IMA 208

Christophe Kervazo
christophe.kervazo@telecom-paris.fr

Pedagogic team: A. Dev Parakkat, C. Kervazo, L. Le Folgoc

General presentation of IMA 208 topics

- Objectives:
  - In most previous classes, images have been considered as simple 2-dimensional (2D) objects
  - In IMA 208, we go a step further by learning how to take into account the 3D nature of objects and how to process videos:

- The class is divided into two mains sub-modules:
  - Motion estimation (C. Kervazo)
  - Object tracking (L. Le Folgoc)

  } **<u>Video processing</u>** (scenes evolve with time)

  - Camera calibration (C. Kervazo)
  - From 3D point cloud to mesh (A. Dev Parakkat)

  } **<u>3D</u>** (scenes are 3-dimensional)

Class modality

- Class modality: IMA 208 will occur every Monday, with 2 TH:
  - 1 TH of lecture
  - 1 TH of practical work (TP)
- All lectures will be run as **inverted classes** (excepted the 1st one)
- **It means that we expect a lot of involvement from your side !!!**
- Therefore, attendance is mandatory for all lectures and TPs.

Inverted class principles

- **Before the 21/02/2025**: please create some groups of 3 persons and write the names of the three persons in the following online form: **https://docs.google.com/forms/d/e/1FAIpQLScc2AuZZlAhZK_aT8hqYaddWO5Ndwpf-ePn-wPzTMrKAhKT5g/viewform**
- You will work with this group during the whole IMA 208

- **Before each lecture**

  - One week before each class, we will provide :
    - some slides to be read before the class (mandatory)
    - some supplementary (optional) documents (text book chapter / article / video…)

  - Each of you will prepare 3 questions related to the class. It can be, according to your personal choice:
    - Something that you did not understand in the slides
    - Something you would like to have more details on
    - Generally speaking, any question you have on the technical content of the course

  - You will have to upload your 3 questions in an online form at most two days before the class (that is, on Saturday 23h59). **The links for the online forms can be found on ecampus** (there is a link per class, pay attention to upload on the right link)

Inverted class principles

- **During the lecture (1h30)**
  - Bring a laptop / a way to connect to ecampus to read the questions

  - Online mini-quizz (5 minutes): just to check that everyone read the slides. Make sure to arrive on time, otherwise you will miss the quizz…

  - First hour of the class:
    - 9 questions (yours + ours) will be randomly attributed to each group using the « Questions to be answered » sheet on Ecampus

    - you will answer to all of them by groups.
      How? Ask teachers, have a look to the supplementary material… but please don't use generative AI (otherwise it's ChatGPT that works, not you…)

  - Second part of the class: each group will select one question and present its prepared answer in front of the whole class. The teacher will correct if relevant

- **After the lecture (optional)**

  - During the week following the class, we will post a few « capsules videos » on ecampus, summarizing some of the important concepts. Feel free to watch them if it helps you.

Why inverted class?

- Inverted class requires more work during the semester **but** it also enables you to master the notions progressively => little work to prepare the exam

- It is an introduction of your future professional life: try to learn some new knowledge from a small set of documents, with the help of some colleagues. Then share the information to everyone

- Pedagogical interest: work in group, learn more autonomy, more involvement in learning

- Please note that it does not mean that it is easier, it requires involvement

Grade

- **12 points**: « practical exam »
    - The exam is a mix of practical work and theoretical questions.
    - It will be done on computers **without internet** (so please be sure to be prepared by seriously doing the TPs by yourself)
    - No electronic device will be allowed but you might bring a single double-sided handwritten sheet of paper of personal notes.

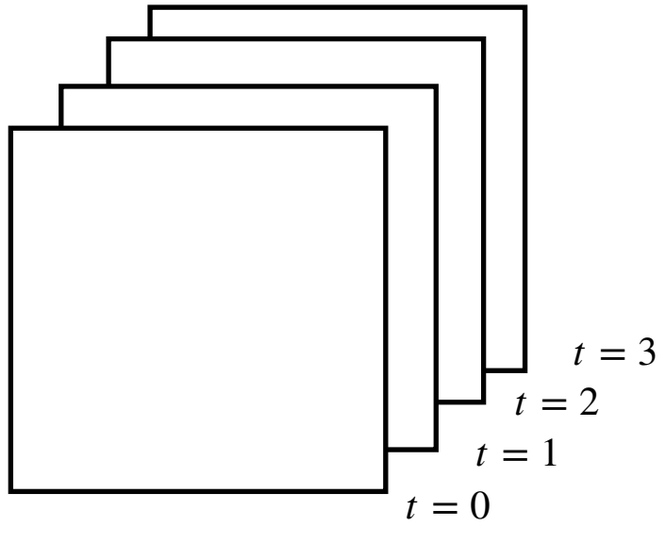
- **8 points:**
    - Question preparation before each class (2 points)
    - Multiple choice questions at the beginning of the class (2 points)
    - TP (4 points). To be handled back at most one day before next lecture (that is, on Sunday 23h59)

Motion estimation in videos

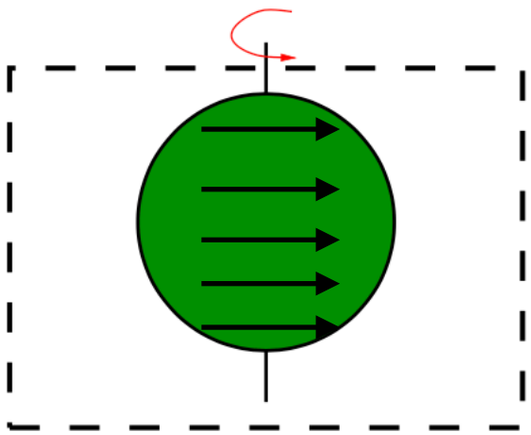
Christophe Kervazo
christophe.kervazo@telecom-paris.fr

Motion ?

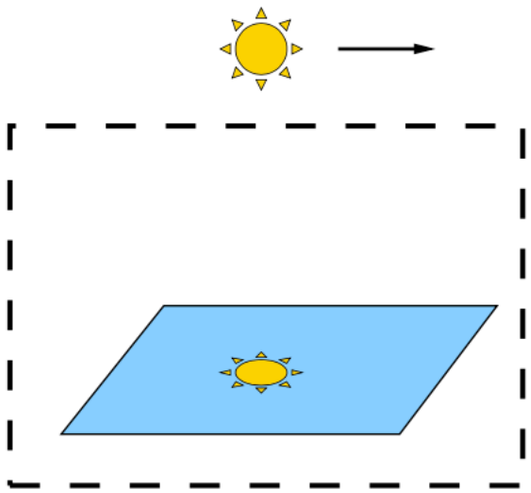
- Video = data cube
- Focus on grayscale images



- Two closely related but different notions
  - 2D motion field : projection of the 3D movement to a plane
  - Optical flow : apparent motion of a luminance pattern



Purely rotating ball: Null optical flow, real motion is not null

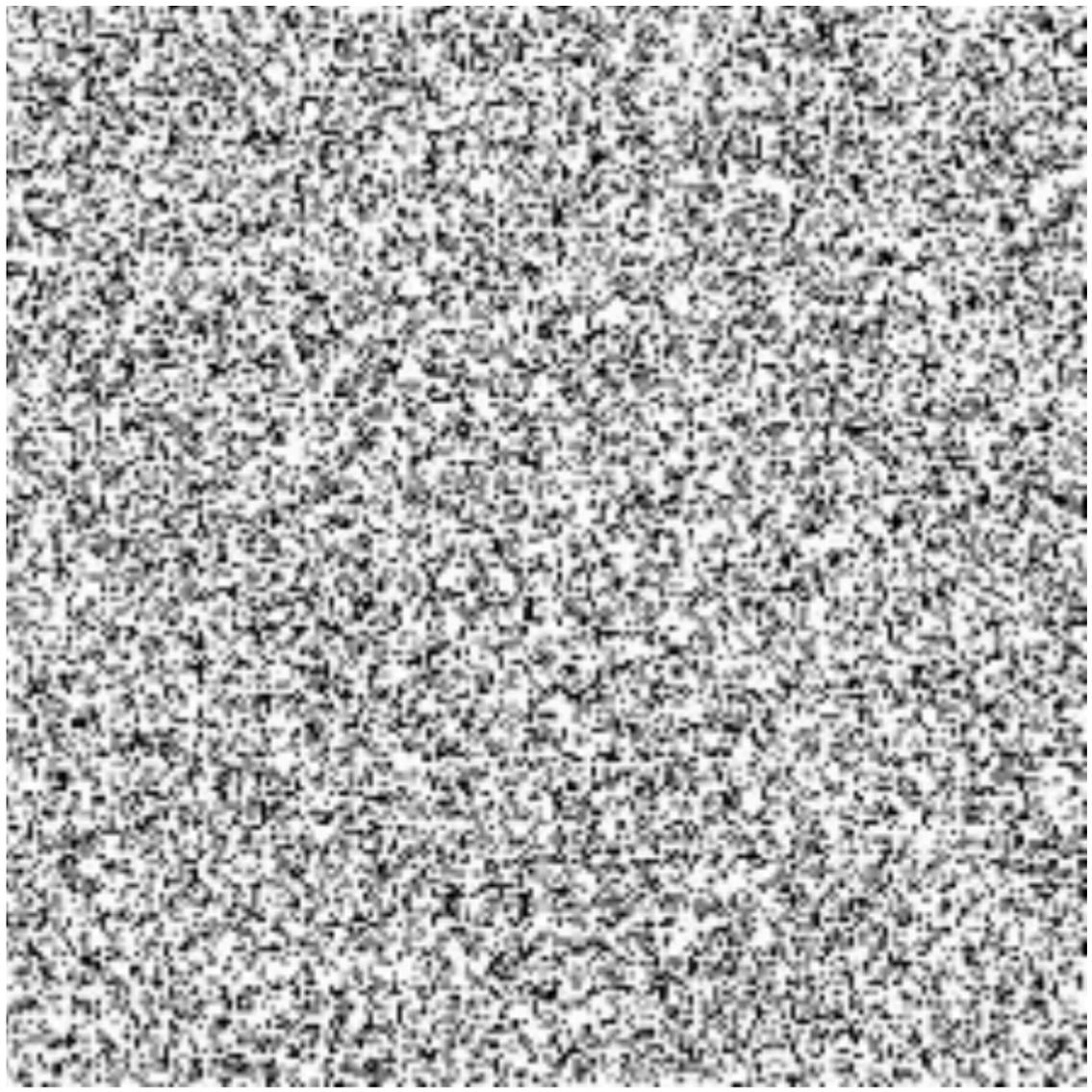


Reflecting surface illuminated by moving source: The optical flow is not null, but these is no physical motion

- Here, focus on optical flow

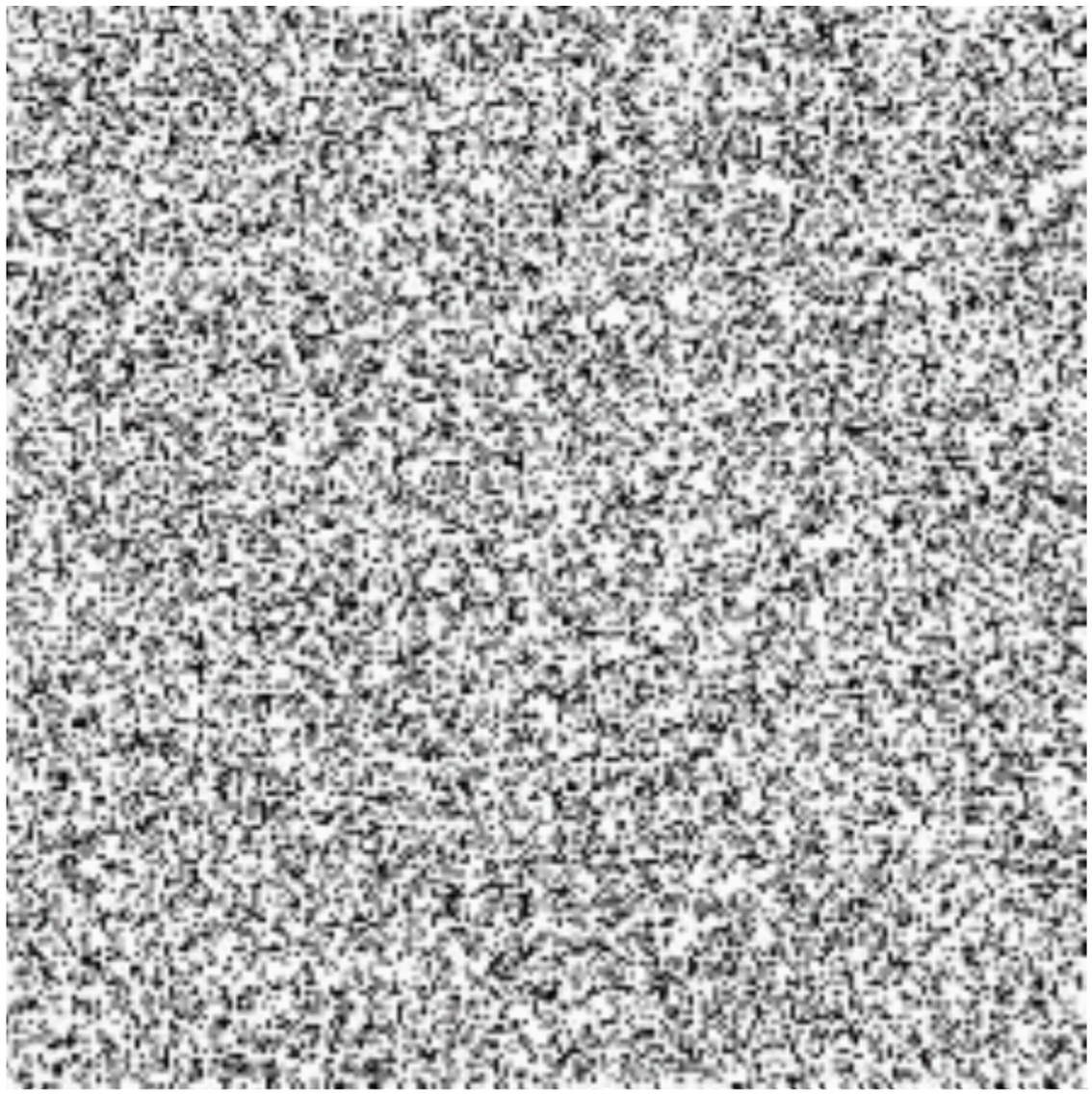
Why estimate motion ?

- Sometimes, motion is the only information we have about a scene



Why estimate motion ?

- Sometimes, motion is the only information we have about a scene



Motion estimation : applications

- Video shot boundary detection

=> enable to summarize videos

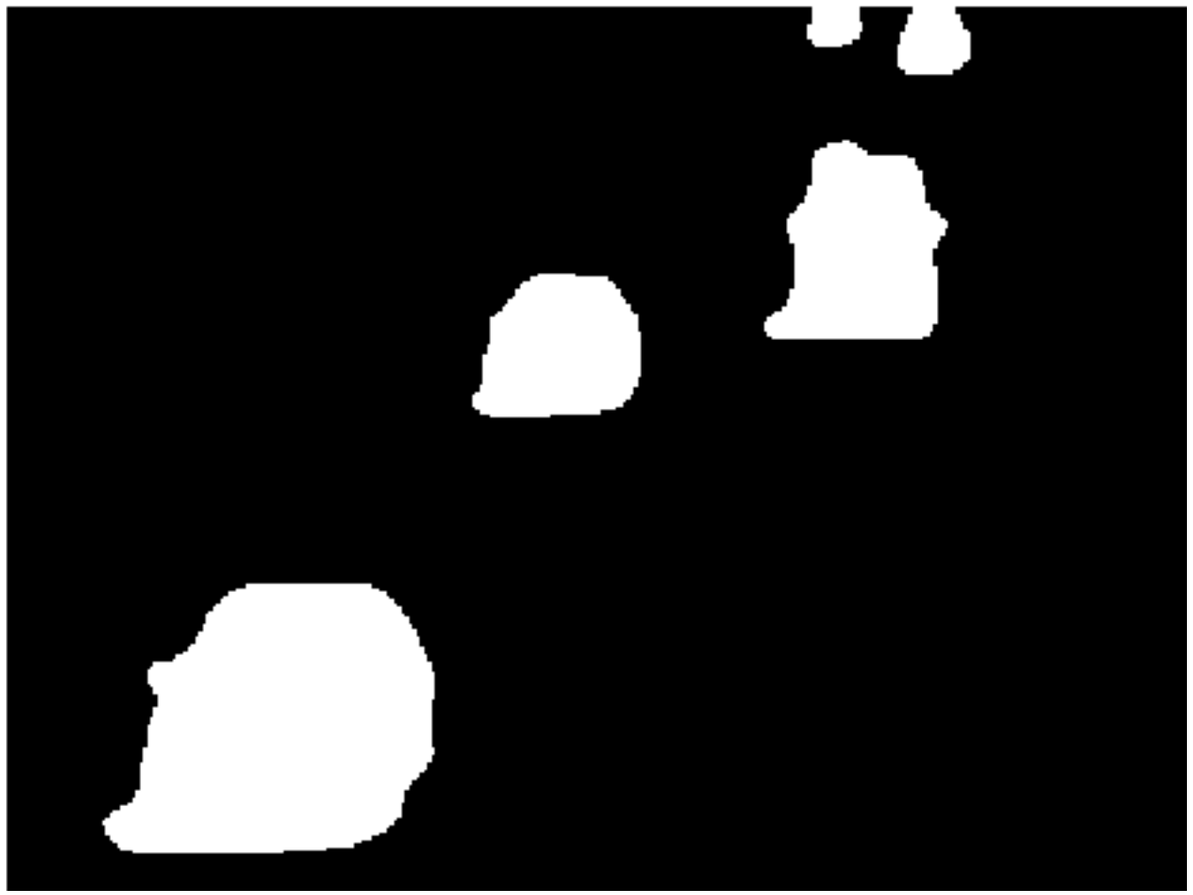
(a) Abrupt Transition

(b) Dissolve Transition

(c) Fade Transition

Motion estimation : applications

- Background subtraction



Motion estimation : applications

- Image segmentation

=> segment the image into coherently moving parts

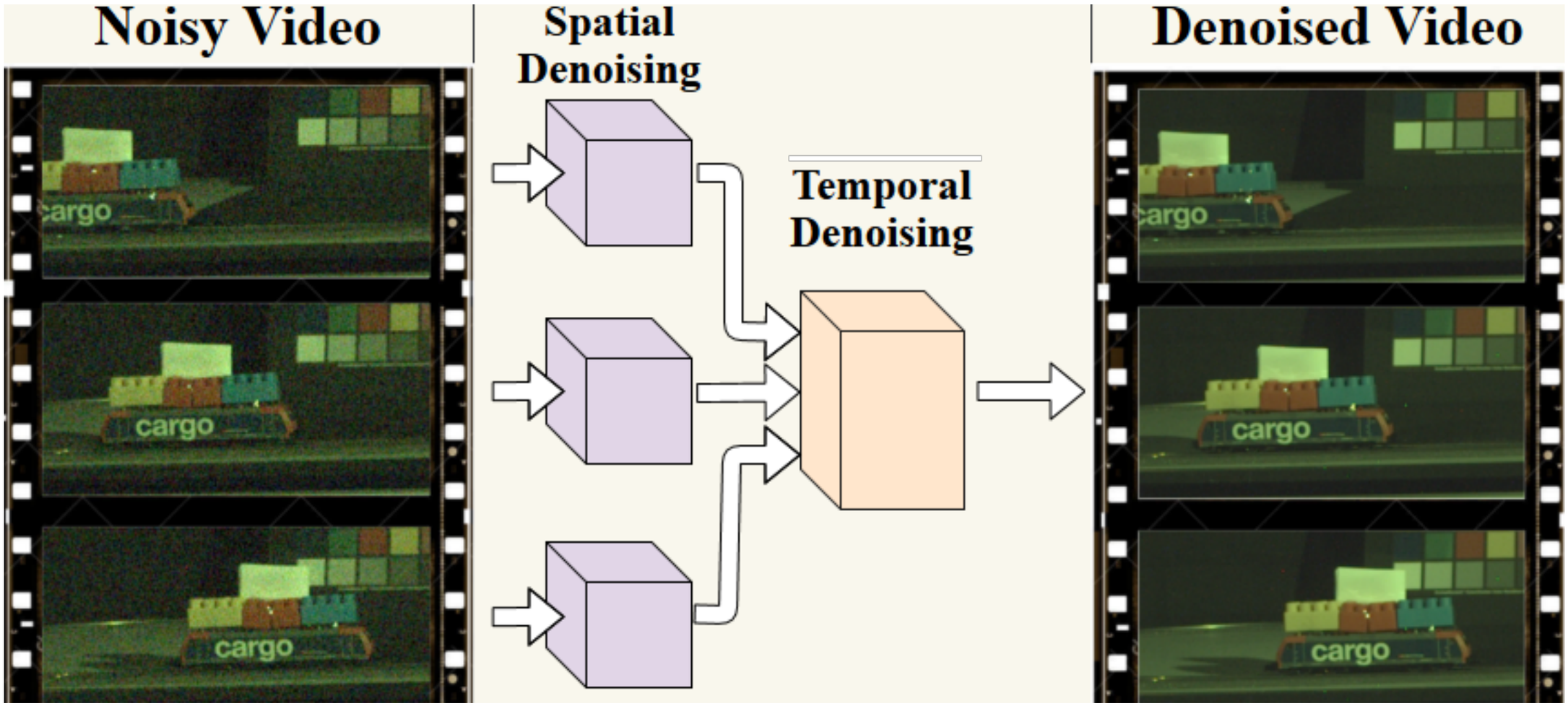
Motion estimation : applications

- Trajectory estimation

Motion estimation : applications

- Image restoration

=> e.g. denoising

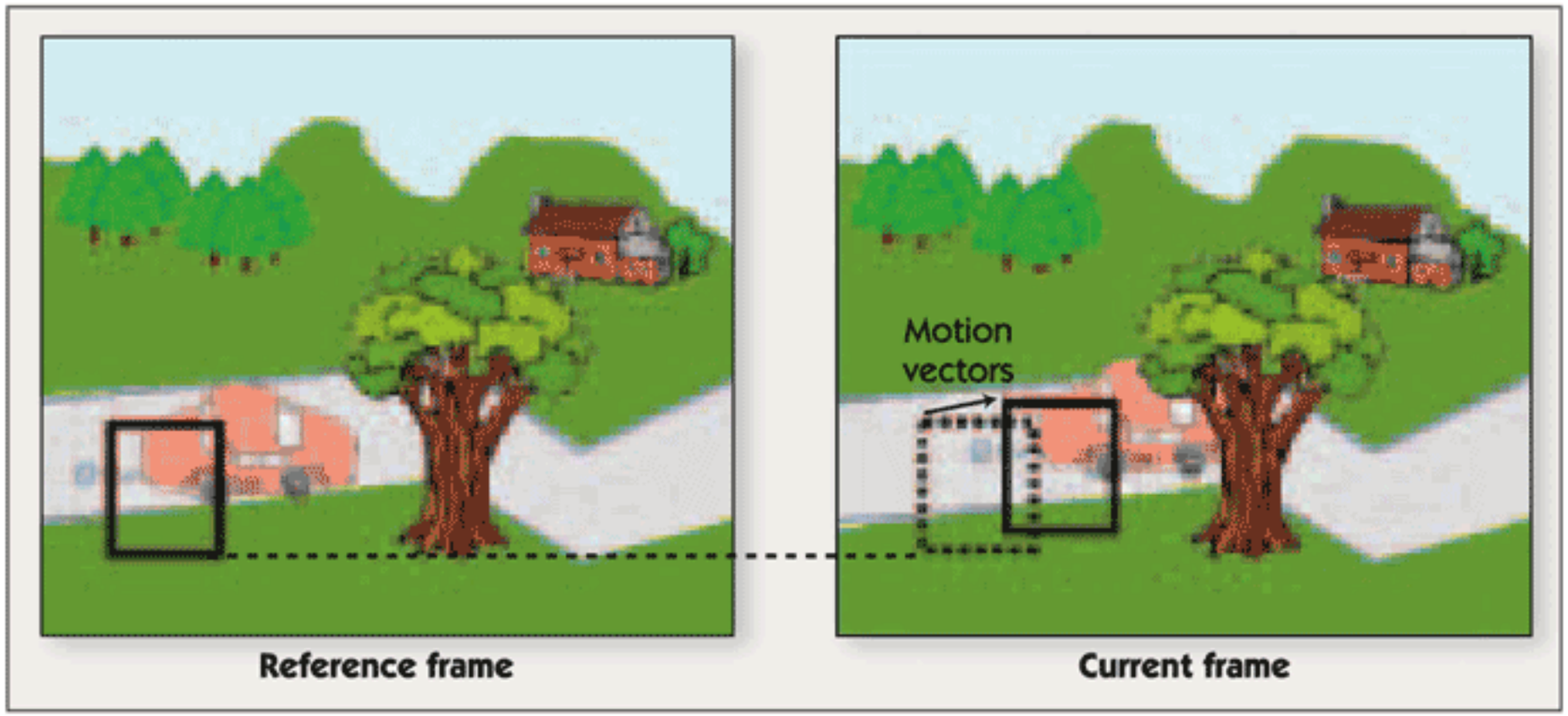


Motion estimation : applications

- Compression

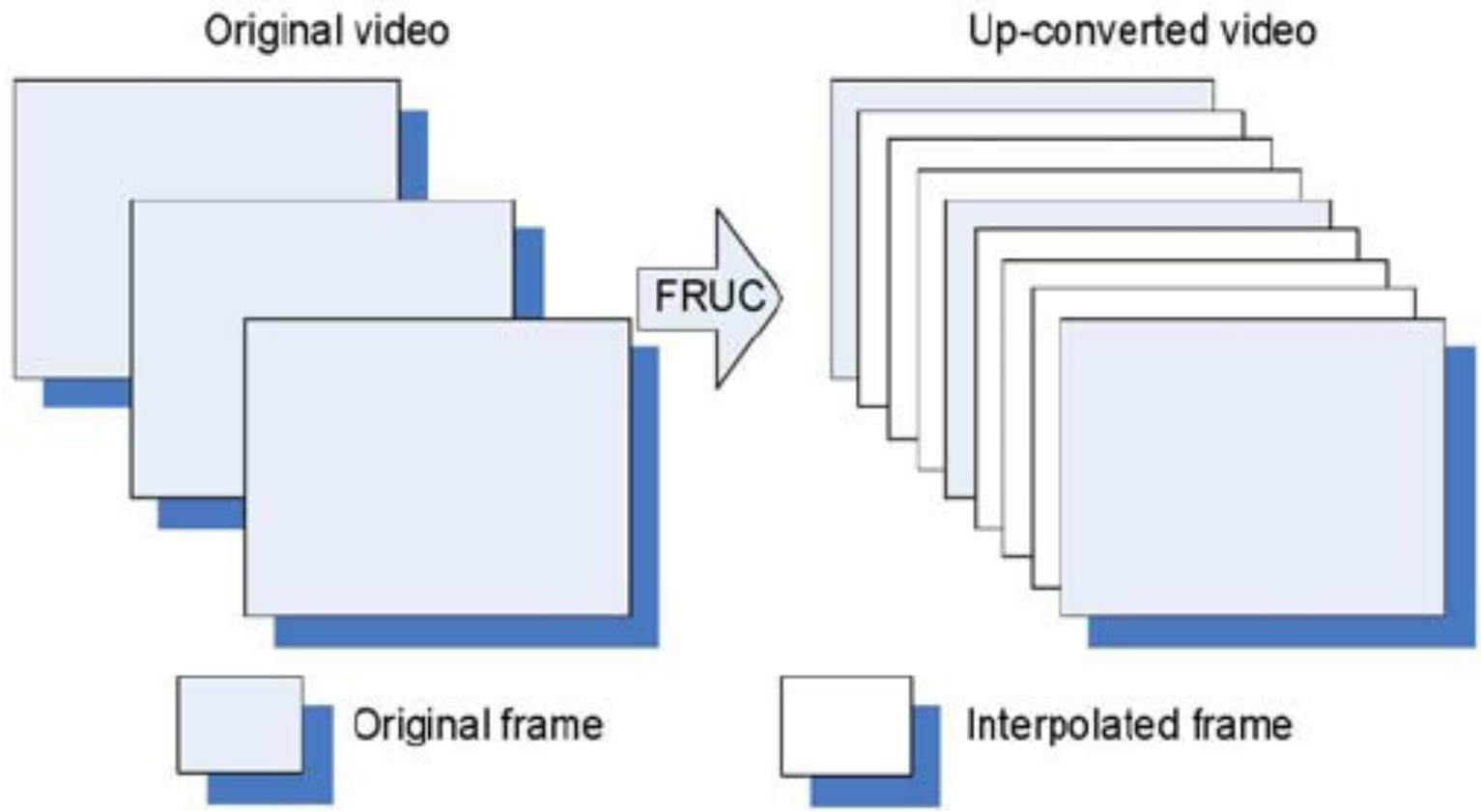
  - Motion estimation allows producing a prediction of the current image
  - Instead of sending the image, we only send the prediction error

  => If the prediction error is low, encoding it requires less bits than encoding the image



Motion estimation : applications

- Frame rate up conversion
  - Increase the number of frames in a video sequence (i.e. the temporal resolution)



Other applications

- 3D structure estimation
- learning dynamical models : how things move
- Recognizing activities
- Motion stabilization…

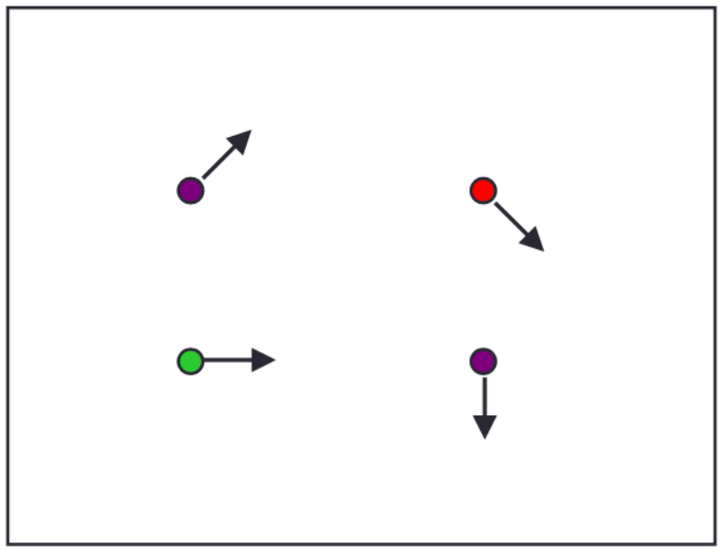
Course outline

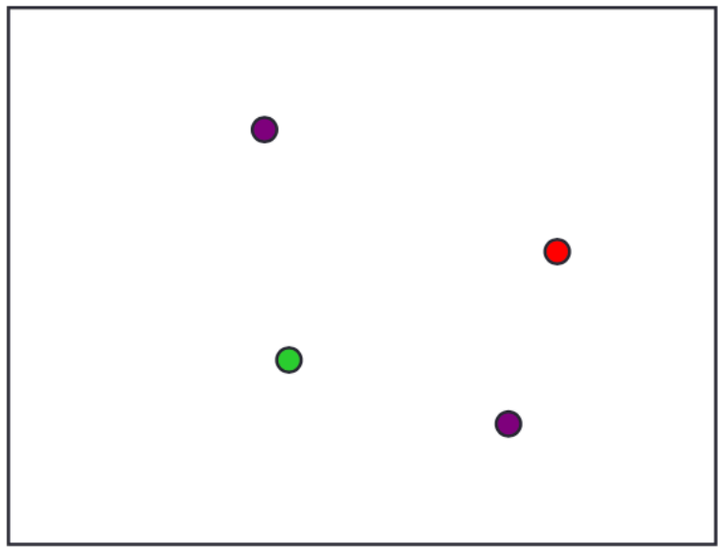
- **Optic flow and variational approaches**
- **Block matching methods**
- **Parametric motion estimation**
- **Deep learning methods**

Optic flow and variational approaches

Optical flow

- Estimate the motion for each pixel





$f(x, y, t)$ $f(x, y, t + 1)$

- **Motion is observed based on color / luminance variations**

- Requires two assumptions
  - **Constant color / illumination hypothesis** (CIH) : the object appearance does not change too much with time
  - **Small motion** : points do not move too far

More detailed hypotheses

- **Luminance does not change** along motion trajectory

$$f(x, y, t + T) = f(x - c(x, y), y - d(x, y), t)$$

- In practice, CIH is not always valid (sampling in time and space + noise, aliasing…)

- We assumed a **small** motion: doing a Taylor expansion,

$$f(x, y, t + T) = f(x, y, t) - c(x, y)\frac{\partial f}{\partial x}(x, y, t) - d(x, y)\frac{\partial f}{\partial y}(x, y, t)$$

$$\Rightarrow \frac{f(x, y, t + T) - f(x, y, t)}{T} = -\frac{c(x, y)}{T}\frac{\partial f}{\partial x}(x, y, t) - \frac{d(x, y)}{T}\frac{\partial f}{\partial y}(x, y, t)$$

$$\xrightarrow[T \to 0]{} \quad f_t + uf_x + vf_y = 0 \quad \textbf{Optic flow equation}$$

$$\text{with: } f_x = \frac{\partial f}{\partial x} \text{ (idem for } f_y \text{ and } f_t\text{) and } u = \lim_{T \to 0} \frac{c(x, y)}{T},\ v = \lim_{T \to 0} \frac{d(x, y)}{T}$$

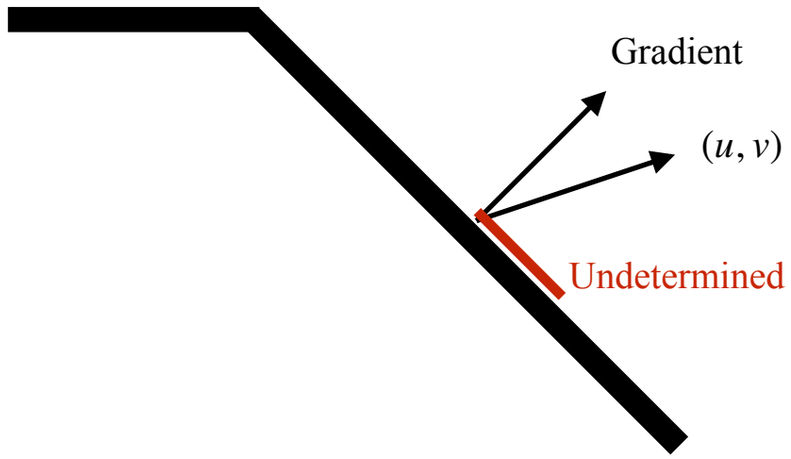
Optic flow : indeterminacies

$$f_t + uf_x + vf_y = 0$$

Per pixel, a single equation for two unknowns…

Rewriting $f_t + [u, v] \cdot \begin{bmatrix} f_x \\ f_y \end{bmatrix} = f_t + [u, v] \cdot \nabla f = 0$

=> The component of the flow perpendicular to ∇f cannot be determined.



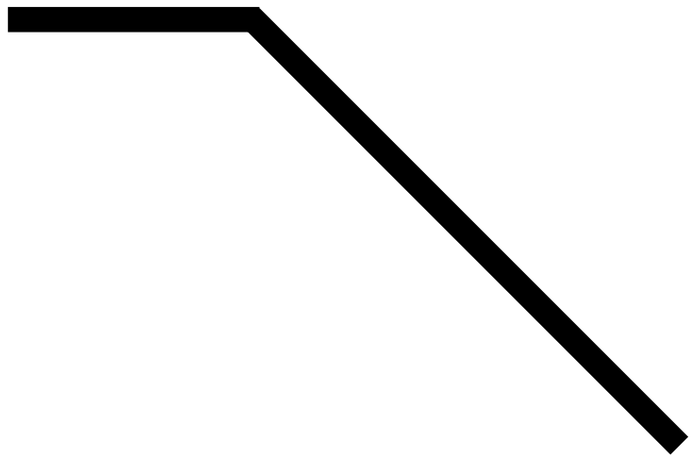
Optic flow : indeterminacies

$$f_t + uf_x + vf_y = 0$$

Per pixel, a single equation for two unknowns…

Rewriting $f_t + [u, v] \cdot \begin{bmatrix} f_x \\ f_y \end{bmatrix} = f_t + [u, v] \cdot \nabla f = 0$

=> The component of the flow perpendicular to ∇f cannot be determined.



=> Also known as the aperture problem

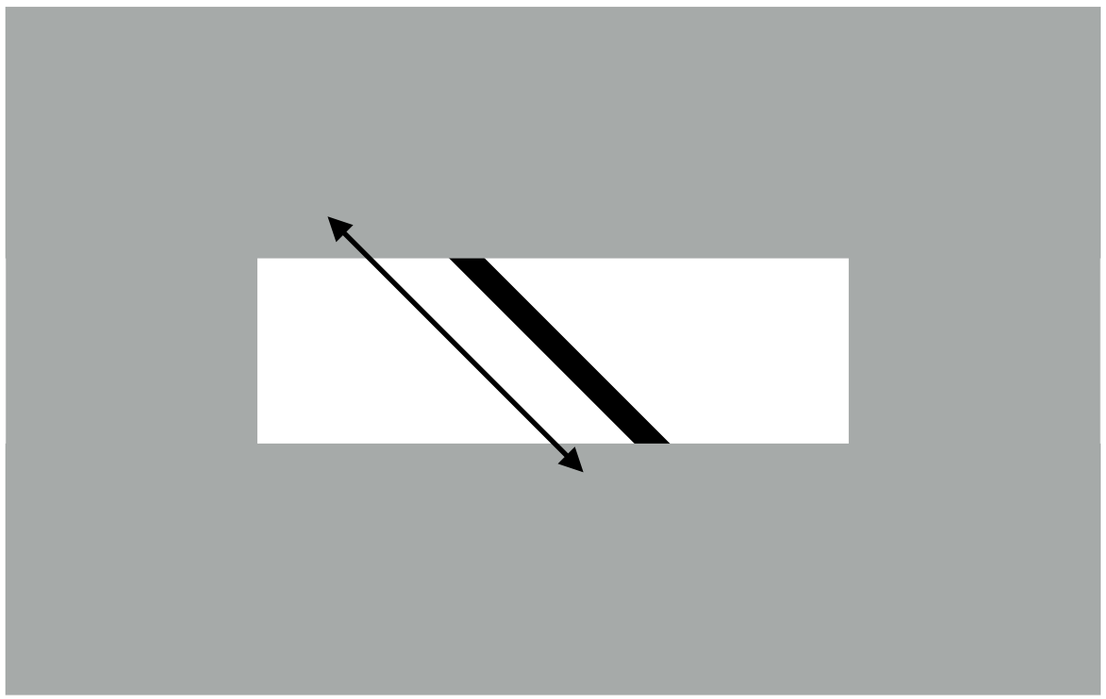
Optic flow : indeterminacies

$$f_t + uf_x + vf_y = 0$$

Per pixel, a single equation for two unknowns…

Rewriting $f_t + [u, v] \cdot \begin{bmatrix} f_x \\ f_y \end{bmatrix} = f_t + [u, v] \cdot \nabla f = 0$

=> The component of the flow perpendicular to ∇f cannot be determined.



=> Also known as the aperture problem

Variational approach

$$f_t + uf_x + vf_y = 0$$

- In practice, there is noise in real data
  => don't set the left-hand term to zero but minimize its energy

- We have seen the indeterminacies
  => need additional constraints to make the problem better posed.

- The problem is recast as an optimization / variational problem of the form

$$\arg\min_{u,v} \{D[f_0, f_1, (u, v)] + R(u, v)\}$$

Data fidelity term

Regularization term

Horn and Schunck algorithm

$$f_t + uf_x + vf_y = 0$$

- Horn and Schunck introduced a total variation (TV) constraint on the velocities (u, v) over a region $\mathscr{R}$:

$$\min_{u,v} \iint_{\mathscr{R}} (uf_x + vf_y + f_t)^2 dxdy \qquad \text{s.t.} \qquad \iint_{\mathscr{R}} \|\nabla u\|^2 + \|\nabla v\|^2 dxdy \leq \tau$$

- This is rewritten using a Lagrange multiplier as

$$\min_{u,v} \iint_{\mathscr{R}} (uf_x + vf_y + f_t)^2 dxdy + \lambda \iint_{\mathscr{R}} \|\nabla u\|^2 + \|\nabla v\|^2 dxdy$$

- A solution can be found by solving

$$\begin{cases} \lambda \nabla^2 u = \dfrac{\partial f}{\partial x}\left[u\dfrac{\partial f}{\partial x} + v\dfrac{\partial f}{\partial y} + \dfrac{\partial f}{\partial t}\right], \\ \lambda \nabla^2 v = \dfrac{\partial f}{\partial y}\left[u\dfrac{\partial f}{\partial x} + v\dfrac{\partial f}{\partial y} + \dfrac{\partial f}{\partial t}\right]. \end{cases} \quad \text{where} \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Horn and Schunck algorithm : limitations

- Solve

$$\left|\begin{array}{l} \lambda\nabla^2 u = \dfrac{\partial f}{\partial x}\left[u\dfrac{\partial f}{\partial x} + v\dfrac{\partial f}{\partial y} + \dfrac{\partial f}{\partial t}\right], \\ \lambda\nabla^2 v = \dfrac{\partial f}{\partial y}\left[u\dfrac{\partial f}{\partial x} + v\dfrac{\partial f}{\partial y} + \dfrac{\partial f}{\partial t}\right]. \end{array}\right. \quad \text{where} \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Difficulties

- Complex resolution: requires to solve a system of partial differential equations
- Critical choice of λ (smoothness + numerical stability)
- The theory does not help for finding λ

Lucas Kanade algorithm

Other approach than HS algorithm:

- does not add directly a smoothness term
- instead, it is assumed that the optical flow is approximately constant in a small neighborhood

Main idea

- Recall the aperture problem: too much variables in $f_t + uf_x + vf_y = 0$

- To increase the number of equations, pretend (u, v) is constant in a pixel neighborhood of size n

$$\begin{bmatrix} f_x(x_1, y_1) & f_y(x_1, y_1) \\ f_x(x_2, y_2) & f_y(x_2, y_2) \\ & \cdots \\ f_x(x_n, y_n) & f_y(x_n, y_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_t(x_1, y_1) \\ f_t(x_2, y_2) \\ \cdots \\ f_t(x_n, y_n) \end{bmatrix}$$

- For instance, if you use 5x5 windows, you get $n = 25$ equations !

Lucas Kanade algorithm

$$\begin{bmatrix} f_x(x_1,y_1) & f_y(x_1,y_1) \\ f_x(x_2,y_2) & f_y(x_2,y_2) \\ & \cdots \\ f_x(x_n,y_n) & f_y(x_n,y_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_t(x_1,y_1) \\ f_t(x_2,y_2) \\ \cdots \\ f_t(x_n,y_n) \end{bmatrix}$$

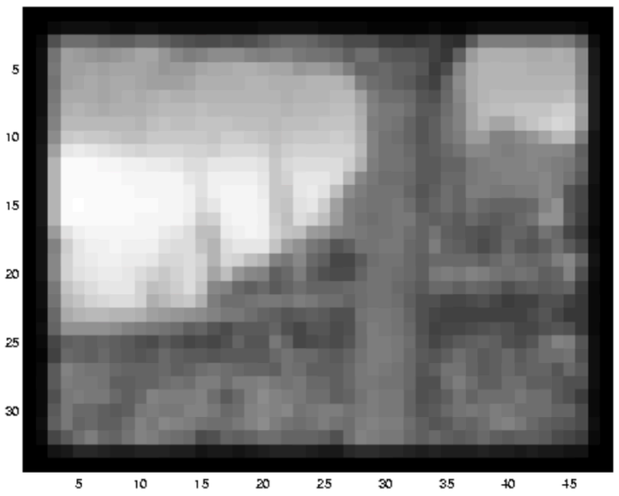
- More equations than unknowns (over-constrained problem)
- Solved by least-square approach

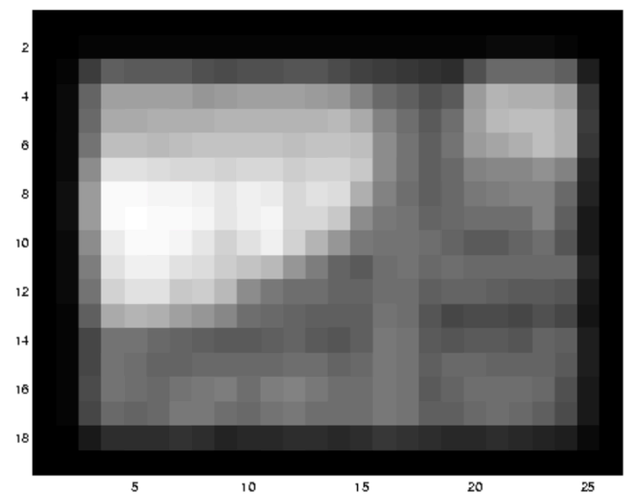
$$\min_{u,v} \sum_{i=1\ldots n} w_i \left(u \frac{\partial f}{\partial x}(x_i,y_i) + v \frac{\partial f}{\partial y}(x_i,y_i) + \frac{\partial f}{\partial t}(x_i,y_i) \right)^2$$

where the w_i are additional weights used to give more influence to the center of $\mathscr{R}$

Lucas Kanade algorithm : dealing with larger motion

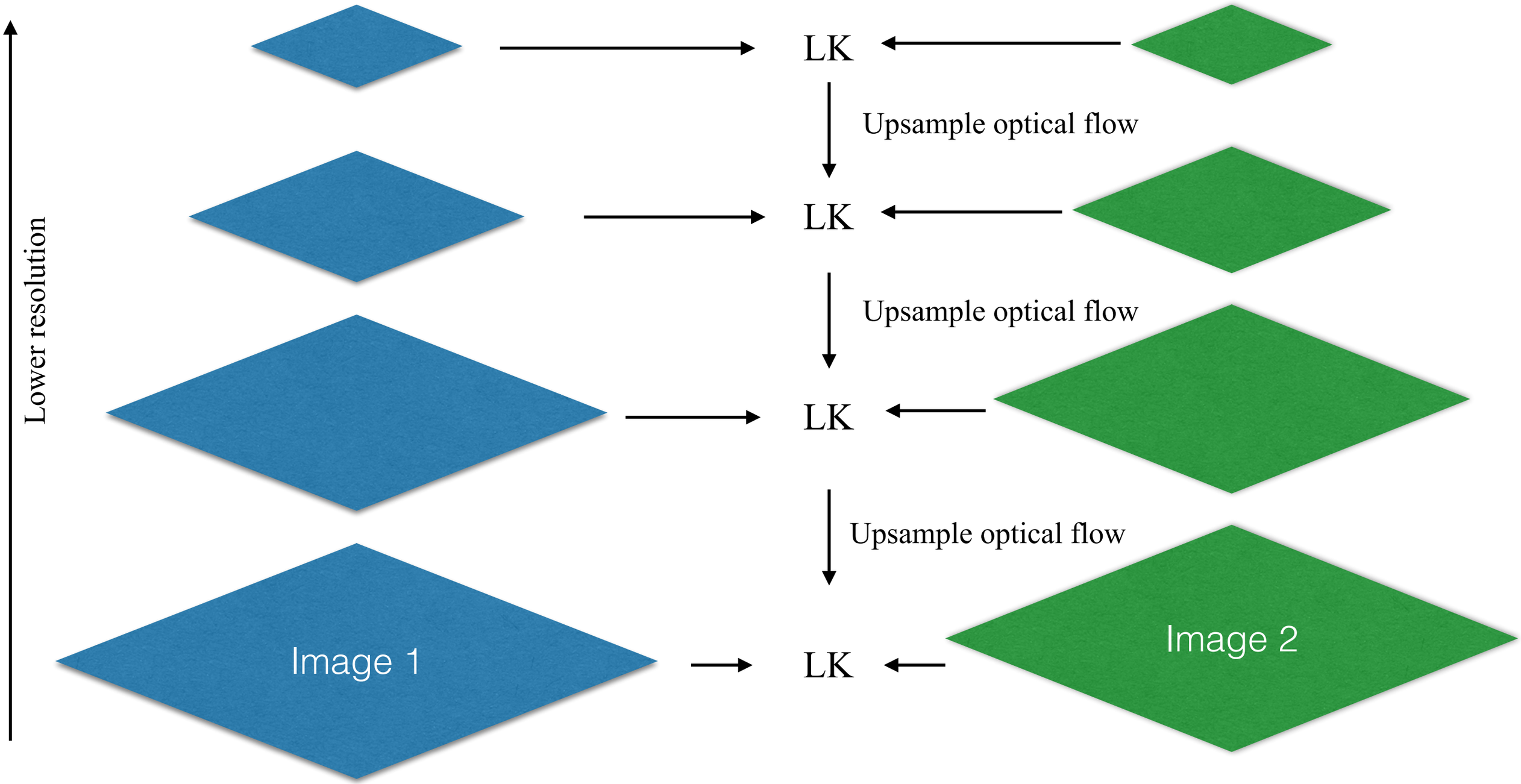
- Lucas Kanade method is based on a first order development, which is valid for only small motions. How to deal with larger ones ?
- Main idea: process iteratively by reducing the images resolutions



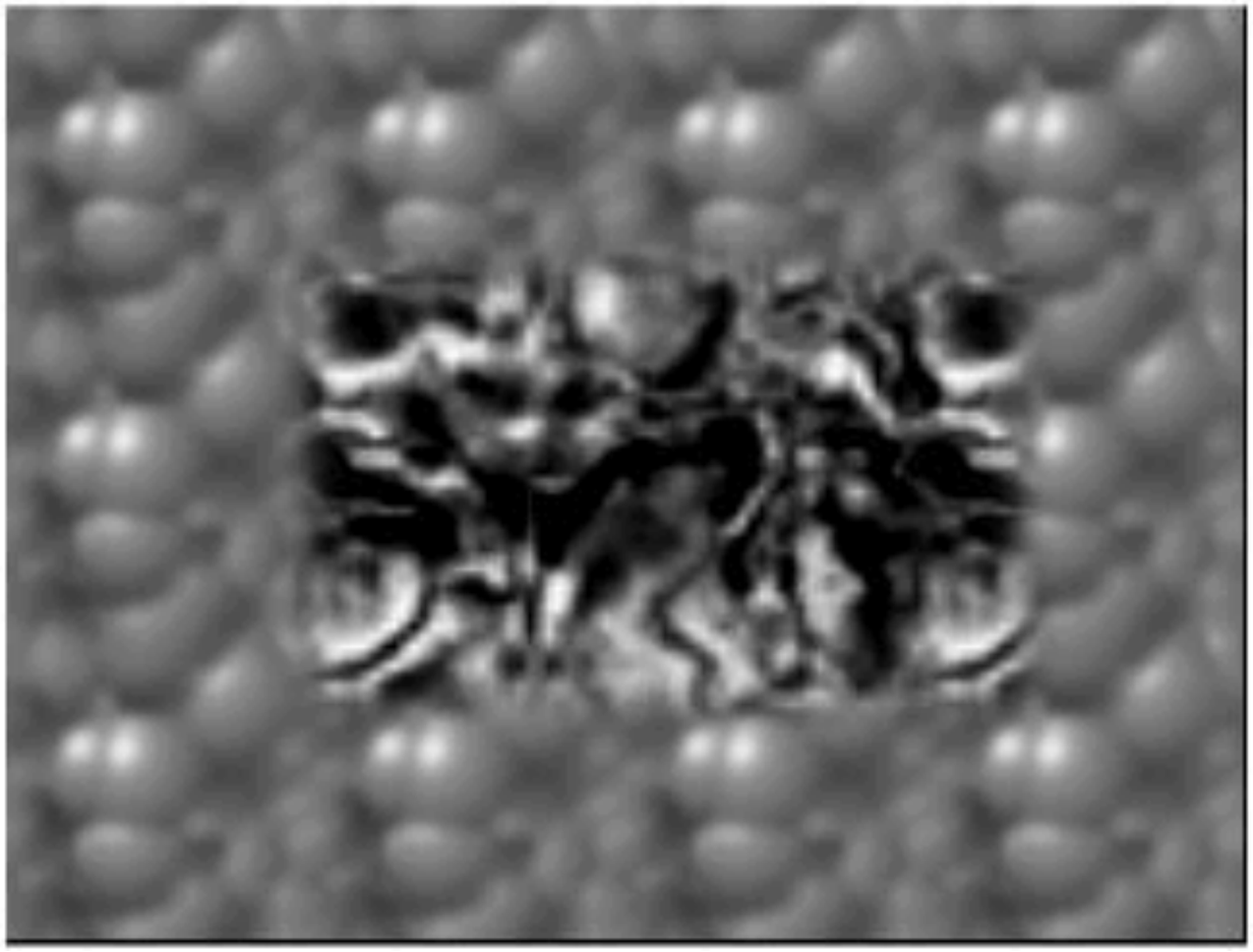


Hierarchical Lucas Kanade algorithm

- Main idea: process iteratively by reducing the images resolutions

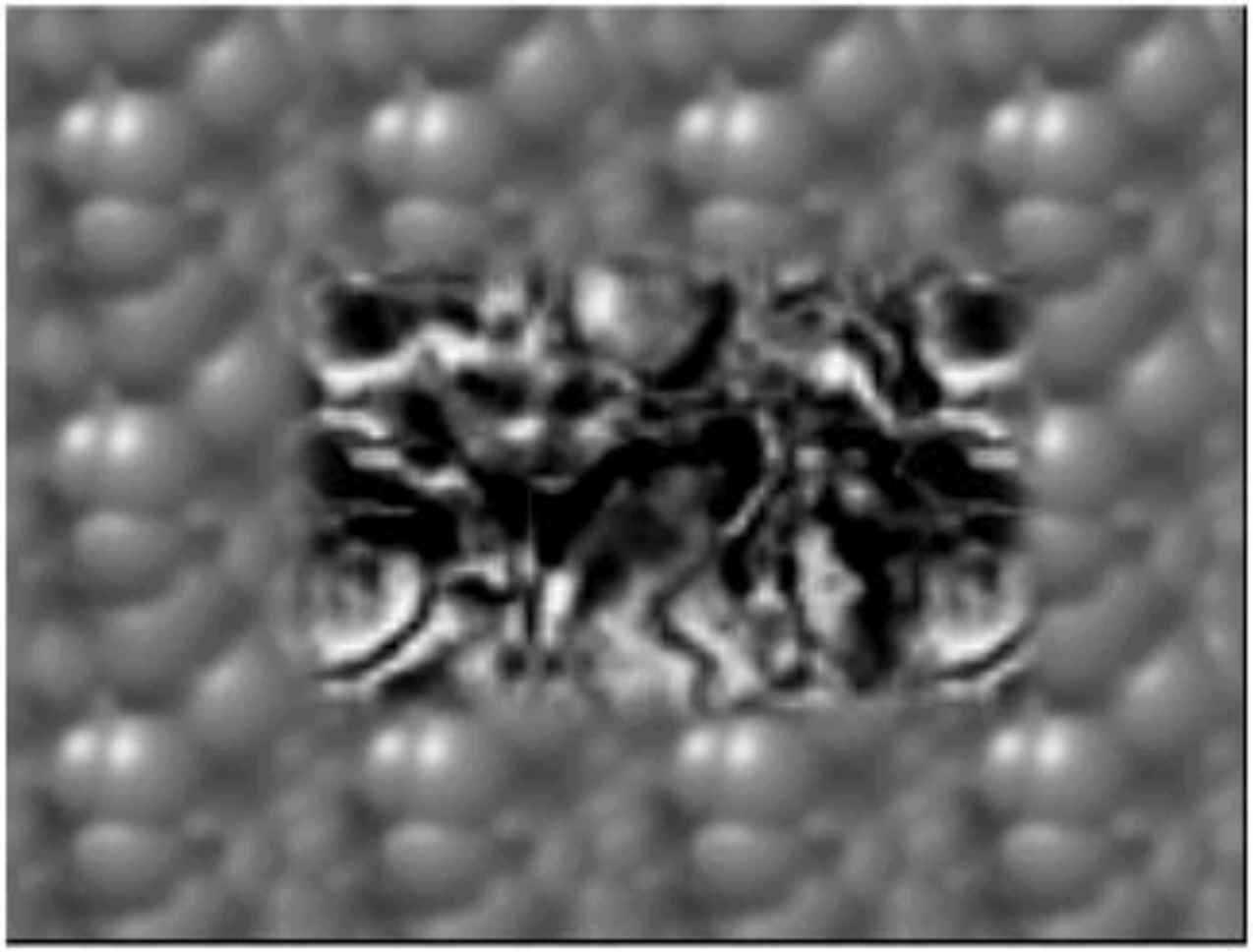


Hierarchical Lucas Kanade algorithm: example



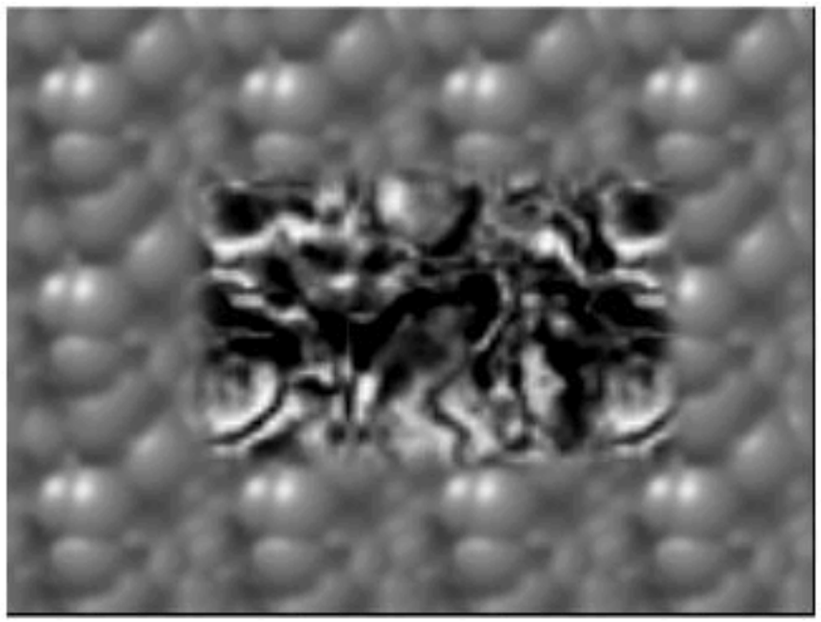
Target image / initial image

Hierarchical Lucas Kanade algorithm: example

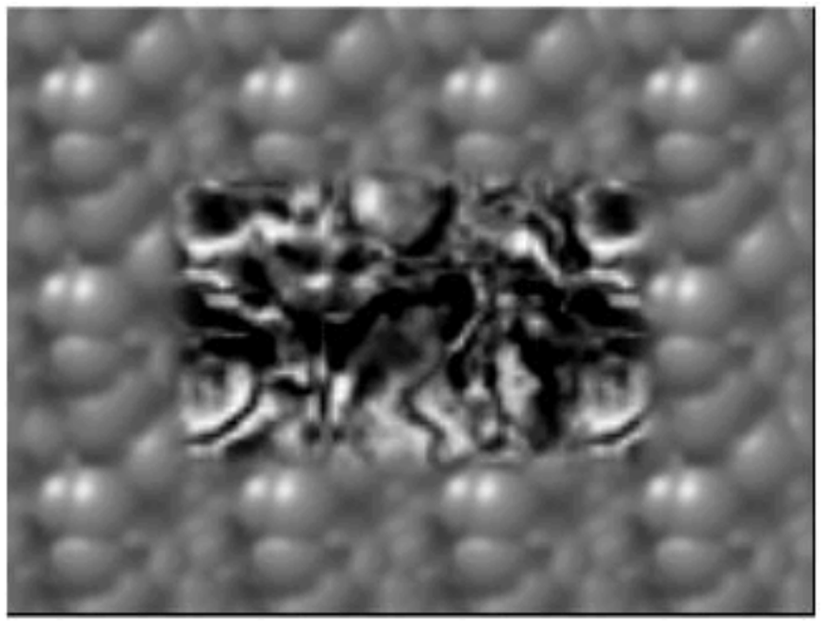


Target image / initial image

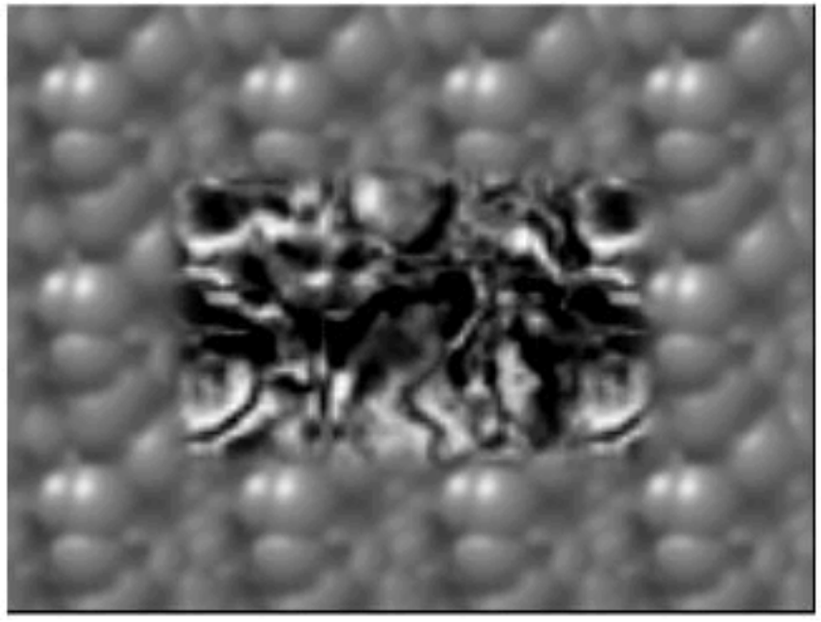
Hierarchical Lucas Kanade algorithm: example of results



Target image / initial image

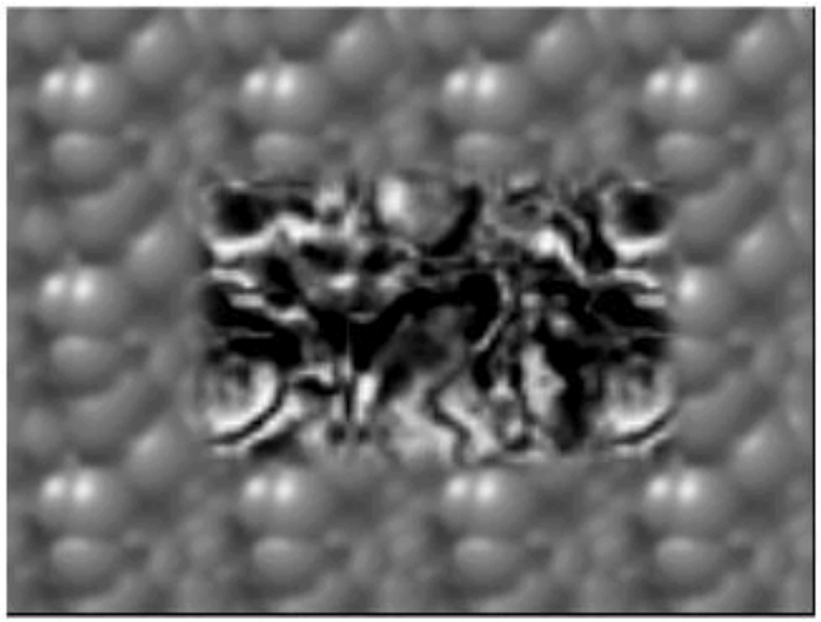


Target image / warped image from LK optic flow

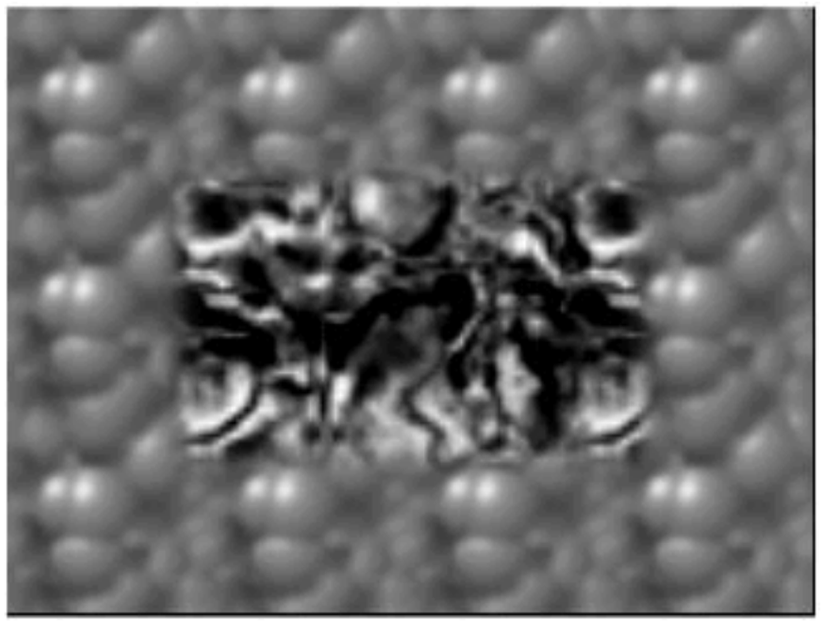


Target image / warped image from hierarchical LK optic flow (4 levels)

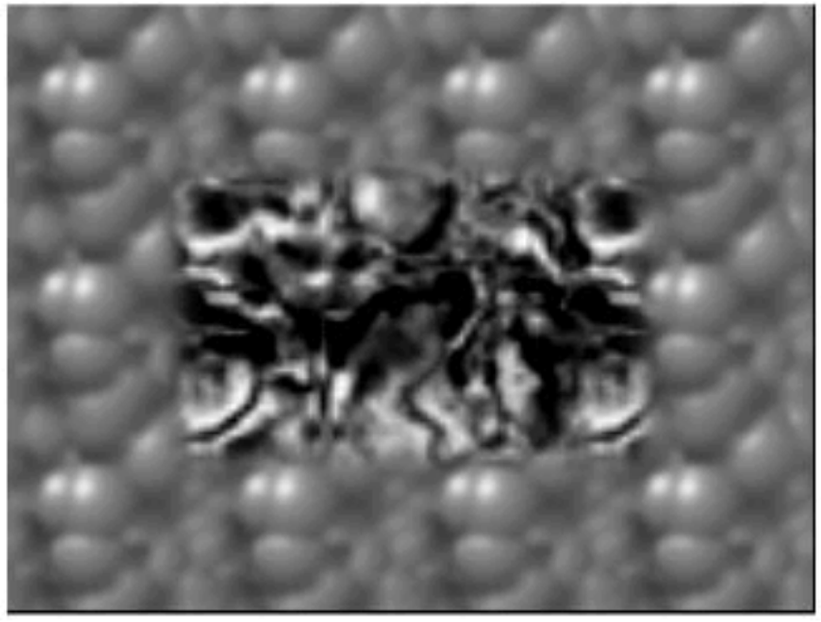
Hierarchical Lucas Kanade algorithm: example of results



Target image / initial image

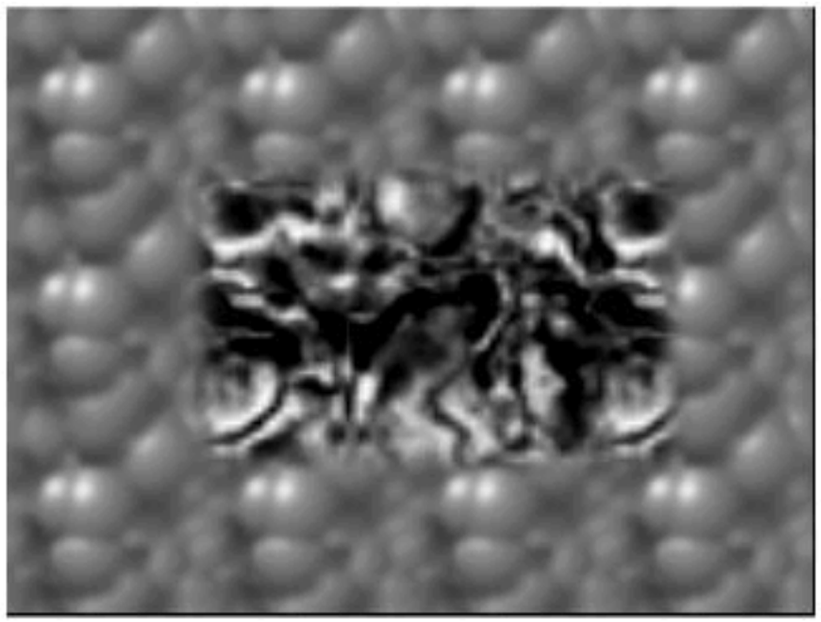


Target image / warped image from LK optic flow

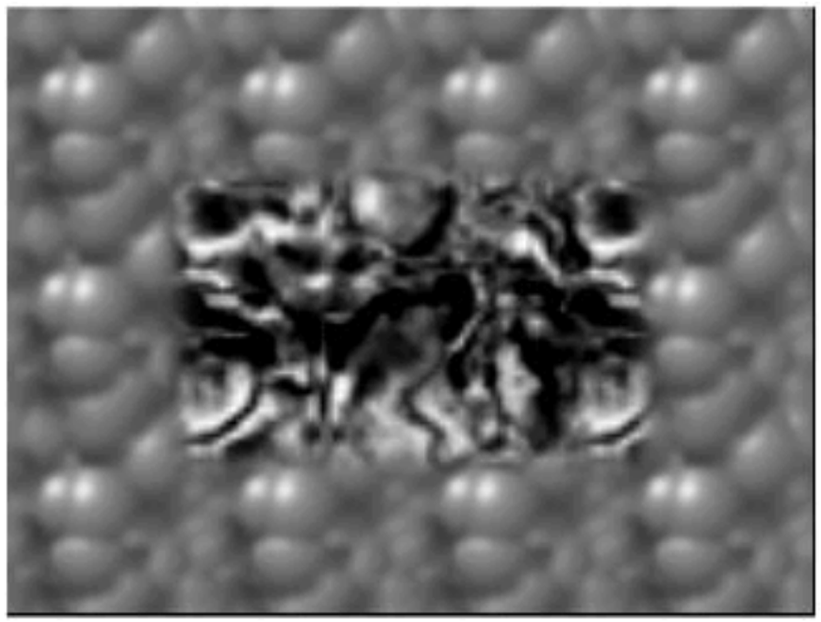


Target image / warped image from hierarchical LK optic flow (4 levels)

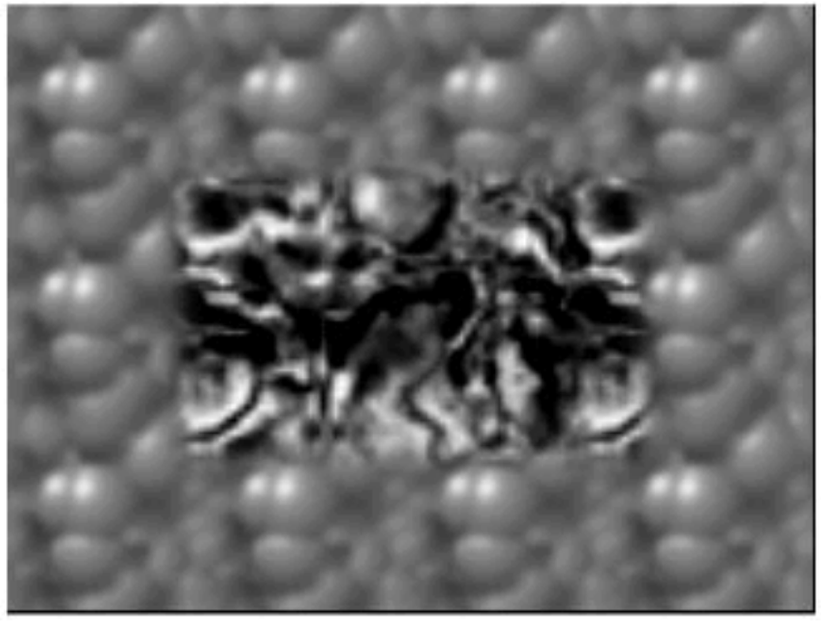
Hierarchical Lucas Kanade algorithm: example of results



Target image / initial image

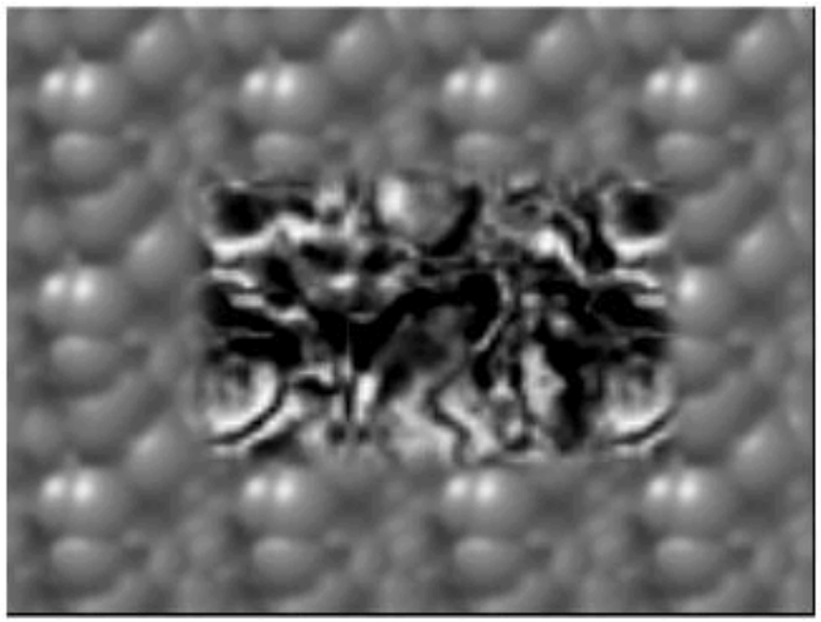


Target image / warped image from LK optic flow

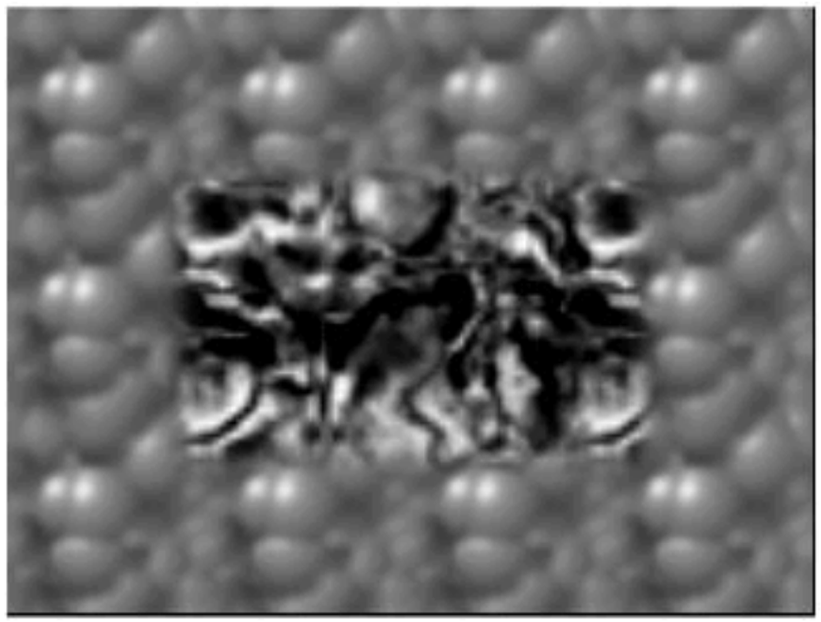


Target image / warped image from hierarchical LK optic flow (4 levels)

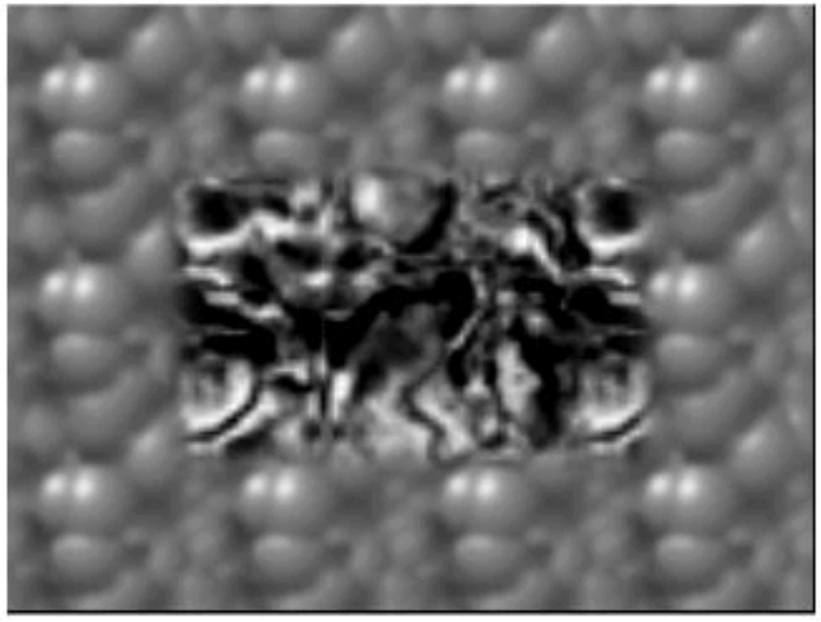
Hierarchical Lucas Kanade algorithm: example of results



Target image / initial image



Target image / warped image from LK optic flow



Target image / warped image from hierarchical LK optic flow (4 levels)

Optical flow and variational approach : summary

- The optical flow computes a *dense* motion estimation : a vector per pixel is estimated. Useful for denoising, restoration…

- Both HS and LK produce bad results close to object boundaries.

- They are based on a first order development, which is valid for only small motions.
  => But hierarchical adaptations might help

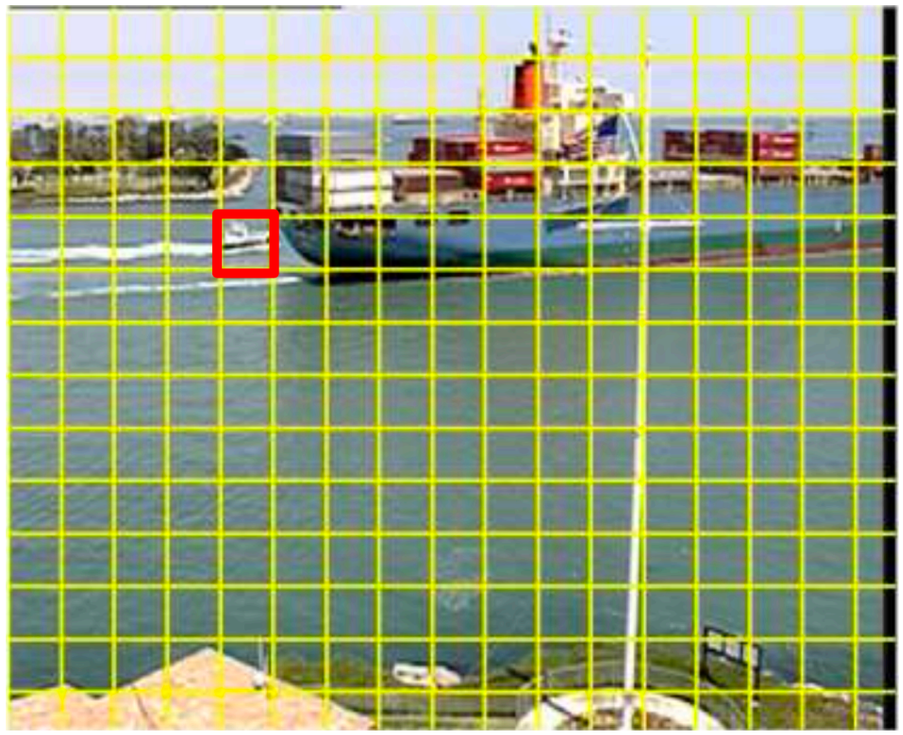
- Gradient computation might be sensitive to noise

- Also, it is not clear how to choose $\mathscr{R}$.

Block matching methods

Motivation

- The optical flow computes a *dense* motion estimation : a vector per pixel is estimated… This is not very robust to noise
- In block matching methods (BM), a motion vector is estimated over a whole block
- Using block is an additional regularity constraint, making the problem better-posed
- BM is extremely popular in video compression



Notations

- We consider images of size $N \times M$
- Block $B_{p,q}$: set of indices starting at pixel (p, q)

$$B_{p,q} = \{p, p+1, \ldots, p+P-1\} \times \{q, q+1, \ldots, q+Q-1\}$$

- Le $\mathbf{f}_t$ be a PQ vector of the luminance values in the different pixels of a block

$$\mathbf{f}_t(B_{p,q}) = \begin{bmatrix} f(p,q,t) \\ f(p+1,q,t) \\ \cdots \\ f(p+P-1,q,t) \\ f(p,q+1,t) \\ \cdots \\ f(p+P-1,p+Q-1,t) \end{bmatrix}$$

- BM consists in finding the motion vector minimizing a *dissimilarity measure* between $\mathbf{f}_t(B_{p,q})$ and $\mathbf{f}_k(B_{p,q})$

Motivation

$\mathbf{f}_t(B_{p,q})$

$\mathbf{f}_k(B_{p,q})$ $\mathbf{f}_k(B_{p+\hat{i},q+\hat{j}})$

- Where $(\hat{i}, \hat{j})$ is the estimated motion vector of the $B_{p,q}$ block from the image t to the image k

BM : global picture

- BM consists in solving for each block $B_{p,q}$ of the current image t :

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} d\left[\mathbf{f}_t(B_{p,q}), \mathbf{f}_k(B_{p-i,q-j})\right]$$
$$= \texttt{argmin}_{(i,j)\in\mathscr{W}} J(i,j)$$

- Different BM techniques depending on
  - The dissimilarity measure d
  - The search windows in $\mathscr{W}$ and the way $\mathscr{W}$ is scanned
  - The block size and shape

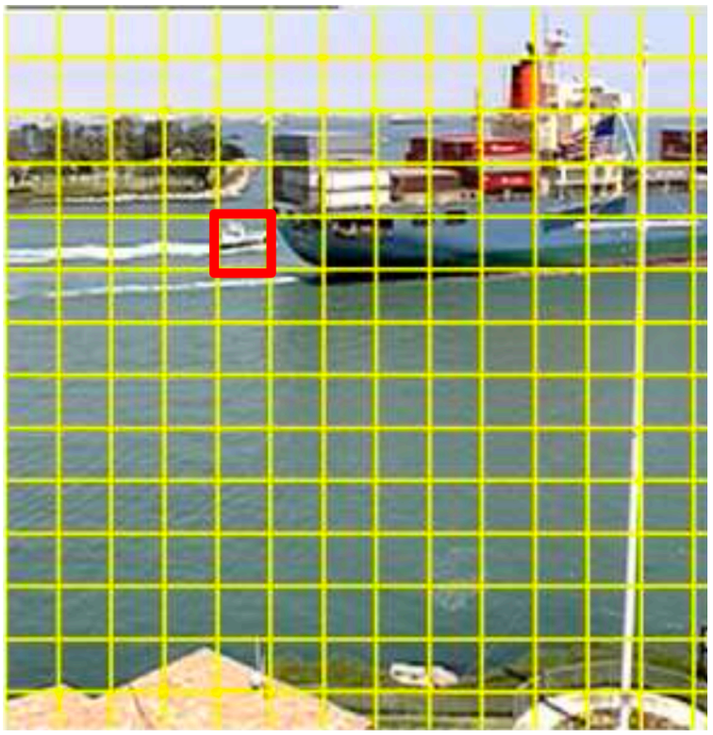


Image at t time

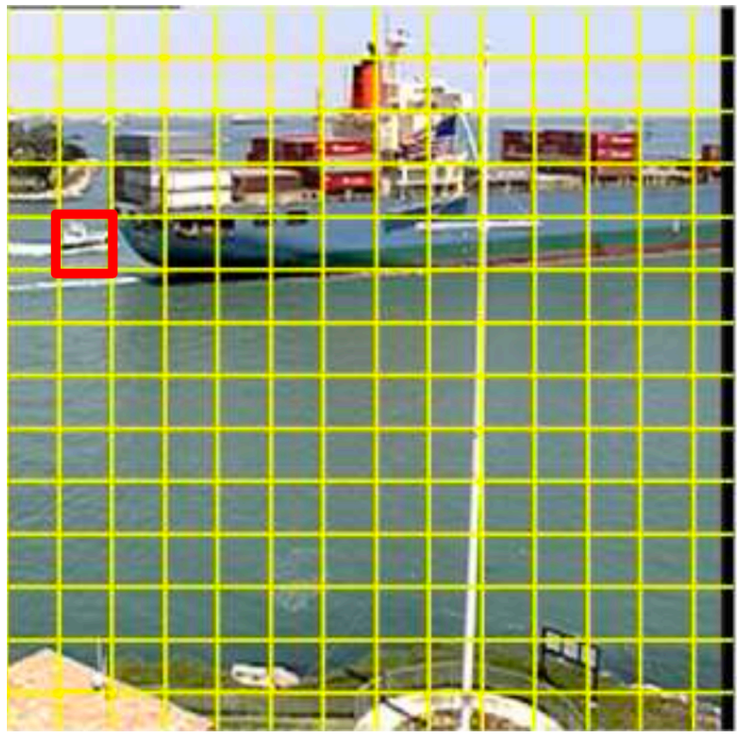


Image at k time

N.B. 1 : this is a specific case in which the stride between blocks is equal to the block sizes. But the blocks can overlap

N.B. 2 : the movement is not necessarily a multiple of the block size

Dissimilarity measure : SSD

- Easy choice of d : Euclidean norm (*i.e.* minimization of energy)

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q}) - \mathbf{f}_k(B_{p-i,q-j})\|_2^2$$

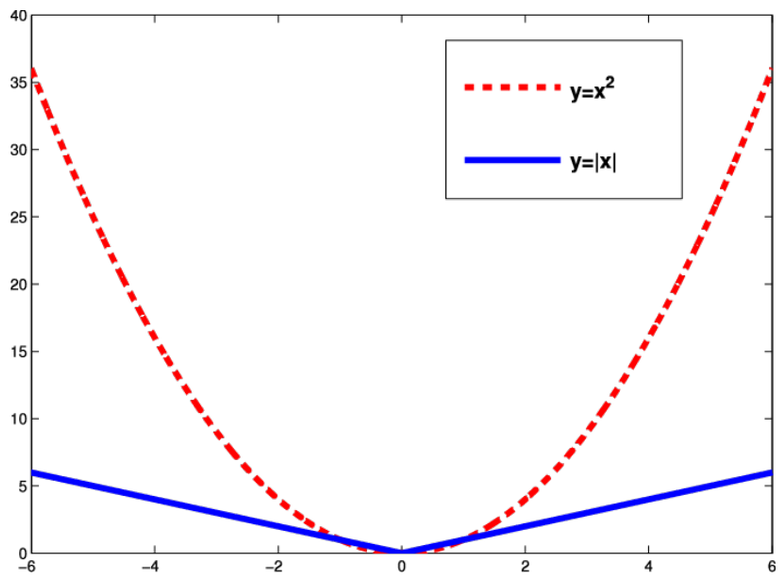
- Called sum-of-squared differences (SSD)

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(n,m)\in B_{b,q}} [f(n,m,t) - f(n-i,m-j,h)]^2$$

- The compensated image is given by replacing $\mathbf{f}_t(B_{p,q})$ by $\mathbf{f}_k(B_{p-i,q-j})$

Drawbacks

- Highlight errors (not robust to outliers)
- Does not take into account global illumination changes



Dissimilarity measure : SAD

- Robustness to outliers : use ℓ_1 norm

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q}) - \mathbf{f}_k(B_{p-i,q-j})\|_1$$

- Called sum-of-absolute differences (SAD)

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(n,m)\in B_{b,q}} |f(n,m,t) - f(n-i,m-j,h)|$$

- The prediction error is larger than with ℓ_2 norm
- The motion field is usually smoother than with the ℓ_2-norm

SSD

SAD

Dissimilarity measure : SAD

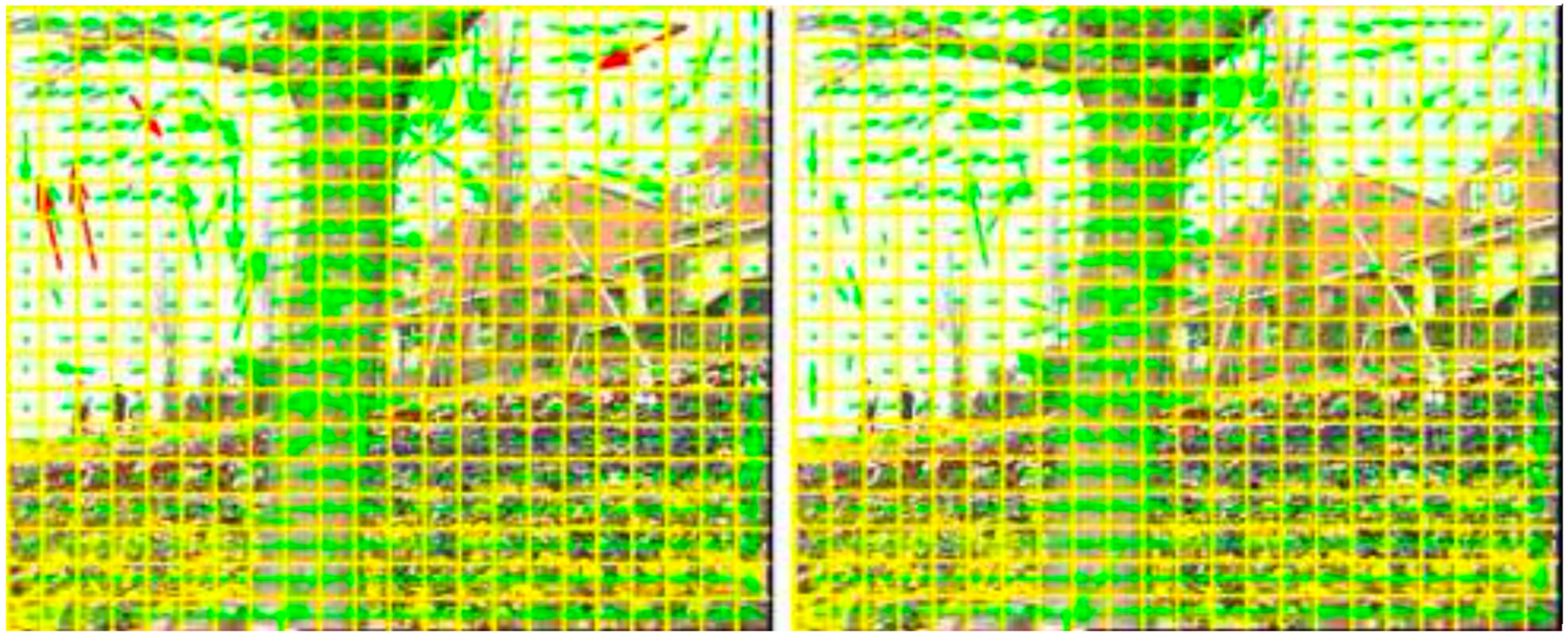
- Robustness to outliers : use ℓ_1 norm

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q}) - \mathbf{f}_k(B_{p-i,q-j})\|_1$$

- Called sum-of-absolute differences (SAD)

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(n,m)\in B_{b,q}} |f(n,m,t) - f(n-i,m-j,h)|$$

- The prediction error is larger than with ℓ_2 norm
- The motion field is usually smoother than with the ℓ_2-norm



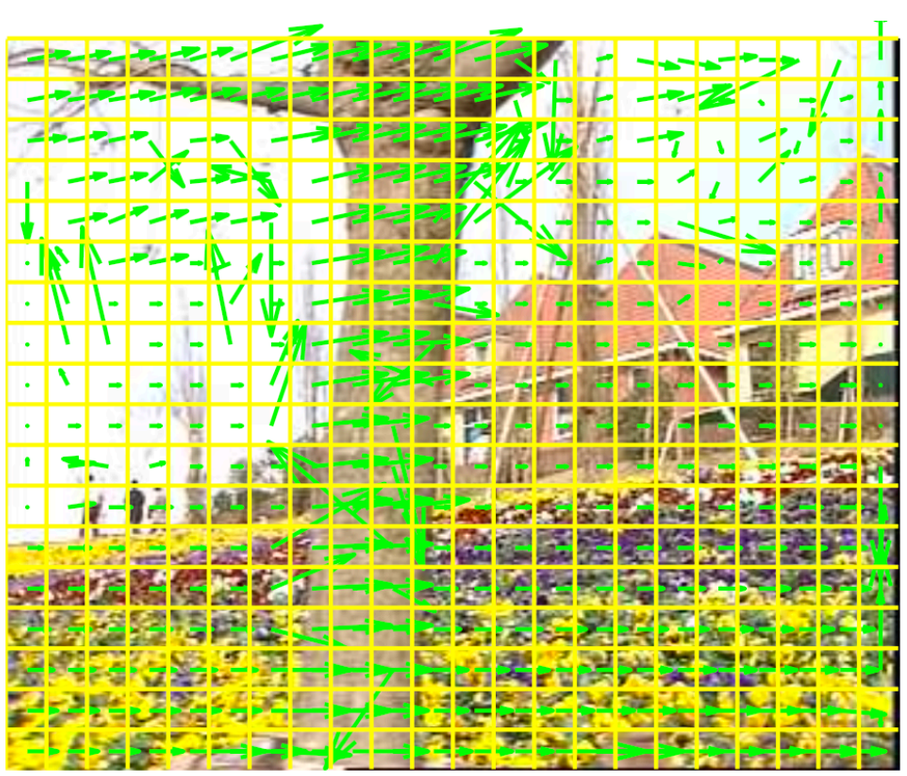
SSD SAD

Dissimilarity measure : regularization

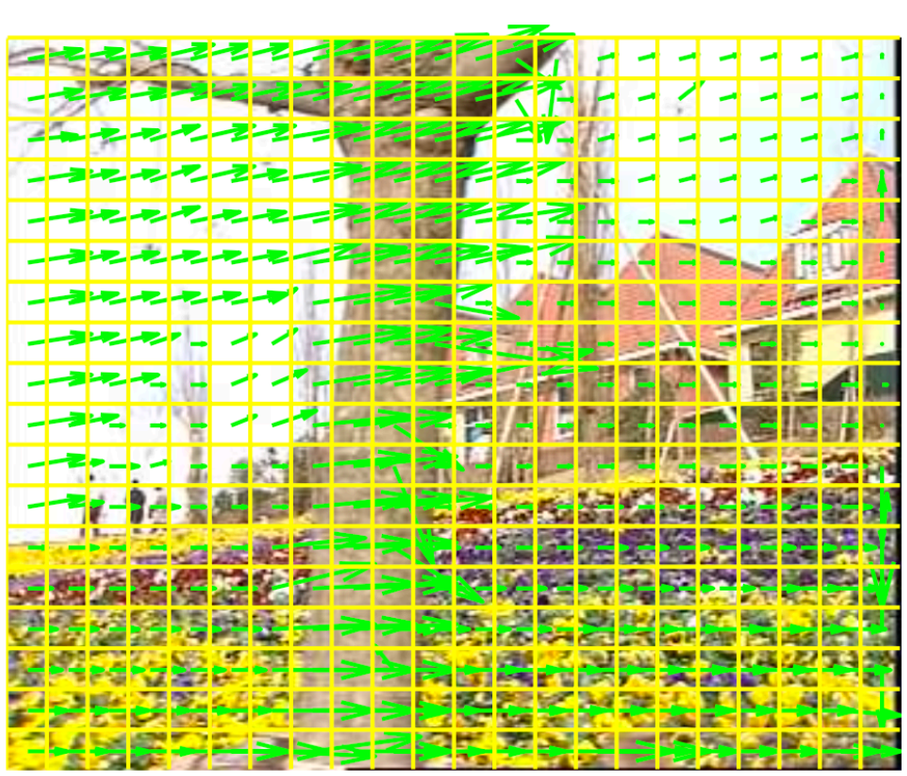
- An irregular MVF implies a higher coding cost and creates artifacts
- Solution : regularize the cost function :

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q}) - \mathbf{f}_k(B_{p-i,q-j})\|_p^p + \lambda R(i,j)$$

- R : regularizer introducing an explicit smoothness constraint (*e.g.* difference of (i,j) and a vector representing the neighborhood)
- $\lambda \geq 0$ controls the trade-off between minimization of the norm, *i.e.* the accuracy, and the regularizer, *i.e.* the smoothness



SSD



Regularized SSD

Dissimilarity measure : correlation

- Rewriting the SSD,

$$(\hat{i},\hat{j}) = \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q}) - \mathbf{f}_k(B_{p-i,q-j})\|_2^2$$
$$= \texttt{argmin}_{(i,j)\in\mathscr{W}} \|\mathbf{f}_t(B_{p,q})\|_2^2 - 2\langle \mathbf{f}_t(B_{p,q}) \,|\, \mathbf{f}_k(B_{p-i,q-j})\rangle + \|\mathbf{f}_k(B_{p-i,q-j})\|^2$$

- If the illumination does not depend on the patch, minimizing the SSD amounts to maximize the cross-correlation

$$(\hat{i},\hat{j}) = \texttt{argmax}_{(i,j)\in\mathscr{W}} \langle \mathbf{f}_t(B_{p,q}) \,|\, \mathbf{f}_k(B_{p-i,q-j})\rangle$$

- Advantage : fast to compute using FFT.
- Disadvantage : the hypothesis that $\|\mathbf{f}_k(B_{p-i,q-j})\|^2$ is constant is unrealistic

=> rather use normalized correlation

Normalized correlation

- Use normalized block $\tilde{\mathbf{f}}_k(B_{p,q})[\ell] = \mathbf{f}_k(B_{p,q})[\ell] - \frac{1}{PQ}\sum_{h=1}^{PQ} \mathbf{f}_k(B_{p,q})[h]$
- Cross-correlation

$$J_{\text{N-CORR}} = \frac{\sum_{\ell=1}^{PQ} \tilde{\mathbf{f}}_k(B_{p,q})[\ell]\tilde{\mathbf{f}}_h(B_{p-i,q-j})[\ell]}{\left(\sum_{\ell=1}^{PQ} \tilde{\mathbf{f}}_k^2(B_{p,q})[\ell] \cdot \sum_{\ell=1}^{PQ} \tilde{\mathbf{f}}_h^2(B_{p-i,q-j})[\ell]\right)^{1/2}}$$