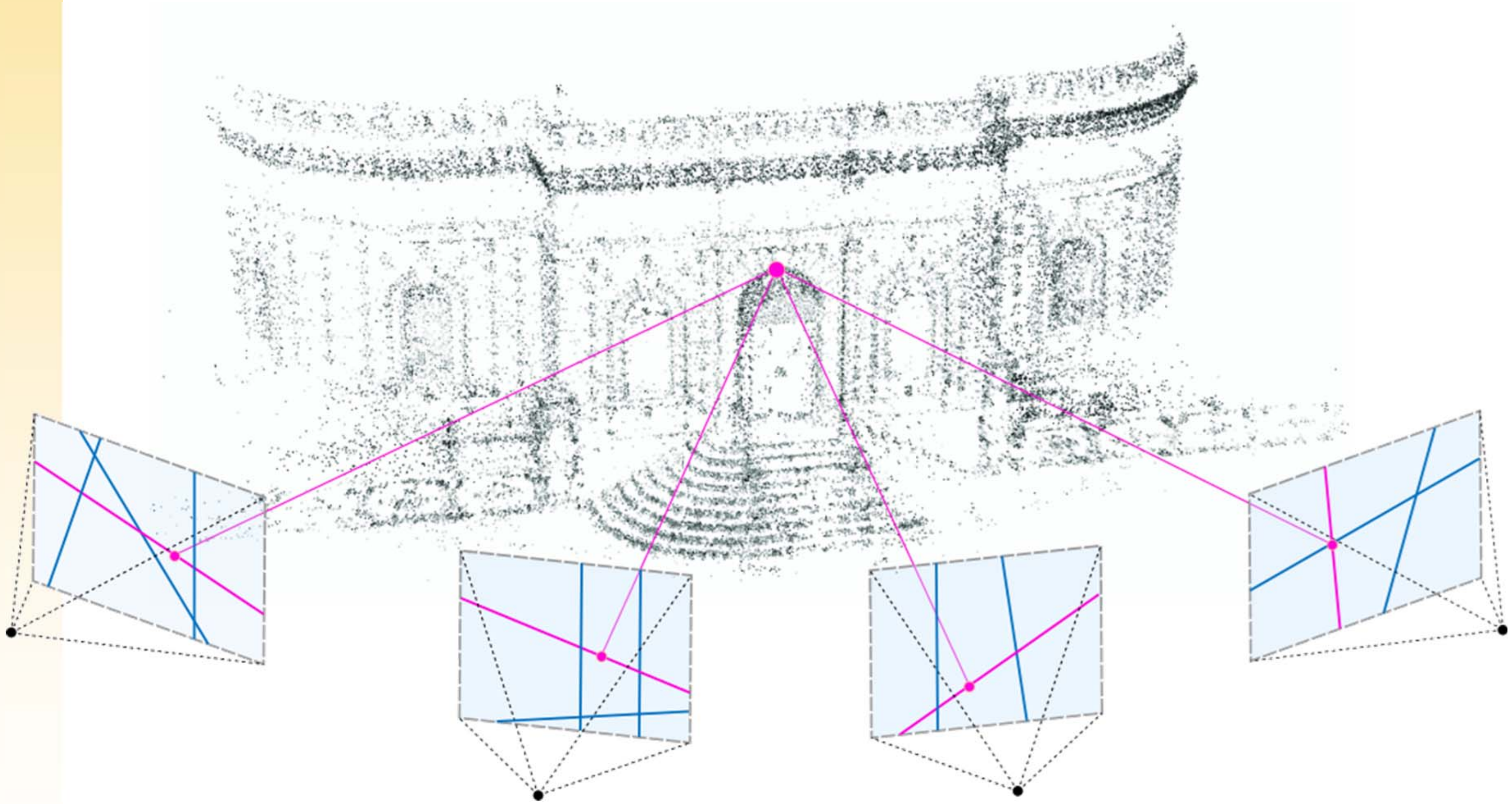
Image registration

Medical image registration

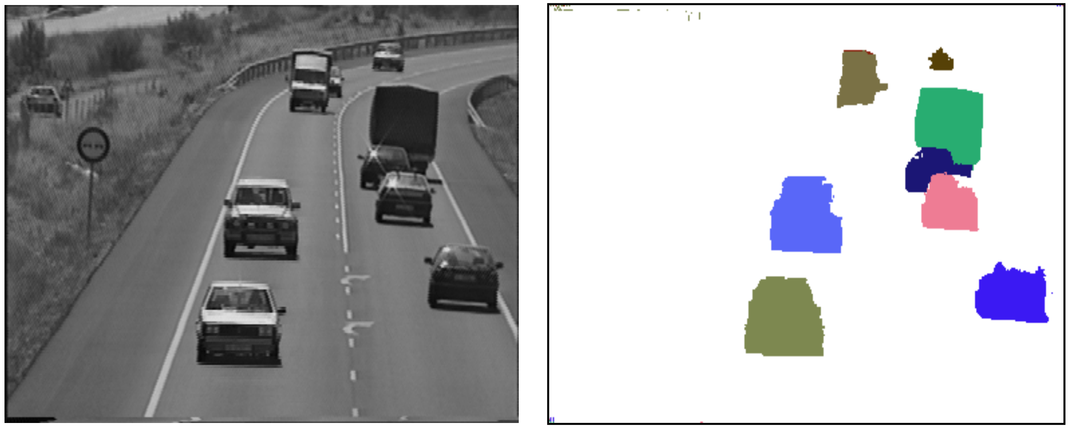


Overlapping with reference image (maps).

Structure from motion



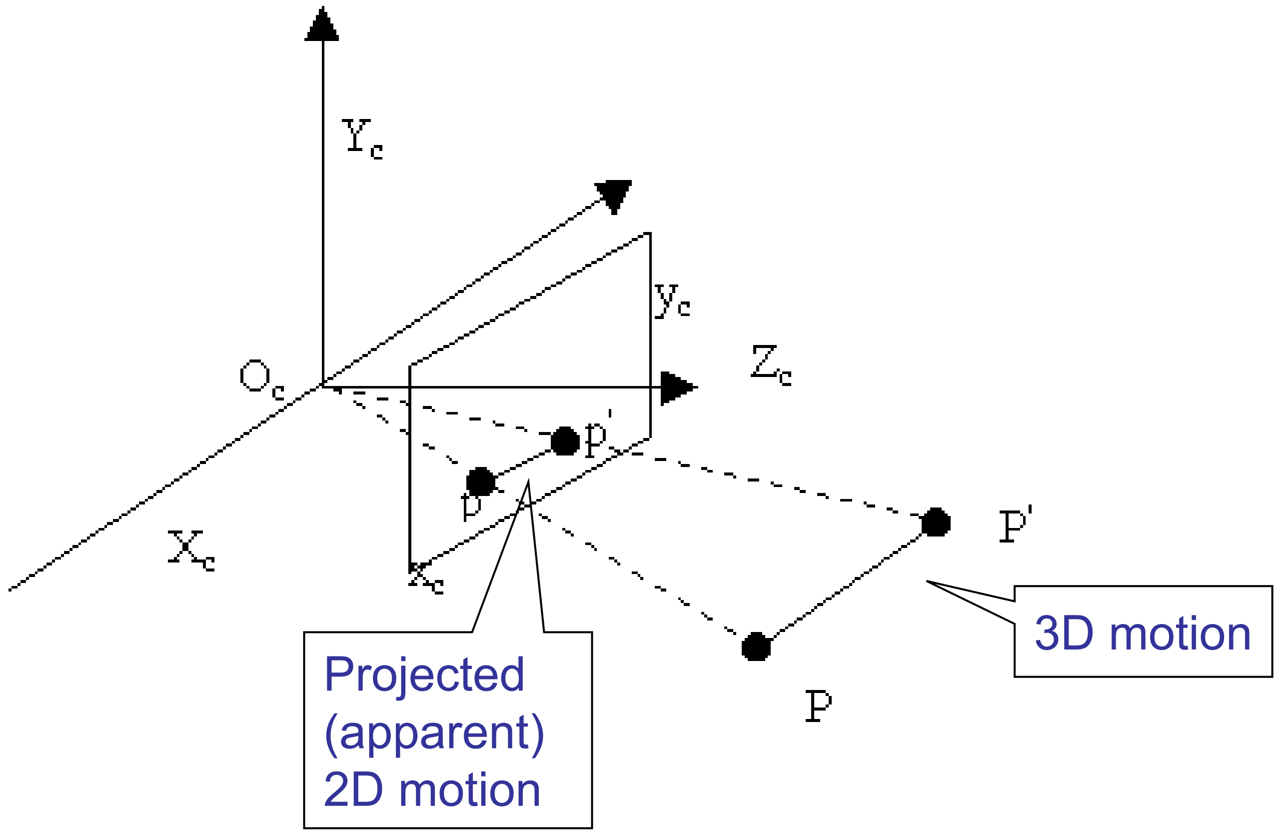
Motion segmentation and tracking



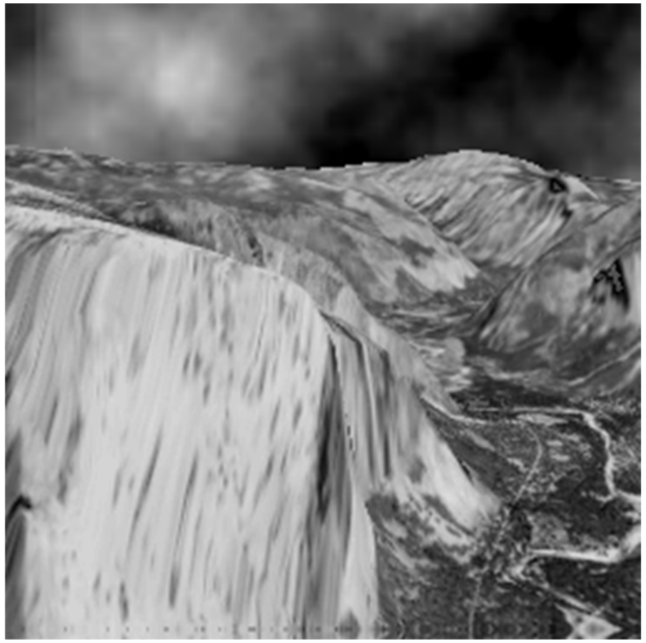
Index

- Application examples.
- Optical flow.
- Global motion: image registration.
  - Gray level-based registration.
  - Criterion functions.
  - Optimization methods.

From 3D to 3D motion

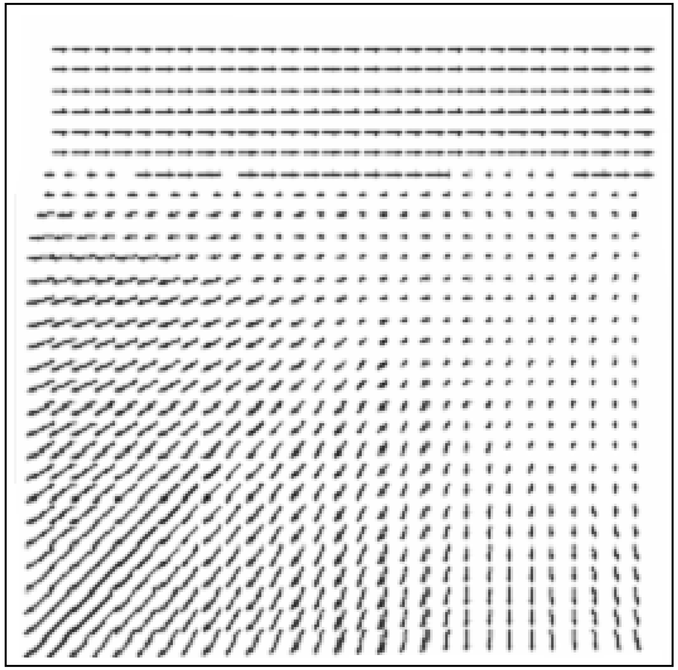


Optical flow



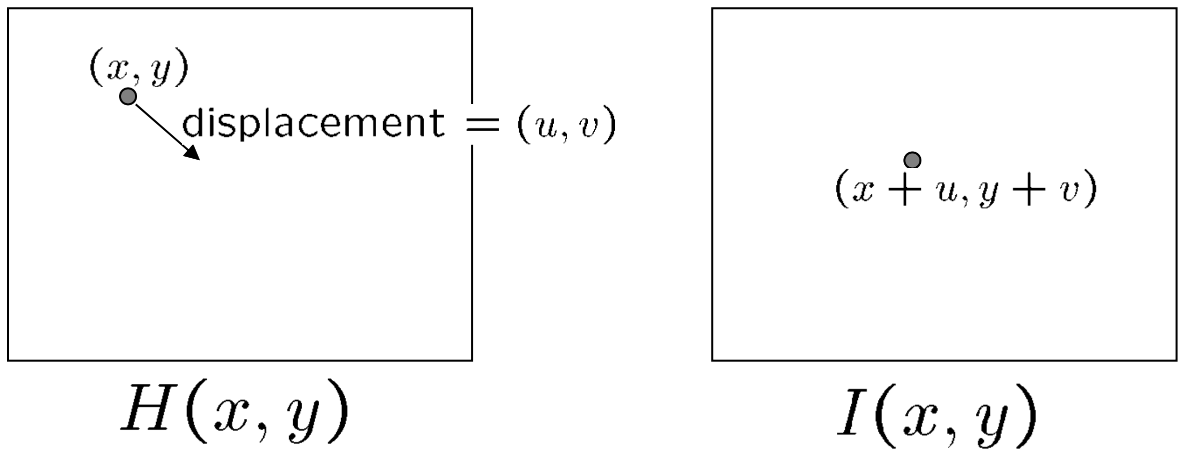
Yosemite sequence

Optical Flow or velocity field



Motion estimation of every pixel (dense motion field)

Optical flow estimation



- Assumptions:
  - Brightness (grey level) constancy assumption.

$$0 = I(x + u, y + v) - H(x, y)$$

  - Small *(u,v)* pixel motion.

Optical flow equation

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$
$$\approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$I_x = \frac{\partial I}{\partial x}$$

$$0 = I(x+u, y+v) - H(x,y)$$
$$\approx I(x,y) + I_x u + I_y v - H(x,y)$$
$$\approx (I(x,y) - H(x,y)) + I_x u + I_y v$$
$$\approx I_t + I_x u + I_y v$$
$$\approx I_t + \nabla I \cdot [u \;\; v]$$

At the limit, (u,v) $\rightarrow$ 0

$$0 = I_t + \nabla I \cdot [\tfrac{\partial x}{\partial t} \;\; \tfrac{\partial y}{\partial t}]$$

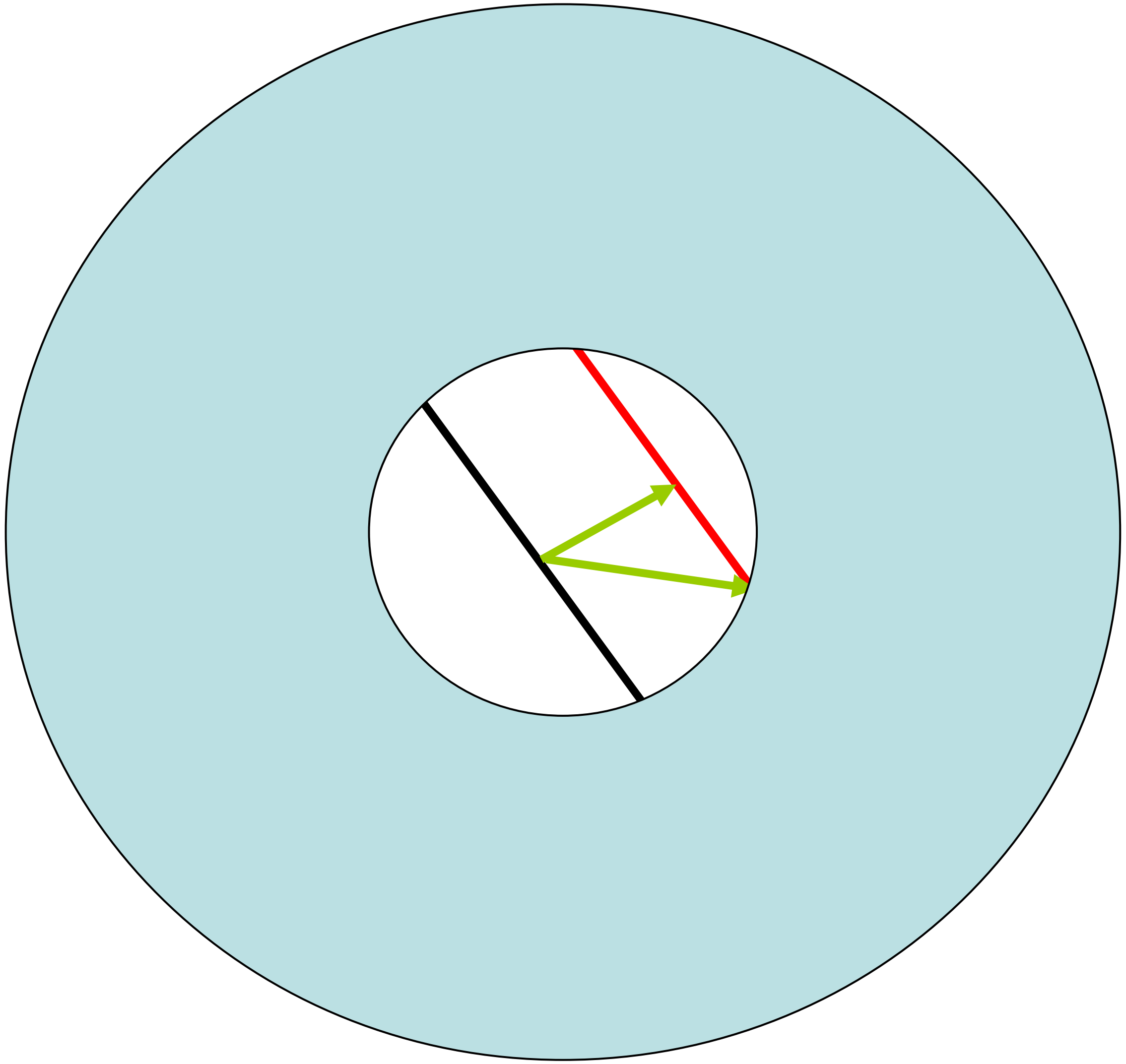
UNIVERSITAT JAUME I

Optical flow equation

$$0 = I_t + \nabla I \cdot [u \;\; v]$$

- One equation and two unknowns *(u,v)*
- Meaning:
  - We can only estimate the optical flow component in the gradient direction.
  - The so-called ”aperture problem”.
- Additional constraints are needed.

The aperture problem



Optical flow estimation

- Avoid the aperture problem.
- Additional constraints:
  - Optical flow is locally smooth
    - Assume pixels in a window have the same *(u,v)*

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \ \ v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{A \\ 25\times2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2\times1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{b \\ 25\times1}}$$

Example: window 5x5= 25 pixels

Lukas-Kanade algorithm

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

Two unknowns d=(u,v)

- Solution: least squares

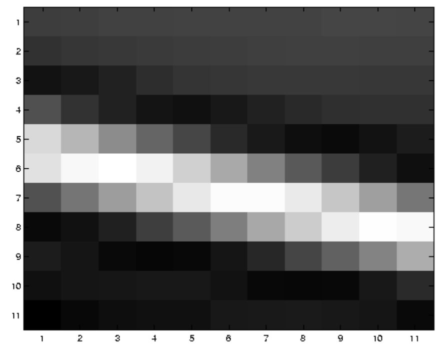
$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

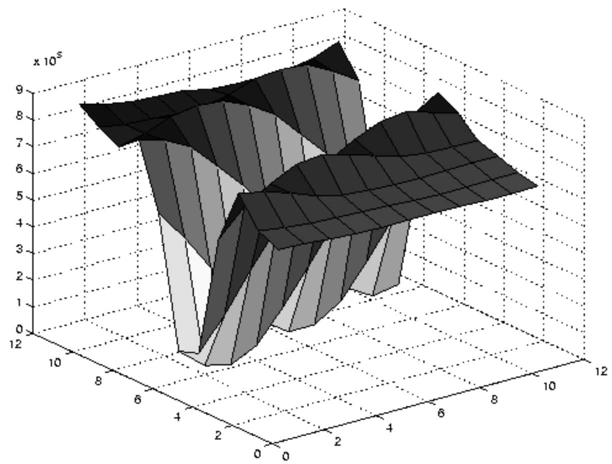
Sum up for all pixels in the window

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Must be invertible.
Not very small eigen values.
One eigen value larger than the other one

Edge pixels



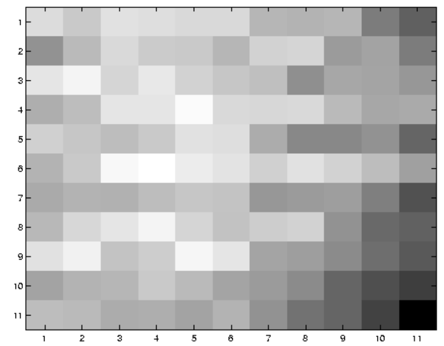


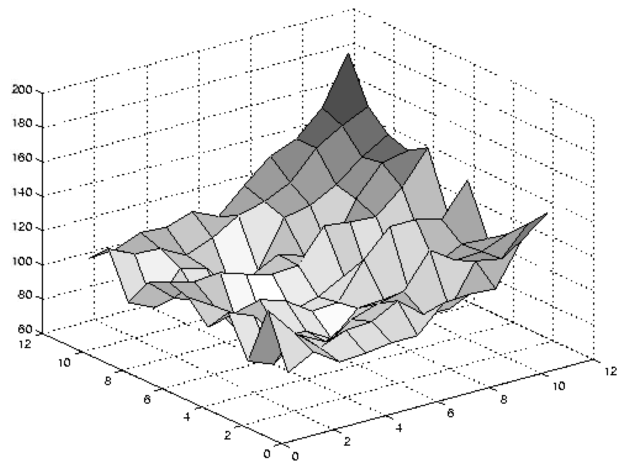
$$\sum \nabla I (\nabla I)^T$$

– large gradients along borders.

– λ_1 large, λ_2 small.

Low textured regions



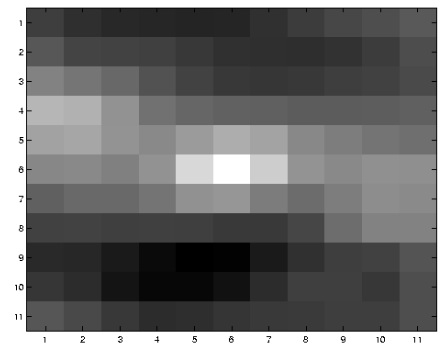


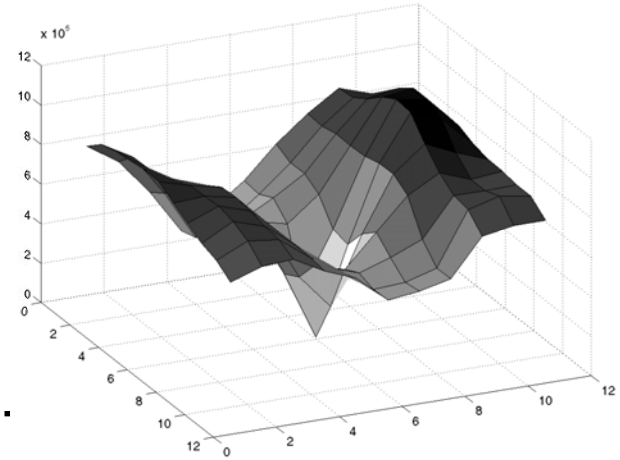
$$\sum \nabla I (\nabla I)^T$$

– small gradient magnitude.

– λ_1 small, λ_2 small

High textured regions





$$\sum \nabla I (\nabla I)^T$$

– different gradients, large magnitudes.

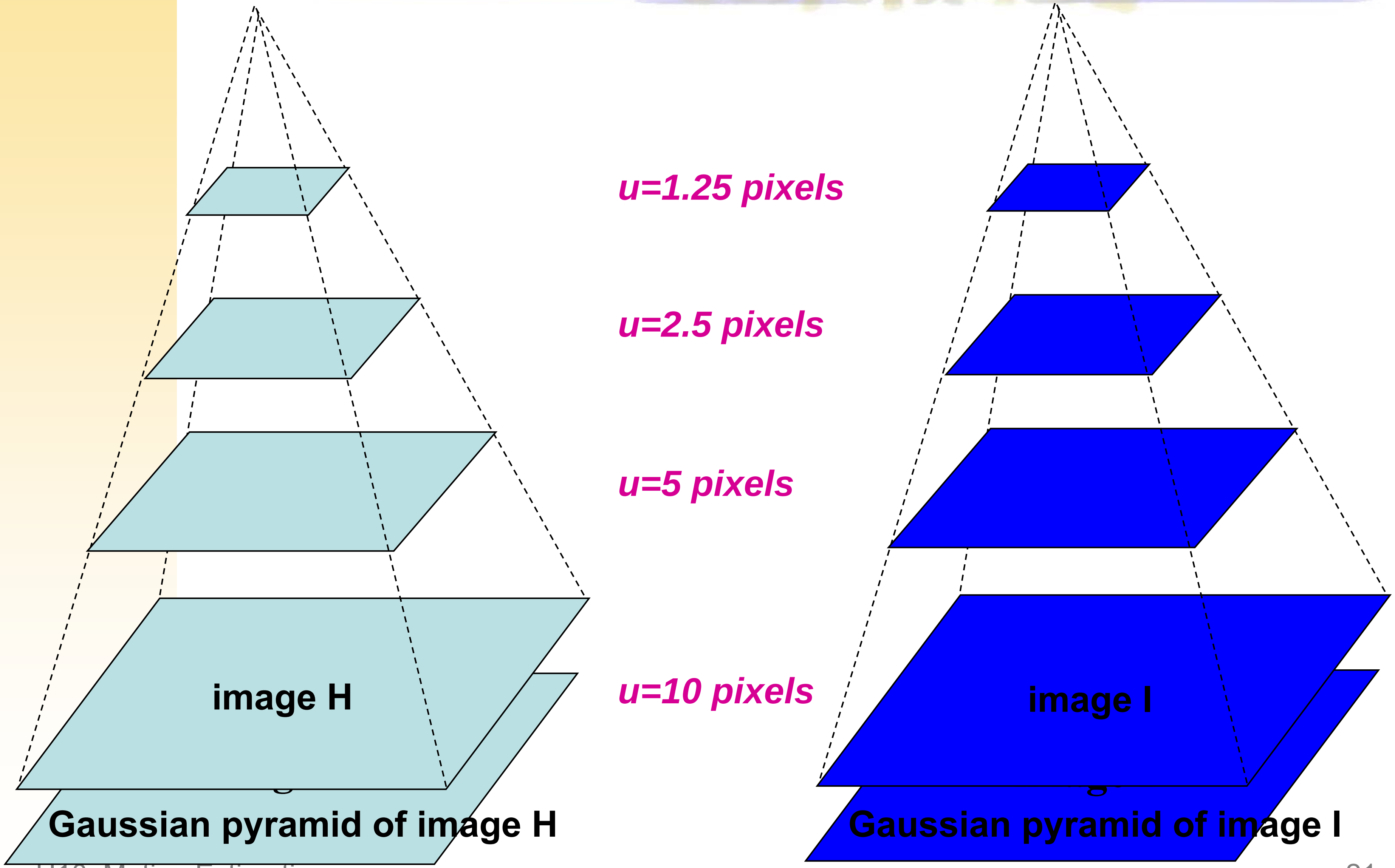
– λ_1, large, λ_2 large

UNIVERSITAT JAUME I

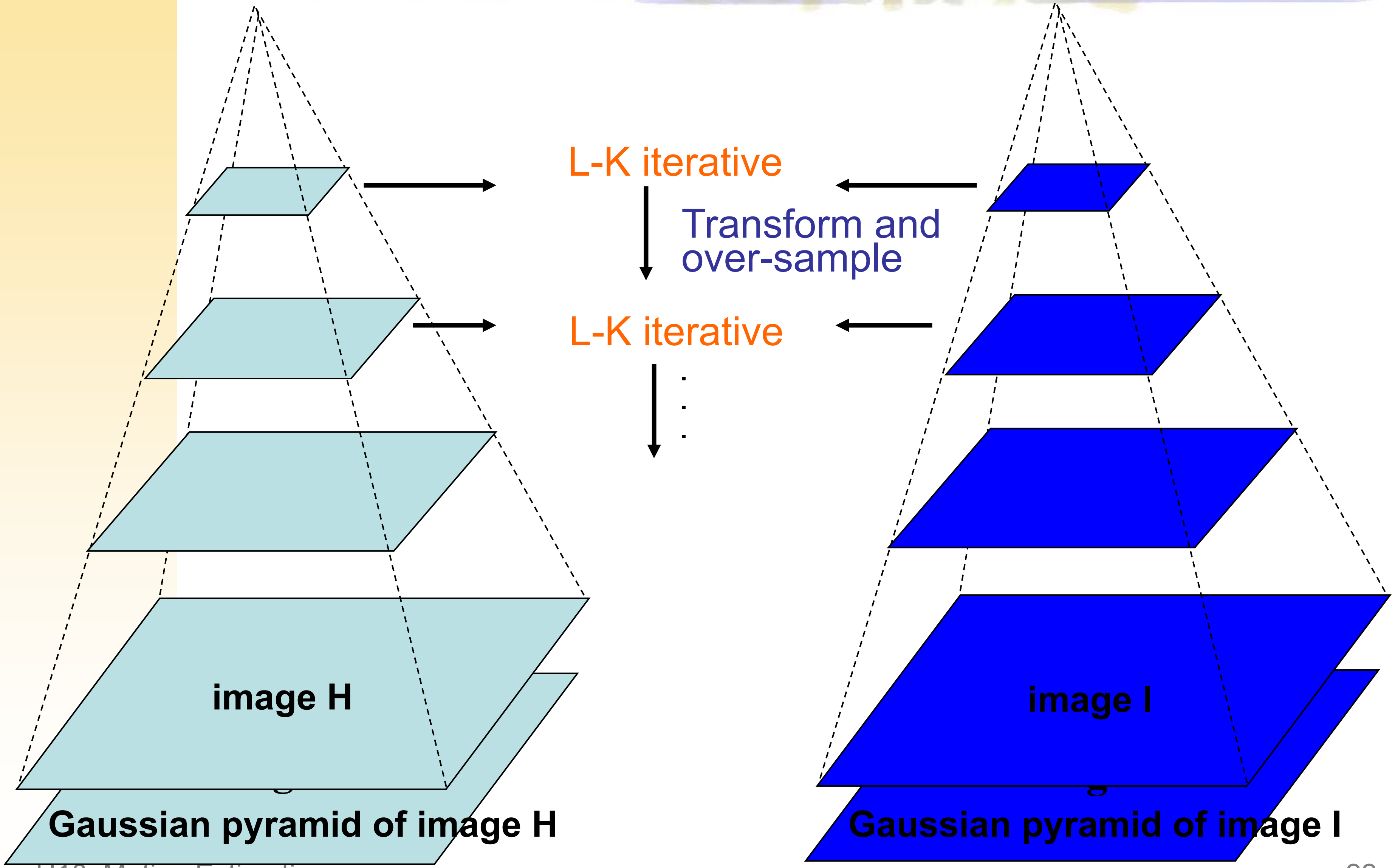
Iterative Lucas-Kanade algorithm

- Velocity estimation of each pixel using the Lucas-Kanade algorithm.
- Transform H into I using the estimated optical flow:
  - Calculate resulting images using interpolation techniques.
- Repeat until convergence.

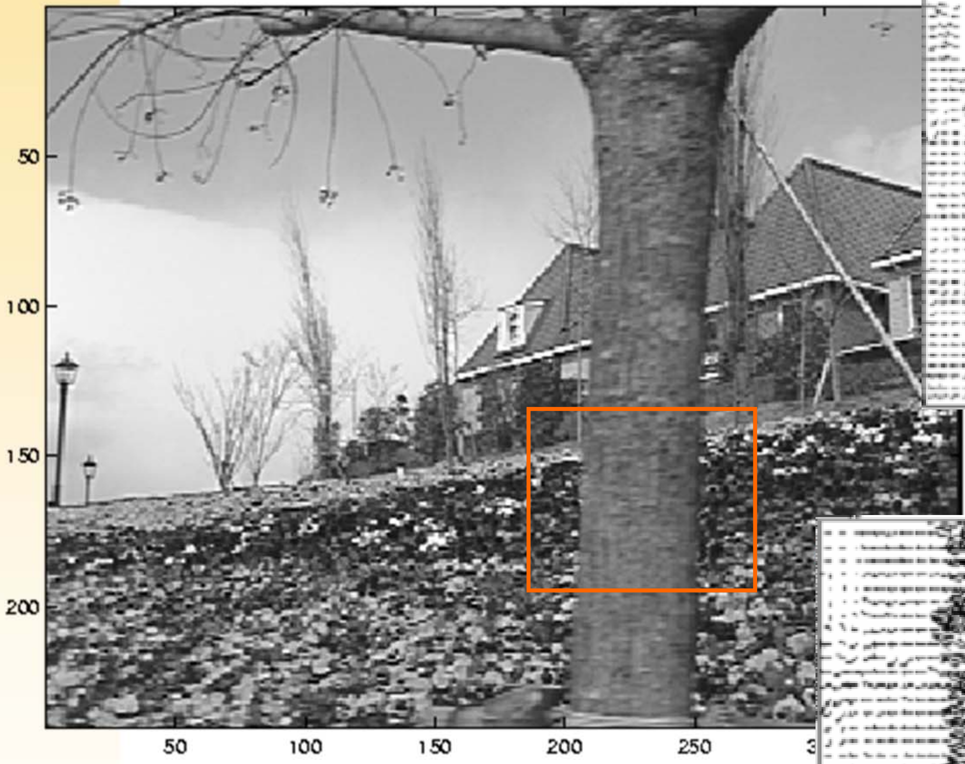
Multiresolution estimation

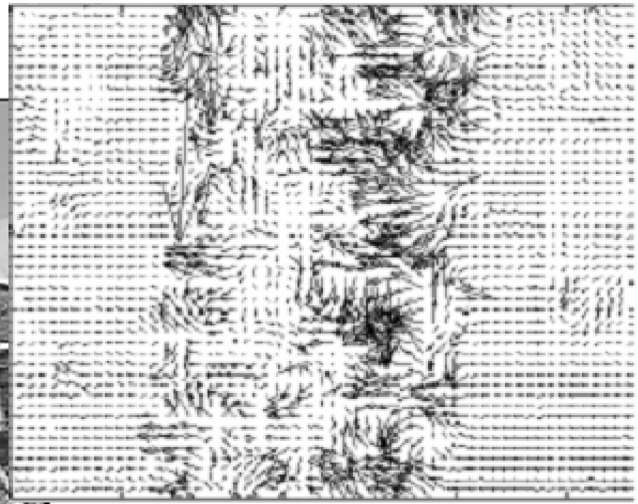


Multiresolution estimation

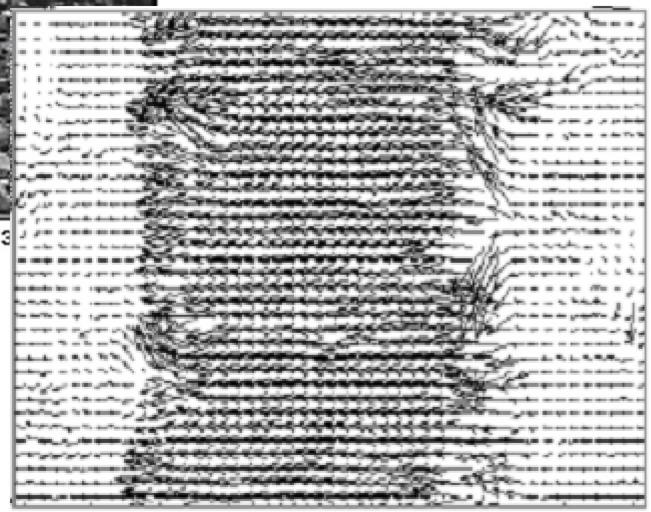


Example





Without multiresolution



With multiresolution

Index

- Application examples.
- Optical flow.
- Global motion: image registration.
  - Gray level-based registration.
  - Criterion functions.
  - Optimization methods.

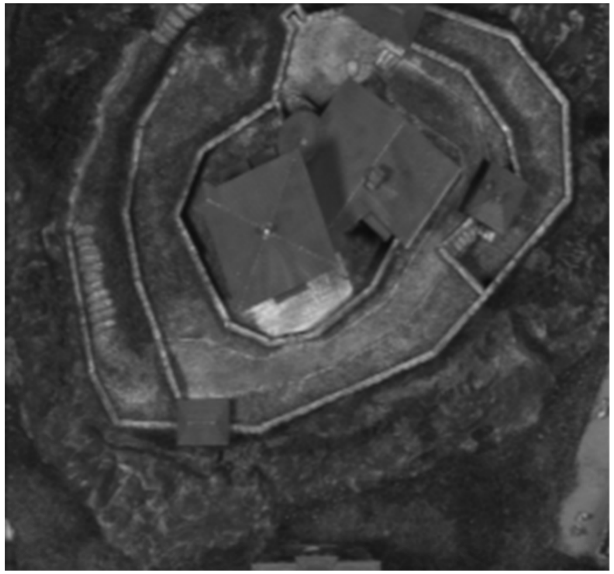
UNIVERSITAT JAUME I

Image registration

- Multiview analysis: Mosaicking, Satellite images, ...
- Temporal analysis: Video segmentation, tracking, ...
- Multimodal analysis: Different sensors (CT-MR, ...)
- Scene-to-model: GIS, Medical imaging, ...

"Find the correspondence between pixels in two images"

"Find a unique geometric transformation that better fits motion of all pixels between two images

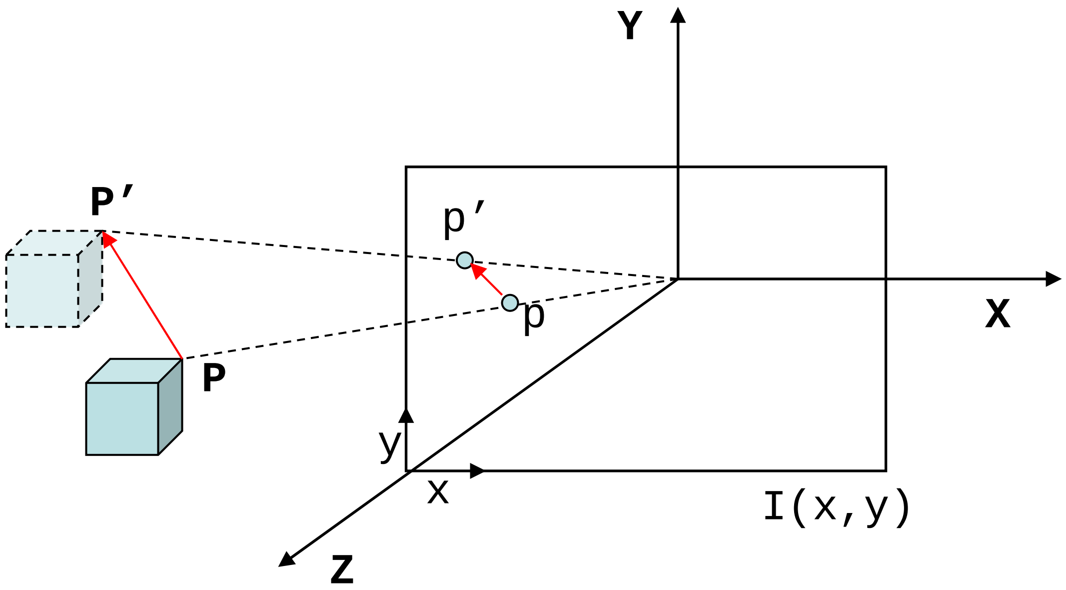


Techniques

- Feature-based methods:
  - Uses feature or interest points:
    - Reduced information.
  - Computationally efficient.

- “Grey level” based methods:
  - Uses raw image information:
    - Overdetermined data.
  - Better accuracy.
  - Optimization methods:
    - Criterion function.

From 3D to 2D motion



- Motion projection of 3D point in a 2D image plane.

UNIVERSITAT JAUME I

2D motion models

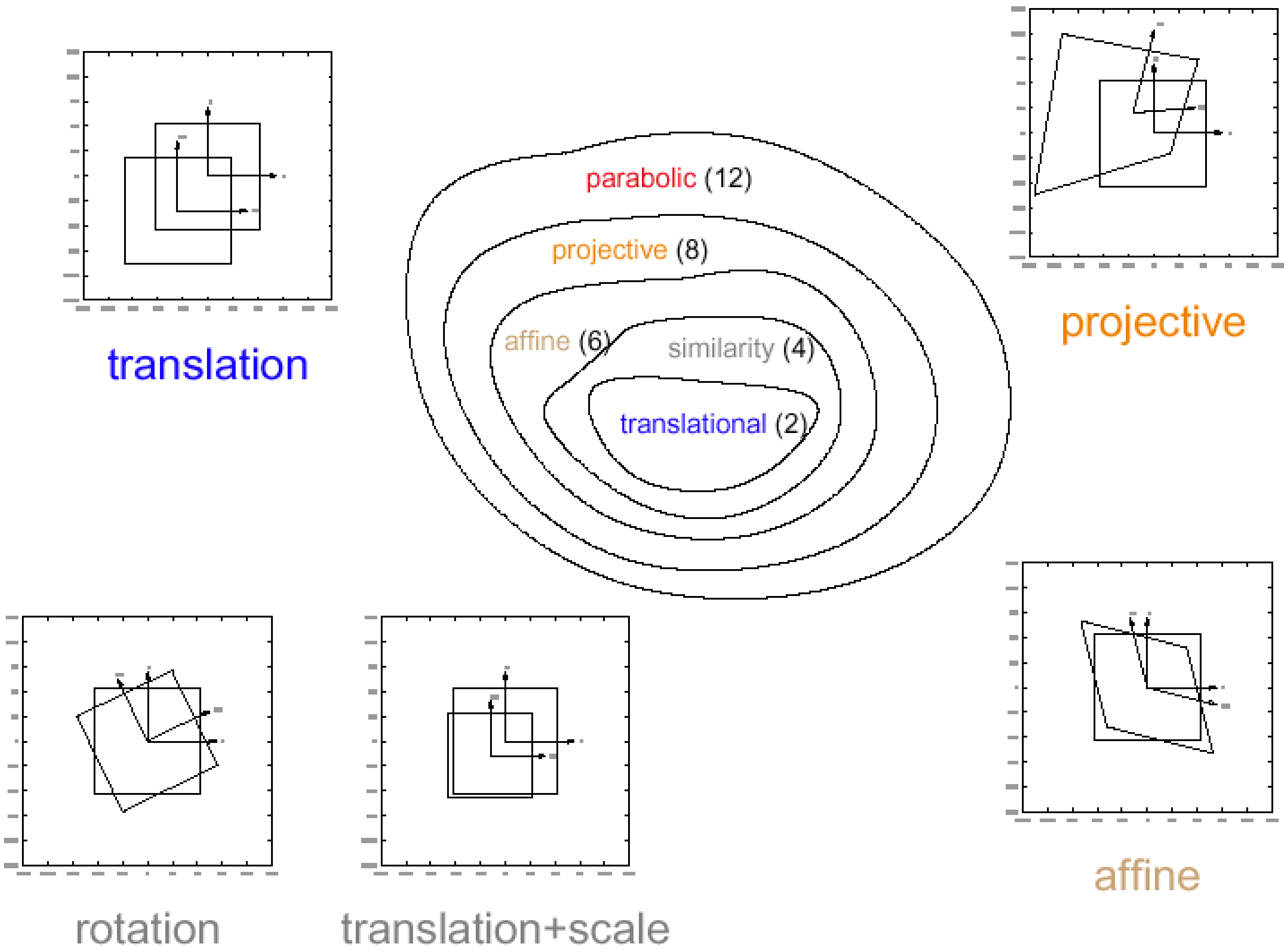
- Affine model:
  - Corresponds to an a la orthographic projection of a 3D plane motion:

$$(x',y')=T(x,y)\begin{cases} x' = a1\ x + a2\ y + a3; \\ y' = a4\ x + a5\ y + a6; \end{cases}$$

- Perspective transformation:
  - Corresponds to a perspective projection of a 3D plane motion.

$$(x',y')=T(x,y)\begin{cases} x' = \dfrac{a1\ x + a2\ y + a3}{a7\ x + a8\ y + 1} \\ y' = \dfrac{a4\ x + a5\ y + a6}{a7\ x + a8\ y + 1} \end{cases}$$

2D motion models



Criterion function

- Define a criterion function:

$$F_i(\chi, L_i)$$

- Estimate parameters which optimize the criterion function:
  - Choose a strategy or minimization method.
  - Example:
    - minimization using least squares:

$$\Theta = \sum_i (F_i(\chi, L_i))^2$$

Grey level-based registration

- Brightness constancy assumption (*BCA*)

$$\Theta_{BCA} = \sum_{(x_i, y_i) \in R} \left(I_1(x'_i, y'_i) - I_2(x_i, y_i) \right)^2$$

- Linearization: Optical flow equation

$$\Theta_{OF} = \sum_{x_i, y_i \in R} \left(I_t + u_{x_i} I_x + u_{y_i} I_y \right)^2$$

Criterion function

- Linearization: Optical flow equation

$$\Theta_{OF} = \sum_{x_i, y_i \in R} (I_t + u_{x_i} I_x + u_{y_i} I_y)^2$$

- Affine motion model

$$x'_i = a_1 x_i + b_1 y_i + c_1$$
$$y'_i = a_2 x_i + b_2 y_i + c_2$$

- Optical flow:

$$u_{x_i} = x'_i - x_i = (a_1 - 1)x_i + b_1 y_i + c_1$$
$$u_{y_i} = y'_i - y_i = a_2 x_i + (b_2 - 1) y_i + c_2$$

Optimization methods

- Ordinary Least Squares (OLS):
  - For lineal functions. Closed-form solution.
  - Sensitive to "outliers".

$$\Theta = \sum_i (F_i(\chi, L_i))^2$$

$$\Theta_{OF} = \sum_{x_i, y_i \in R} (I_t + u_{x_i} I_x + u_{y_i} I_y)^2$$

Jacobian of the criterion function. Matrix (r x p) (observations x parameters)

$$A\chi = d$$

Independent term of the criterion function. Vector (r x 1)

$$\chi = (A^t A)^{-1} A^t d$$

Parameters vector to be estimated. Vector (p x 1)

Optimization methods

- Ordinary Least Squares (OLS):
  - Non linear functions:
    - Newton-Gauss iterative method.

$$\Theta = \sum_i (F_i(\chi, L_i))^2$$

$$\Theta_{BCA} = \sum_{(x_i, y_i) \in R} (I_1(x_i', y_i') - I_2(x_i, y_i))^2$$

$$\chi_j = \chi_{j-1} + \Delta\chi$$

Until increment is small or zero

$$\Delta\chi = (A^t A)^{-1} A^t d$$

Optimization methods

- Newton-Gauss iterative algorithm:

  - Initialize $\chi_0 \quad j = 0$
  - Repeat

$$j = j + 1$$

$$\Delta\chi = (A^t A)^{-1} A^t d$$

$$\chi_j = \chi_{j-1} + \Delta\chi$$

  - Until $|\Delta\chi|$ is sufficiently small.

Optimization methods

- Robust: estimators:
  - Bounded “outliers” influence.
  - M-estimators:
    - Converted into iterative weighted LS estimators.

- Generalized Least Squares (GLS):
  - Use orthogonal (”geometric”) distances.
  - Non linear functions.
  - Residual minimization of observations.
  - Can estimate parameters and modify observation values.

Examples




Examples

Examples

UNIVERSITAT JAUME I

Examples


GLS (BCA)


OLS (OF)


RLS (BCA)


GLS (BCA)


OLS (OF)


RLS (BCA)

References

- Optical flow:
  - D. J. Fleet. “Measuring the image velocity”. Kluwer Academic Publishers. 1992.
  - J.K. Aggarwal, N. Nandhakumar. “On the computation of motion from sequences of images – a review”. Proceeding of the IEEE 76:8, 917-935. 1988.
  - B.K.P. Horn, B. Schunck. “Determining optical flow”. Artificial Intelligence 17, 185-204.1981.

- Segmentation and tracking:
  - Badenas, J.; Sanchiz, J.M. and Pla, F., “Motion-Based Segmentation and Region-Tracking in Image Sequences”, Pattern Recognition, 2001, No. 34, pp. 661-670.

References

- Image registration:
  - J.R. Bergen, P.J. Burt, R.Hingorani, and S. Peleg, “Athree-frame algorithm for estimating two-component image motion,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, pp. 886–896, 1992.
  - J.M. Odobez and P. Bouthemy, “Robust multiresolution estimation of parametric motion models,” Int. J. Visual Communication and Image Representation, vol. 6, no. 4, pp. 348–365, 1995.
  - Montoliu, R. and Pla, F. “Multiple Parametric Motion Model Estimation and Segmentation”, International Conference on Image Processing, ICIP, Thessaloniki, Greece, 2001, ISBN 0-7803-6725-1, Vol. II, pp. 933-936.
  - Z. Zhang, “Parameter-estimation techniques: A tutorial with application to conic fitting,” Image and Vision Computing, Vol 15, no 1, pp. 59–76, 1997.