Master IMA-DIGIT, VISION
Optical flow estimation from a sequence of images

Dominique.Bereziat@lip6.fr
http://perso.lip6.fr/Dominique.Bereziat/VISION

2025-2026

Sorbonne Université

General informations

- 4 sessions focused on 2-D motion estimation
- session 1 (2025-12-16, 1:00 PM): lecture part I, tutorial work
- session 2 (2026-01-06, 1:45 PM): lab 1, to submit on Moodle
- session 3 (2026-01-13, 1:45 PM): lecture part II
- session 4 (2026-01-20, 1:45 PM): lab 2, to submit on Moodle
- Labs 1 and 2 will be assessed and graded

Part I: definition, formalization, earlier methods, local and global approaches

Content

Motion estimation, several problems

- The question: from a sequence of images, how to:
  - detect motion of objects within in images? (what are the moving objects or image structures ?)
  - quantify this motion? (measure of velocity)
  - recover the *real* motion of these objects (*i.e.* in the 3-D scene)

Motion estimation, several problems (cont'd)

- How to define the motion?
- How to recover the motion (depending on the image acquisition context)?
  - natural images: human vision?
  - medical images: physical interaction between photon and matter
  - satellite images: same
  - ...
- Some low-level aspects (recover a dense map of velocity), or high-level aspects (segmentation of moving objects)
- This lecture: centered on the dense estimation of 2-D motion of image structures (also called *optical flow*)

Optical flow: some technical issues to fix

- Robustness w.r.t. change of brightness
- Deformable/rigide objects
- Basic or complex motion
- Occluding
- Large displacements

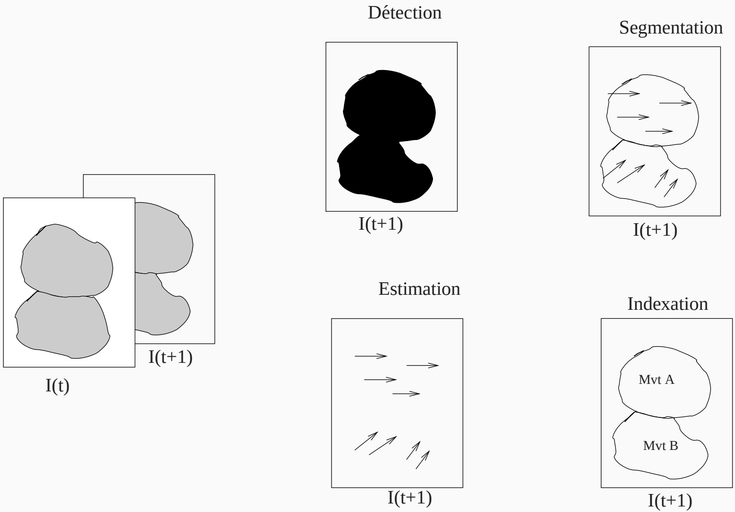
Optical flow: some societal issues

- An old problem (early researches started in 1970)
- Nowadays still an active area of researches
- Motion is every where as a temporal extension of images
- Many industrial domains implications:
    - medical imaging
    - military
    - in remote sensing (oceanography, meteorology, land use, ecology...)
    - remote monitoring (crowd, road, street...)
    - ...

Optical flow: some applications

- Objects tracking (military, video monitoring, robotic, medical. . .)
- Stereovision (disparity map)
- Human movement modeling
- Human behavior analysis, gesture recognition
- Cardiac dynamics analysis
- Video compression
- Motion compensation
- Obstacle detection (autonomous driving/robot)
- 3-D motion reconstruction (autonomous robot, drone)
- Sea surface circulation
- Cellular division analysis, cells tracking

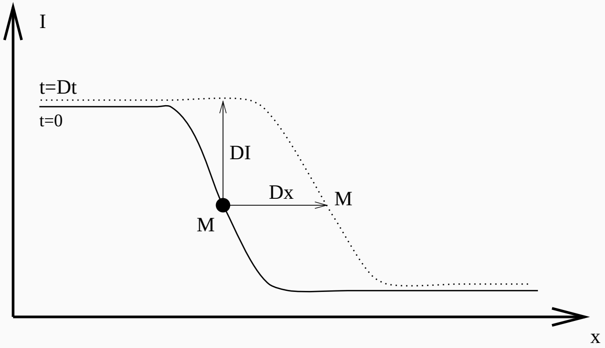
Some problems



Content

Optical flow: definition

- Optical flow: apparent motion of objects/structures/edges perceptible in images, and caused by relative movement between the camera and the object in the scene
- Movement induces a variation of pixels brightness in time and in space



Optical flow: definition (cont'd)

- The basic hypothesis in this lecture is the **brightness constancy**: we assume that pixels inside moving objects keep the same image brightness (color):

$$I(\mathrm{x} + \Delta\mathrm{x}, t + \Delta t) = I(\mathrm{x}, t)$$

with:

- $\mathrm{x} = (x, y) \in \Omega$: spatial position[1]
- Ω: spatial domain (a bounded subspace of $\mathbb{R}^2$) and spatial support of I
- $t \geq 0$: time
- I: sequence of images (function on space and time)
- $\Delta\mathrm{x} = (\Delta x, \Delta y)$: displacement at pixel x
- Δt: time occurring between 2 successive acquisitions

[1] The both notations, x or (x, y), will be used. These notations can also be omitted.

Content

Motion in Fourier space

- Principle: analyze displacements in Fourier space
- Suitable only with a rigid and global motion
- Between times t and $t + \Delta t$ assume an uniform translation (τ_x, τ_y) of the whole image:

$$I(x, y, t) = I(x + \tau_x, y + \tau_y, t + \Delta t)$$

- Apply a Fourier transform on the previous equation:

$$\hat{I}(u, v, t) = \iint_\Omega I(x + \tau_x, y + \tau_y, t + 1) e^{-2i\pi(xu+yv)} dxdy$$

  Ω is the image domain

- Change of variables under the integral: $x' = x + \tau_x, y' = y + \tau_y$:

$$\hat{I}(u, v, t) = \underbrace{\hat{I}(u, v, t + 1) e^{2i\pi(\tau_x u + \tau_y v)}}_{\hat{J}}$$

- Retrieve the translation value by correlation:

$$\text{corr}(\hat{I}, \hat{J}) = \frac{\hat{I}\bar{\hat{J}}}{|\hat{I}||\hat{J}|} = \exp(2i\pi(u\tau_x + v\tau_y))$$

Motion in Fourier space (cont'd)

- Algorithm
  1. Compute FT of images at times t and $t + \triangle t$
  2. Computer correlation
  3. Apply inverse FT on correlation. We should have:

$$\iint_{\Omega}^{(2i\pi(u\triangle x+v\triangle y))} e^{2i\pi(xu+yv)} dudv = \delta_{\tau_x}(x)\delta_{\tau_y}(y)$$

  4. Resulting image should contain 2 symmetric Dirac peaks (local maxima) whose coordinates are component of translation τ
- Extension to rigid deformations:
  - rotation: again with FT, one can retrieve a global rotation
  - change of scale: use a Mellin transform[2]

[2] $\mathcal{M}f(t) = \int_0^{+\infty} x^{t-1} f(x) dx$

Content

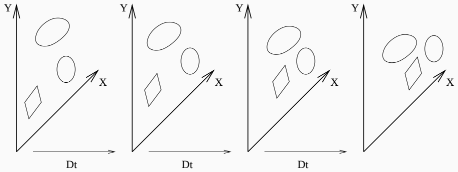
Hough based methods

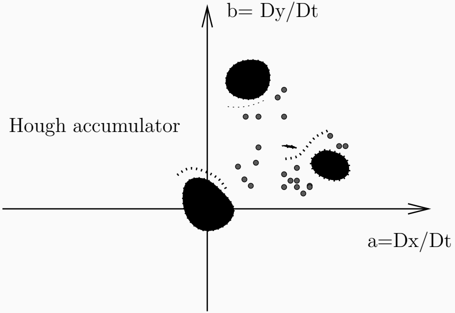
- Principle: vote for all possible displacements in an accumulator space, then analyze the accumulator space
- Again, we assume a brightness constancy moving structures
- Algorithm:
  - Consider two successive frames I_t and$I_{t+\Delta t}$,
  - Init $H_t(a, b)$ the space accumulator: $\forall(a, b), H_t(a, b) = 0$
  - For each pixel $(x, y) \in \Omega_I$ do:
    - For all possible displacement $(a, b) \in \Omega_H$ do:
      If $I_t(x, y) \simeq I_{t+\Delta t}(x + a, y + b)$ then:

$$H_t(a, b) = H_t(a, b) + 1$$

Hough based methods (cont'd)

- Example on a basic example





Hough based methods (cont'd)

- Pro:
  - suited for solid objects
  - algorithm cost independent of the number of objects
  - easy to distribute on many core/thread (warning there is a bottleneck: a reduction on $H_t(a, b)$)
- Cons:
  - high complexity! $O(n^2)$ (but one can restrict Ω_H)
  - many false positives and noise in the Hough accumulator space. Analyze (localization of local maximum) may be challenging
  - Lose of spatial localization

Content

Image Difference based methods

- Historically, the first approaches
- Principle: analyze the temporal difference, pixel by pixel, between two images
- A simplistic hypothesis: temporal difference is caused by a moving object
- Possible definition for image difference:

$$DP_{jk}(x,y) = \begin{cases} 1 & \Leftrightarrow |I(x,y,j) - I(x,y,k)| > \gamma \\ 0 & \end{cases}$$

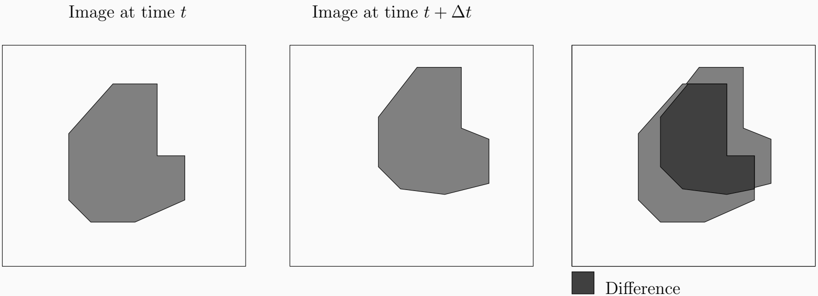


Image Difference based methods (cont'd)

- Other choices possible:
  - adding a connectivity constraint
  - having a local threshold. For instance here, based on a Student test:

$$\gamma(x, y) = \frac{\frac{\sigma_1^2 + \sigma_2^2}{2} + \left(\frac{\mu_1 - \mu_2}{2}\right)^2}{\sigma_1 \sigma_2}$$

    with:

    - (μ_1, σ_1): mean and variance of the first image
    - (μ_2, σ_2): mean and variance of the second image
  - This is mainly used in change detection (land use)
  - But, insufficient to estimate a motion

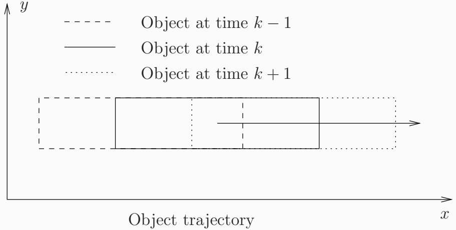
Cumulative Image Difference

- Image difference can be improved to detected quantitatively a movement
- Idea: use an reference image (a scene without moving objects) and accumulate at various time the image difference
- Definition of the cumulative image difference at time index k w.r.t. the reference image r:

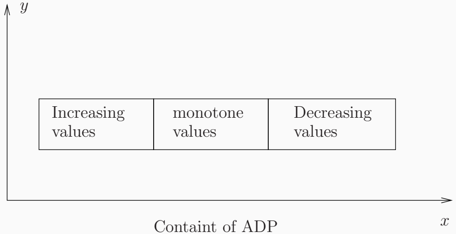
$$\begin{cases} ADP_0(x,y) = 0 \\ ADP_k(x,y) = ADP_{k-1}(x,y) + DP_{rk}(x,y) \end{cases}$$

- The ADP_k image contains the trace of moving objects

Cumulative Image Difference (cont'd)



Object trajectory



Containt of ADP

Cumulative Image difference (cont'd)

- Displacement must be smaller than the length of the object
- Regions with null value: no motion
- Regions with positive values:
  - rate of decreasing and increasing parts give the velocity (for a constant motion)
  - main direction of constant regions give the direction (for rectilinear motion)
- Conclusion:
  - Estimation of motion possible (intensity and direction) if the movement is rectilinear and uniform
  - Low computational cost
  - Historically used for ballistic tracking
  - No change of brightness

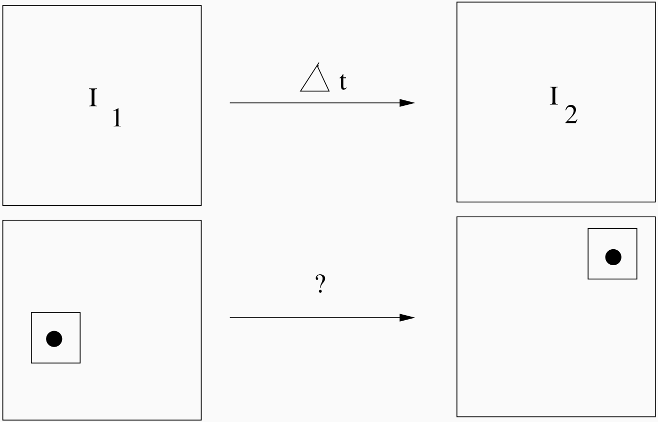
Content

Block-matching methods

- Find the most similar block centered on object to track in the next image
- Well suited for rigid objects
- For example: track a car in a video traffic
- Need to have a *similarity* measure
- Not necessary to have a segmentation of objects: divide the image in several blocks and find matches

Tracking with BM method

- Principle:



Block-matching: formalization

- $\mathrm{x} = (x, y)$ pixel coordinates
- $\delta = (\triangle x, \triangle y)$ displacement vector
- $I(\mathrm{x}, t)$ gray level value of pixel x at time t
- Similarity at pixel x for a displacement δ

$$S(\mathrm{x}, \delta) = \sum_{m \in W(\mathrm{x})} [\quad F(I(\mathrm{x} + m, t) \diamondsuit \\ F(I(\mathrm{x} + m + \delta, t + 1)]$$

- F: filter
- $\diamondsuit$: similarity operator
- $W(\mathrm{x})$: window centered on pixel x

Block-matching: algorithm

- Descriptor S should be the highest for the two most similar regions
- Algorithm:
    - For all pixel x of I at time t:
      find δ maximizing $S(\mathrm{x}, \delta)$
- In practical case: restrict the domain of δ

Some examples of similarities (1)

- Covariance:

$$C_M(\mathrm{x},\delta) = \sum_m W(m)(I(\mathrm{x}+m,t) - \bar{I}(\mathrm{x},t)) \\ \times (I(\mathrm{x}+m+\delta,t+1) - \bar{I}(\mathrm{x}+\delta,t+1))$$

- Correlation:

$$C_V(\mathrm{x},\delta) = \frac{C_M(\mathrm{x},\delta)}{\overline{Var}(I(\mathrm{x},t)) \times \overline{Var}(I(\mathrm{x}+\delta,t+1))}$$

- Binary correlation:

$$C_B(\mathrm{x},\delta) = \sum W(m) B_{I_t}(\mathrm{x}+m) \times B_{I_{t+1}}(\mathrm{x}+m+\delta)$$

with B_I binarization of image I (threshold, edges map...)

Some examples of similarities (2)

- Laplacian correlation:

$$C_L(\mathrm{x}, \delta) = \sum W(m) \triangle I(\mathrm{x} + m, t) \times \triangle I(\mathrm{x} + m + \delta, t + 1)$$

  with $\triangle I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

- Quadratic error (statistic of moment 1)

$$C_E(\mathrm{x}, \delta) = \sum W(m)(I(\mathrm{x} + m, t) - I(\mathrm{x} + m + \delta, t + 1))^2$$

- Crossed entropy, ..., many possible criteria depending on the image structures properties

Block-matching: pros and cons

- The window W size should be well chosen
- Easy to implement
- Cannot provide a dense velocity map (too costly):
    - complexity in $O(n \times p)$, n is the image size, p the window size
    - reduction of the domain of δ
    - look in the right direction (use of a Kalman filter)
- Well suited for tracking rigid objects
- The standard in MPEG4 compression (motion compensation)

Block-matching: example (1)

Loading data ...

Figure 1: Hambourg's Taxis 1

Block-matching: example (2)

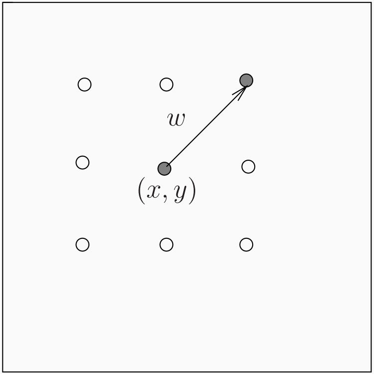
Loading data ...

Figure 2: Hambourg's Taxis 2

Content

Optical flow: definition and formalization

- Hypothesis: along the trajectory of a moving object, intensities are constant



- This apparent motion is called “optical flow”

Optical flow: definition and formalization (cont'd)

- Transport of image values:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t), \forall (x, y) \in \Omega \qquad (1)$$

with:

  - Ω: image domain (a closed set of $\mathbb{R}^2$)
  - $(\delta x, \delta y)$: displacement of point (x, y) at time t

- This is a non linear equation: I is not explicitly defined, it is the data (a sequence of images)!
- Taylor expansion at first order in a neighborhood of (x, y, t):

$$\begin{aligned} I(x + \delta x, y + \delta y, t + \delta t) \approx \quad & I(x, y, t) \\ & + \delta x \frac{\partial I}{\partial x}(x, y, t) \\ & + \delta y \frac{\partial I}{\partial y}(x, y, t) \\ & + \delta t \frac{\partial I}{\partial t}(x, y, t) \end{aligned} \qquad (2)$$

Optical flow: definition and formalization (cont'd)

- Replace right member of (1) by (2) and divide by δt leads to

$$\frac{\partial I}{\partial x}(x,y,t)\frac{\delta x}{\delta t}+\frac{\partial I}{\partial y}(x,y,t)\frac{\delta y}{\delta t}+\frac{\partial I}{\partial t}(x,y,t)=0$$

- Passage to the limit: $\delta t \to 0$,

$$\frac{\partial I}{\partial x}(x,y,t)\frac{\partial x}{\partial t}+\frac{\partial I}{\partial y}(x,y,t)\frac{\partial y}{\partial t}+\frac{\partial I}{\partial t}(x,y,t)=0$$

- We note $(u,v)=\begin{pmatrix}\frac{\partial x}{\partial t} & \frac{\partial y}{\partial t}\end{pmatrix}$, and $\frac{\partial I}{\partial x}=I_x$:

$$I_x(x,y,t)u(x,y,t)+I_y(x,y,t)v+I_t(x,y,t)=0$$

- In the following, we omit (x,y,t);

$$I_x u+I_y v+I_t=0$$

$\Rightarrow$ advection equation

Optical flow: definition and formalization (cont'd)

- Alternative writing:

$$\nabla I \cdot w + I_t = 0$$

with $w = \begin{pmatrix} u & v \end{pmatrix}^T$ (velocity vector), $\nabla I = \begin{pmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{pmatrix}^T$ (image gradient), and $\cdot$ dot product

- This equation also derives from **brightness invariance over time** with chain rule:

$$\frac{dI}{dt} = 0 \quad \textit{(brightness invariance)}$$
$$\Leftrightarrow \frac{dI}{dt}(x(t), y(t), t) = 0$$
$$\Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

Optical flow: definition and formalization (cont'd)

- To summarize, two models:
  - non linear brightness constancy:

$$I(\mathrm{x} + w\delta t, t + \delta t) = I(\mathrm{x}, t) \tag{3}$$

    with $\mathrm{x} = (x, y)$, and $\delta\mathrm{x} = w\delta t$
  - linear brightness constancy, also called "optical flow constraint equation":

$$\nabla I \cdot w + I_t = 0 \tag{4}$$

- Solving linear constraints are easier than non linear ones!

Solving the optical flow constraint equation

- One equation, one vector of $\mathbb{R}^2$:

$$\text{determine } u, v \text{ such as } I_x u + I_y v + I_t = 0$$

is an under-determined problem: one infinity number of solutions:

$$\forall v \;\; u = \frac{-I_y v + I_t}{I_x}$$

- **ill-posed** problem
- It is only possible to determine one direction of vector w...

Solving the optical flow constraint equation (cont'd)

- Let's project w on the image gradient direction: $\frac{\nabla I}{\|\nabla I\|}$

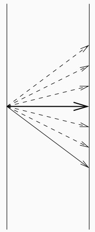
$$\begin{aligned} w_{\nabla I} &= \left\langle \frac{\nabla I}{\|\nabla I\|}, w \right\rangle \frac{\nabla I}{\|\nabla I\|} \\ &= \frac{I_x u + I_y v}{\|\nabla I\|} \frac{\nabla I}{\|\nabla I\|} \\ &= \frac{-I_t}{\|\nabla I\|} \frac{\nabla I}{\|\nabla I\|} \end{aligned}$$

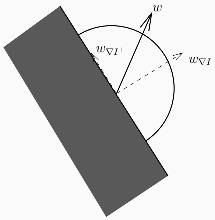
- It is the component of w in the direction of spatial gradient, also called *motion index*
- If the images are sufficiently **textured**, one can theoretically has an unique solution (as solution of the non-linear constancy equation, a matching problem over pixels). Practically, it never happens
- Sufficiently textured: a unique configuration of spatio-temporal gradient in the neighborhood of each pixels

Content

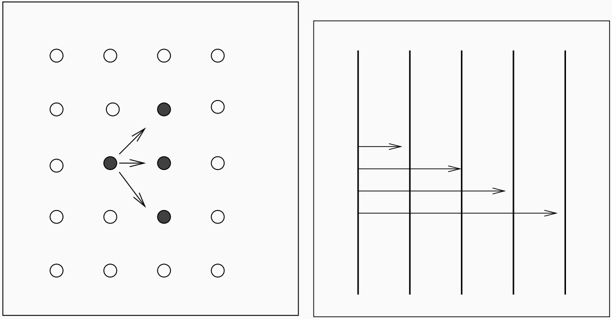
The aperture problem

- The optical flow equation (even for the non linear equation) is not sufficient to determine the motion map
- Locally (i.e. in a neighboring of a pixel), there is an ambiguity to determine the true motion (infinity numbers of solutions)
- *Aperture problem* (Marr 1981): only the component of velocity normal to the local orientation is accessible (motion index)

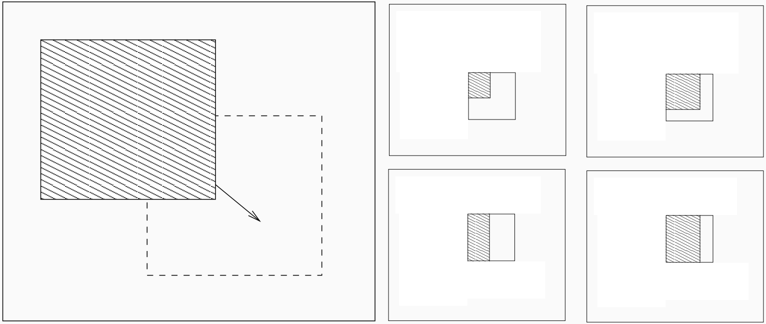




The aperture problem (cont'd)



The aperture problem: what is missing



Additional constraints

- Theoretically: a second constraint on w is necessary
- Two constraints linearly independent: an invertible system
- What constraint? Not an universal answer.
- A first solution: several optical flow constraints:

$$\nabla I^i w + I^i_t = 0, i \in \{1, 2, \cdots\}$$

  I^i are various acquisitions of a same scene:
    - multispectral images
    - images at several point of view (stereovision)
    - filtered images (for instance: Laplacian)
- See [Tistarelli, 1994]

Additional constraints (cont'd)

- Study case: filters to derive several optical flow constraints
- Let's consider the following system:

$$\begin{cases} I_x^1 u + I_y^1 v + I_t^1 = 0 \\ I_x^2 u + I_y^2 v + I_t^2 = 0 \end{cases} \Leftrightarrow Aw = F$$

with

$$A = \begin{pmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} -I_t^1 \\ -I_t^2 \end{pmatrix}$$

is invertible iff $\det(A) \neq 0$ and then if $I_x^1 I_y^2 \neq I_x^2 I_y^1$

- No linear dependencies between ∇I^1 and ∇I^2

Additional constraints (cont'd)

- Example: choose $I^1 = I_x$ and $I^2 = I_y$

$$\begin{cases} (I_x)^1_x u + (I_x)^1_y v + (I_x)^1_t = 0 \\ (I_y)^2_x u + (I_y)^2_y v + (I_y)^2_t = 0 \end{cases}$$

- Then: $A = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$ (Hessian matrix)
- $\det(A) = I_{xx} I_{yy} - I_{xy}^2 \neq 0$
- Issue: second derivatives are very sensitive to noise
- See [Tretiak and Pastor, 1984]
- Rarely, it is possible to consider additional physical constraints (for instance Navier-Stocke for image of pressure)

Solving in Wavelet spaces

- Wavelets: an orthogonal basis allowing both spatial and frequency localization
- Recall: Fourier space is a pure frequency representation (= image structure analysis according to their size)
- Wavelets are compromise between Fourier analysis and spatial analysis (see TADI course)
- Image are projected on the following orthogonal basis:

$$\begin{cases} \psi^s_{jk}(x) = 2^j \psi^s(2^j x - k) \\ (\psi^s)_{s=1\dots S}\text{: a set of mother wavelets} \end{cases}$$

Solving in Wavelet spaces (cont'd)

- Basis ψ_{jk}^{s} is designed to be orthogonal
- The optical flow constraint equation is projected onto the basis:

$$\left\langle \nabla I, \psi_{jk}^{s} \right\rangle \cdot w + \left\langle I_t, \psi_{jk}^{s} \right\rangle = 0, \forall s = 1...S$$

  with $\langle f, g \rangle = \int_{\Omega} f(x) g(x) d\mu(x)$ scalar product associated to the space of integrable functions

- This system of equations is free (linearly independent): it could be solved!
- See C. Bernard thesis: [Bernard, 1999]

Reducing the space of solution

- The space of solution is huge (a functional space of infinite dimension)
- One can reduce it: for instance the subspace of piecewise affine functions

$$w(x,y) = \begin{cases} ax + by + c \\ dx + ey + f \end{cases}$$

  the dimension is now finite (here 6)
- Another spaces are possible: Wavelet, PCA, polynomial...
- Functional subspace with some convenient properties, for instance smooth functions: regularization

Content

[Lucas and Kanade, 1981]

- Initially proposed as a rigid registration method, but suitable to provide a dense velocity map
- Formalization:

  In a neighborhood of x find w_x such as $I_2(x + w_x) = I_1(x) \forall x$

- Minimize of all pixel x:

$$E(w_x) = \sum_{y \in \mathcal{W}_{\mathbf{x}}} \left(I_2(y + w_x) - I_1(y)\right)^2 \quad (5)$$

  where $\mathcal{W}_x$ is a window centered on pixel x

- w is a vector of same size of image, a brute force approach is too costly

Lucas and Kanade method (cont'd)

- The method is said *local*: for a pixel x, the solution is determined in the neighborhood of x
- To avoid brute force: make the problem linear
- Taylor expansion at first order:

$$I_2(\mathrm{y} + w_\mathrm{x}) \sim I_2(\mathrm{y}) + \langle \nabla I_2(\mathrm{y}), w_\mathrm{x} \rangle$$

- Equation (5) writes:

$$E(w_\mathrm{x}) = \sum_{\mathrm{y} \in \mathcal{W}_\mathbf{x}} \left(I_2(\mathrm{y}) - I_1(\mathrm{y}) + \langle \nabla I_2, w_\mathrm{x} \rangle \right)^2 \tag{6}$$

Lucas and Kanade method (cont'd)

- Let's define $I_{21}(\mathrm{y}) = I_2(\mathrm{y}) - I_1(\mathrm{y})$: approximation of temporal derivative
- Term $I_{21}(\mathrm{y}) + \langle \nabla I_2(\mathrm{y}), w_\mathrm{x} \rangle$ is linear w.r.t. w_x
- $\langle \nabla I_2(\mathrm{y}), w_\mathrm{x} \rangle = \nabla I_2(\mathrm{y})^T w_\mathrm{x}$
- I_2 is a row vector such as $(I_2(\mathrm{y}), \quad \mathrm{y} \in \mathcal{W}_\mathrm{x})$, same of I_{21}
- If $\mathcal{W}_\mathrm{x}$ is of size n, the ∇I_2 is a $2 \times n$ matrix:

$$\nabla I_2 = \begin{pmatrix} \frac{\partial I}{\partial x}(\mathrm{y}_1) & \cdots & \frac{\partial I}{\partial x}(\mathrm{y}_n) \\ \frac{\partial I}{\partial y}(\mathrm{y}_1) & \cdots & \frac{\partial I}{\partial y}(\mathrm{y}_n) \end{pmatrix}$$

- I_{21}, ∇I_2 are data, w is the unknown

Lucas and Kanade method (cont'd)

- Eq. (6) writes

$$\begin{aligned} E(w_x) &= \sum_{y \in \mathcal{W}_\mathbf{x}} (I_{21}(y) + \langle \nabla I_2, w_x \rangle)^2 \\ &= \sum_{y \in \mathcal{W}_\mathbf{x}} (B + A w_x)^2 \\ &\text{with} \quad A = \nabla I_2^T, B = I_{21} \end{aligned}$$

- A linear regression!
- Min of E is solution of $Aw_x + B = 0$ but A not a square matrix
- Pseudo inverse:

$$\begin{aligned} Aw_x &= -B \\ A^T A w_x &= -A^T B \\ w_x &= -(A^T A)^{-1} A^T B \end{aligned}$$

  $A^T A$ is now square and invertible if non singular
- Size of $A^T A$: $2 \times 2 \Rightarrow$ determinant formulae to compute the inverse matrix

Lucas and Kanade method: examples

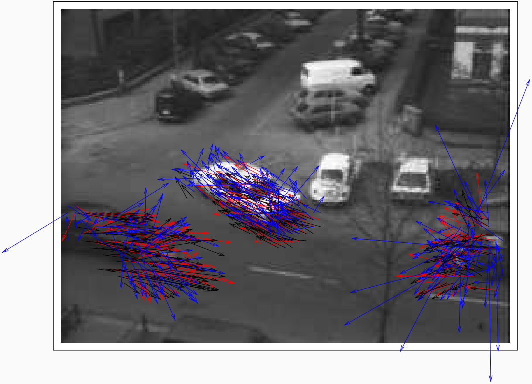


Figure 3: $t_b = 0.9$, $t_h = 5$. Blue: $W = 5$, red: $W = 10$, black: $W = 15$

Lucas and Kanade method: examples (cont'd)

Loading data ...

Figure 4: $W = 10$

Lucas and Kanade method: concluding remarks

- One parameter: the window size
  - robustness to noise if the window is sufficiently large
  - window too large: loss of accuracy (strong regularization)
  - the window size should be adapted to the image structure to analyze
- Fast method (for each pixel, solve a linear system of dimension 2)
- Other types of window are possible: for instance a Gaussian one, to be rotation invariant

Other parametric methods

- Lucas-Kanade can easily extent to polynomial models
- Here piecewise affine motion:

$$w(x, y) = \begin{cases} a + bx + cy \\ d + ex + fy \end{cases}$$

- Lucas-Kanade may be sufficient for translation of rigid objects, affine model is suitable for deformable objects and rotational movements
- Using linear algebra, affine motion writes:

$$w(x, y) = B(x, y)A$$

with

$$B(x, y) = \begin{pmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{pmatrix}$$

and

$$A = (a, b, c, d, e, f)^T$$

Other parametric methods (cont'd)

- Parametric motion models can be extended to capture a *constant* variation of brightness i.e.:

$$\frac{dI}{dt} = \nabla I \cdot w + I_t = -\xi$$

- With the affine model, we have:

$$\nabla I^T B(x,y)A + I_t + \xi = 0$$

with $\Theta^T = (A^T, \xi)$ (7 parameters)

- Cost function to minimize:

$$E(\Theta) = \sum_{i=1}^{n} (\nabla I^T(x_i,y_i)B(x_i,y_i)A + I_t(x_i,y_i) + \xi)^2$$

Other parametric methods (cont'd)

- Let's introduce:

$$\begin{cases} \mathcal{X}_i = (\nabla I(x_i, y_i) B(x_i, y_i), 1) \\ \mathcal{Y}_i = -I_t(x_i, y_i) \end{cases}$$

- Then E writes:

$$E(\Theta) = \sum_{i=1}^{n} (\mathcal{X}_i \Theta - \mathcal{Y}_i)^2$$

- Pseudo inverse to derive the solution of $\operatorname{argmin}_\Theta E(\Theta)$:

$$\widehat{\Theta} = \left(\sum_i \mathcal{X}_i^T \mathcal{X}_i \right)^{-1} \sum_i \mathcal{X}_i^T \mathcal{Y}_i$$

Content

[Horn and Schunck, 1981]

- Historically, the first *variational* algorithm determining optical flow
- Model: a cost function to minimize with
  - a **data term** modeling the brightness constancy
  - a **regularization term** to constraint solution to be smooth

$$E(w) = \underbrace{\int_\Omega (\nabla I \cdot w + I_t)^2 \, dxdy}_{\text{data term}} + \alpha \underbrace{\int_\Omega \|\nabla w\|^2 \, dxdy}_{\text{regularization term}} \tag{7}$$

- E is a *functional* (a function taking the function w as input):

$$E : L^2(\Omega) \to [0, +\infty[$$

  $f \in L^2(\Omega) \Leftrightarrow \int_\Omega f(x,y)^2 dxdy < \infty$
- The minimization of E is achieved using a *gradient descent* method
- A *variational* method: the gradient of E is derived using *calculus of variations*
- A *global* method: E is determined on the whole domain Ω

Horn and Schunck's cost function

$$E(w) = \int_\Omega (\nabla I \cdot w + I_t)^2 \, dxdy + \alpha \int_\Omega \|\nabla w\|^2 \, dxdy$$

- E is a cost function (valued in $\mathbb{R}^+$) and measures a compromise between a data fidelity and solution regularity
- w is the control variable
- α tunes the importance of the regularization
- The space of solution is a vector space of smooth functions (a set of infinite dimension)
- Probabilistic interpretation. Let's suppose:

$$\begin{aligned} \nabla I(\mathrm{x}, t).w(\mathrm{x}, t) + I_t(\mathrm{x}, t) &= \varepsilon_d(\mathrm{x}) \\ \nabla w(\mathrm{x}, t) &= \varepsilon_r(\mathrm{x}) \end{aligned}$$

with ε_d et ε_r two independent Gaussian noises. The solution minimizing (7) is the maximal likelihood estimator of $P(w|I)$

Horn and Schunck's cost function (cont'd)

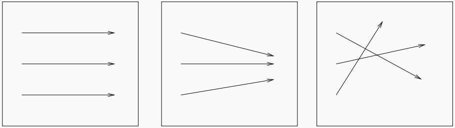
- Intuitively, E small when
  1. the data term
     $$\int_{\Omega} (\nabla I \cdot w + I_t)^2 dx$$
     is small: the optical flow constraint is respected
  2. and the regularization term
     $$\int_{\Omega} \|\nabla w\|^2 dx$$
     is small: meaning??

Horn and Schunck's cost function: meaning of regularization

- Regularization term: $\int_\Omega \|\nabla w\|^2 dx$
- By definition:

$$\begin{aligned}\|\nabla w\|^2 &\equiv \|\nabla u\|^2 + \|\nabla v\|^2 \\ &\equiv \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \\ &\equiv u_x^2 + u_y^2 + v_x^2 + v_y^2\end{aligned}$$

- A small regularization term = velocity vector map locally constant (or close of be null): the vector field is smooth.
- Action of hyper parameter α:



Horn and Schunck: determination of a solution

- E is a convex functional (sum of two quadratic terms)
- Then, a solution of
$$\nabla E(w) = 0$$
is solution of the convex optimization problem
$$\text{argmin}_w E(w)$$
- Gradient of a functional? Gâteau derivative:
$$\langle \nabla E(w), f \rangle = \lim_{h \to 0} \frac{E(w + hf) - E(w)}{h}$$
- Gâteau derivative extent the directional derivative for functional
- $w = (u, v) : \mathbb{R}^2 \to \mathbb{R}^2$: we first determine
$$\left\langle \frac{\partial E(u, v)}{\partial u}, f_u \right\rangle = \lim_{h} \frac{E(u + hf_u, v) - E(u, v)}{h}$$
then the derivative w.r.t. v

Horn and Schunck: obtain the gradient

- From the Gâteau definition, we apply a integration by part and derive (seen in Tutorial work):

$$\nabla E(w) = \begin{pmatrix} \frac{\partial E(u,v)}{\partial u} \\ \frac{\partial E(u,v)}{\partial v} \end{pmatrix} = \begin{pmatrix} 2I_x\left(I_x u + I_y v + I_t\right) - 2\alpha\triangle u \\ 2I_y\left(I_x u + I_y v + I_t\right) - 2\alpha\triangle v \end{pmatrix}$$

- System to be solved:

$$I_x\left(I_x u + I_y v + I_t\right) - \alpha\triangle u = 0$$
$$I_y\left(I_x u + I_y v + I_t\right) - \alpha\triangle v = 0$$

- Approximation of Laplacian: $\triangle u \simeq \bar{u} - u$ with $\bar{u}$(x) the average of neighborhood of x excluding x
- Previous system can be rewritten as (see Tutorial work):

$$(\alpha + I_x^2 + I_y^2)(u - \overline{u}) = -I_x(I_x\overline{u} + I_y\overline{v} + I_t) \tag{8}$$
$$(\alpha + I_x^2 + I_y^2)(v - \overline{v}) = -I_y(I_x\overline{u} + I_y\overline{v} + I_t) \tag{9}$$

Horn and Schunck: determine zero's gradient

- Fixed-point theorem: if the sequence (u_k, v_k) defined such as

$$
\begin{aligned}
(\alpha + I_x^2 + I_y^2)(u^{k+1} - \overline{u}^k) &= -I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t) \\
(\alpha + I_x^2 + I_y^2)(v^{k+1} - \overline{v}^k) &= -I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t) \\
u_0 = v_0 &= 0
\end{aligned}
$$

  converges, its limit is solution of system (8,9)
- Discretization of operator $\overline{f}$:

$$
\begin{aligned}
\overline{f}_{i,j} &= \frac{1}{6}\{f_{i-1,j} + f_{i,j+1} + f_{i+1,j} + f_{i,j-1}\} \\
&\quad + \frac{1}{12}\{f_{i-1,j-1} + f_{i-1,j+1} + f_{i+1,j+1} + f_{i+1,j-1}\}
\end{aligned}
$$

Horn and Schunck's: algorithm

1. Determine spatio-temporal gradients of I: (I_x, I_y, I_t)
2. Choose a number of iterations (in Practical work one could verify that converge is slow...)
3. Choose a suitable value for hyperparameter α (it could be calibrated a ground truth for each type of data...)
4. $u^0 = v^0 = 0$
5. Iterate:

$$
\begin{aligned}
u^{k+1} &= \overline{u}^k + \frac{-I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha + I_x^2 + I_y^2} \\
v^{k+1} &= \overline{v}^k + \frac{-I_{y(}I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha + I_x^2 + I_y^2}
\end{aligned}
$$

Horn and Schunck's: results

Figure 5: Data: Hamburg's taxis, 100 iterations, $\alpha = 30$

Horn and Schunck's: results (cont'd)

Loading data ...

Figure 6: Hamburg: animation

Horn and Schunck's: results (cont'd)

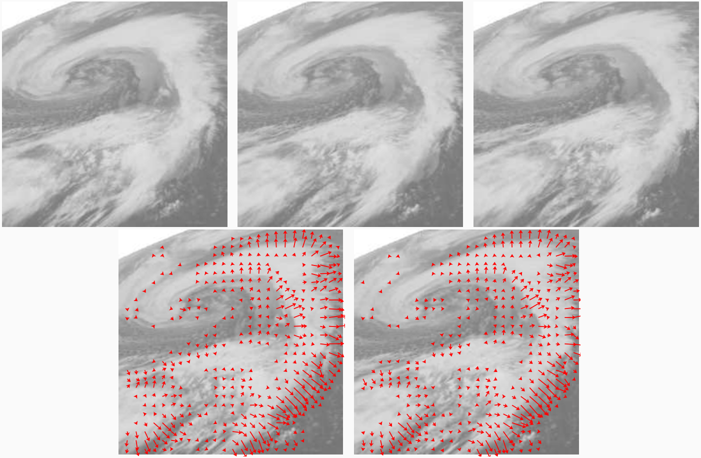


Figure 7: Meteorological infrared images: 100 iterations, $\alpha = 30$

Horn and Schunck's: results (cont'd)

Loading data ...

Figure 8: Meteo: animation

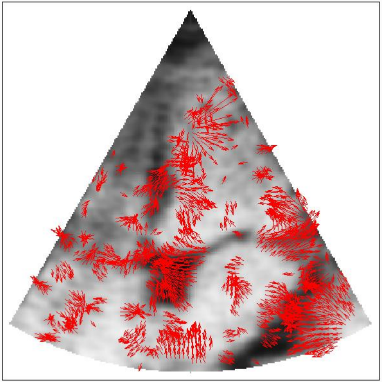


Figure 9: Echocardiography infrasound images (200 iterations, $\alpha = 20$)

Horn and Schunck's: results (cont'd)

Loading data ...

Figure 10: Echo: animation

Horn and Schnuck's: concluding remarks

- Pros:
  - robust method: practically α can be fixed for a class of images
  - fast and easy to implement
  - fine for fluid flow with small displacement
- Cons:
  - number of iterations should be high (at least 100 iterations), can be improved
  - not robust to change of illumination (due to optical flow constraint, example: echocardiography images)
  - regularization: quadratic norm, smooth solutions, does not preserve discontinuities (see Hamburg sequence)
  - linear optical flow constraint remains an approximation (only suitable for displacements up to 2 pixels), how to deal with large displacements?
  - Occlusion?
- These issues will be discussed in the second part of the course

Appendix

Bernard, C. (1999).
***Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière*.**
PhD thesis, École Polytechnique.

Horn, B. and Schunck, B. (1981).
Determining optical flow.
In *Artificial Intelligence*, volume 17, pages 144–156.

Lucas, B. and Kanade, T. (1981).
An iterative image registration technique with an application to stereo vision.
In *IJCAI*, pages 674–679.

Tistarelli, M. (1994).
Multiple constraints for optical flow.
In *Proceeding of ECCV*, pages 61–70.

Tretiak, O. and Pastor, L. (1984).

Velocity estimation from image sequences with second order differential operators.
In *International Conference on Pattern Recognition*, pages 16–19.