Optical flow estimation from a sequence of images

Dominique.Bereziat@lip6.fr
2022-2023
Sorbonne Université
Introduction and definitions
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/rigid 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 everywhere 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
Introduction and definitions

Introduction

Definition

Early approaches

Dense estimation

Horn and Schunck's Issues

Evaluation

Alternatives approaches

Appendix
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):

\[ I(x + \Delta x, t + \Delta t) = I(x, t) \]

with:
- \( x = (x, y) \in \Omega \): spatial position\(^1\)
- \( \Omega \): 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)
- \( \Delta x = (\Delta x, \Delta y) \): displacement at pixel \( x \)
- \( \Delta t \): time occurring between 2 successive acquisitions

\(^1\)The both notations, \( x \) or \((x, y)\), will be used. These notations can also be omitted.
Early approaches
Content

Introduction and definitions

Early approaches

Fourier based methods

Hough based methods

Image Difference based methods

Block-matching

Dense estimation

Horn and Schunck’s Issues

Evaluation

Alternatives approaches

Appendix
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 $(\tau_x, \tau_y)$ 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:

\[
\hat{I}(u, v, t) = \int\int_{\Omega} I(x + \tau_x, y + \tau_y, t + 1)e^{-2i\pi(ux + vy)}\,dx\,dy
\]

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

\[
\hat{I}(u, v, t) = \hat{\hat{I}}(u, v, t + 1)e^{2i\pi(\tau_x u + \tau_y v)}
\]

- Retrieve the translation value by correlation:

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

  $$\int\int_{\Omega} e^{2i\pi(xu+vy)} du dv = \delta_{\tau x}(x)\delta_{\tau y}(y)$$

  4. Resulting image should contain 2 symmetric Dirac peaks (local maxima) whose coordinates are component of translation $\tau$

- **Extension to rigid deformations:**
  
  - rotation: again with FT, one can retrieve a global rotation
  - change of scale: use a Mellin transform

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

Early approaches

Fourier based methods

**Hough based methods**

Image Difference based methods

Block-matching

Dense estimation

Horn and Schunck’s Issues

Evaluation

Alternatives approaches

Appendix
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

\[
\begin{align*}
X & \quad Y \\
\downarrow & \quad \downarrow \\
X & \quad Y
\end{align*}
\]

\[
\begin{align*}
b = \frac{Dy}{Dt} \\
a = \frac{Dx}{Dt}
\end{align*}
\]

Hough accumulator
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 $\Omega_H$)
  - many false positives and noise in the Hough accumulator space. Analyze (localization of local maximum) may be challenging
Introduction and definitions

Early approaches

Fourier based methods
Hough based methods

Image Difference based methods
Block-matching

Dense estimation

Horn and Schunck's Issues

Evaluation

Alternatives approaches

Appendix
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:

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

![Image at time t](image1.png)
![Image at time t + Δt](image2.png)

Difference
• 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:
  • \((\mu_1, \sigma_1)\): mean and variance of the first image
  • \((\mu_2, \sigma_2)\): 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 detect quantitatively a movement.
- Idea: use a reference image (a scene without moving objects) and accumulate at various times 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

Decreasing values
monotone values
Increasing values
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
Introduction and definitions

Early approaches

Fourier based methods
Hough based methods
Image Difference based methods

Block-matching

Dense estimation

Horn and Schunck’s Issues

Evaluation

Alternatives approaches

Appendix
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:

\[ I_1 \]

[Diagram showing two images labeled $I_1$ and $I_2$, with a triangle symbol representing time difference $\triangle t$ and a question mark between the images.]
Block-matching: formalization

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

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

- $F$: filter
- $\diamond$: similarity operator
- $W(x)$: 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 $\delta$ maximizing $S(x, \delta)$
- In practical case: restrict the domain of $\delta$
Some examples of similarities (1)

- Covariance:
  \[ C_M(x, \delta) = \sum_m W(m)(I(x + m, t) - \bar{I}(x, t)) \times (I(x + m + \delta, t + 1) - \bar{I}(x + \delta, t + 1)) \]

- Correlation:
  \[ C_V(x, \delta) = \frac{C_M(x, \delta)}{\text{Var}(I(x, t)) \times \text{Var}(I(x + \delta, t + 1))} \]

- Binary correlation:
  \[ C_B(x, \delta) = \sum_m W(m)B_{I_t}(x + m) \times B_{I_{t+1}}(x + m + \delta) \]

with \( B_I \) binarization of image \( I \) (threshold, edges map...).
Some examples of similarities (2)

- Laplacian correlation:
  \[ C_L(x, \delta) = \sum W(m) \Delta I(x + m, t) \times \Delta I(x + m + \delta, t + 1) \]
  where \( \Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \)

- Quadratic error (statistic of moment 1)
  \[ C_E(x, \delta) = \sum W(m)(I(x + m, t) - I(x + m + \delta, t + 1))^2 \]

- Crossed entropy, …, many possible criteria depending on the image structures properties
• 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 $\delta$
  • look in the right direction (use of a Kalman filter)
• Well suited for tracking rigid objects
• The standard in MPEG4 compression (motion compensation)
Figure 1: Hambourg’s Taxis 1
Figure 2: Hambourg’s Taxis 2
Dense estimation
Content

Introduction and definitions

Early approaches

Dense estimation

Optical flow: definition and formalization

The Aperture Problem

Local regularization based approaches

Global regularization based approaches

Horn and Schunck's Issues

Evaluation

Alternatives approaches

Appendix
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:

\[ I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t), \forall (x, y) \in \Omega \]  \hspace{1cm} (1)

with:

- \( \Omega \): 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 \((x, y, t)\):

\[
I(x + \delta x, y + \delta y, t + \delta t) \approx I(x, y, t) \\
+ \delta x \frac{\partial I}{\partial x}(x, y, t) \\
+ \delta y \frac{\partial I}{\partial y}(x, y, t) \\
+ \delta t \frac{\partial I}{\partial t}(x, y, t) \]  \hspace{1cm} (2)
Optical flow: definition and formalization (cont’d)

- Replace right member of (1) by (2) and divide by $\delta t$ leads to

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

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

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

- We note $(u, v) = \left( \frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right)$, and $\frac{\partial l}{\partial x} = l_x$:

$$l_x(x, y, t)u(x, y, t) + l_y(x, y, t)v + l_t(x, y, t) = 0$$

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

$$l_x u + l_y v + l_t = 0$$

$\Rightarrow$ advection equation
Optical flow: definition and formalization (cont’d)

• Alternative writing:

\[ \nabla I \cdot w + l_t = 0 \]

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

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

\[
\frac{dl}{dt} = 0 \quad (brightness \ invariance)
\]

\[
\Leftrightarrow \frac{dl}{dt}(x(t), y(t), t) = 0
\]

\[
\Leftrightarrow \frac{\partial l}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial l}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial l}{\partial t} = 0
\]
To summarize, two models:

- Non linear brightness constancy:

\[
I(x + w\delta t, t + \delta t) = I(x, t)
\]  

with \( x = (x, y) \), and \( \delta x = w\delta t \)

- Linear brightness constancy, also called “optical flow constraint equation”:

\[
\nabla I \cdot w + I_t = 0
\]  

Solving linear constraints are easier than non linear ones!
Solving the optical flow constraint equation

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

  
  determine $u, v$ 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: \( \frac{\nabla I}{\|\nabla I\|} \)

\[
w_{\nabla I} = \left\langle \frac{\nabla I}{\|\nabla I\|}, w \right\rangle \frac{\nabla I}{\|\nabla I\|} = \frac{l_x u + l_y v}{\|\nabla I\|} \frac{\nabla I}{\|\nabla I\|} = -\frac{l_t}{\|\nabla I\|} \frac{\nabla I}{\|\nabla I\|}
\]

- 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 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
Introduction and definitions

Early approaches

Dense estimation

Optical flow: definition and formalization

The Aperture Problem

Local regularization based approaches

Global regularization based approaches

Horn and Schunck's Issues

Evaluation

Alternatives approaches

Appendix
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:

\[
\begin{align*}
I_1^x u + I_1^y v + I_1^t &= 0 \\
I_2^x u + I_2^y v + I_2^t &= 0
\end{align*}
\]

\[\iff \quad Aw = F\]

with

\[
A = \begin{pmatrix}
I_1^x & I_1^y \\
I_2^x & I_2^y
\end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix}
-I_1^t \\
-I_2^t
\end{pmatrix}
\]

is invertible iff \(\det(A) \neq 0\) and then if \(I_1^x I_2^y \neq I_2^x I_1^y\)

• No linear dependencies between \(\nabla I^1\) and \(\nabla I^2\)
• Example: choose $I^1 = I_x$ and $I^2 = I_y$

$$
\begin{align*}
(l_x)^1_x u + (l_x)^1_y v + (l_x)^1_t &= 0 \\
(l_y)^2_x u + (l_y)^2_y v + (l_y)^2_t &= 0
\end{align*}
$$

• Then: $A = \begin{pmatrix} l_{xx} & l_{xy} \\ l_{xy} & l_{yy} \end{pmatrix}$ (Hessian matrix)

• $\det(A) = l_{xx} l_{yy} - l_{xy}^2 \neq 0$

• 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:

\[
\begin{align*}
\psi_{jk}^s(x) &= 2^j \psi^s(2^j x - k) \\
(\psi^s)_{s=1...S} &\text{: a set of mother wavelets}
\end{align*}
\]
Solving in Wavelet spaces (cont’d)

• Basis $\psi^s_{jk}$ is designed to be orthogonal
• The optical flow constraint equation is projected onto the basis:

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

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
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 \( x \):

\[
E(w_x) = \sum_{y \in \mathcal{W}_x} (I_2(y + w_x) - I_1(y))^2
\]

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
• 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:

$$l_2(y + w_x) \sim l_2(y) + \langle \nabla l_2(y), w_x \rangle$$

• Equation (5) writes:

$$E(w_x) = \sum_{y \in \mathcal{W}_x} (l_2(y) - l_1(y) + \langle \nabla l_2, w_x \rangle)^2$$  \(6\)
Let’s define \( l_{21}(y) = l_2(y) - l_1(y) \): approximation of temporal derivative

- Term \( l_{21}(y) + \langle \nabla l_2(y), w_x \rangle \) is linear w.r.t. \( w_x \)
- \( \langle \nabla l_2(y), w_x \rangle = \nabla l_2(y)^T w_x \)
- \( l_2 \) is a row vector such as \( (l_2(y), y \in \mathcal{W}_x) \), same of \( l_{21} \)
- If \( \mathcal{W}_x \) is of size \( n \), the \( \nabla l_2 \) is a \( 2 \times n \) matrix:

\[
\nabla l_2 = \begin{pmatrix}
\frac{\partial l}{\partial x}(y_1) & \cdots & \frac{\partial l}{\partial x}(y_n) \\
\frac{\partial l}{\partial y}(y_1) & \cdots & \frac{\partial l}{\partial y}(y_n)
\end{pmatrix}
\]

- \( l_{21}, \nabla l_2 \) are data, \( w \) is the unknown
Lucas and Kanade method (cont’d)

- Eq. (6) writes

\[
E(w_x) = \sum_{y \in W_x} (l_{21}(y) + \langle \nabla l_2, w_x \rangle)^2
\]

\[
= \sum_{y \in W_x} (B + A w_x)^2
\]

with \( A = \nabla l_2^T, B = l_{21} \)

- A linear regression!
- Min of \( E \) is solution of \( A w_x + B = 0 \) but \( A \) not a square matrix
- Pseudo inverse:

\[
A w_x = -B
\]

\[
A^T A w_x = -A^T B
\]

\[
w_x = -(A^T A)^{-1} A^T B
\]

\( A^T A \) is now square and invertible if non singular

- Size of \( A^T A \): \( 2 \times 2 \) ⇒ determinant formulae to compute the inverse matrix
Figure 3: \( t_b = 0.9, t_h = 5 \). Blue: \( W = 5 \), red: \( W = 10 \), black: \( W = 15 \)
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 is easy to extent to polynomial models
- Here piecewise affine motion:

\[
\begin{align*}
w(x, y) &= \begin{cases} 
  a + bx + cy \\
  d + ex + fy
\end{cases}
\end{align*}
\]

- 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

\[
A = (a, b, c, d, e, f)^T
\]
• Parametric motion models can be extended to capture a constant variation of brightness i.e.:

\[
\frac{dl}{dt} = \nabla l \cdot w + l_t = -\xi
\]

• With the affine model, we have:

\[
\nabla l^T B(x, y) A + l_t + \xi = 0
\]

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

• Cost function to minimize:

\[
E(\Theta) = \sum_{i=1}^{n} (\nabla l^T (x_i, y_i) B(x_i, y_i) A + l_t(x_i, y_i) + \xi)^2
\]
Other parametric methods (cont’d)

- Let’s introduce:
  \[
  \begin{align*}
  x_i &= (\nabla I(x_i, y_i)B(x_i, y_i), 1) \\
  y_i &= -I_t(x_i, y_i)
  \end{align*}
  \]

- Then \( E \) writes:
  \[
  E(\Theta) = \sum_{i=1}^{n} (x_i\Theta - y_i)^2
  \]

- Pseudo inverse to derive the solution of \( \arg\min_{\Theta} E(\Theta) \):
  \[
  \hat{\Theta} = \left( \sum_i x_i^T x_i \right)^{-1} \sum_i x_i^T y_i
  \]
Introduction and definitions

Early approaches

Dense estimation
  Optical flow: definition and formalization
  The Aperture Problem
  Local regularization based approaches
    Global regularization based approaches

Horn and Schunck's Issues

Evaluation

Alternatives approaches

Appendix
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

\[
E(w) = \int_{\Omega} (\nabla I \cdot w + I_t)^2 \, dxdy + \alpha \int_{\Omega} \| \nabla w \|^2 \, dxdy
\]

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

\[
E : L^2(\Omega) \rightarrow [0, +\infty[ \\
\] 

\[ f \in L^2(\Omega) \Leftrightarrow \int_{\Omega} f(x, y)^2 \, dxdy < \infty \]

- 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 \( \Omega \)
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
- \( \alpha \) 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:

\[ \nabla I(x, t).w(x, t) + I_t(x, t) = \varepsilon_d(x) \]
\[ \nabla w(x, t) = \varepsilon_r(x) \]

with \( \varepsilon_d \) et \( \varepsilon_r \) 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
     $$\int_{\Omega} \left( \nabla I \cdot w + I_t \right)^2 d\mathbf{x}$$
     is small: the optical flow constraint is respected
  2. and the regularization term
     $$\int_{\Omega} \| \nabla w \|^2 d\mathbf{x}$$
     is small: meaning??
Horn and Schunck’s cost function: meaning of regularization

- Regularization term: \( \int_{\Omega} \| \nabla w \|^2 dx \)
- By definition:
  \[
  \| \nabla w \|^2 \equiv \| \nabla u \|^2 + \| \nabla v \|^2 \\
  \equiv \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \\
  \equiv u_x^2 + u_y^2 + v_x^2 + v_y^2
  \]

- A small regularization term = velocity vector map locally constant (or close of be null): the vector field is smooth.
- Action of hyper parameter \( \alpha \):
  
  ![Diagram](image-url)
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
  \[
  \arg\min_w E(w)
  \]
- 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):

\[
\nabla E(w) = \left( \frac{\partial E(u,v)}{\partial u}, \frac{\partial E(u,v)}{\partial v} \right) = \left( \begin{array}{c}
2l_x (l_x u + l_y v + l_t) - 2\alpha \triangle u \\
2l_y (l_x u + l_y v + l_t) - 2\alpha \triangle v
\end{array} \right)
\]

- System to be solved:

\[
l_x (l_x u + l_y v + l_t) - \alpha \triangle u = 0
\]
\[
l_y (l_x u + l_y v + l_t) - \alpha \triangle v = 0
\]

- 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):

\[
\begin{align*}
(\alpha + l_x^2 + l_y^2)(u - \bar{u}) &= -l_x(l_x \bar{u} + l_y \bar{v} + l_t) \\
(\alpha + l_x^2 + l_y^2)(v - \bar{v}) &= -l_y(l_x \bar{u} + l_y \bar{v} + l_t)
\end{align*}
\]
Horn and Schunck: determine zero’s gradient

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

\[
(\alpha + l_x^2 + l_y^2)(u^{k+1} - \bar{u}^k) = -l_x(l_x \bar{u}^k + l_y \bar{v}^k + l_t)
\]

\[
(\alpha + l_x^2 + l_y^2)(v^{k+1} - \bar{v}^k) = -l_y(l_x \bar{u}^k + l_y \bar{v}^k + l_t)
\]

\[u_0 = v_0 = 0\]

converges, its limit is solution of system (8,9)

• Discretization of operator \(\bar{f}\):

\[
\bar{f}_{i,j} = \frac{1}{6}\{f_{i-1,j} + f_{i,j+1} + f_{i+1,j} + f_{i,j-1}\}
\]

\[+ \frac{1}{12}\{f_{i-1,j-1} + f_{i-1,j+1} + f_{i+1,j+1} + f_{i+1,j-1}\}\]
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 \( \alpha \) (it could be calibrated a ground truth for each type of data...)

4. \( u^0 = v^0 = 0 \)

5. Iterate:

\[
\begin{align*}
u^{k+1} &= \bar{u}^k + \frac{-I_x(I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\alpha + I_x^2 + I_y^2} \\
v^{k+1} &= \bar{v}^k + \frac{-I_y(I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\alpha + I_x^2 + I_y^2}
\end{align*}
\]
Horn and Schunck’s: results

Figure 5: Data: Hamburg’s taxis, 100 iterations, $\alpha = 30$
Figure 6: Hamburg: animation
Figure 7: Meteorological infrared images: 100 iterations, $\alpha = 30$
Figure 8: Meteo: animation
Figure 9: Echocardiography infrasound images (200 iterations, $\alpha = 20$)
Horn and Schunck’s: results (cont’d)

**Figure 10:** Echo: animation
Horn and Schnuck’s: concluding remarks

• Pros:
  • robust method: practically $\alpha$ 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 are now discussed in the following
Horn and Schunck’s Issues
Content

Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck’s Issues
  Guided regularizations
  Large displacements
  Illumination changes and occlusion

Evaluation

Alternatives approaches

Appendix
\textbf{L}_2 \text{ regularization is a smoothing process}

- The regularization term (a L_2 norm on velocity gradient) is a smoothing process:

\[
E(w) = \int_{\Omega} (\nabla I \cdot w + I_t)^2 \, dx + \int_{\Omega} \alpha \|\nabla w\|^2 \, dx
\]

\(E_{\text{data}}\) and \(E_{\text{regul}}\)

Gradient: \(\nabla E(w) = 2 \nabla I (\nabla I \cdot w + I_t) - 2 \alpha \Delta w\)

- Let's consider the family \((w_\tau)_{\tau \geq 0}\) of functions defined by:

\[
w(x, 0) = 0 \quad x \in \Omega
\]

\[
\frac{\partial w_\tau(x, t)}{\partial \tau} + \alpha \Delta w_\tau(x, t) = \nabla I (\nabla I \cdot w_\tau(x, t) + I_t) \quad (10)
\]

- Stationary solutions of (10) (i.e. do not depend on \(\tau\)) are solution of \(\nabla E(w) = 0\) (as \(\frac{\partial w_\tau}{\partial \tau} = 0\))

- \(w_\infty = \lim_{\tau \rightarrow \infty} w_\tau\) is a stationary solution
\( L_2 \) regularization is a smoothing process (cont’d)

- Equation (10) is called \textit{Euler-Lagrange} equation associated to the problem of minimizing \( E \) (Eq. (7))
- This is a diffusion equation (see my lecture in TADI on scales spaces) with a forcing term (right member of Eq. (10))
- Discretization of Eq. (10)) leads to a Gauss-Seidel method (iterative methode for matrix inversion, similar to Horn and Schunk method, Eqs. (8) and (9))
- Avoid the smoothing effect induced by diffusion: use non-linear diffusion (guided or not by the image values)
Oriented regularization: [Nagel, 1987]

- Preserved velocity map discontinuities by smoothing along edges contours
- Regularizing term in Horn and Schunck cost function is rewritten as:

\[ E_{\text{regul}} = \int_{\Omega} \alpha \, \text{tr} \left( (\nabla w)^T \nabla w \right) \, dx \, dy \]

- Indeed:
  - \( \nabla w = \begin{pmatrix} \nabla u & \nabla v \end{pmatrix} = \begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix} \)
  - \( \nabla w^T \nabla w = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix} = \begin{pmatrix} u_x^2 + u_y^2 & \ldots \\ \ldots & v_x^2 + v_y^2 \end{pmatrix} \)
  - \( \text{tr}(\nabla w^T \nabla w) = u_x^2 + u_y^2 + v_x^2 + v_y^2 \)
Nagel oriented regularization (cont’d)

• Nagel considers the following norm:

\[ E_{\text{regul}} = \int_{\Omega} \alpha \text{tr} ((\nabla w)^T V \nabla w) \, dx \, dy \]

with \( V \) a \( 2 \times 2 \) matrix such as:

\[ V = \frac{1}{\|\nabla I\|^2 + 2\delta} W \]

\[ W = \begin{pmatrix} I_y^2 + \delta & -I_y I_x \\ -I_x I_y & I_x^2 + \delta \end{pmatrix} \]

• Parameter \( \delta \) allows \( V \) to be invertible: \( \delta = 0 \Rightarrow \det(W) = 0 \)

• \( W \) is divided by a normalization term
• With $\delta > 0$, $V$ is always well defined
• In the following, consider $\delta = 0$
  • $W$ writes:
    \[
    W = \begin{pmatrix} -l_y \\ l_x \end{pmatrix} \begin{pmatrix} -l_y & l_x \end{pmatrix} = U^T U
    \]
    with $U = \begin{pmatrix} l_y & l_x \end{pmatrix}$
  • The regularization term now writes:
    \[
    \int_{\Omega} \text{tr} \left( (U \nabla w)^T (U \nabla w) \right) dx
    \]
    and after expansion:
    \[
    \int_{\Omega} \text{tr} \begin{pmatrix} (-l_y u_x + l_x u_y)^2 & \cdots \\ \cdots & (-l_y v_x + l_x v_y)^2 \end{pmatrix} dx
    \]
Nagel oriented regularization (cont’d)

- Interpretation:
  - if $\nabla u$ and $\nabla v$ have the same direction than $\nabla I$, the regularization is close to zero
  - In this case: no diffusion, no smoothing, discontinuities of $w$ are preserved
  - Along edges no smoothing, outside velocity map is smoothed

- Alternative writing:

$$E_{\text{regul}}(w) = \int_{\Omega} \frac{\alpha}{\|\nabla I\|^2 + 2\delta} \left[ (-l_y u_x + l_x u_y)^2 + (-l_y v_x + l_x v_y)^2 + \delta(\nabla u^2 + \nabla v^2) \right] dx \, dy$$

a combination of an uniform smoothing and an oriented diffusion tuned by $\delta$ and guided by image configuration

- Two parameters drive the regularization: $\alpha$ and $\delta$
Nagel oriented regularization (cont’d)

- Associated Euler-Lagrange equations:

\[
\begin{align*}
  u^{k+1} &= \eta(u^k) - l_x \frac{l_x \eta(u^k) + l_y \eta(v^k) + l_t}{\alpha + l_x^2 + l_y^2} \\
  v^{k+1} &= \eta(v^k) - l_y \frac{l_x \eta(u^k) + l_y \eta(v^k) + l_t}{\alpha + l_x^2 + l_y^2}
\end{align*}
\]

with:

\[
\begin{align*}
  \eta(f) &= \bar{f} - 2l_x l_y f_{xy} - q \nabla f \\
  q &= \frac{1}{l_x^2 + l_y^2 + 2\delta} \nabla l^T \left[ \begin{pmatrix} l_{yy} & -l_{xy} \\ -l_{xy} & l_{xx} \end{pmatrix} + 2 \begin{pmatrix} l_{xx} & l_{xy} \\ l_{xy} & l_{yy} \end{pmatrix} \right]
\end{align*}
\]
Nagel oriented regularization: results

Figure 11: Hamburg’s Taxi sequence, $\delta = 10, \alpha = 25$
Figure 12: Animation
Nagel oriented regularization: results (cont’d)

Figure 13: Hamburg’s Taxi sequence, $\delta = 10$, $\alpha = 1$
Figure 14: Animation
Nagel oriented regularization: concluding remarks

- Allow discontinuities
- Nagel regularization is a non linear diffusion (see TADI, scales spaces)
- Another non linear diffusion schemes, guided by image configurations, are possible:
  - isotropic diffusion [Alvarez et al., 1999]:
    \[ E_{\text{regul}} = \int \phi(|\nabla I|) \| \nabla w \|^2 d\mathbf{x} \]
    with \( \phi \) decreasing function
  - ... 
- Norms can also depends only on velocity map configuration (flow-guided)...
$L_1$ versus $L_2$ norms

- The quadratic term strongly penalizes discontinuities
Consider the minimization problem:

\[ E(w) = \int_{\Omega} (\nabla I \cdot w + I_t)^2 dx + \alpha \int_{\Omega} \left( \sqrt{u_x^2 + u_y^2} + \sqrt{v_x^2 + v_y^2} \right) dx \]

Gradient of \( L_1 \):

\[
\frac{\partial L_1}{\partial u} = -\frac{\partial}{\partial x} \left( \frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) - \frac{\partial}{\partial y} \left( \frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right)
\]

\[
\frac{\partial L_1}{\partial v} = -\frac{\partial}{\partial x} \left( \frac{v_x}{\sqrt{u_x^2 + u_y^2}} \right) - \frac{\partial}{\partial y} \left( \frac{v_y}{\sqrt{u_x^2 + u_y^2}} \right)
\]
Euler-Lagrange equations associated to Cohen’s cost function can be approximated using numerical scheme proposed in Perona and Malik [Perona and Malik, 1990], or Rudin et al. [Rudin et al., 1992]

$L_1$ norm is a particular case of a norm family writing $\Psi(\|\nabla f\|)$ with $\Psi$ a monotone increasing real function

$L_2$ norm: $\Psi(s) = s^2$, $L_1$ norm: $\Psi(s) = |s| = \sqrt{s^2}$

Huber $L_1$ norm (a smooth and derivable $L_1$ norm):

$$\Psi(s) = \begin{cases} 
\frac{z^2}{2\mu} & \text{if } |z| \leq \mu \\
|z| - \frac{\mu}{2} & \text{otherwise}
\end{cases}$$

alternative writing: $\Psi(s) = \sqrt{s^2 + \epsilon}$

Geman-McClure norm: $\Psi(s) = \frac{s^2}{\mu^2 + s^2}$

Lorentz norm: $\Psi(s) = \log(1 + \frac{s^2}{\sigma^2})$
Norm robust to discontinuities

Figure 15: Geman (green), Lorentz (red), $L_1$ (blue)

- Geman and Lorentz norms don’t penalize discontinuities
Robust norms

- General formulation:

\[ E(w) = \int_{\Omega} \psi_1(\nabla I \cdot w + I_t)^2 \, dx + \alpha \int_{\Omega} \psi_2(\|\nabla w\|)^2 \, dx \]

- Assume \( \psi^1 \) and \( \psi^2 \) derivable:

\[ \nabla E(w) = \nabla I \psi'_1(\nabla I \cdot w + I_t) - \alpha \nabla \cdot \left( \frac{\nabla w}{\|\nabla w\|} \psi'_2(\|\nabla w\|) \right) \]

- A robust norm for the data term: allow to be robust to noise and not penalize large deviation to optical flow constraint

- Issue: introducing non linear terms lead to a non convex optimization problem
Consider the non convex cost function:

\[
E(w) = \int_{\Omega} (|\nabla I \cdot w + I_t| + \alpha \|\nabla w\|)dx
\]  \hspace{1cm} (11)

Idea: transform the non convex optimization problem into a series of convex optimization problems

Introduce the auxiliary variable \(w'\) and the new cost function:

\[
E_\theta(w, w') = \int_{\Omega} \left( |\nabla I \cdot w' + I_t| + \frac{1}{2\theta} \|w - w'\|^2 + \alpha \|\nabla w\| \right) dx
\]

When \(\theta\) tends to zero, \(E_\theta\) becomes an approximation of (11) and \(w'\) tends to \(w\)

\(E_\theta\) can be decoupled into two convex optimization problems
Non convex optimization: Zach et al. (cont’d)

- Minimize $E_\theta(w, w')$ w.r.t. to $w$ and $w'$ is equivalent to alternatively minimize the two following convex problems:
  
  1. $w'$ fixed, find $w$ minimizing:
     $$
     \int_\Omega \left( \frac{1}{2\theta} \| w - w' \|^2 + \alpha \| \nabla w \| \right) \, dx \tag{12}
     $$
  
  2. $w$ fixed, find $w'$ minimizing:
     $$
     \int_\Omega \left( |\nabla I \cdot w' + l_t| + \frac{1}{2\theta} \| w - w' \|^2 \right) \, dx \tag{13}
     $$

- Problem (12) has been studied by Rudin et al. [Rudin et al., 1992] in a context of image denoising
- Problem (13) can be solved in a direct way
Zach et al. method: results

- Numerical schemes are available in [Chambolle, 2004]
- Source code: http://www.ipol.im/pub/art/2013/26

**Figure 16:** Horn & Schunck, Nagel, Zack
Zach et al. method: concluding remarks

- Zach et al. deal with a non convex optimization, solved using the split Bregman technique
- $L_1$ norm on $\nabla w$ allows to reconstruct velocity map with discontinuities
- $L_1$ norm on data term: robust to noise and lack of contrast (black taxi velocity better estimated)
- In practical case: the convergence is fast
Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck's Issues
  Guided regularizations
  Large displacements
  Illumination changes and occlusion

Evaluation

Alternatives approaches

Appendix
Large displacements

- The linear optical flow constraint
  \[ \nabla I \cdot w + I_t = 0 \]

  is an approximation of the non linear transport equation
  \[ I(x + w\delta t, t + \delta t) = I(x, t) \]

- In practical case, only available for small displacements \((w\delta t \leq 2)\)
- \(\delta t\) is given by experimental condition, it is not an hyper parameter
- How to deal with large displacements?
  - For instance: can we try to solve the non linear optical flow equation?
  - and is it possible in a variational framework?
Large displacements: solving non linear optical flow equation

- Yes, it is if we can determine the gradient of

\[ E_{\text{data}}(w) = \int_{\Omega} (I(x + w\delta t, t + \delta t) - I(x, t))^2 \, dx \]

- Gâteau derivative:

\[ \lim_{\gamma \to 0} \frac{E_{\text{data}}(u + \gamma f, v) - E_{\text{data}}(u, v)}{\gamma} \]

- Previous expression contains a term in \( \gamma f \) that tends to zero (limit): one can introduce a linear Taylor expansion without error.

- Finally we can derive:

\[ \frac{\partial E_{\text{data}}}{\partial w}(x) = 2\delta t \nabla I(x + w\delta t, t + \delta t)[I(x + w\delta t, t + \delta t) - I(x, t)] \]

- It is not magic: \( \nabla I \) is not explicitly given and obtained by approximation.
Large displacements: solving non linear optical flow equation

- \( \delta t = 1: \) minimize \( \int_{\Omega} (I(x + w, t + 1) - I(x, t))^2 + \alpha \| \nabla w \|^2 \) \( dx \)

- Euler-Lagrange associated equations:

\[
\frac{\partial w}{\partial \tau} = \nabla I(x + w_\tau, t + 1)[I(x + w_\tau, t + 1) - I(x, t)] - \alpha \Delta w_\tau
\]

- Approximated by an Euler scheme:

\[
\frac{\partial w}{\partial \tau}(k\lambda) \simeq \frac{w^{k+1} - w^k}{\lambda}
\]

and a semi-implicit scheme\(^3\):

\[
w^{k+1} + \lambda \Delta w^{k+1} = w^k + \alpha \lambda \nabla I(x + w^k, t + 1)[I(x + w^k, t + 1) - I(x, t)]
\]

- Need to evaluate \( I(x + w^k, t + 1) \) and \( \nabla I(x + w^k, t + 1) \) using bilinear interpolation\(^4\)

\(^3\)Due to numerical considerations, see TADI lecture on scales spaces

\(^4\)\( x + w^k \) do not belong to the spatial grid
Large displacements: multi-resolution approaches

[Anandan, 1989, Black and Anandan, 1991]

- Principle of multi-resolution/multi-grid approaches:
  - from data, build a hierarchy of resolution (as a series of low-pass filter and $2 \times 2$ subsampling),
  - start from the lowest resolution, compute a first guest
  - from a coarse resolution to the next finer: compute an accurate solution

- Applied to optical flow estimation: at each resolution the hypothesis of small displacements (linear optical flow) holds:
  1. At the coarsest resolution (image of size $2 \times 2$): the linear optical flow equation is correct (at most displacement of one pixel)
  2. From a resolution to the next fine: the upsampled optical flow is refined with a $2 \times 2$ local estimation

```
coarse resolution                                  coarse optical flow estimation
                                      2 \times 2 downsampling
fine resolution                                  2 \times 2 upsampling
                                      fine optical flow estimation
```
Large displacements: building the pyramid of resolutions

- \( I(x, y, t) \) original image (level 0, finest resolution): \( I^0(x, y, t) \)
- level \( k \) to level \( k + 1 \):
  \[
  I^{k+1}(x, y, t) = \downarrow (I^k * G_\sigma)(x, y, t)
  \]
- \( \downarrow \) downsampling operator (keep 1 pixel over 4)
- Anti-aliasing filter: Gaussian smoothing with standard deviation of \( \sigma = 2 \)
- \( \Omega^k \) spatial domain of level \( k \) verifying:
  \[
  \Omega^N \subset \cdots \subset \Omega^{k+1} \subset \Omega^k \subset \cdots \subset \Omega^0
  \]
- Minimal resolution, level \( N \): an image reduced to \( 2 \times 2 \) pixels
- We have \( N = \log_2 |\Omega| - 1 \).
Large displacements: building the pyramid of resolutions (cont’d)

Pyramid of resolutions (here two levels):

\[ w^0 = \uparrow w^1 + dw^0 \]
Large displacements: compute velocity at level $k$ from level $k + 1$

- Notation: $w^k$ velocity at level $k$
- $dw^k$: increment of velocity computed at level $k$ such that:

$$\uparrow w^{k+1} + dw^k = w^k$$

with $\uparrow$ the upsampling operator

- Non linear optical flow constraint at level $k$:

$$D^k(x, t) = l^k(x + w^k dt, t + dt) - l^k(x, t)$$
$$= l^k(x + (\uparrow w^{k+1} + dw^k) dt, t + dt) - l^k(x, t)$$
$$= 0$$

- $w^{k+1}$ is given (estimation at coarse resolution), $dw^k$ explains velocity difference between levels $k + 1$ and $k$: $2 \times 2$ upsampling, so $|du^k|, |dv^k| \leq 2$, the linear optical flow is a correct approximation
Large displacements: compute velocity at level $k$ from level $k + 1$ (cont’d)

• We write:

$$I^k(x + (\uparrow w^{k+1} + dw^k)dt, t + dt) = I^k((x + \uparrow w^{k+1} dt) + dw^k dt, t + dt)$$

• First order Taylor expansion of $I^k$ at pixel $x + \uparrow w^k + 1 dt$:

$$I^k((x + \uparrow w^{k+1} dt) + dw^k dt, t + dt) \simeq I^k(x + \uparrow w^{k+1} dt, t + dt) + \nabla I(x + \uparrow w^{k+1}, t + dt) dw^k dt$$

• $D^k$ becomes:

$$D^k(x, t) = I^k(x + \uparrow w^{k+1} dt, t + dt) - I^k(x, t)$$

$$+ \nabla I(x + \uparrow w^{k+1} dt, t + dt) dw^k dt$$

$$= 0$$
Large displacements: compute velocity at level $k$ from level $k+1$ (cont’d)

- Let’s introduce the shifted image difference between level $k$ and $k+1$:

$$I^k_{\text{shift}}(x, w^{k+1}, t) = I^k(x + \uparrow w^{k+1} dt, t + dt) - I^k(x, t)$$

- Equation $D^k = 0$ writes:

$$\frac{1}{dt} I^k_{\text{shift}}(x, w^{k+1}, t) + \nabla I^k(x + \uparrow w^{k+1} dt, t + dt) dw^k = 0 \quad (14)$$

- Eq. (14) is called *incremental optical flow equation*, it is of same nature that the linear optical flow equation Eq (4) with a shifted spatial gradient and a shifted temporal gradient as data

- $dw^k$ can be obtained with one of the optical flow methods previously studied, for instance (Horn & Schunk, Lucas & Kanade...), see:
  - global approach: [Proesmans et al., 1994]
  - local approach: [Bergen et al., 1992]
1. Build the pyramid of resolution $I^k$

2. Coarse level $N$: $w^N = \vec{0}$, estimation of $dw^N$

3. Level $k$: estimation of $dw^k$ from $w^{k+1}$ and $I^k$ by solving:

$$\frac{1}{dt} l_{shift}^k(x, \uparrow w^{k+1}, t) + \nabla l_{shift}^k(x, \uparrow w^{k+1}, t) dw^k = 0$$

4. Update $w^k = w^{k+1} + dw^k$, $k = k - 1$

5. Iterate steps 3. and 4. up to $k = 0$
Content

Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck’s Issues
  Guided regularizations
  Large displacements
  Illumination changes and occlusion

Evaluation

Alternatives approaches

Appendix
Global illumination change is a common issue.

Model: \[ I(x, t + 1) = I(x + w, t) + c, \ c \text{ biais} \]

Simple remark:
\[ I(x, t + 1) = I(x + w, t) + c \implies \nabla I(x, t + 1) = \nabla I(x + w, t) \]

Brox et al proposal: two data term in the cost function
  - one for brightness constancy
  - one for gradient brightness constancy

Two constraints, but the problem remains ill-posed, why?

Cost function:
\[
E(w) = \int \| I(x + w, t + 1) - I(x, t) \|^2 dx
+ \int \gamma \| \nabla I(x + w, t + 1) - \nabla I(x, t) \|^2 dx
+ \int \alpha \| \nabla w \|^2 dx
\]
Object occlusion

- Occlusion occurs when an object is in front of another one
- The optical flow equation does not hold for occluded objects
- What can we do?
  - detect regions of occlusion: estimation of velocity will be not relevant in these regions
  - extrapolate, interpolate velocity map on these regions
A 2-stage algorithm:

1. detection of the occlusion regions:
   - Estimation of optical flow between images 1 and 2: \( w_{12} \)
   - Estimation of retrograd optical flow, i.e. between images 2 and 1: \( w_{21} \)
   - Occlusion at pixel \( x \) if \( w_{12} \) is significantly different from \( -w_{21} \)

2. Estimation of velocity inside the occlusion regions:
   - Use of an in-painting method (see TADI lecture on scale space): use of guided norm, \( w \) is smoothing in the direction of \( w \) inside these regions

- Stages 1. and 2. are repeated until convergence
- Joint estimation of optical flow and inpainting is also possible
Horn and Schunck issues: concluding remarks

**Figure 17:** Large displacements, discontinuous vector field, occlusions

**Figure 18:** Ground truth, Horn and Schunk (1981), Sun *et al* (2010)
Evaluation
Evaluation: introduction

- Difficulty of evaluation without ground truth
- Ground truth remains possible in some cases
  - synthetic images (coming from computer graphics)
    - useful as proof of concept
    - but not always realistic
  - real data
    - possible in some cases (controlled or known rigid/articulated motions, \textit{in situ} measures)
    - but costly and complex to set up
- others and general cases
  - human validation: measuring displacement of objects/regions/points of interest
Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck’s Issues

Evaluation

Introduction

Qualitative evaluation (visualization)

Quantitative evaluation (benchmarks)

Alternatives approaches

Appendix
Evaluation by visualization: vector field

- Comparison of sparse vector field
- Example of Hamburg’s taxis (no ground truth)

Figure 19: Horn and Schunck (red), Zack et al (blue)

- With Matlab/Matplotlib: quiver()
Evaluation by visualization: Middlebury colormap

- Dense representation: Middlebury colormap\(^5\)
- Color wheel: velocity direction, color saturation: velocity magnitude

**Figure 20:** \(L_2\) (left), TV-\(L_1\) (right)

\(^5\)http://vision.middlebury.edu/flow/
Evaluation by visualization: velocity magnitude

- Dense representation with velocity magnitude (norm):
  \[ \| w \| = \sqrt{u^2 + v^2} \]

**Figure 21:** $L_2$ (left), TV-$L_1$ (right)
Evaluation by visualization: stream lines

- Stream lines: trajectory of point $x_0 \in \mathbb{R}^2$ transported by a static vector field $w(x)$ (here, stationary velocities, no time)

- Solve:

  \[
  \frac{\partial x}{\partial s}(s) = w(x(s)) \quad s \in [0, 1] \\
  x(0) = x_0
  \]  

  (15)

- Solution (integration):

  \[
  x(s) = x_0 + \int_0^s w(x(u)) du
  \]

- Resolution using a 4-order Range-Kutta scheme (i.e. $w(x(u))$ is evaluated by bilinear interpolation)
Evaluation by visualization: stream lines (cont’d)

- Stream lines:

Figure 22: function stream2() (Matlab) streamplot() (Matplotlib)
Evaluation by visualization: Line Integral Convolution

- Line Integral Convolution (LIC): dense visualization of stream lines
- Determination of stream lines, Eq (15)
- Integration using the following way:

\[
\text{LIC}(x_0) = \int_{\mathbb{R}} k(u - u_0) \ T(x(u)) \ du
\]

\[
x_0 = x(u_0)
\]

- \(T\): image of texture (uniform noise)
- Convolution kernel \(k\) determines a window over the stream line:
  - \(k(u) = \frac{1}{2L} 1_{[-L,+L]} \)
  - or \(k\) Gaussian kernel of variance \(L\)
Figure 23: $L_2$ (left), TV-$L_1$ (left)
Evaluation by visualization: trajectories

- Temporal trajectory: points transported by a non stationary velocity field $w(x, t)$

- Modification of Eq (15):
  \[
  \frac{\partial x}{\partial t} = w(x, t) \\
  x(0) = x_0
  \]

- Integration:
  \[
  x(t) = x_0 + \int_0^t w(x(u), u)\,du
  \]

- Use of 4-order Rung-Kutta scheme
Evaluation by visualization: trajectories (cont’d)

- Matlab: `stream3()`, Matplotlib?
- Can be combined with LIC sort a dense visualization

Figure 24: $L_2$ (red), TV-$L_1$ (blue)
Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck’s Issues

Evaluation
  - Introduction
  - Qualitative evaluation (visualization)
    - Quantitative evaluation (benchmarks)

Alternatives approaches

Appendix
Quantitative evaluation

- Comparison with the **ground truth**
- How to compare?
  visually, with statistics
- How to obtain a ground truth? Use of *twin* experiments

![Diagram showing optical flow estimation and comparison with ground truth]
Quantitative evaluation: error measurements

- Let $w$ be the reference, $\tilde{w}$ the estimated
- Angular error: $\varepsilon_{AE} = \langle w, \tilde{w} \rangle = \arccos\left( \frac{w^T \tilde{w}}{\|w\| \|\tilde{w}\|} \right)$
- Angular error in space-time ([Fleet and Jepson, 1990]):
  $\varepsilon'_{AE} = \langle (w, 1), (\tilde{w}, 1) \rangle = \arccos\left( \frac{1 + w^T \tilde{w}}{\sqrt{(1 + \|w\|^2)(1 + \|\tilde{w}\|^2)}} \right)$

Figure 25: Angular error in space, and in space-time
Quantitative evaluation: error measurements (cont’d)

- Relative Norm Error: \( \varepsilon_{RNE} = \frac{\|w\| - \|\tilde{w}\|}{\|w\| + \varepsilon} \)
- End Point Error: \( \varepsilon_{EPE} = \|w - \tilde{w}\| \) warning: an absolute error, relevant for comparison.
- Relative End Point Error: \( \frac{\|w - \tilde{w}\|}{\|w\| + \varepsilon} \)
- Final statistics: mean and standard deviation of these error maps
Quantitative evaluation: Benchmarks

- First database for ranking optical flow algorithms: Baron et al [Barron et al., 1994]
  - a survey (about ten methods)
  - synthetic data with ground truth
  - evaluation using previous statistics

Figure 26: Synthetic data (with ground truth)

---

6https://www-pequan.lip6.fr/~bereziat/barron/
Quantitative evaluation: Benchmarks (cont’d)

• and also true data:

Figure 27: True data with ground truth
Quantitative evaluation: Middlebury

- Middlebury database, [Baker et al., 2011][7]
- Synthetic and true data with ground truth known for tuning, and hidden for performance ranking

---

Figure 28: Example of synthetic data with ground truth

Quantitative evaluation: Middlebury (cont’d)

Figure 29: Example of true data with ground truth

- black areas: occluding regions
Figure 30: True and synthetic data with hidden ground truth

- Characteristics: large displacements, discontinuous velocity field, occluding
- Other databases are available: Sintel Flow Database\(^8\), KITTI (road traffic)\(^9\)...

\(^8\)http://sintel.is.tue.mpg.de/
\(^9\)http://www.cvlibs.net/datasets/kitti/
Without ground truth? One can verify the estimated velocity map transport correctly image $I_1$ to $I_2$

**Figure 31**: reconstruction error
Quantitative evaluation without ground truth (cont’d)

- Reconstructed image: \( I_1^{\text{warped}}(x + w(x)\delta t) = I_1(x) \)
- Issue: this process leaves uninitialized pixels in \( I_1^{\text{warped}} \) because the mapping \( x \mapsto x + w \) is not bijective application in a discrete world
- Possible solutions:
  - Initialize \( I_1^{\text{warped}}(x) = I_1(x) \) before mapping. Drawback: introduce false discontinuities
  - Fill in holes with inpainting technique. Drawback: no more false discontinuities, but not necessarily correct values
  - \( I_1^{\text{warped}}(x) = I_1(x' + w(x')) \) where \( x' \) is the pixel in \( I_1 \) that is mapped to \( x \) in \( I_2 \) Drawback: issue if \( x \) has several antecedents
- Error measurement: \( \| I_2 - I_1^{\text{warped}} \| \)
Alternatives approaches
Introduction and definitions

Early approaches

Dense estimation

Horn and Schunck’s Issues

Evaluation

Alternatives approaches
  Neural networks

Appendix
Limitations of variational approaches

- Model used (brightness constancy, regularization) remain imperfect and not always justified: need of more general models, whose parameters would be learned with supervised machine learning techniques

- Short state-of-the-art:
  - Black et al [Black et al., 1997]: PCA computed on a training set, motion is seen as a linear combination of eigenvector. The optical flow equation is projected onto the PCA basis leading to a linear regression problem. No regularization.
  - Rosenbraum et al [Rosenbraum et al., 2013]: motion models as a Gaussian mixture
  - Sun et al [Sun et al., 2008]: image are pre-processed with a bank of FIR filters, filters are learned (by likelihood maximization) before compute the optical flow
Deep neural networks

- Following Sun’s idea, CNN can be used to learn motion estimation filter
  - at first order: the linear optical flow equation, as well regularisation, use differential operators that can be learned with convolution kernels ⇒ convolutional networks
  - at second order: universal approximation theorem, [Hornik et al., 1989], a network with an hidden layer can approximate any continuous function ⇒ deep networks
- Availability of huge databases for motion estimation (KITTI, SINTEL...) permits to train deep CNN, with a limitation, these databases being synthetic and a lack of realism
Flownet [Dosovitskiy et al., 2015]

- FlownetS (Simple) and FlownetC (Correlated)

**Figure 32:** Both figures from [Dosovitskiy et al., 2015]

- Details of the green box:
Flownet [Dosovitskiy et al., 2015]

- “U-Net” architecture:
  - Encoder to a latent space, then decoder
  - Skip connection between each resolution downsample
- Encoder, two versions:
  - FlownetS ('Simple'): input data are stacked into channels (2 consecutive RGB images = 6 channels) than encoded
  - FlownetC: ('Correlation'): two separate stages, one by images. Then features are merged with a correlation product (unlearned) before to be encoded to the latent space
- The network learns the evolution law between a pair of images: richer than the advection
- The encoder/decoder architecture mimics a multiresolution scheme
- Loss function: EPE, $\mathcal{L}(w, \hat{w}) = \|w - \hat{w}\|$ (supervised training)
- Better results for FlownetC than FlownetS
The “Flying chairs” database

- Dataset of 45 Gb, semi-synthetic images

- Size required to train correctly FlowNet.

- [https://lmb.informatik.uni-freiburg.de/resources/datasets/FlyingChairs.en.html](https://lmb.informatik.uni-freiburg.de/resources/datasets/FlyingChairs.en.html)

- Train: several hours on a huge GPU

- still outperformed by the best variationnal approaches (on small displacements specially)
Spynet [Ranjan and Black, 2017], Flownet2 [Ilg et al., 2017]

- Spynet: standard multiresolution pyramid without latent space

![Diagram of Spynet and Flownet2](image)

**Figure 33:** From [Ilg et al., 2017]

- Outperforms Flownet
- Flownet2: combination of several FlownetS and FlownetC, with a module dedicated to “small displacements” outperforms Spynet
RAFT [Teed and Deng, 2020]

- Recurrent All-Pairs Field Transforms for Optical Flow

![Diagram]

**Figure 34:** From [Teed and Deng, 2020]

- **Architecture:**
  - a Feature encoder, similar to FlownetC, but the correlation is 4D between all-pairs of pixel feature of the two input images
  - Iterative update: a multiresolution strategy ($w^{k+1} = \triangle w + w^{k-1}$), obtained from successive pooling of 4D correlation and a GRU module
  - Loss: weighted $L_1$ EPE on each $w^k$, supervised.
Unsupervised training

- Training sets are not always realist, how to train without ground truth?
- Change the loss function: consider the reconstruction error instead of EPE

Figure 35: From [Yu et al., 2016]
Unsupervised training (cont’d)

- The optical flow constraint is embedded in the loss
  \[ \mathcal{L}_{EPE}(w, w^{GT}) = \| w - w^{GT} \| \Rightarrow \mathcal{L}_{warp}(I_1, I_2, w) = \| I_1 - \text{Warp}(I_2, w) \| \]

- Remains ill posed! regularization is required:
  \[ \mathcal{L}_{\text{smooth}}(w) = \| \nabla w \| \]

- Issue: the optical flow constraint must be verified for correct performance

- At this moment, unsupervised approaches remain less accurate, work in progress (Deep Image Prior….)
Semi-supervised training with GAN [Lai et al., 2017]

- Generator: a NN $G(l_1, l_2)$ producing a velocity $\tilde{w}$ minimizing the warping loss
- Idea for a discriminator $D$: train a NN such a:

$$D(\tilde{w}) = \begin{cases} 
1 & \text{if } \tilde{w} \text{ produced from a ground truth} \\
0 & \text{if } \tilde{w} \text{ is computed by } G 
\end{cases}$$

- $D$ knows the ground truths, $G$ is trained to make $D$ wrong


Hierarchical model-based motion estimation.

*Ondelettes et problèmes mal posés : la mesure du flot optique et l’interpolation irrégulière.*

*Robust dynamic motion estimation over time.*

*Learning parametrized models of image motion.*
In *Computer Vision and Pattern Recognition*.

*High accuracy optical flow estimation based on a theory for warping.*
An algorithm for total variation minimization and applications.  

Nonlinear variational method for optical flow computation.  
In SCIA, pages 523–530.

Flownet: Learning optical flow with convolutional networks.  
In International Conference on Computer Vision.

Computation of component image velocity from local phase information.  

Determining optical flow.


*LNCS*, 801.
Optical flow estimation using a spatial pyramid network.  
In *cvpr*.

Learning the statistics of optical flow.  
In *NIPS*.

Nonlinear total variation based noise removal algorithms.  

Learning optical flow.  
In *European Conference on Computer Vision*.

Raft: Recurrent all pairs field transforms for optical flow.  
In *European Conference on Computer Vision*.

Multiple constraints for optical flow.
In Proceeding of ECCV, pages 61–70.

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

Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness.
In European Conference on Computer Vision.

A duality based approach for realtime tv-1 optical flow.