

# 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

**Part II: Fix Horn and Schunk's  
issues, recent methods,  
evaluation**

---

## Part II: Fix Horn and Schunck's issues, recent methods, evaluation

### Horn and Schunck's Issues

#### Guided regularizations

#### Large displacements

#### Illumination changes and occlusion

#### Alternatives approaches

#### Evaluation

## Appendix

## $L_2$ regularization is a smoothing process

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

$$E(w) = \underbrace{\int_{\Omega} (\nabla I \cdot w + I_t)^2 dx}_{E_{\text{data}}} + \underbrace{\int_{\Omega} \alpha \|\nabla w\|^2 dx}_{E_{\text{regul}}} \quad (1)$$

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:

$$\begin{aligned} w(x, 0) &= 0 \quad x \in \Omega \\ \frac{\partial w_{\tau}(x, t)}{\partial \tau} + \alpha \Delta w_{\tau}(x, t) &= \nabla I(\nabla I \cdot w_{\tau}(x, t) + I_t) \end{aligned} \quad (2)$$

- Stationary solutions of (2) (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 (2) is called *Euler-Lagrange* equation associated to the problem of minimizing  $E$  (Eq. (1))
- This is a diffusion equation (see my lecture in TADI on scales spaces) with a forcing term (right member of Eq. (2))
- Discretization of Eq. (2)) leads to a Gauss-Seidel method (iterative method for matrix inversion, similar to Horn and Schunk numerical scheme (see Eqs. (8,9) in previous lecture)
- Avoid the smoothing effect induced by diffusion: use non-linear diffusion (guided or not by the image values)

## Oriented regularization: [Nagel, 1987]

- Idea: to preserve velocity map discontinuities avoid smoothing along edges contours
- Regularizing term in Horn and Schunck cost function is rewritten as:

$$E_{\text{regul}} = \int_{\Omega} \alpha \text{tr} ((\nabla w)^T \nabla w) dx dy$$

- Indeed:

- $\nabla w = \begin{pmatrix} \nabla u & \nabla v \end{pmatrix} = \begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix}$
- $\nabla w^T \nabla w = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix} = \begin{pmatrix} u_x^2 + u_y^2 & \dots \\ \dots & v_x^2 + v_y^2 \end{pmatrix}$
- $\text{tr}(\nabla w^T \nabla w) = u_x^2 + u_y^2 + v_x^2 + v_y^2$

## Nagel oriented regularization (cont'd)

- Nagel considers the following norm:

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

with  $V$  a  $2 \times 2$  matrix such as:

$$V = \frac{1}{\|\nabla I\|_2^2 + 2\delta} W$$
$$W = \begin{pmatrix} I_y^2 + \delta & -I_x I_y \\ -I_x I_y & I_x^2 + \delta \end{pmatrix}$$

- Parameter  $\delta$  allows  $V$  to be invertible:  $\delta = 0 \Rightarrow \det(W) = 0$
- $W$  is divided by a normalization term

## Nagel oriented regularization (cont'd)

- With  $\delta > 0$ ,  $V$  is always well defined
- In the following, consider  $\delta = 0$ 
  - $W$  writes:

$$W = \begin{pmatrix} -I_y \\ I_x \end{pmatrix} \begin{pmatrix} -I_y & I_x \end{pmatrix} = U^T U$$

with  $U = \begin{pmatrix} -I_y & I_x \end{pmatrix}$

- The regularization term now writes:

$$\int_{\Omega} \text{tr} \left( (U \nabla w)^T (U \nabla w) \right) dx$$

and after expansion:

$$\int_{\Omega} \text{tr} \begin{pmatrix} (-I_y u_x + I_x u_y)^2 & \dots \\ \dots & (-I_y v_x + I_x v_y)^2 \end{pmatrix} dx$$

## 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
  - Velocity map is smoothed except along edges
- Alternative writing:

$$E_{\text{regul}}(w) = \int_{\Omega} \frac{\alpha}{\|\nabla I\|_2^2 + 2\delta} [(-I_y u_x + I_x u_y)^2 + (-I_y v_x + I_x v_y)^2 + \delta(\nabla u^2 + \nabla v^2)] dx dy$$

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:

$$u^{k+1} = \eta(u^k) - l_x \frac{l_x \eta(u^k) + l_y \eta(v^k) + l_t}{\alpha + l_x^2 + l_y^2}$$

$$v^{k+1} = \eta(v^k) - l_y \frac{l_x \eta(u^k) + l_y \eta(v^k) + l_t}{\alpha + l_x^2 + l_y^2}$$

with:

$$\eta(f) = \bar{f} - 2l_x l_y f_{xy} - q \nabla f$$

$$q = \frac{1}{l_x^2 + l_y^2 + 2\delta} \nabla l^T \left[ \begin{pmatrix} l_{yy} & -l_{xy} \\ -l_{xy} & l_{xx} \end{pmatrix} + 2 \begin{pmatrix} l_{xx} & l_{xy} \\ l_{xy} & l_{yy} \end{pmatrix} V \right]$$

# Nagel oriented regularization: results



**Figure 1:** Hamburg's Taxi sequence,  $\delta = 10$ ,  $\alpha = 25$

Loading data ...

**Figure 2:** Animation

## Nagel oriented regularization: results (cont'd)



**Figure 3:** Hamburg's Taxi sequence,  $\delta = 10$ ,  $\alpha = 1$

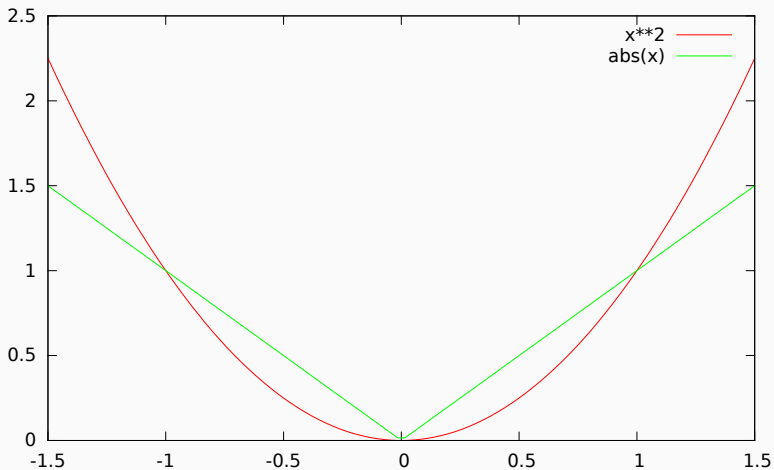
Loading data ...

**Figure 4:** 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]:
$$E_{\text{regul}} = \int \phi(|\nabla I|) \|\nabla w\|^2 dx, \text{ with } \phi \text{ decreasing function}$$
  - ...
- Norms can also depends only on velocity map configuration (*flow-guided*)...

## $L_1$ versus $L_2$ norms



- The quadratic term strongly penalizes discontinuities

## $L_1$ regularization: [Cohen, 1993]

- Consider the minimization problem:

$$E(w) = \int_{\Omega} (\nabla I \cdot w + I_t)^2 dx + \alpha \underbrace{\int_{\Omega} \left( \sqrt{u_x^2 + u_y^2} + \sqrt{v_x^2 + v_y^2} \right) dx}_{L_1}$$

- Gradient of  $L_1$ :

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

$$\frac{\partial L_1}{\partial v} = -\frac{\partial}{\partial x} \left( \frac{v_x}{\sqrt{v_x^2 + v_y^2}} \right) - \frac{\partial}{\partial y} \left( \frac{v_y}{\sqrt{v_x^2 + v_y^2}} \right)$$

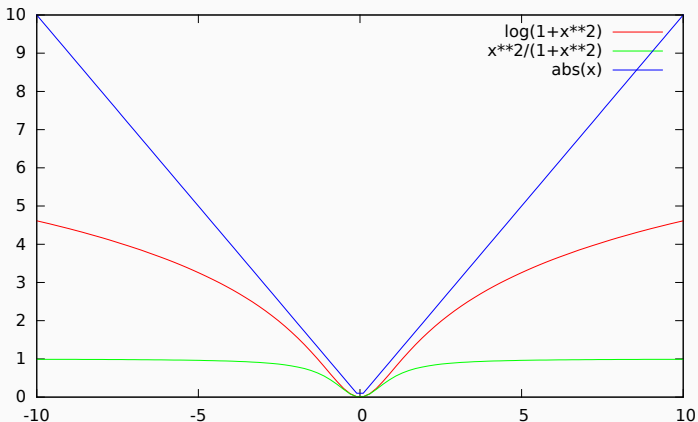
## $L_1$ regularization

- 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], see my lecture on scales spaces (TADI)
- $L_1$  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):

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

- alternative writing:  $\Psi(s) = \sqrt{s^2 + \epsilon}$
- Geman-McClure norm:  $\Psi(s) = \frac{s^2}{\mu^2 + s^2}$
- Lorentz norm:  $\Psi(s) = \log(1 + \frac{s^2}{\sigma^2})$

## Norms robust to discontinuities



**Figure 5:** Geman (green), Lorentz (red),  $L_1$  (blue)

- Geman and Lorentz norms don't penalize discontinuities

- 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: robust to noise as it don't penalize large deviation to optical flow constraint
- Issue: introducing non linear terms lead to a non convex optimization problem

## Non convex optimization: [Zach et al., 2007]

- Consider the non convex cost function:

$$E(w) = \int_{\Omega} (|\nabla I \cdot w + I_t| + \alpha \|\nabla w\|) dx \quad (3)$$

- 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 (3) and  $w'$  tends to  $w$
- $E_{\theta}$  can be decoupled into two convex optimization problems

## Non convex optimization: Zack *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:

$$\int_{\Omega} \left( \frac{1}{2\theta} \|w - w'\|^2 + \alpha \|\nabla w\| \right) dx \quad (4)$$

2.  $w$  fixed, find  $w'$  minimizing:

$$\int_{\Omega} \left( |\nabla I \cdot w' + I_t| + \frac{1}{2\theta} \|w - w'\|^2 \right) dx \quad (5)$$

- Problem (4) has been studied by Rudin *et al.* [Rudin et al., 1992] in a context of image denoising
- Problem (5) 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>

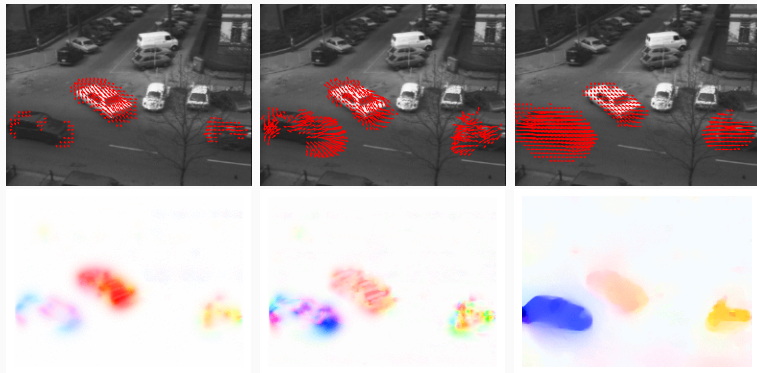


Figure 6: Horn & Schunck, Nagel, Zach

## 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

## Part II: Fix Horn and Schunk's issues, recent methods, evaluation

### Horn and Schunck's Issues

Guided regularizations

**Large displacements**

Illumination changes and occlusion

Alternatives approaches

Evaluation

### 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 ( $\|w\|\delta t < 2$ )
- $\delta t$  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! if we can determine the gradient of

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

- Gâteaux derivative:

$$\lim_{\gamma \rightarrow 0} \frac{E_{\text{data}}(u + \gamma f, v) - E_{\text{data}}(u, v)}{\gamma}$$

- 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:  $I(x + w\delta t, t + \delta t)$  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:

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

- 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<sup>1</sup>:

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

- Need to evaluate  $I(x + w^k, t + 1)$  and  $\nabla I(x + w^k, t + 1)$  using bilinear interpolation<sup>2</sup>

<sup>1</sup>Due to numerical considerations, see my TADI lecture on scales spaces

<sup>2</sup> $x + w^k$  do not belong to the spatial grid

## Large displacements: iterative Lucas-Kanade

- [Lucas and Kanade, 1981]
- Back to LK notation:  $I_2(x + w_x) = I_1(x)$
- Assume having a first guess  $w_b$  for  $w_x = w_b + \delta w$
- First order Taylor expansion of  $I_2$  at point  $x + w_b$ :

$$I_2(x + w_b + \delta w) - I_1(x) \approx I_2(x + w_b) - I_1(x) + \nabla I_2(x + w_b) \delta w$$

- We solved now an *incremental* optical flow equation:

$$I_2(x + w_b) - I_1(x) + \nabla I_2(x + w_b) \cdot \delta w = 0 \quad (6)$$

- Cost function :

$$E(\delta w) = \sum_{x \in \mathcal{W}_x} (I_2(x + w_b) - I_1(x) + \nabla I_2(x + w_b) \cdot \delta w)^2$$

- $\delta w$ : incremental displacement
- Apply Lucas Kanade method between images  $I_1$  and a shifted version of  $I_2$  given by  $I_2(x + w_b)$  and interpolation

# Large displacement: iterative Lucas-Kanade (cont'd)

- Algorithm

1.  $w = \vec{0}$
2. Shift  $I_2$  with  $w$ :  $I_2(x + w)$
3. Compute gradient of  $I_2(x + w)$  and time derivative  $I_2(x + w) - I_1(x)$
4. Apply Lucas Kanade:  
$$\delta w = \text{LucasKanade}(\nabla I_2(x + w), I_2(x + w) - I_1(x))$$
5. Update  $w$ :  $w = w + \delta w$
6. Back to 2.

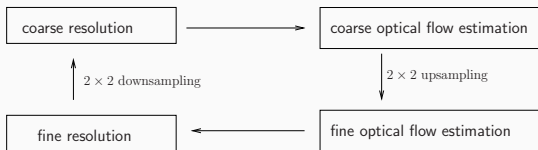
- Issues:

- No convergence guarantee
- Hard to determine the optimal number of iterations

# 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 guess
  - 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



# 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  $k + 1$ :

$$I^{k+1}(x, y, t) = \downarrow (I^k \star G_\sigma)(x, y, t)$$

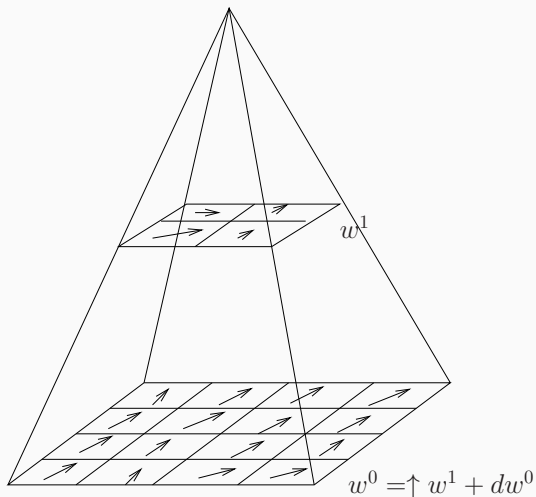
- $\downarrow$ :  $2 \times 2$  downsampling operator (keep 1 pixel over 4)
- Anti-aliasing filter: Gaussian smoothing with standard deviation of  $\sigma = 2$
- $\Omega^k$  spatial domain of level  $k$  verifying:

$$\Omega^N \subset \dots \subset \Omega^{k+1} \subset \Omega^k \subset \dots \subset \Omega^0$$

- 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):



# 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  $2 \times 2$  upsampling operator

- **Important:** one-pixel displacement at level  $k + 1$  implies two-pixel displacement at level  $k$ ! the upsampling operator  $\uparrow$  has to **multiply by 2** the velocity magnitude
- Non linear optical flow constraint at level  $k$ :

$$\begin{aligned} D^k(\mathbf{x}, t) &= I^k(\mathbf{x} + w^k dt, t + dt) - I^k(\mathbf{x}, t) \\ &= I^k(\mathbf{x} + (\uparrow w^{k+1} + dw^k)dt, t + dt) - I^k(\mathbf{x}, t) = 0 \end{aligned}$$

- $w^{k+1}$  is given (estimation at coarse resolution),  $dw^k$  explains velocity difference between levels  $k + 1$  and  $k$ :  $2 \times 2$  upsampling, so  $|du^k|, |dv^k| < 2$ , the linear optical flow is an admissible approximation

## Large displacements: compute velocity at level $k$ from level $k + 1$ (cont'd)

- We write:

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

and apply the same idea of iterative Lucas Kanade

- First order Taylor expansion of  $I^k$  at point  $\mathbf{x} + \uparrow w^{k+1}dt$ :

$$\begin{aligned} I^k((\mathbf{x} + \uparrow w^{k+1}dt) + dw^k dt, t + dt) &\simeq \\ I^k(\mathbf{x} + \uparrow w^{k+1}dt, t + dt) + \nabla I(\mathbf{x} + \uparrow w^{k+1}, t + dt) \cdot dw^k dt \end{aligned}$$

- $D^k$  becomes:

$$\begin{aligned} D^k(\mathbf{x}, t) &= I^k(\mathbf{x} + \uparrow w^{k+1}dt, t + dt) - I^k(\mathbf{x}, t) \\ &\quad + \nabla I^k(\mathbf{x} + \uparrow w^{k+1}dt, t + dt) \cdot dw^k dt \\ &= 0 \end{aligned}$$

## Large displacements: compute velocity at level $k$ from level $k + 1$ (cont'd)

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

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

- Equation  $D^k = 0$  writes:

$$\frac{1}{dt} I_{\text{shift}}^k(\mathbf{x}, w^{k+1}, t) + \nabla I^k(\mathbf{x} + \uparrow w^{k+1} dt, t + dt) \cdot dw^k = 0 \quad (7)$$

- Eq. (7) is of same nature of (6): an *incremental optical flow equation* 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]

# Large displacements and multiresolution approaches: algorithm

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:

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

4. Update  $w^k = w^{k+1} + dw^k$ ,  $k = k - 1$
5. Iterate steps 3. and 4. up to  $k = 0$

## Part II: Fix Horn and Schunk's issues, recent methods, evaluation

### Horn and Schunck's Issues

Guided regularizations

Large displacements

**Illumination changes and occlusion**

Alternatives approaches

Evaluation

### Appendix

## Illumination change: [Brox et al., 2004]

- Global illumination change is a common issue
- Model:  $I(x, t + 1) = I(x + w, t) + c$ ,  $c$  bias
- Simple remark:

$$I(x, t + 1) = I(x + w, t) + c \Rightarrow \nabla I(x, t + 1) = \nabla I(x + w, t)$$

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

$$\begin{aligned} E(w) = & \int \|I(x + w, t + 1) - I(x, t)\|^2 dx \\ & + \int \gamma \|\nabla I(x + w, t + 1) - \nabla I(x, t)\|^2 dx \\ & + \int \alpha \|\nabla w\|^2 dx \end{aligned}$$

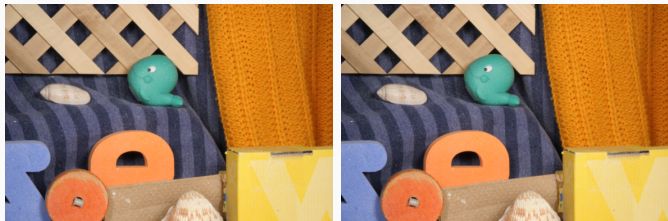
- Two constraints, but the problem remains ill-posed, why?

- Occlusion occurs when an object is in front of an 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

# Object occlusion: [Ince and Konrad, 2008]

- 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 my TADI lecture on scales spaces):  
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 7:** Large displacements, discontinu vector field , occlusions



**Figure 8:** Ground truth, Horn and Schunck (1981), Sun *et al* (2010)

## Part II: Fix Horn and Schunk's issues, recent methods, evaluation

Horn and Schunck's Issues

Alternatives approaches

**Data assimilation**

Neural networks

Evaluation

Appendix

# Data assimilation approach

- A formalism for inverse problems: knowing some partial observation of a physical system and a background, knowing physics of the system (time evolution), and knowing statistics of errors (covariances matrices), how to retrieve the system?
- a state vector  $X_t \in \mathbb{R}^n$  describes the physical system over time  $t \in [0, T]$
- a model  $\mathbb{M}$  (physics) rules the time evolution of  $X$ :

$$X_{t+1} = \mathbb{M}_t X_t$$

- we have a first guess (background) of the initial condition of  $X$ :

$$X_0 = X_b + \epsilon_B, \quad \epsilon_B \text{ assumed Gaussian of covariance } B$$

- we have partial observation  $Y \in \mathbb{R}^d$  of  $X$ :

$$Y_t = \mathbb{H}_t X_t + \epsilon_{R_t} \quad R_t \text{ assumed Gaussian of covariance } R_t$$

$\mathbb{H}$  is called *observation operator*. As  $d < n$ , it is non invertible

# Data assimilation approach: 4DVar formalism

- The question: how to retrieve the initial condition  $X_0$  satisfying the system?

$$X_0 = X_b + \epsilon_B \quad (8)$$

$$X_{t+1} = \mathbb{M}_t X_t \quad (9)$$

$$Y_t = \mathbb{H}_t X_t + \epsilon_{R_t} \quad (10)$$

- From Eq. (9):  $X_t = \mathbb{M}_{t-1} \cdots \mathbb{M}_1 \mathbb{M}_0 X_0 = \mathbb{M}_{0 \rightarrow t} X_0$   
 $\Rightarrow X_t$  only depends on  $X_0$
- To answer to the question: find  $X_0$  that minimizes

$$J(X_0) = \underbrace{\|X_0 - X_b\|_B^2}_{\epsilon_B} + \sum_{t=0}^T \underbrace{\|Y_t - \mathbb{H}_t X_t\|_{R_t}^2}_{\epsilon_{R_t}}$$

s.t. Eq. (9)

in a variational framework (4DVar=space and time, variational)

Notation:  $\|\epsilon\|_A^2 = \int \epsilon^T(x) A^{-1}(x) \epsilon(x) dx$

## Minimize the 4DVar cost function

$$J(X_0) = \|X_0 - X_b\|_B^2 + \sum_{t=0}^T \|Y_t - \mathbb{M}_t X_t\|_{R_t}^2$$

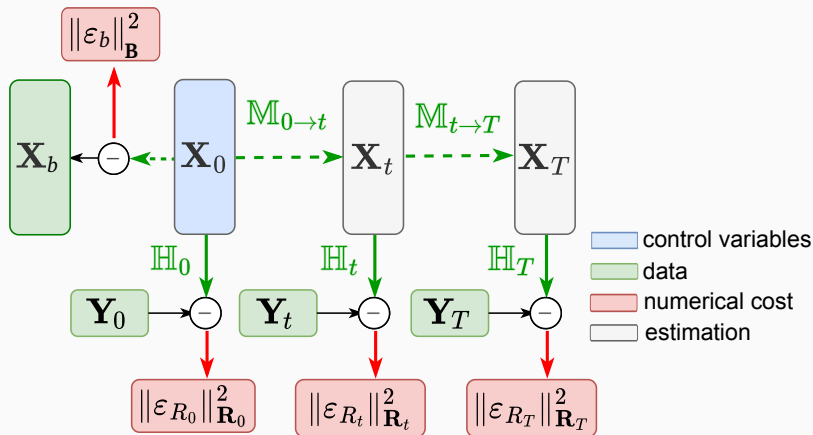
- Gradient of  $J$  (assuming  $\mathbb{M}$  and  $\mathbb{H}$  linear) is:

$$\begin{aligned} \nabla J(X(t_0)) = & 2B^{-1}(X(t_0) - X_b) \\ & + 2 [\mathbb{H}_0^T R_0^{-1} D_0 + \\ & \mathbb{M}_1^T [\mathbb{H}_1^T R_1^{-1} D_1 \\ & + \dots \\ & + \mathbb{M}_T^T \mathbb{H}_T^T R_T^{-1} D_T] \dots] \end{aligned}$$

with  $D_t = Y_t - \mathbb{H}_t X_t$

- In practise, gradient can also be obtained using *automatic differentiation* (ex: Autograd for Pytorch)
- Minimum of  $J$  is achieved by steepest descent with a Quasi-Newton solver (ex: L-BFGS)

# 4DVar diagram (in a Deep Learning spirit)



## Application to optical flow estimation

- Physical system: a scalar map,  $I$ , is advected by a velocity map,  $w$

$$X = \begin{pmatrix} I & w \end{pmatrix}^T$$

- From this system we observe the scalar map at various acquisition dates  $t$ :  $Y_t = I(., t)$ , and we want to retrieve  $w$
- Observation operator is then the projection of  $X$  into the subspace  $\mathbb{R}^d$  of observation:  $\mathbb{H}X = Y$
- Advection is the physical process ruling the state vector in time:

$$\frac{\partial I}{\partial t} + \nabla I(t) \cdot w(t) = 0 \quad (11)$$

while velocity will be supposed stationary

- After time discretisation,  $\mathbb{M}$  writes such as:

$$I_{t+1} = I_t + \Delta t \nabla I_t \cdot w_t \quad (\text{advection}) \quad (12)$$

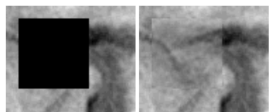
$$w_{t+1} = w_t \quad (\text{stationarity}) \quad (13)$$

- $\Delta t$  is the time step

## Application to optical flow estimation (cont'd)

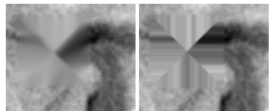
- Possible choice of background:  $X_b = (\vec{0} \quad I_0)^T$ 
  - we never observe  $w$
- We assume no spatial correlation on  $Y$ :
  - $R_t$  is diagonal and  $R_t(x)$  gives the variance noise acquisition as pixel  $x$
- Missing data: set  $R_t^{-1}(x) = 0$ 
  - we can use 4DVar for inpainting!
- No observation at time  $t$ : set  $R_t^{-1} \equiv 0$ :
  - we control the number of time steps, no need of multi-resolution scheme to respect the optical flow assumption ( $t + 1 = t + N \times \Delta t$ , with  $\Delta t = 1/N$  arbitrarily small)
- Optical flow estimation, even in a 4DVar formalism, remains ill-posed and need regularization
  - Regularization of  $w_0$  can be embedded in matrix  $B$ :  
derivation being linear, it exists  $B$  such as:  $\|X_0 - X_b\|_B^2 = \alpha \|\nabla w_0\|^2$

# Some results: Inpainting on ocean images



(a) Noisy image

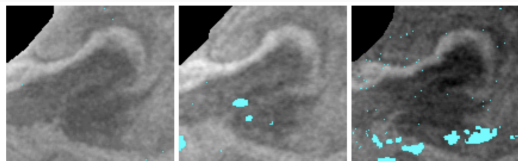
(b) Our approach



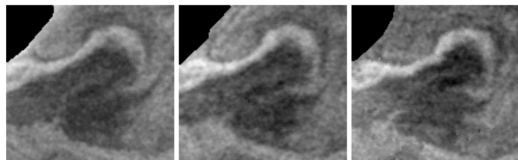
(c) Bertalmio *et al*

(d) Tschumperlé *et al*

Evaluation on ground  
truth



(c) Observations: frames 4, 5 and 6

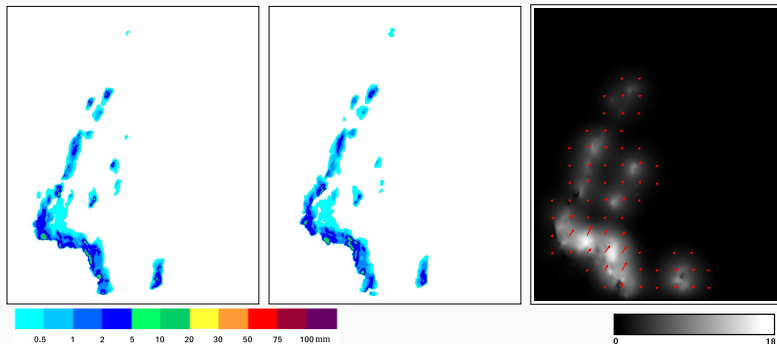


(d) Results: frames 4, 5 and 6

**Fig. 8.** Observations and results.

Cloud cover inpainting

## Some results: Rain nowcasting



**Figure 9:** left and middle: two radar rainmaps for two successive times acquisition; right: the motion and its intensity estimated from these observation

- Forecast is done by applying  $\mathbb{M}_{0 \rightarrow t}$  on last observation and the estimated velocity map

## Part II: Fix Horn and Schunk's issues, recent methods, evaluation

Horn and Schunck's Issues

Alternatives approaches

Data assimilation

Neural networks

Evaluation

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

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

# Flownet [Dosovitskiy et al., 2015]

- FlownetS (Simple) and FlownetC (Correlated)

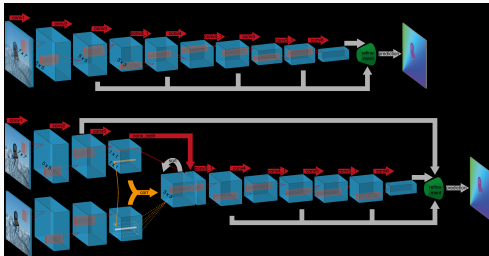
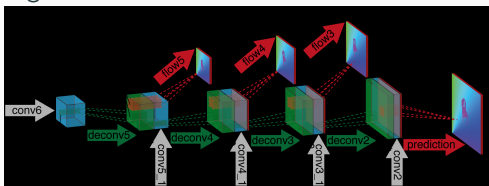


Figure 10: Both figures from [Dosovitskiy et al., 2015]

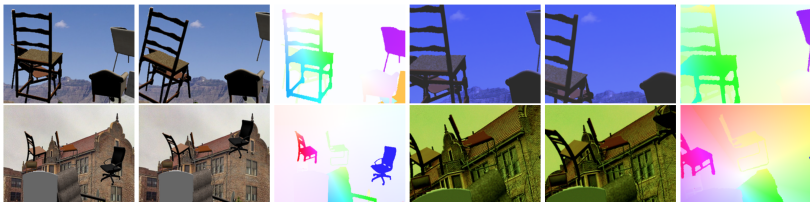
- Details of the green box:



- “U-Net” architecture:
  - Encoder into 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) then encoded
  - FlownetC: ('Correlation'): two separate stages, one by images. Then features are merged with a correlation product (unlearned) before to be encoded into 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>
- Train: several hours on a huge GPU
- still outperformed by the best variational approaches (on small displacements specially)

# Spynet [Ranjan and Black, 2017], FlowNet2 [Ilg et al., 2017]

- Spynet: standard multiresolution pyramid without latent space

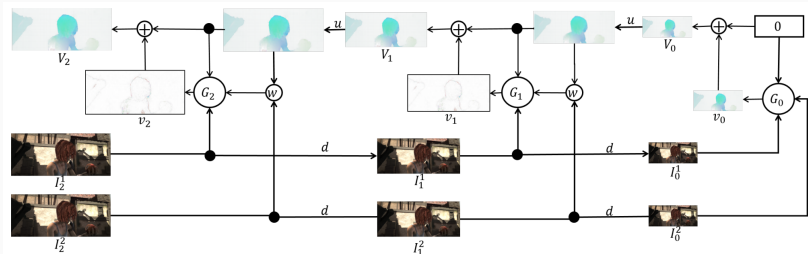


Figure 11: From [Ranjan and Black, 2017]

- Loss function: EPE for each resolution
- Outperforms FlowNet
- FlowNet2: combination of several FlowNetS and FlowNetC, with a module dedicated to “small displacements” outperforms Spynet

- Recurrent All-Pairs Field Transforms for Optical Flow

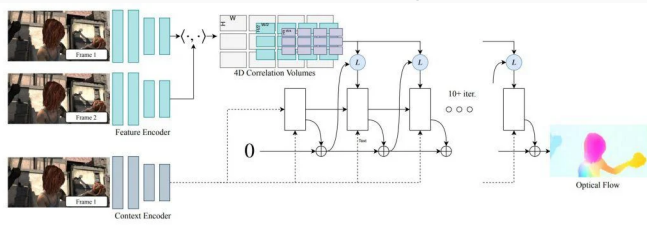


Figure 12: From [Teed and Deng, 2020]

- 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 ( $w^{k+1} = \Delta w + w^{k-1}$ ), 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 realistic, how to train without ground truth?
- Change the loss function: consider the reconstruction error instead of EPE

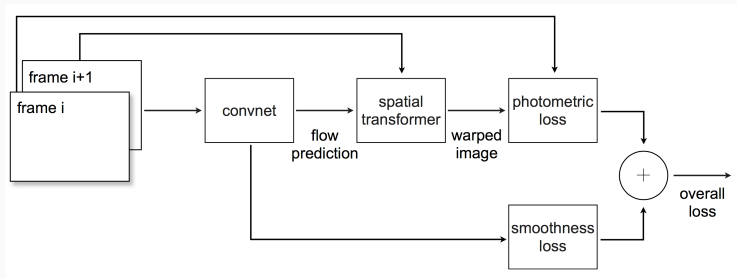


Figure 13: From [Yu et al., 2016]

## Unsupervised training (cont'd)

- The optical flow constraint is embedded in the loss

$$\mathcal{L}_{\text{EPE}}(w, w^{\text{GT}}) = \|w - w^{\text{GT}}\| \Rightarrow \mathcal{L}_{\text{warp}}(I_1, I_2, w) = \|I_1 - \text{Warp}(I_2, w)\|$$

- 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...)

- Generator: a NN  $G(I_1, I_2)$  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

# 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, *in situ* measures)
    - but costly and complex to set up
  - others and general cases
    - human validation: measuring displacement of objects/regions/points of interest

## Part II: Fix Horn and Schunck's issues, recent methods, evaluation

Horn and Schunck's Issues

Alternatives approaches

### Evaluation

Introduction

**Qualitative evaluation (visualization)**

Quantitative evaluation (benchmarks)

Appendix

## Evaluation by visualization: vector field

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

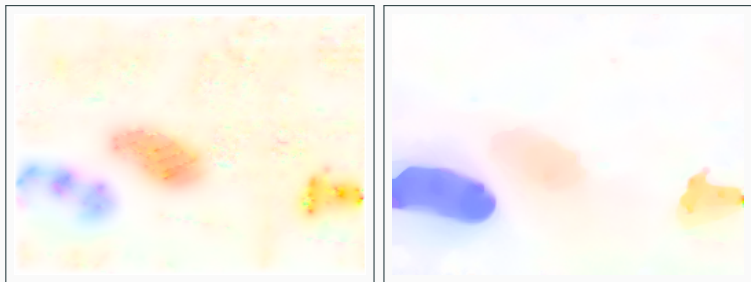


Figure 14: Horn and Schunck (red), Zack *et al* (blue)

- With Matlab/Matplotlib: `quiver()`

## Evaluation by visualization: Middlebury colormap<sup>3</sup>

- Dense representation: Middlebury colormap<sup>3</sup>
- Color wheel: velocity direction, color saturation: velocity magnitude



**Figure 15:**  $L_2$  (left), TV- $L_1$  (right)

---

<sup>3</sup><http://vision.middlebury.edu/flow/>

## Evaluation by visualization: velocity magnitude

- Dense representation with velocity magnitude (norm):

$$\|w\| = \sqrt{u^2 + v^2}$$

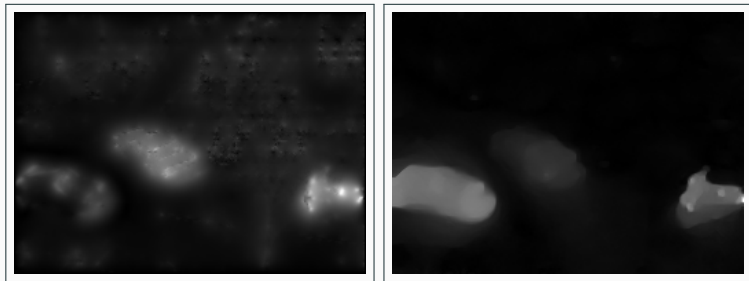


Figure 16:  $L_2$  (left),  $TV-L_1$  (right)

## 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:

$$\begin{aligned}\frac{\partial x}{\partial s}(s) &= w(x(s)) \quad s \in [0, 1] \\ x(0) &= x_0\end{aligned}\tag{14}$$

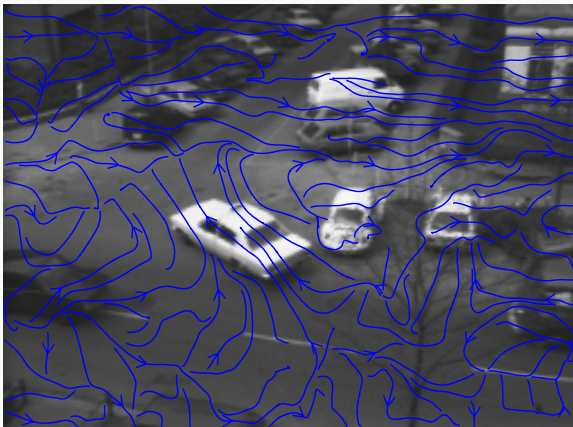
- Solution (integration):

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

- 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:



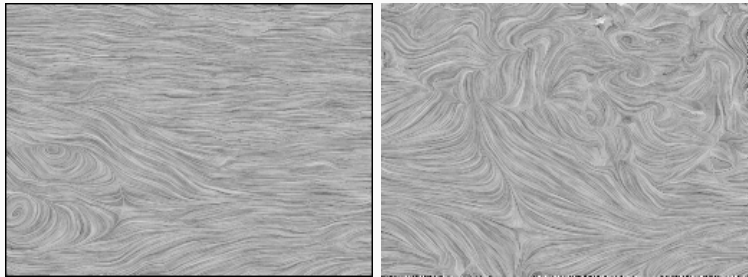
**Figure 17:** function `stream2()` (Matlab) `streamplot()` (Matplotlib)

# Evaluation by visualization: Line Integral Convolution

- Line Integral Convolution (LIC): dense visualization of stream lines
- Determination of stream lines, Eq (14)
- Integration using the following way:

$$\begin{aligned} \text{LIC}(x_0) &= \int_{\mathbb{R}} k(u - u_0) T(x(u)) du \\ x_0 &= x(u_0) \end{aligned}$$

- $T$ : image of texture (uniform noise)
- Convolution kernel  $k$  determine a window over the stream line:
  - $k(u) = \frac{1}{2L} \mathbb{1}_{[-L, +L]}$
  - or  $k$  Gaussian kernel of variance  $L$



**Figure 18:**  $L_2$  (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 (14):

$$\begin{aligned}\frac{\partial x}{\partial t} &= w(x, t) \\ x(0) &= x_0\end{aligned}$$

- 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

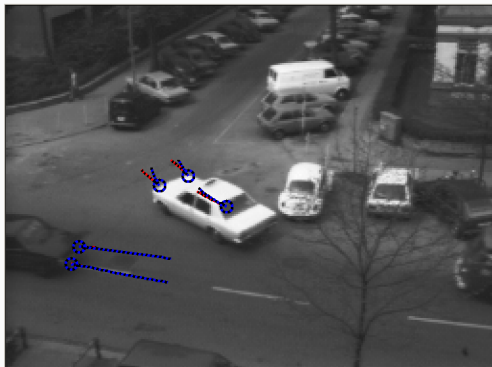


Figure 19:  $L_2$  (red), TV- $L_1$  (blue)

## Part II: Fix Horn and Schunck's issues, recent methods, evaluation

Horn and Schunck's Issues

Alternatives approaches

**Evaluation**

Introduction

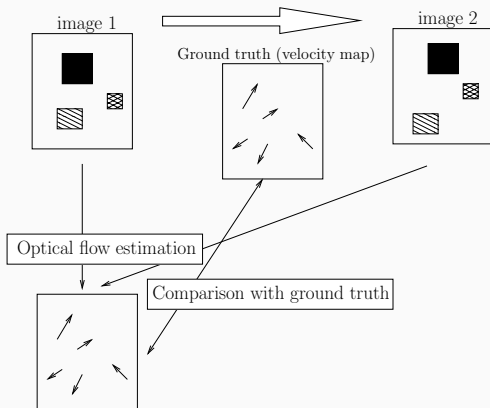
Qualitative evaluation (visualization)

Quantitative evaluation (benchmarks)

Appendix

# Quantitative evaluation

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



## Quantitative evaluation: error measurements

- Let  $w$  be the reference,  $\tilde{w}$  the estimated
- Angular error:  $\varepsilon_{\text{AE}} = \langle \widehat{w}, \widehat{\tilde{w}} \rangle = \arccos \left( \frac{w^T \tilde{w}}{\|w\| \|\tilde{w}\|} \right)$
- Angular error in space-time ([Fleet and Jepson, 1990]):

$$\varepsilon'_{\text{AE}} = \langle (\widehat{w}, 1), (\widehat{\tilde{w}}, 1) \rangle = \arccos \left( \frac{1 + w^T \tilde{w}}{\sqrt{(1 + \|w\|^2)(1 + \|\tilde{w}\|^2)}} \right)$$

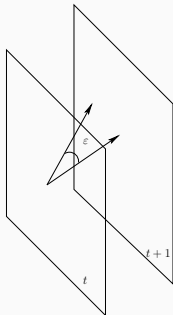
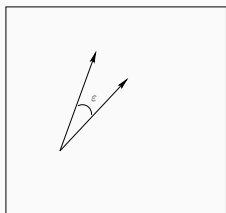


Figure 20: Angular error in space, and in space-time

## Quantitative evaluation: error measurements (cont'd)

- Relative Norm Error:  $\varepsilon_{\text{RNE}} = \frac{\|w\| - \|\tilde{w}\|}{\|w\| + \epsilon}$
- End Point Error:  $\varepsilon_{\text{EPE}} = \|w - \tilde{w}\|$  warning: an absolute error, relevant for comparison.
- Relative End Point Error:  $\frac{\|w - \tilde{w}\|}{\|w\| + \epsilon}$
- 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 <sup>4</sup>
  - evaluation using previous statistics

Loading data ...

Loading data ...

**Figure 21:** Synthetic data (with ground truth)

---

<sup>4</sup><https://perso.lip6.fr/Dominique.Bereziat/barron/>

- and also true data:

Loading data ...

Loading data ...

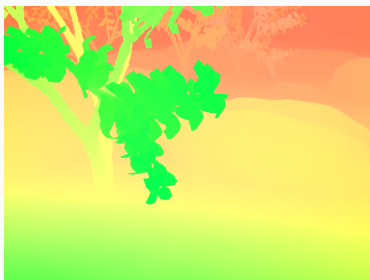
Loading data ...

**Figure 22:** True data with ground truth

# Quantitative evaluation: Middlebury

- Middlebury database<sup>5</sup>, [Baker et al., 2011]
- Synthetic and true data with ground truth known for tuning, and hidden for performance ranking

Loading data ...



**Figure 23:** Example of synthetic data with ground truth

---

<sup>5</sup><http://vision.middlebury.edu/flow>

## Quantitative evaluation: Middlebury (cont'd)

Loading data ...



**Figure 24:** Example of true data with ground truth

- black areas: occluding regions

Loading data ...

Loading data ...

**Figure 25:** True and synthetic data with hidden ground truth

- Characteristics: large displacements, discontinuous velocity field, occluding
- Other databases are available: Sintel Flow Database<sup>6</sup>, KITTI (road traffic)<sup>7</sup>...

---

<sup>6</sup><http://sintel.is.tue.mpg.de/>

<sup>7</sup><http://www.cvlibs.net/datasets/kitti/>

# Quantitative evaluation without ground truth

- Without ground truth? One can verify the estimated velocity map transport correctly image  $I_1$  to  $I_2$

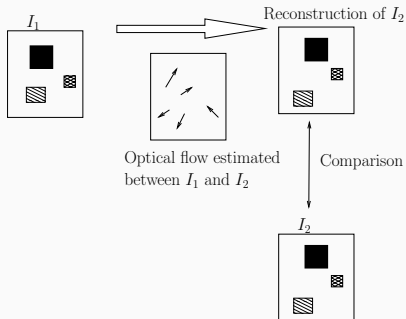


Figure 26: reconstruction error

## 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
  - Fillin 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:  $\|I_2 - I_1^{\text{warped}}\|$

# Appendix

---



Alvarez, L., Esclarin, J., Lefebure, M., and Sanchez, J. (1999).

**A PDE model for computing the optical flow.**

In *Congreso de ecuaciones diferenciales y aplicaciones*, volume 8, pages 1349–1356.



Anandan, P. (1989).

**A computational framework and an algorithm for the measurement of visual motion.**

*International Journal on Computer Vision*, 2(3):283–310.



Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011).

**A database and evaluation methodology for optical flow.**

*International Journal on Computer Vision*, 92:1–31.



Barron, J., Fleet, D., and Beauchemin, S. (1994).

**Performance of optical flow techniques.**

*International Journal on Computer Vision*.



Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992).

## **Hierarchical model-based motion estimation.**

*European Conference on Computer Vision*, pages 237–252.



Black, M. and Anandan, P. (1991).

## **Robust dynamic motion estimation over time.**

In *Computer Vision and Pattern Recognition*, pages 296–302.



Black, M., Yacoob, Y., Jepson, A., and Fleet, D. (1997).

## **Learning parametrized models of image motion.**

In *Computer Vision and Pattern Recognition*.



Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004).

## **High accuracy optical flow estimation based on a theory for warping.**

In Springer-Verlag, editor, *Proc. of 8th ECCV*, volume 4, pages 25–36, Prague, Czech Republic.



Chambolle, A. (2004).

## **An algorithm for total variation minimization and applications.**

*Journal of Mathematical Imaging and Vision*, 20:89–97.



Cohen, I. (1993).

**Nonlinear variational method for optical flow computation.**

In *SCIA*, pages 523–530.



Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., and Brox, T. (2015).

**FlowNet: Learning optical flow with convolutional networks.**

In *International Conference on Computer Vision*.



Fleet, D. and Jepson, A. (1990).

**Computation of component image velocity from local phase information.**

*International Journal of Computer Vision*, 5(1):77–104.



Hornik, K., Stinchcombe, M., and White, H. (1989).

**Multilayer feedforward networks are universal approximators.**

*Neural Networks*, 2(5):359–366.



Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017).

## **FlowNet 2.0: Evolution of optical flow estimation with deep networks.**

In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Ince, S. and Konrad, J. (2008).

### **Occlusion-aware optical flow estimation.**

*Transactions on Image Processing*, 17(8):1443–1450.



Lai, W., Huang, J., and Yang, M. (2017).

### **Semi-supervised learning for optical flow with generative adversarial networks.**

In *NIPS*.



Lucas, B. and Kanade, T. (1981).

### **An iterative image registration technique with an application to stereo vision.**

In *IJCAI*, pages 674–679.



Nagel, H. (1987).

## **On the estimation of optical flow: relations between different approaches and some new results.**

*Articial Intelligence*, 33(3):298–324.



Perona, P. and Malik, J. (1990).

### **Space scale and edge detection using anisotropic diffusion.**

*Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Diffusion isotropique non linéaire et implémentation.



Proesmans, M., Van Gool, L., and Oosterlinck, A. (1994).

### **Determination of optical flow and its discontinuities using non-linear diffusion.**

*LNCS*, 801.



Ranjan, A. and Black, M. (2017).

### **Optical flow estimation using a spatial pyramid network.**

In *cvpr*.



Rosenbraum, D., Zoran, D., and Weiss, Y. (2013).

### **Learning the statistics of optical flow.**

In *NIPS*.



Rudin, L., Osher, S., and Fatemi, E. (1992).

**Nonlinear total variation based noise removal algorithms.**

*Physica D*, 60:259–268.



Sun, D., Roth, S., Lewis, J., and Black, M. (2008).

**Learning optical flow.**

In *European Conference on Computer Vision*.



Teed, Z. and Deng, J. (2020).

**Raft: Recurrent all pairs field transforms for optical flow.**

In *European Conference on Computer Vision*.



Yu, J., Harley, A., and Derpanis, K. (2016).

**Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness.**

In *European Conference on Computer Vision*.



Zach, C., Pock, T., and Bischof, H. (2007).

**A duality based approach for realtime TV- $l^1$  optical flow.**

In *Ann. Symp. German Association Patt. Recogn.*, pages 214–223.