Laboratoire Électronique Informatique et Image

UNIVERSITÉ DE BOURGOGNE

Optical Flow

Yannick Benezeth

yannick.benezeth@u-bourgogne.fr

Master in Computer Vision – 2016/2017

Schedule….

- 10h (5h today and 5h next week)
- Lectures + 2 practical works + 1 lab session
- Lab session is evaluated (code + small reports)
- Today: Lecture + 1 practical work + begining of lab session.

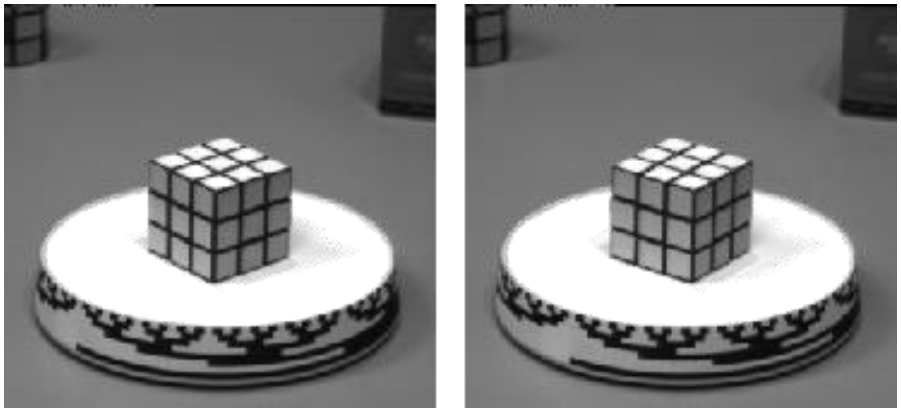
Overview

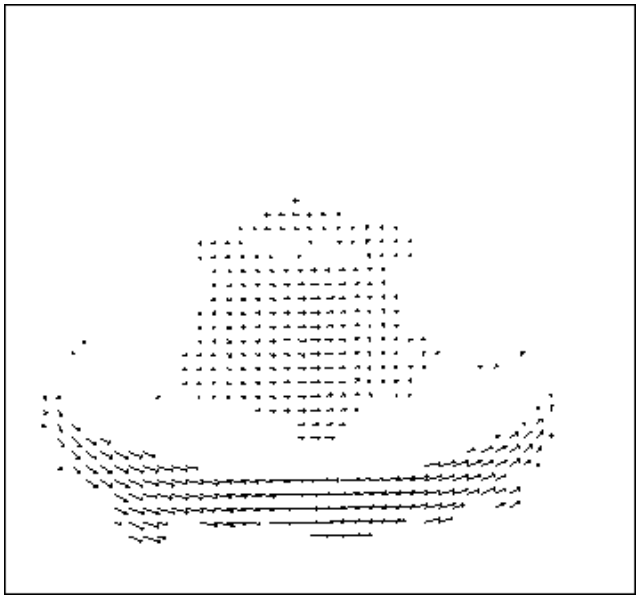
1. Introduction
2. Optical flow estimation
    1. Matching methods
    2. Global methods (Horn & Schunck)
    3. Local methods (Lucas-Kanade)
    4. Refinements
3. Experimental results and comparison

1. Introduction

What is Optical Flow?

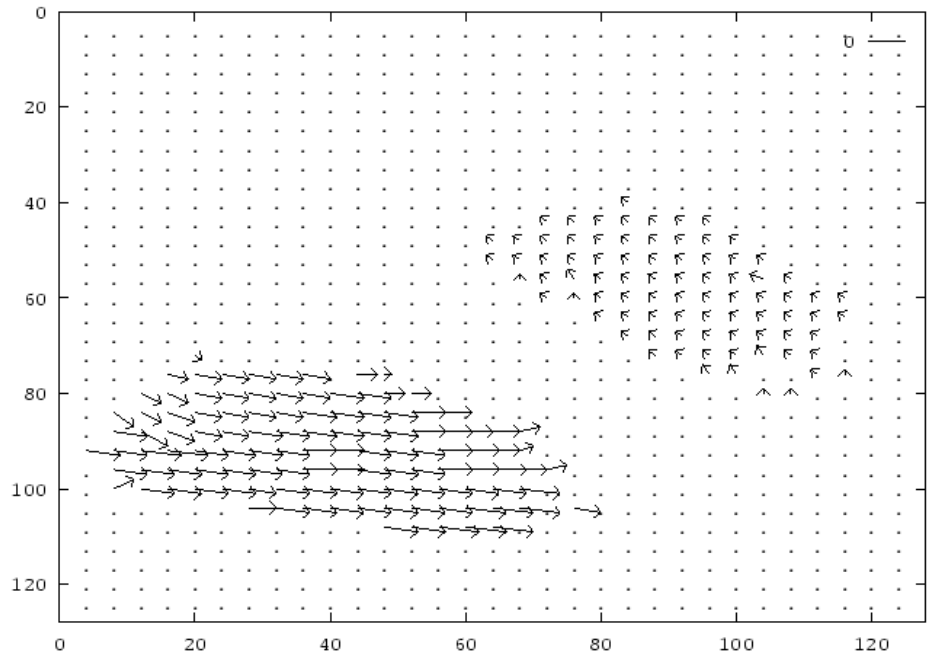
The apparent 2D image motion of pixels from one frame to the next in a video sequence





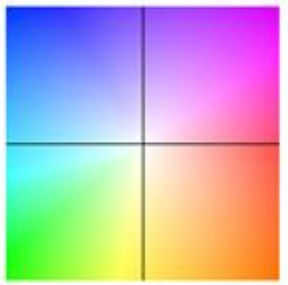
Optical Flow Examples

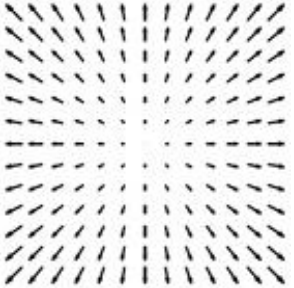
Hamburg Taxi sequence



Optical Flow Examples

Example with color representation of OF

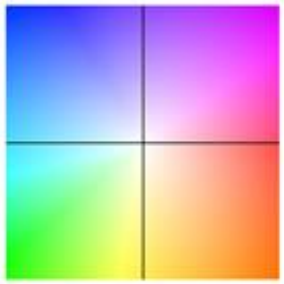


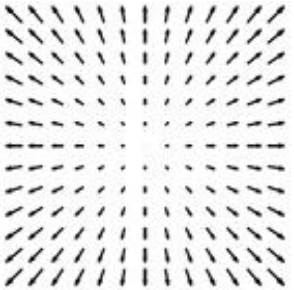


Color: Orientation
Saturation: Magnitude

Optical Flow Examples

Example with color representation of OF





Applications of optical flow

Motion detection: when the camera is static with some moving objects in the scene :

$\|\boldsymbol{v}\| = 0$ for all pixels in the static background

$\|\boldsymbol{v}\| \neq 0$ for all pixels in the moving objects.

Applications of optical flow

Motion detection: What happens when the camera is moving?

- Dominant motion estimation
- We suppose that this motion is due to the camera.
- We detect each pixel which does not undergo this motion!

Applications of optical flow

Video compression (motion compensation)

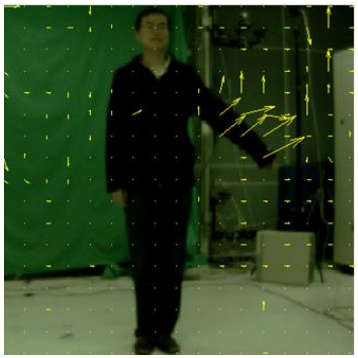
Full original frame

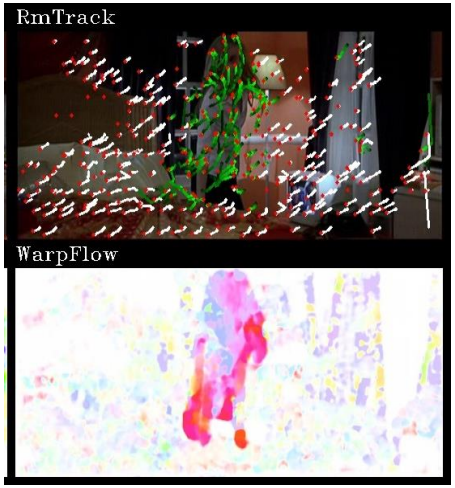
Differences between the original frame and the next frame

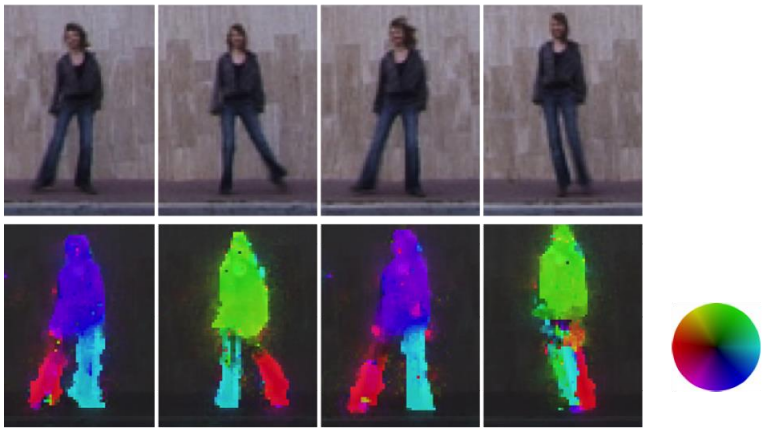
Motion compensated difference

Applications of optical flow

Recognizing events and activities (motion description)

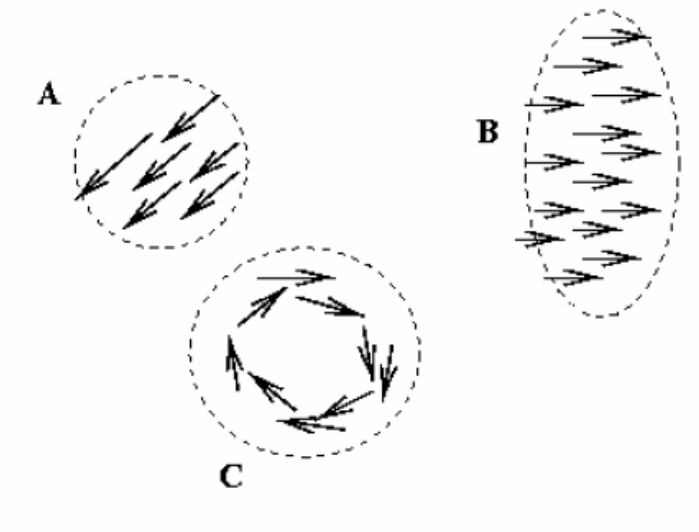




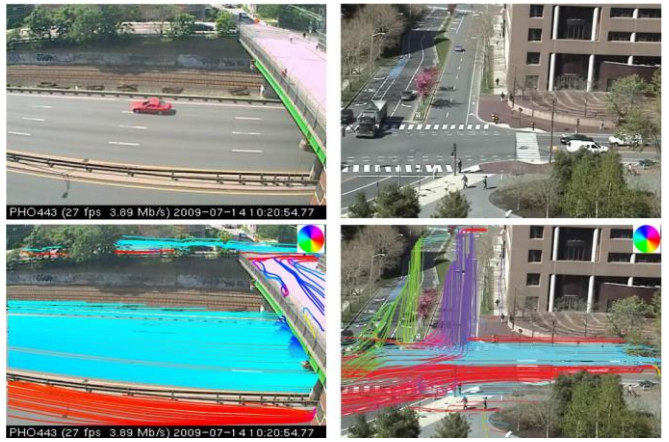


Applications of optical flow

Scene understanding (motion clustering)



Segment the video into multiple coherently moving objects



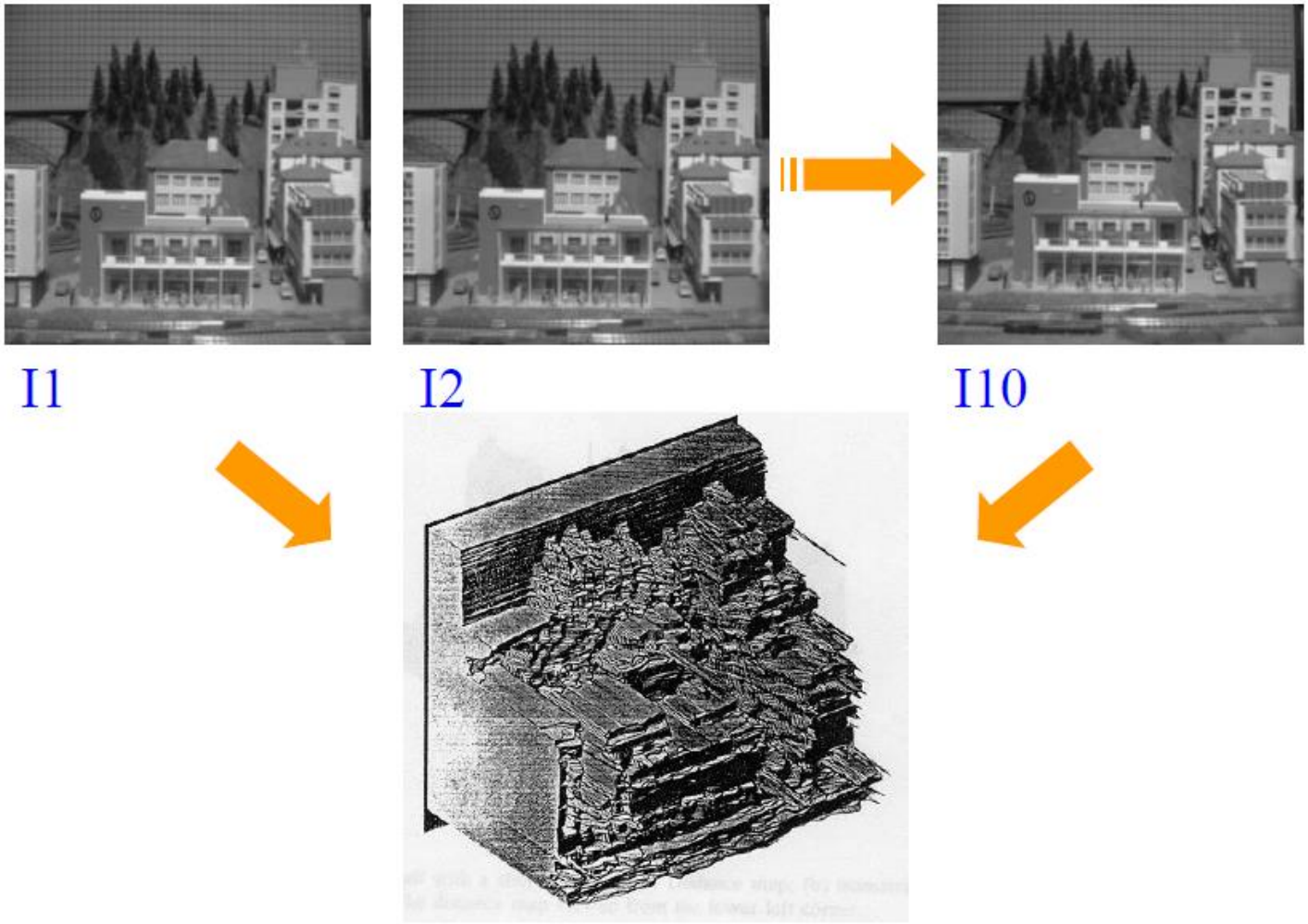
Estimate motion patterns in video surveillance sequences

Applications of optical flow

Improving video quality (video stabilization)

Applications of optical flow

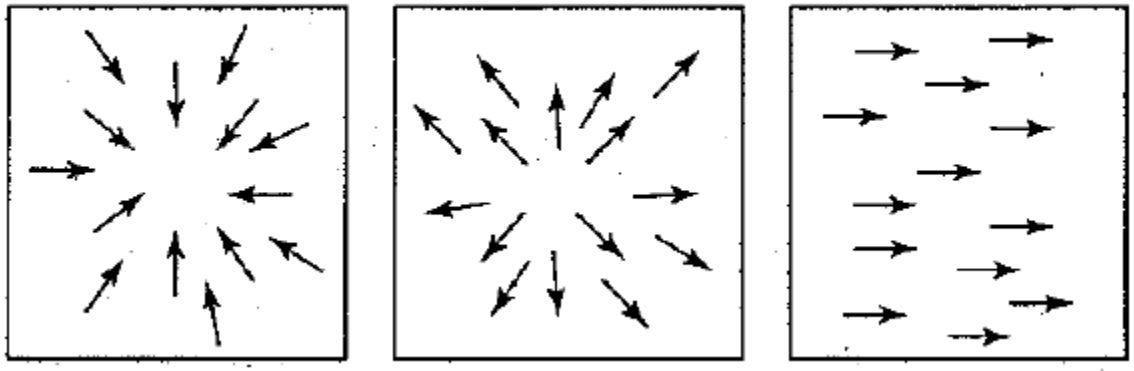
Structure from motion



Applications of optical flow

Robotics

Zoom out **Zoom in** **Pan right to left**



Optical Flow and 3D motion

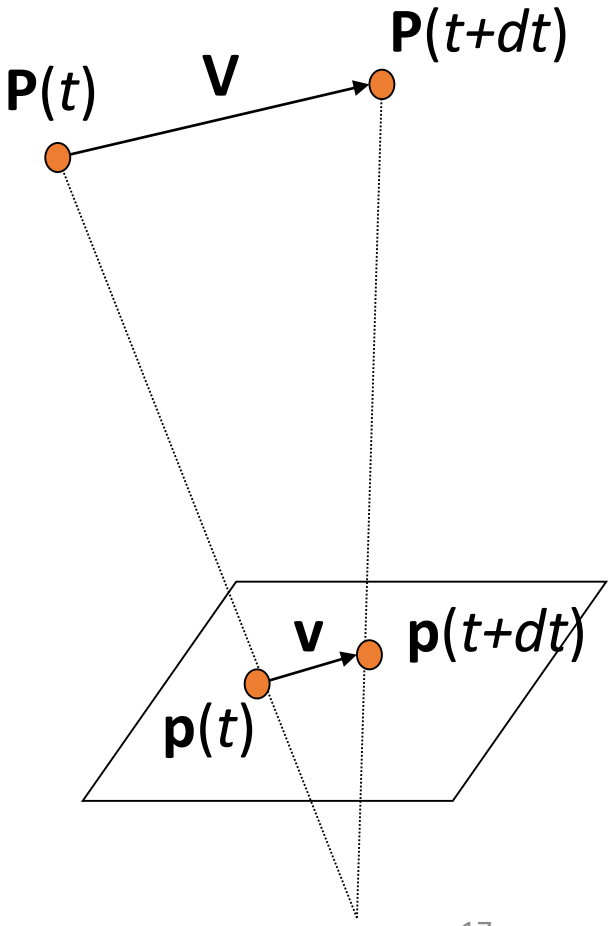
P(t) is a moving 3D point with a velocity **V**.

Let **p**(t) = (x(t),y(t)) the projection of **P** in the image.

The velocity of **p** in the image is given by **v**.

v is a 2D projection of **V** on image plane.

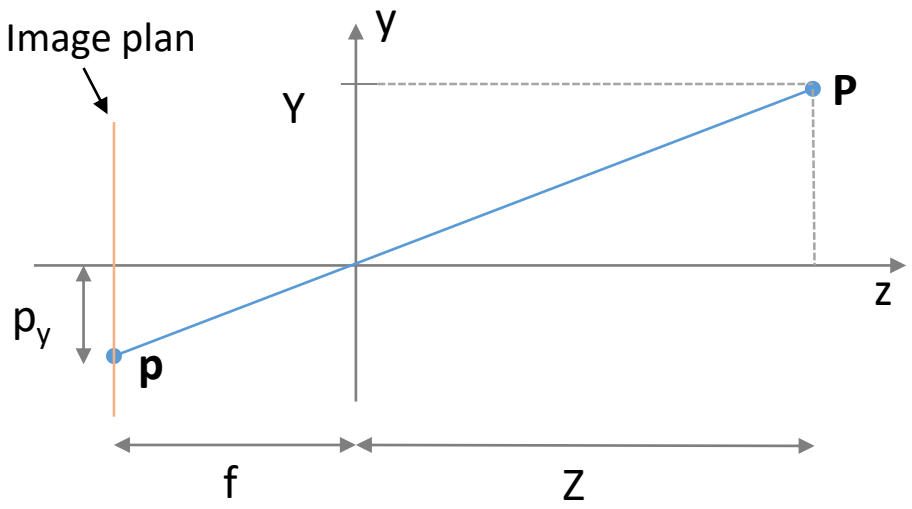
What is the relation between **v** and **V**?



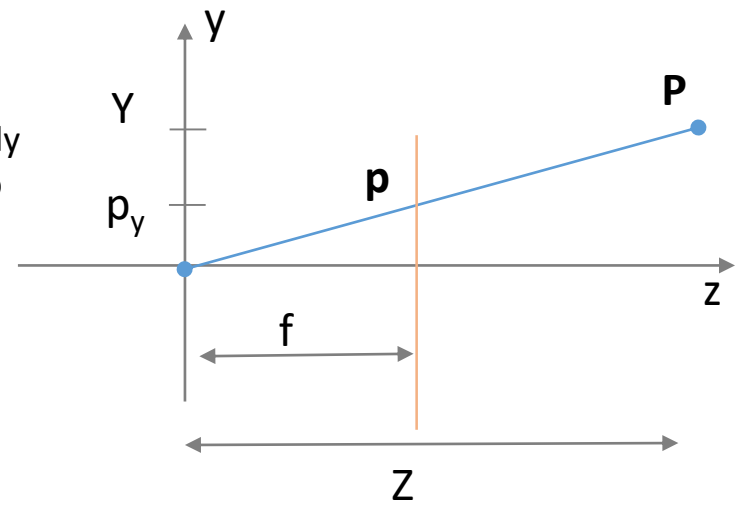
Optical Flow and 3D motion

Relation between **v** and **V**:

- *Pinhole Camera Model* : perspective projection



Mathematically equivalent to



To project **P** to **p**, we can use similar triangles to get $\frac{Z}{f} = \frac{Y}{p_y}$ or $p_y = f\frac{Y}{Z}$.

This is perspective projection -> equivalent to $\boldsymbol{p} = f\frac{\boldsymbol{P}}{Z}$

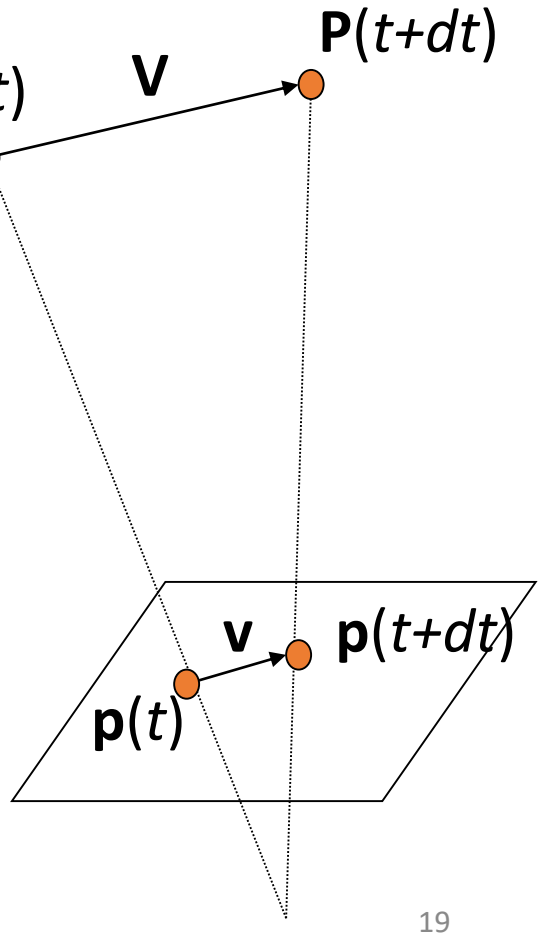
Optical Flow and 3D motion

- Let us pose $\boldsymbol{V} = (V_x, V_y, V_z)$ and $\boldsymbol{v} = \frac{d\boldsymbol{p}}{dt}$
- We know $\boldsymbol{p} = f\frac{\boldsymbol{P}}{Z}$, so:

$$\boldsymbol{v} = f\frac{Z\boldsymbol{V} - V_z\boldsymbol{P}}{Z^2}$$

$$v_x = \frac{fV_x - V_z x}{Z} \qquad v_y = \frac{fV_y - V_z y}{Z}$$

Image motion is a function of both the 3D motion (**V**) and the depth of the 3D point (*Z*)



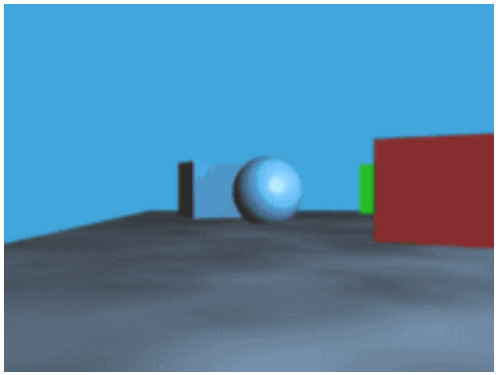
Optical Flow and Motion parallax

If V_Z=0, i.e. motion is parallel to the image plane:

$$\boldsymbol{v} = f\frac{\boldsymbol{ZV} - V_Z\boldsymbol{P}}{Z^2} \quad \Rightarrow \quad \boldsymbol{v} = f\frac{\boldsymbol{V}}{Z}$$

All the motion vectors are parallel.

The length of the motion vectors is inversely proportional to the depth *Z.*



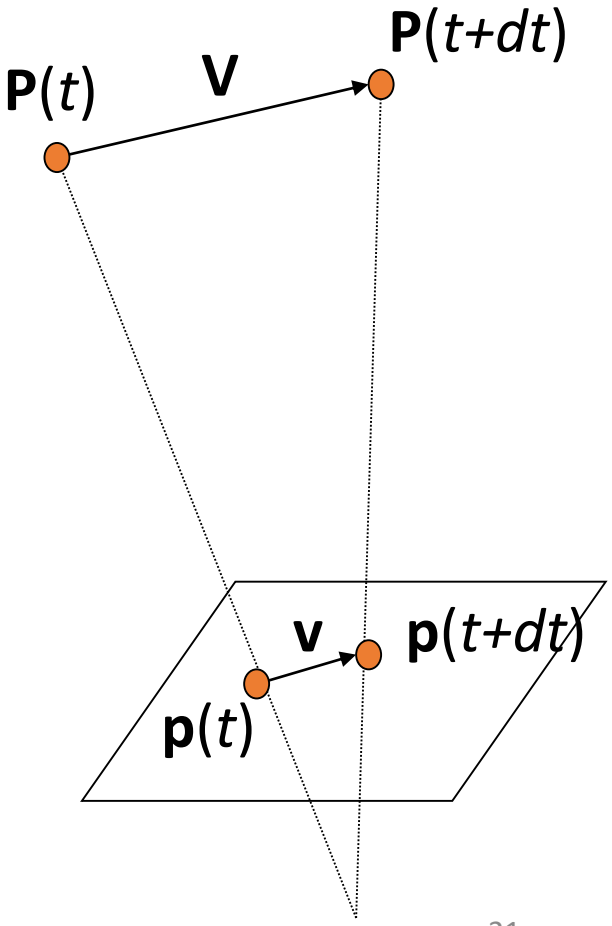
Parallax(*):
The objects in the distance appear to move slower than the objects close to the camera

(*) Wikipedia

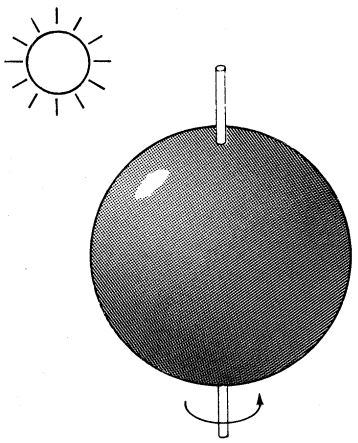
Optical Flow: Definition

Motion Field = Projection of the 3D motion of points in the scene onto the image plane.

Optical Flow = 2D velocity field describing the apparent motion in the image.



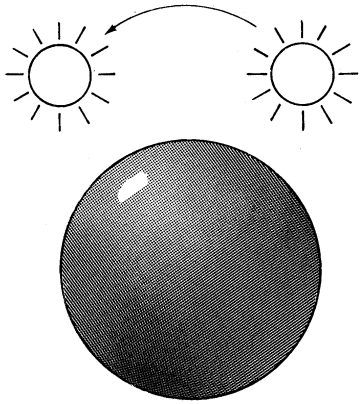
Ambiguity in Optical Flow



- A Lambertian (matte) ball is rotating under constant illumination.
- What does the 2D motion field look like ?
- What does the 2D optical flow field look like ?

- Motion field follows surface points.
- Optical flow is zero since motion is not visible.

Ambiguity in Optical Flow

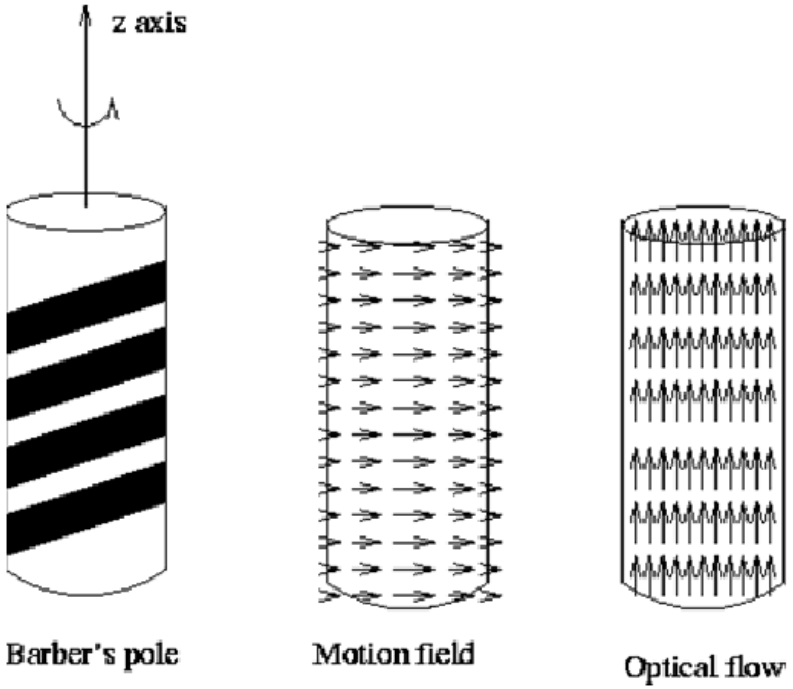


- A fixed Lambertian ball is illuminated by a moving source — the shading of the image changes
- What does the 2D motion field look like ?
- What does the 2D optical flow field look like ?

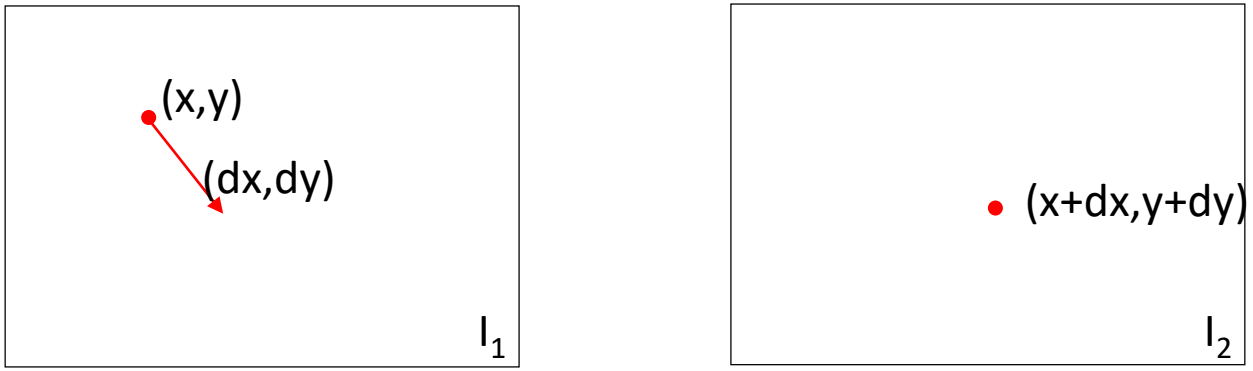
- Thus the motion field is zero, but the optical flow field is not.

Ambiguity in Optical Flow

Barber pole illusion



Brightness Constancy Assumption

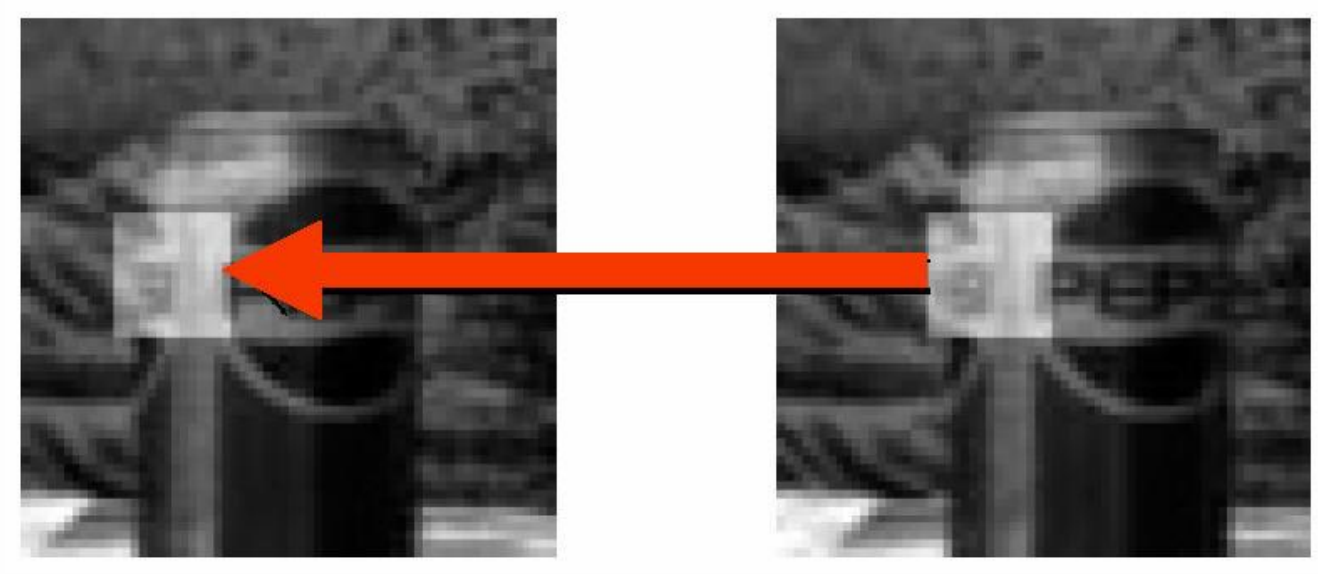


$$I_1(x, y) = I_2(x + dx, y + dy)$$

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Brightness Constancy Assumption

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$



Optical flow constraint equation

- Taylor series of a real function of two variables is given by:

$$f(x+dx, y+dy) = f(x,y) + \left[f_x(x,y)dx + f_y(x,y)dy\right] + Higher\ Order\ Terms$$

with $f_x(x,y) = \frac{\partial f}{\partial x}$ and $f_y(x,y) = \frac{\partial f}{\partial y}$

- Taylor Series of Brightness Constancy Assumption (BCA) equation

BCA equation : $I(x,y,t) = I(x+dx, y+dy, t+dt)$

$$I(x,y,t) = I(x,y,t) + I_x dx + I_y dy + I_t dt$$

Assumption: motion is small

$$0 = I_x dx + I_y dy + I_t dt$$

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

Optical flow constraint equation

Optical flow constraint equation

Everything starts with this equation : $I_x u + I_y v + I_t = 0$

Also written

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v = -I_t$$

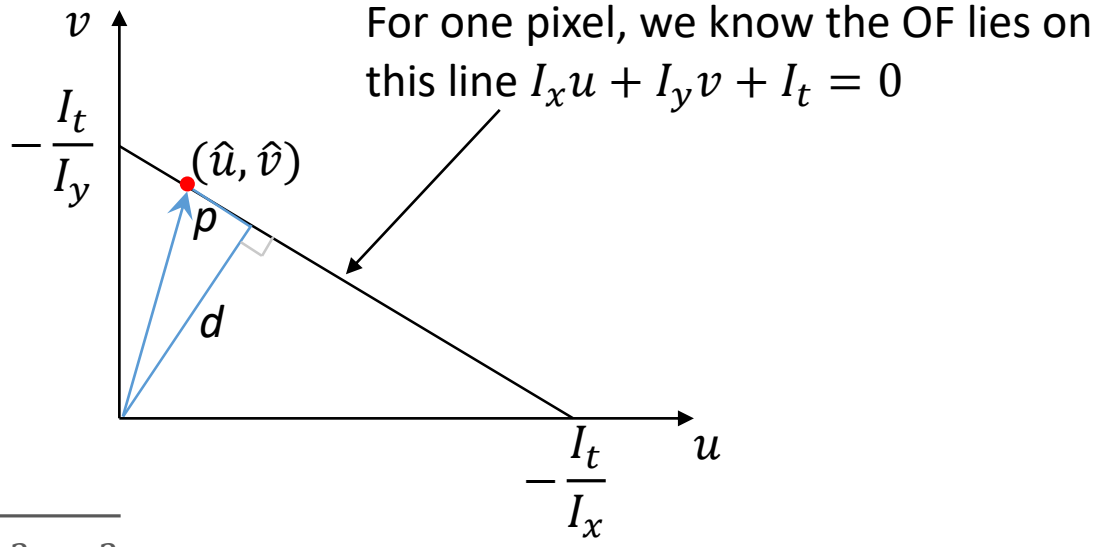
$$\nabla I . \begin{pmatrix} u \\ v \end{pmatrix} = -I_t$$

Interpretation of Optical Flow Constraint Equation

$I_x u + I_y v + I_t = 0$ is the equation of a straight line (in *u* and *v* space)

How many unknowns and equations per pixel ?

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



Let $(\hat{u}, \hat{v})$ be the true OF, it has 2 components:

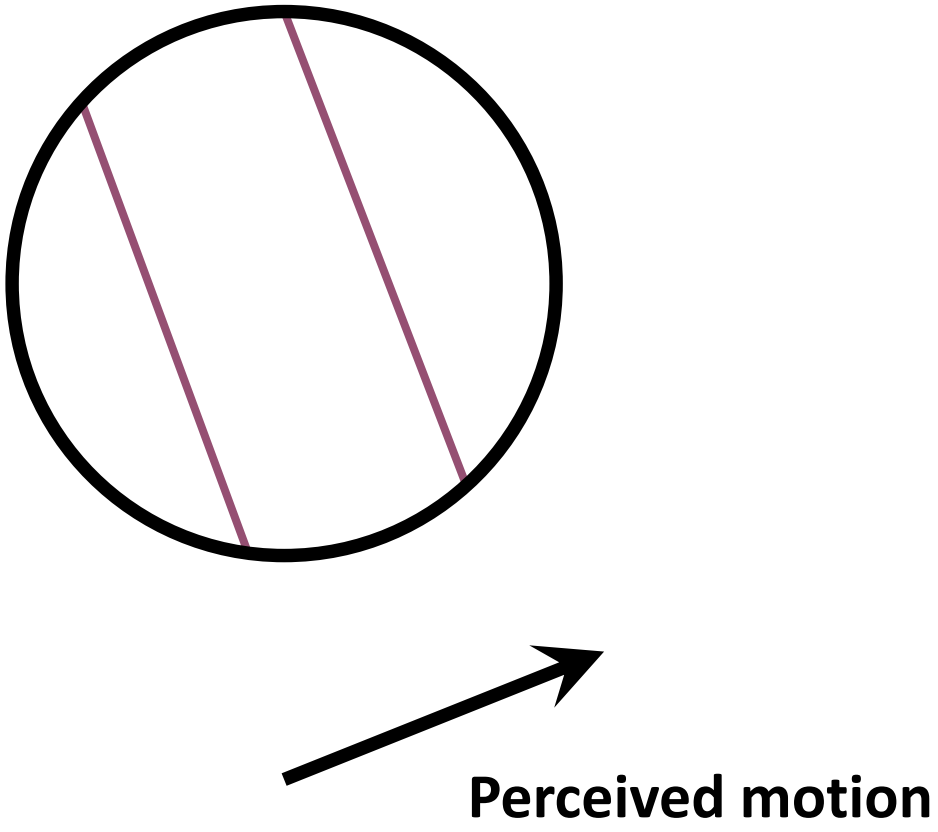
- p - parrallel flow
- d - normal flow

d can be computed $d = I_t / \sqrt{I_x^2 + I_y^2}$, *p* cannot.

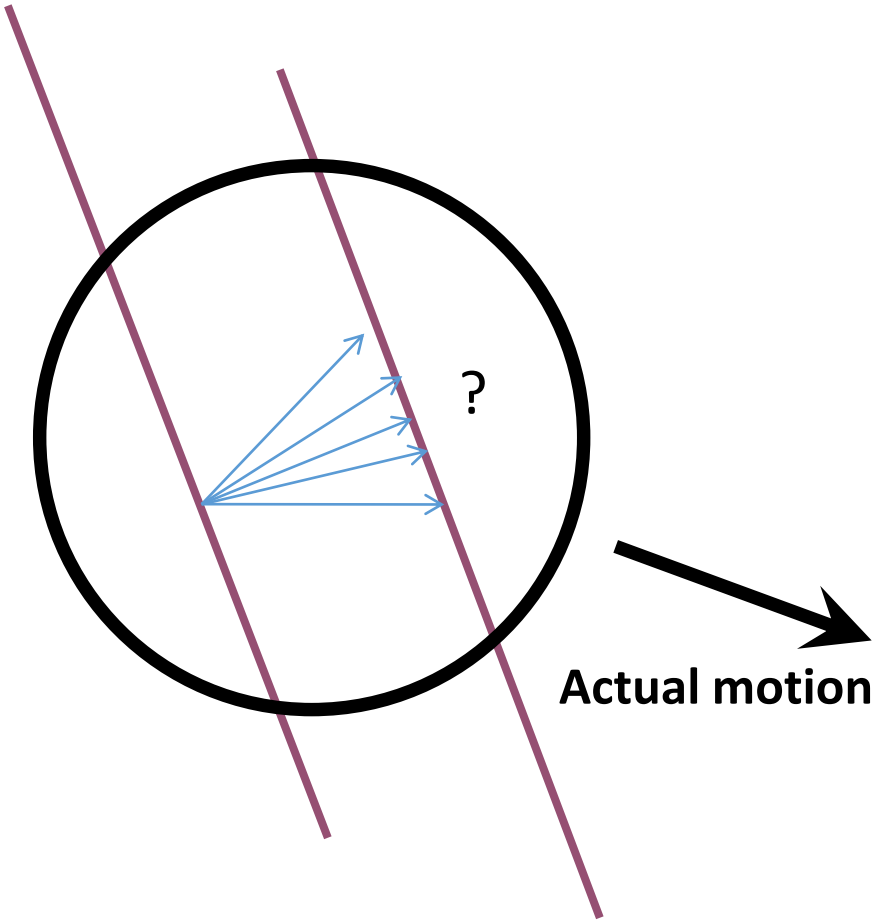
➡ Under-constraint nature of Optical Flow problem.

➡ With one point we can only detect movement perpendicular to the brightness gradient. Known as aperture problem.

Aperture problem

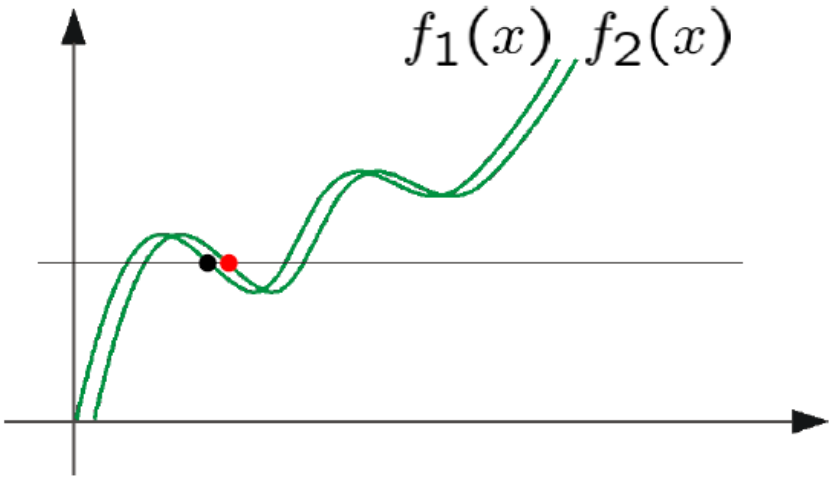


Aperture problem

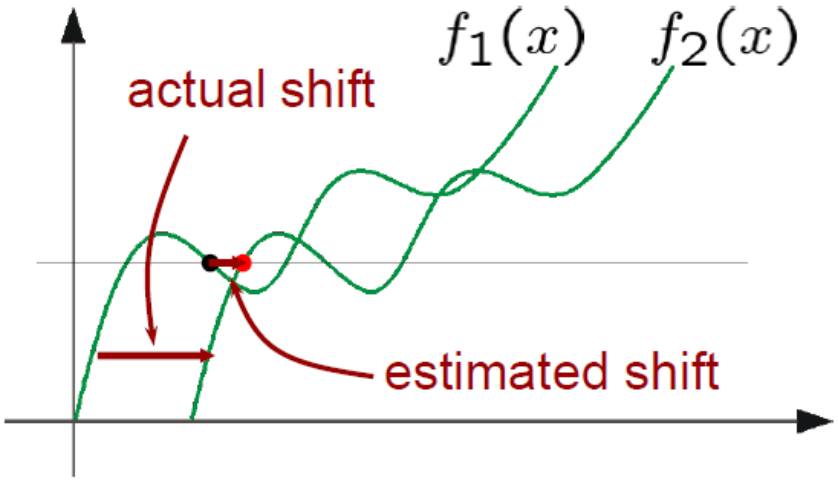


Temporal aliasing problem

- Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity. How do we know which correspondence is correct?



Nearest match is correct
(no aliasing)



Nearest match is incorrect
(aliasing)

In a nutshell

Key assumptions of Optical Flow

- Brightness constancy: a point in I_1 looks the same in I_2

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$ (brightness constancy equation)

- Motion is small: points do not move very far (<1 pixel)

  With this assumption we can use the Taylor Series of the brightness constancy equation to obtain

$$\nabla I . \begin{pmatrix} u \\ v \end{pmatrix} = -I_t$$ (Optical flow constraint equation)

1 equation, 2 unknowns -> we need additional constraints to solve the Optical Flow problem.

2. Optical Flow estimation methods

2.1. Matching Methods

Matching methods

Instead of solving the Optical Flow Constraint Equation (1 eq 2 unknows), the cross-correlation methods propose to solve directly the brightness constancy hypothesis :

$$I(x+u, y+v, t+1) = I(x, y, t)$$

Model the image as a series of patches (blocks) and determine the (u,v) that minimizes the L1 or L2 norm (image difference) between blocks.

$$\min_{u,v} \sum_{i=-N}^{N} \sum_{j=-N}^{N} \left(I(x+i+u, y+j+v, t+1) - I(x+i, y+j, t) \right)^2$$

Sum of Squared Difference (SSD)

Matching methods

- Advantage : works even with large displacements (overcome the temporal aliasing problem).
- Drawbacks :
  - difficult to solve
  - impossible to have a dense estimation in real time
  - sparse estimation !
    - Interest points detection (HARRIS, SIFT, SURF) in 2 consecutive images. The estimation of the optical flow is reduced to match « similar » interest points.

Matching methods

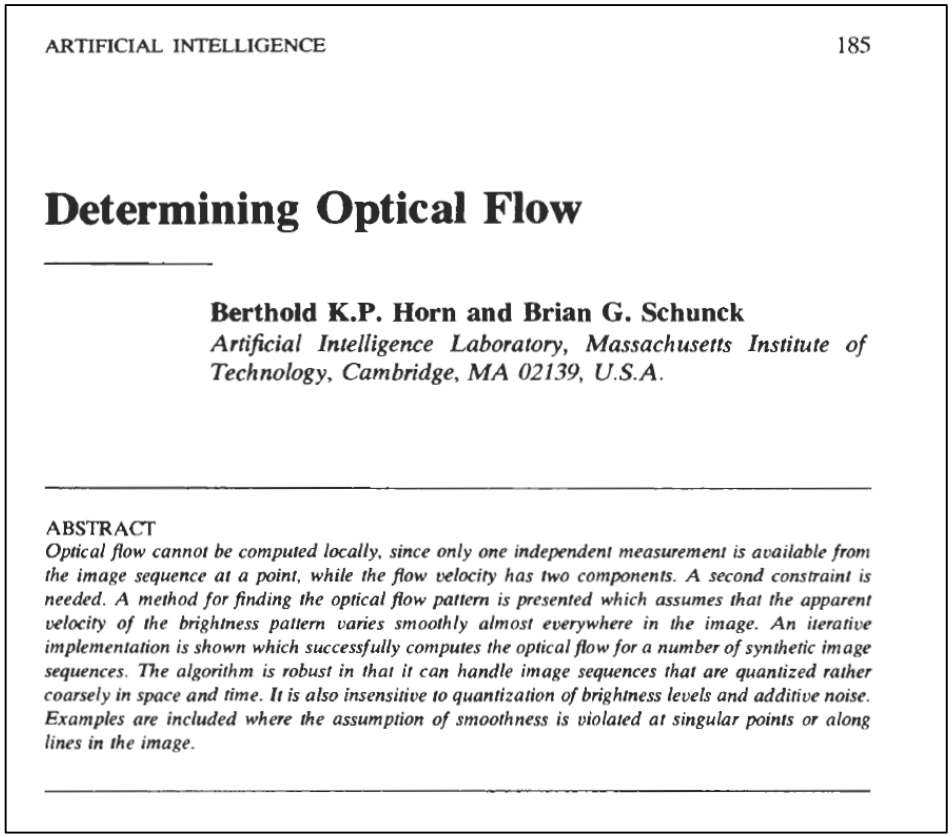
http://cmp.felk.cvut.cz/cmp/courses/Y33ROV/Y33ROV_ZS20082009/Lectures/Motion/klt.pdf

2.2. Horn and Shunck (Global method)

Horn and Shunck

Horn B.K.P and Shunck B.G. Determing optical flow. Artificial Intelligence. Vol 17 pp 185-203

Main idea : the optical flow is smooth !



Determining Optical Flow

Berthold K.P. Horn and Brian G. Schunck
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

ABSTRACT
Optical flow cannot be computed locally, since only one independent measurement is available from the image sequence at a point, while the flow velocity has two components. A second constraint is needed. A method for finding the optical flow pattern is presented which assumes that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. An iterative implementation is shown which successfully computes the optical flow for a number of synthetic image sequences. The algorithm is robust in that it can handle image sequences that are quantized rather coarsely in space and time. It is also insensitive to quantization of brightness levels and additive noise. Examples are included where the assumption of smoothness is violated at singular points or along lines in the image.

Horn and Shunck

OFCE is an under constraint system that cannot be solved for each pixel. Thus, they propose to minimize :

$$E(u,v) = \iint \left\{ \left(I_x u + I_y v + I_t\right)^2 + \lambda^2 \left(u_x^2 + u_y^2 + v_x^2 + v_x^2\right) \right\} dxdy$$

Data term (OFCE)

Smoothness term

Also written:

$$E(u,v) = \iint \left\{ \left(I_x u + I_y v + I_t\right)^2 + \lambda^2 \left(\|\nabla u\|^2 + \|\nabla v\|^2\right) \right\} dxdy$$

Horn and Shunck

$$E(u,v) = \iint \left\{ \left(I_x u + I_y v + I_t\right)^2 + \lambda^2 \left(u_x^2 + u_y^2 + v_x^2 + v_x^2\right) \right\} dxdy$$

To minimize $\mathrm{E}(\mathrm{u},\mathrm{v})$ we can *« differentiate »* with respect to *u* and *v* and *=0*

$$\left(I_x u + I_y v + I_t\right) I_x + \lambda^2 (\nabla^2 u) = 0$$
$$\left(I_x u + I_y v + I_t\right) I_y + \lambda^2 (\nabla^2 v) = 0$$

With $\Delta u = \nabla^2 u = u_{xx} + u_{yy}$, i.e. the Laplacian

More details in TD

Horn and Shunck

Derivative masks (Roberts)

$\begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}$ on I_1 $\quad$ $\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$ on I_1 $\quad$ $\begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}$ on I_1

$\begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}$ on I_2 $\quad$ $\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$ on I_2 $\quad$ $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ on I_2


I_x


I_y


I_t

Laplacian

Works only for small motion

$$\begin{pmatrix} 1/12 & 1/6 & 1/12 \\ 1/6 & -1 & 1/6 \\ 1/12 & 1/6 & 1/12 \end{pmatrix}$$

$$\nabla^2 f = f_{xx} + f_{yy} = \bar{f} - f$$

Horn and Shunck

$$E(u,v) = \iint \left\{ \left(I_x u + I_y v + I_t\right)^2 + \lambda^2 \left(u_x^2 + u_y^2 + v_x^2 + v_x^2\right) \right\} dxdy$$

To minimize $\mathrm{E}(\mathrm{u},\mathrm{v})$ we can *« differentiate »* with respect to *u* and *v* and *=0*

$$\left(I_x u + I_y v + I_t\right) I_x - \lambda^2 (\nabla^2 u) = 0$$
$$\left(I_x u + I_y v + I_t\right) I_y - \lambda^2 (\nabla^2 v) = 0$$

With $\nabla^2 u = u_{xx} + u_{yy}$, i.e. the Laplacian

Using the approximate of the Laplacian, we have:

$$\left(I_x u + I_y v + I_t\right) I_x - \lambda^2 (\bar{u} - u) = 0$$
$$\left(I_x u + I_y v + I_t\right) I_y - \lambda^2 (\bar{v} - v) = 0$$

Can be reformulated by

$$u = \bar{u} - I_x \frac{I_x \bar{u} + I_y \bar{v} + I_t}{\lambda^2 + {I_x}^2 + {I_y}^2} \text{ and } v = \bar{v} - I_y \frac{I_x \bar{u} + I_y \bar{v} + I_t}{\lambda^2 + {I_x}^2 + {I_y}^2}$$

Horn and Shunck

In practice, this optimization is obtained thanks to an iterative scheme:

$$u^{k+1} = \bar{u}^{k} - \frac{I_x\left(I_x\bar{u}^{k} + I_y\bar{v}^{k} + I_t\right)}{\lambda^2 + I_x^2 + I_y^2}$$

$$v^{k+1} = \bar{v}^{k} - \frac{I_y\left(I_x\bar{u}^{k} + I_y\bar{v}^{k} + I_t\right)}{\lambda^2 + I_x^2 + I_y^2}$$

Where k denotes the iteration number

Horn and Shunck

Algorithm

Begin

calculate I_x, I_y *and* I_t using a selected approx formula (masks)

initialize the values *u* and *v* to zero

choose a suitable weighting value λ

k := 1

Repeat until some error measure is satisfied (converges or fix number of iteration)

For each pixel (i,j)

compute $\bar{u}(i,j)$ *and* $\bar{v}(i,j)$

update $u(i,j)$ and $v(i,j)$

End for

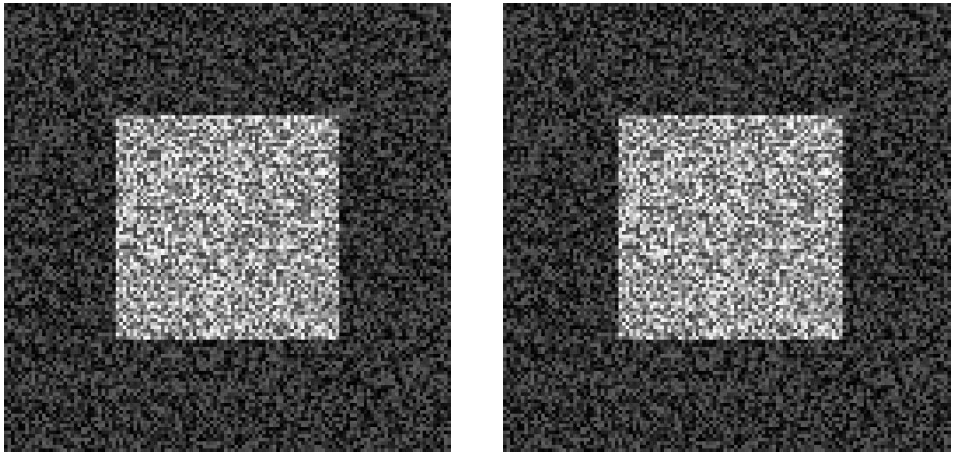
k := *k* + 1

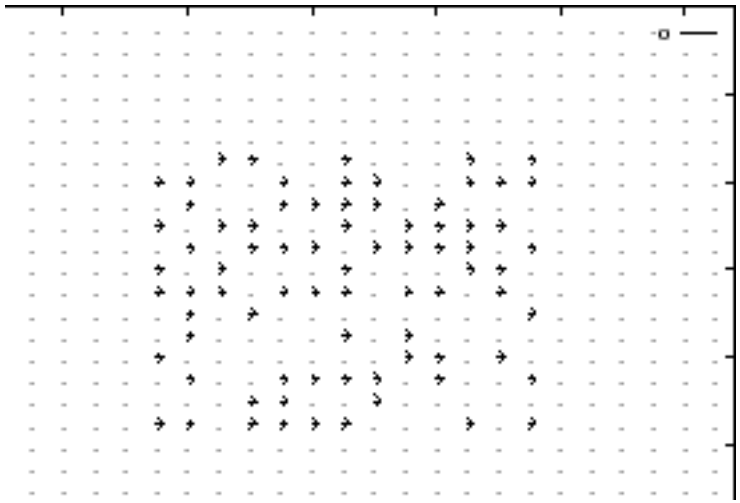
End repeat

End

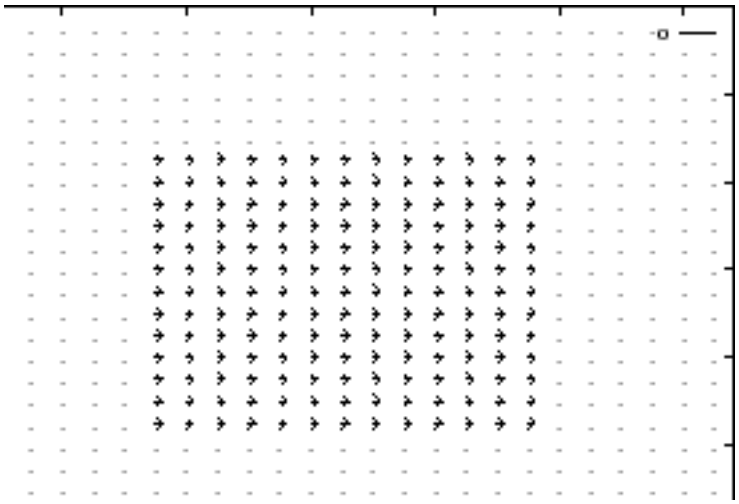
Horn and Shunck

Result on synthetic images





After one iteration

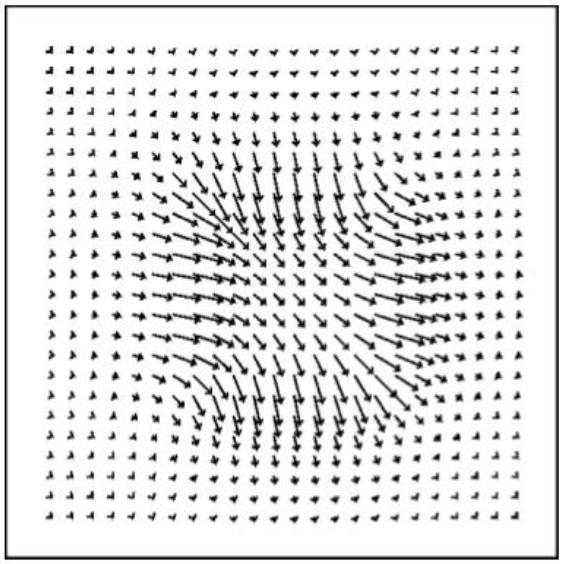


After 10 iterations

* UCF-CRCV Lecture

Horn and Shunck

Result on synthetic images



- This example is very caricatural since we do not have any texture information on the background or on the moving object.
- The motion field is too smooth, there is no discontinuities along the edge of the square!!

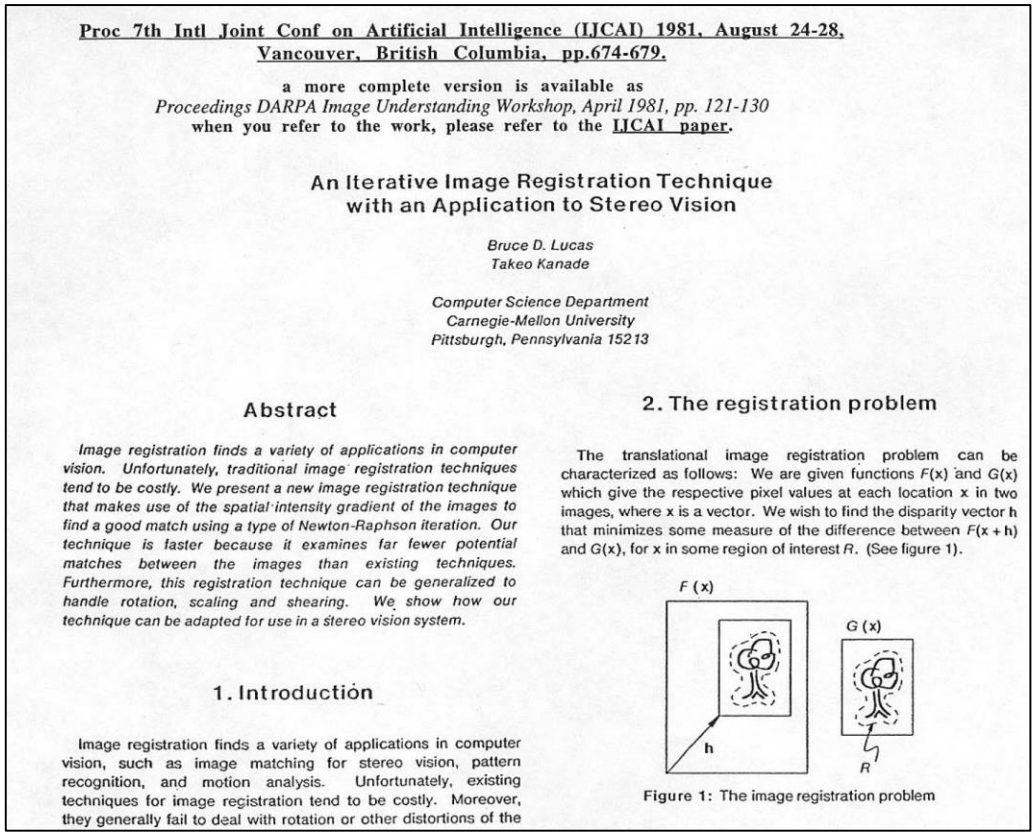
* P. Kornprobst Thesis

2.3. Lucas-Kanade (local method)

Lucas-Kanade

Lucas, B. and Kanade, T., An iterative image registration technique with an application to stereo vision. In *Proc. of the Int. Joint Conf. on Artificial Intelligence, 1981*

Main idea : the optical flow is constant on the neighborhood of the current point (x,y). Each neighbor gives one equation!!



Proc 7th Intl Joint Conf on Artificial Intelligence (IJCAI) 1981, August 24-28, Vancouver, British Columbia, pp.674-679.

a more complete version is available as
Proceedings DARPA Image Understanding Workshop, April 1981, pp. 121-130
when you refer to the work, please refer to the IJCAI paper.

An Iterative Image Registration Technique with an Application to Stereo Vision

Bruce D. Lucas
Takeo Kanade

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Image registration finds a variety of applications in computer vision. Unfortunately, traditional image registration techniques tend to be costly. We present a new image registration technique that makes use of the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. Our technique is faster because it examines far fewer potential matches between the images than existing techniques. Furthermore, this registration technique can be generalized to handle rotation, scaling and shearing. We show how our technique can be adapted for use in a stereo vision system.

1. Introduction

Image registration finds a variety of applications in computer vision, such as image matching for stereo vision, pattern recognition, and motion analysis. Unfortunately, existing techniques for image registration tend to be costly. Moreover, they generally fail to deal with rotation or other distortions of the

2. The registration problem

The translational image registration problem can be characterized as follows: We are given functions $F(x)$ and $G(x)$ which give the respective pixel values at each location x in two images, where x is a vector. We wish to find the disparity vector h that minimizes some measure of the difference between $F(x+h)$ and $G(x)$, for x in some region of interest R. (See figure 1).

Figure 1: The image registration problem

Lucas-Kanade

Optical Flow Constraint Equation: $I_x u + I_y v + I_t = 0$

Hypothesis : the pixel's neighbors have the same velocity *(u,v)*.

For example, if we consider a 5x5 window, we have 25 equations and 2 unknowns:

$$I_{x_1} u + I_{y_1} v = -I_{t_1}$$
$$\vdots$$
$$I_{x_{25}} u + I_{y_{25}} v = -I_{t_{25}}$$

$$\begin{pmatrix} I_{x_1} & I_{y_1} \\ \vdots & \vdots \\ I_{x_{25}} & I_{y_{25}} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -I_{t_1} \\ \vdots \\ -I_{t_{25}} \end{pmatrix}$$

$$\boldsymbol{Au = B}$$

Lucas - Kanade

Thus, we obtain an overdetermined system :

$$A\begin{pmatrix} u \\ v \end{pmatrix} = B$$

A is *[25x2]* matrix, cannot be inverted, we multiply each side by A^T

$$A^T A \begin{pmatrix} u \\ v \end{pmatrix} = A^T B$$

A^TA is *[2x2]* matrix, can be inverted.

Can be solved also with:

$$\min \sum_i \left(I_{x_i} u + I_{y_i} v + I_{t_i}\right)^2$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \left(A^T A\right)^{-1} A^T B$$

Pseudo inverse

Lucas - Kanade

We have $A^T A \begin{pmatrix} u \\ v \end{pmatrix} = A^T B$

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

The matrix $A^T A$ is often called the structure tensor of the image at a particular point (or sometimes *second moment matrix*)

Lucas - Kanade

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

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{pmatrix}^{-1} \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\sum I_x I_x \sum I_y I_y - \left(\sum I_x I_y\right)^2} \begin{pmatrix} \sum I_y I_y & -\sum I_x I_y \\ -\sum I_x I_y & \sum I_x I_x \end{pmatrix} \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix}$$

Lucas - Kanade

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \sum \nabla I . (\nabla I)^T$$

- The structure tensor is a matrix derived from the gradient of a function.
- Its eigenvectors and eigenvalues (λ_1 and λ_2) summarize the distribution of the gradient $\nabla I = (I_x, I_y)$, *(i.e.* edge direction and magnitude) in a specified neighborhood of a point.
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
  - The other eigenvector is orthogonal to it

Lucas - Kanade

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \quad (1)$$

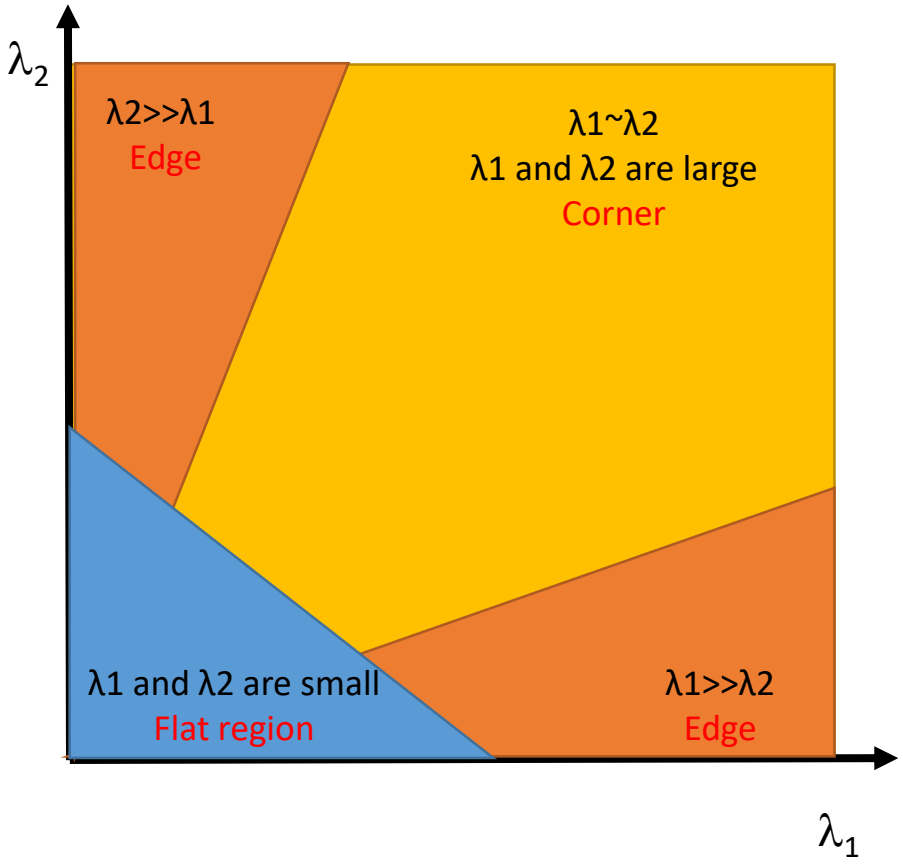
Linear algebra explains that (1) is solvable if

- A^TA should be invertible
- A^TA is a well-conditioned matrix
  - eigenvalues λ_1 and λ_2 of A^TA should not be too small
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

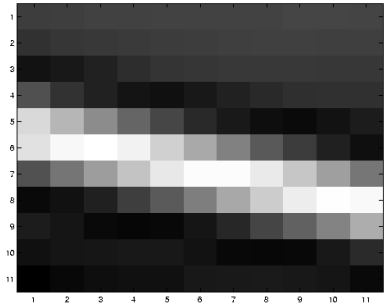
Lucas - Kanade

- Classification of image pixels using eigenvalues of the structure tensor matrix



Lucas - Kanade

Unstable on edges….



- large gradients in one direction
- large λ_1, small λ_2

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Lucas - Kanade

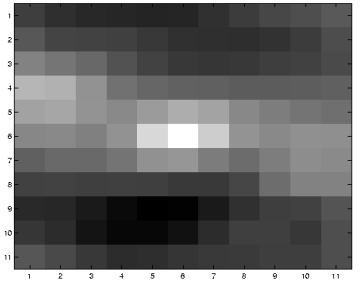
Unstable on low texture patches….

- gradients have small magnitude
- small λ_1, small λ_2

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Lucas - Kanade

Stable on corners….



- gradients are different, large magnitudes
- large λ_1, large λ_2

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Lucas - Kanade

What are the potential causes of errors of LK ?

- A^TA is not easily invertible
- Noise in the image
- When our assumptions are violated
  - Brightness constancy is not satisfied
  - The motion is not small
  - A point does not move like its neighbors (window size is too large…)

Multi-resolution Lucas – Kanade

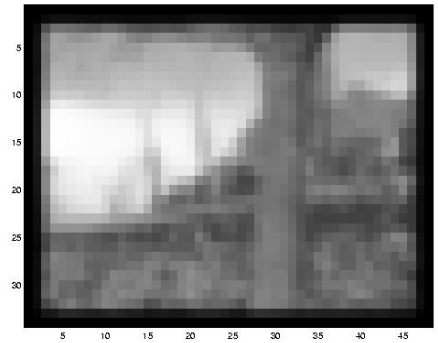
- How to deal with large motion?

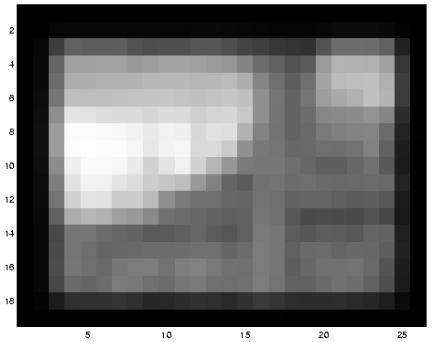
Coarse to fine optical flow estimation!!!

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Multi-resolution Lucas – Kanade

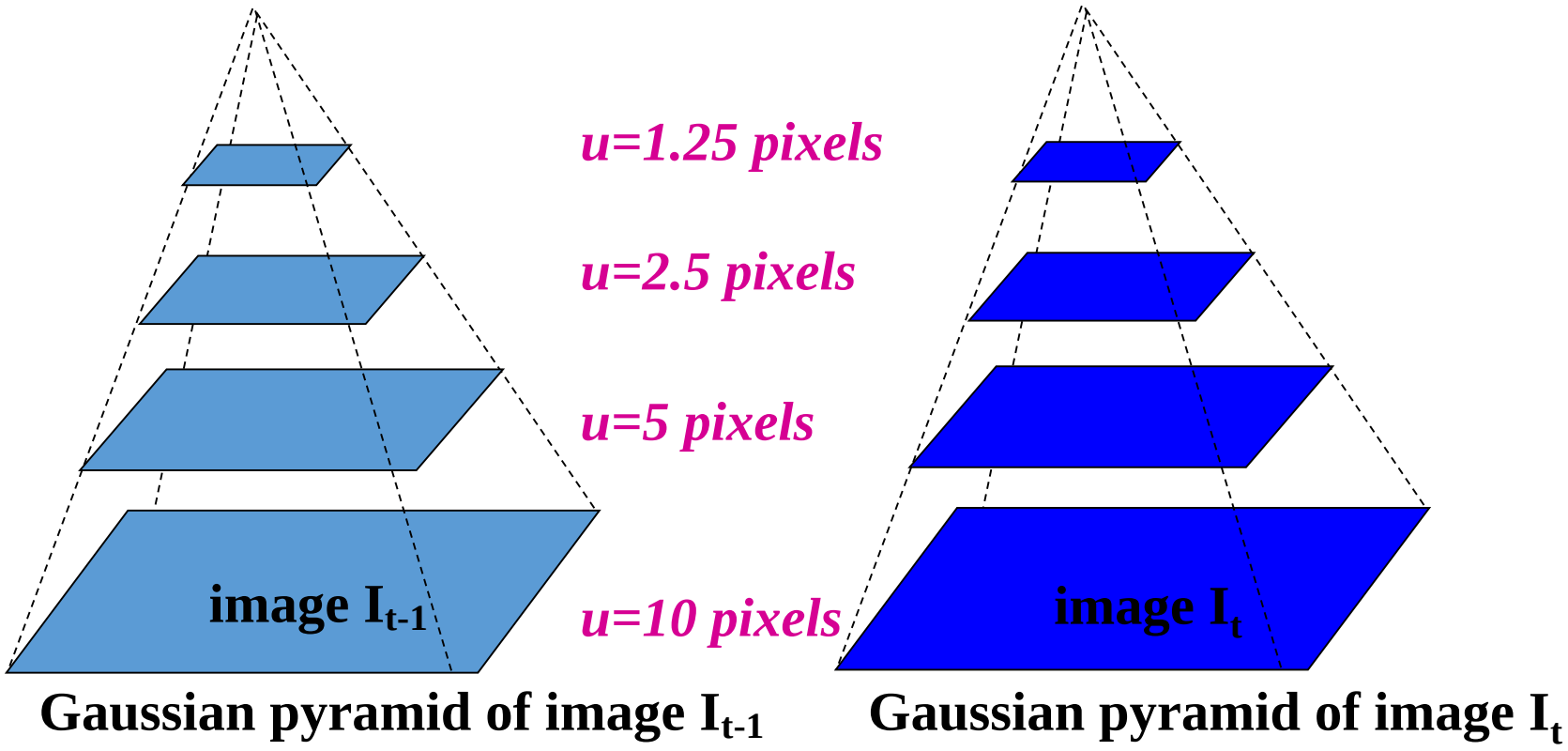
Reduce the resolution





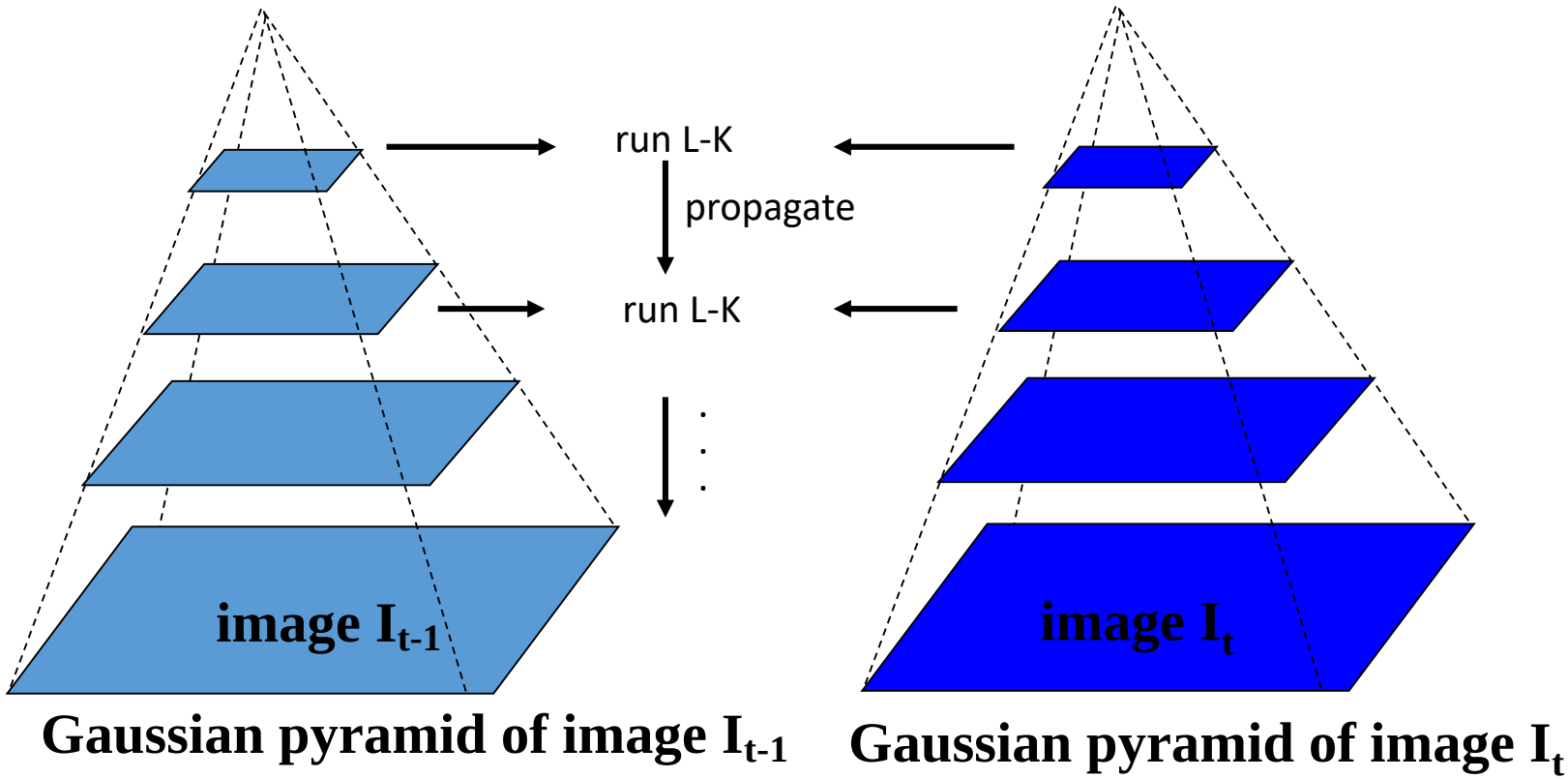
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Multi-resolution Lucas – Kanade



Multi-resolution Lucas – Kanade

Coarse to fine optical flow estimation



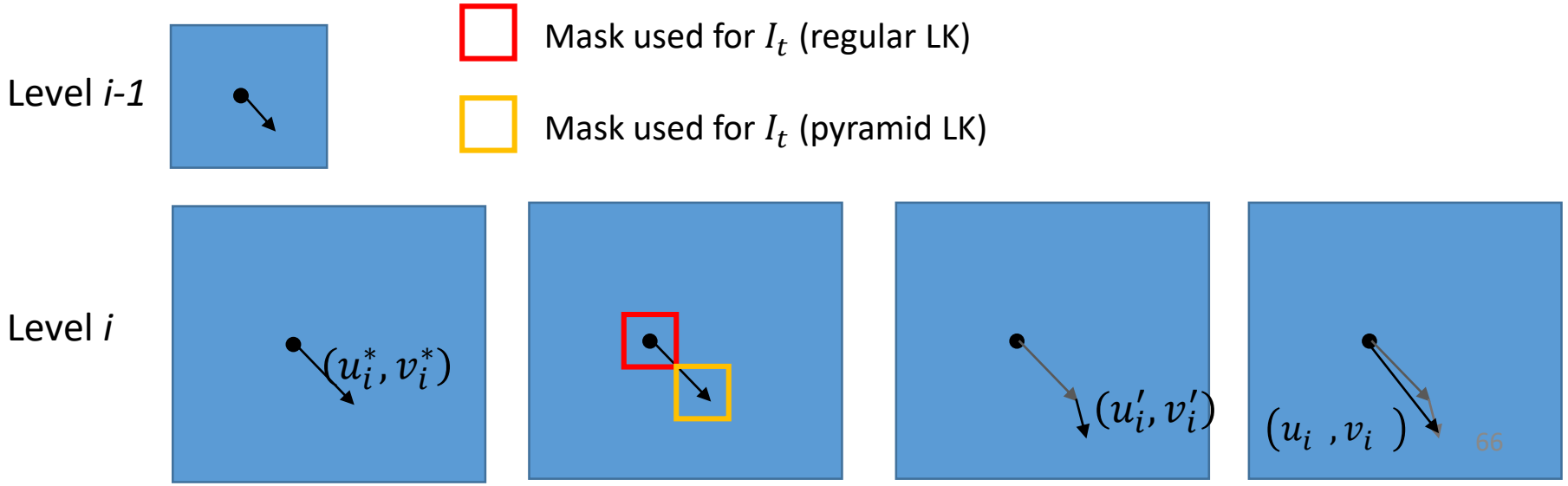
Multi-resolution Lucas – Kanade

Algorithm

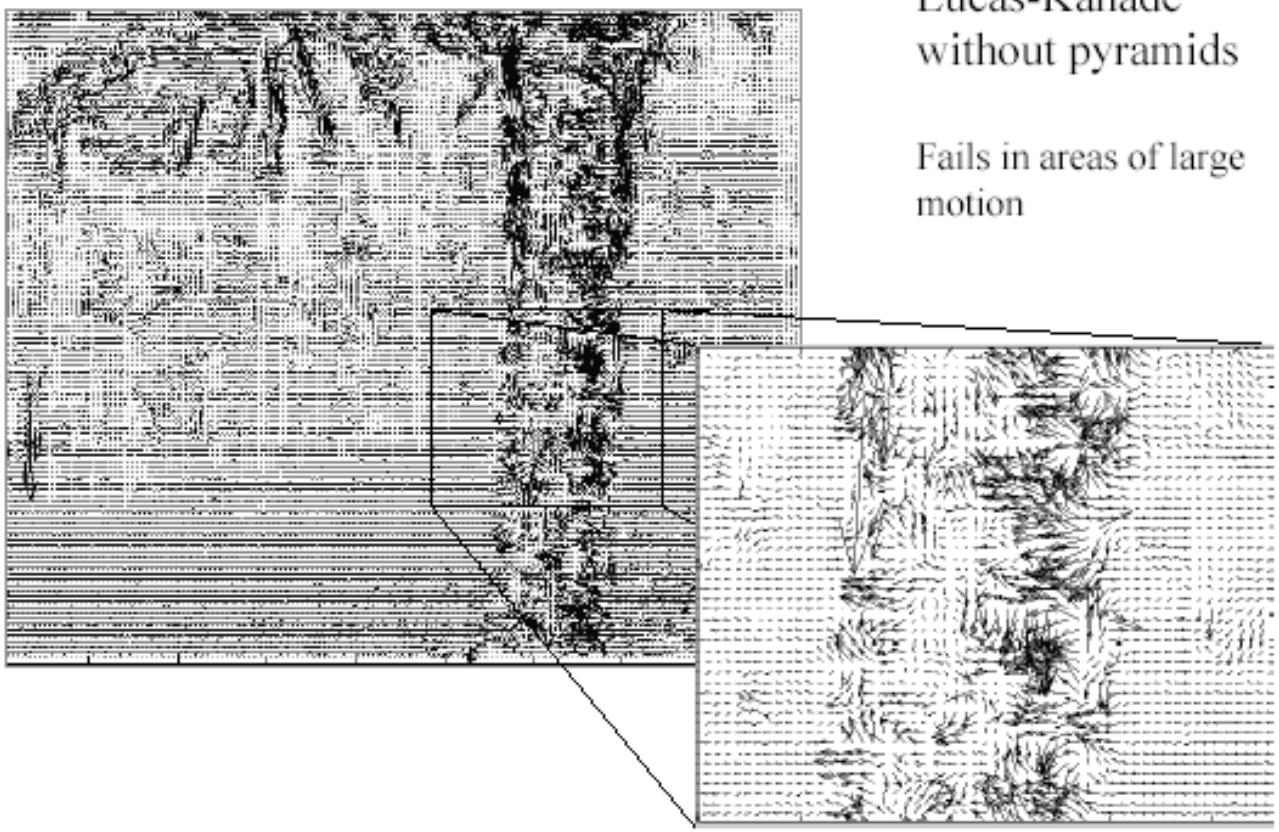
Compute simple LK at highest level

At level i

- Take flow u_{i-1} and v_{i-1} from level i-1, then bilinear interpolate it to create u_i^* and v_i^* matrices of twice resolution for level i.
- Multiply u_i^* and v_i^* by 2.
- **Compute I_t from a block displaced by $u_i^*(\mathbf{x},\mathbf{y})$ and $v_i^*(\mathbf{x},\mathbf{y})$**
- Apply LK to get u_i' and v_i' (the correction in flow)
- Add correction u_i' and v_i', i.e. $u_i = u_i' + u_i^*$ and $v_i = v_i' + v_i^*$

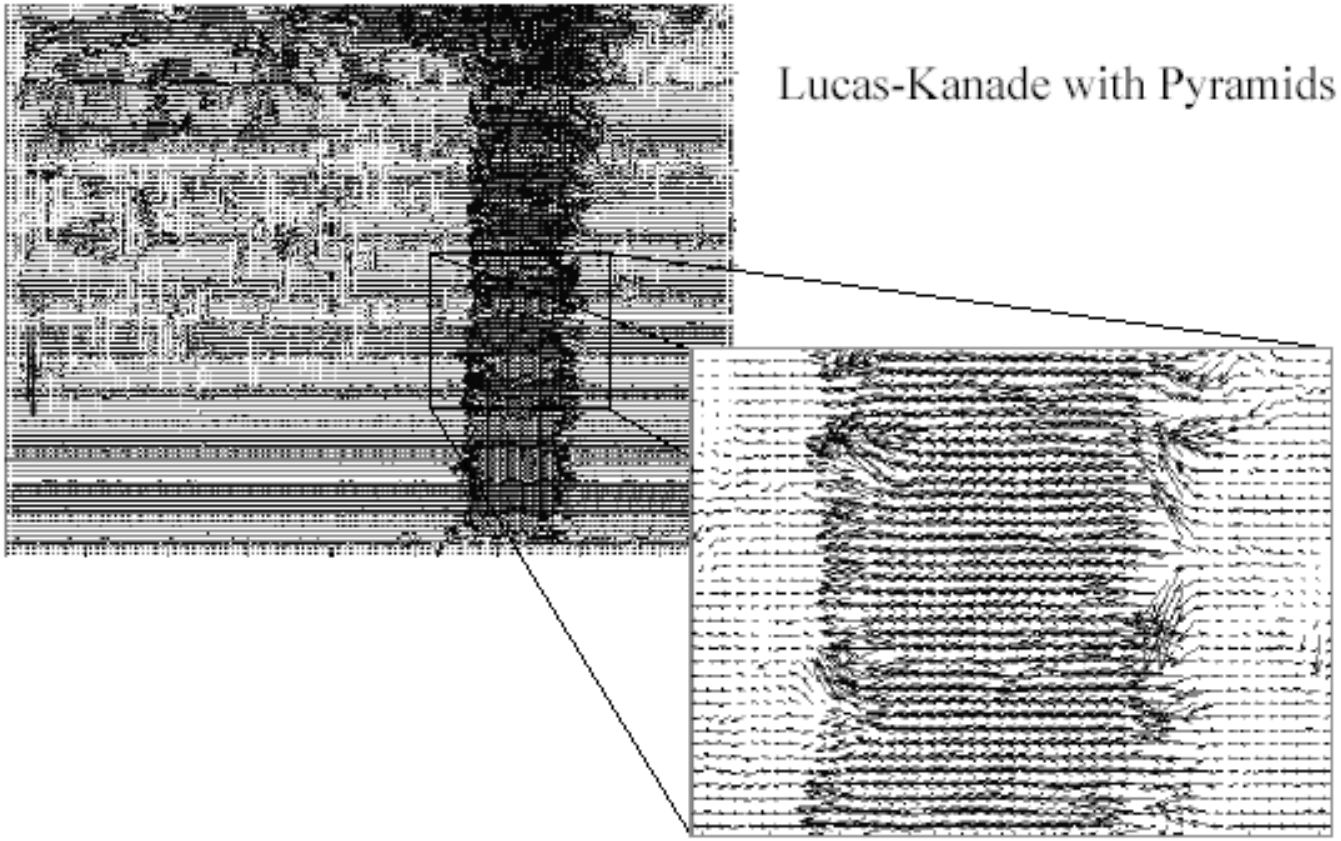


Multi-resolution Lucas – Kanade



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Multi-resolution Lucas – Kanade



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Horn and Shunck vs Lucas – Kanade

Goal : solve the Optical Flow Constraint Equation of unknowns (u,v) :

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

How to overcome the underconstraint problem ?

The Optical Flow is constant around (x,y)

Local methods (Lucas-Kanade)

The Optical Flow is smooth in the image

Global methods (Horn & Shunck)

2.4. Refinements and extensions

Refinements and extensions

- Main methods are based on the minimization of E_{data} (from BCE) and/or $E_{smoothness}$ (e.g. from HS).

- Gradient Constancy Assumption

$$\nabla I(x, y, t) = \nabla I(x + dx, y + dy, t + dt)$$

Brox et al., PAMI, 2011.

  - Can help with cases where there are slight illumination changes
  - The data term can be based on gradient constancy or on a combination of gradient and brightness constancies

$$E_{data} = (I_2(x + dx, y + dy) - I_1(x, y))^2 + \gamma(\|\nabla I_2(x + dx, y + dy) - \nabla I_1(x, y)\|^2)$$

Refinements and extensions

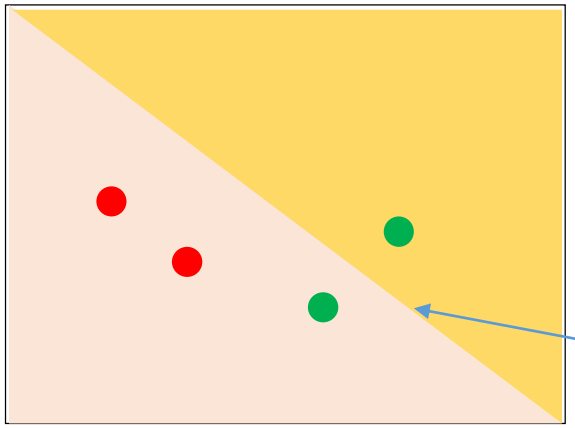
- Combining HS and LK

$$E(u,v) = E_{LK}(u,v) + \alpha E_{smoothness}(u,v)$$

Bruhn et al., IJCV, 2005.

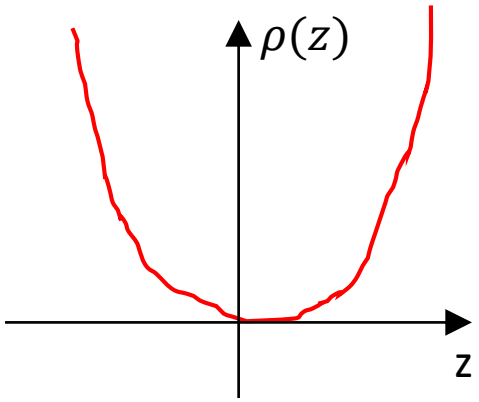
- No smoothness along edges (oriented smoothness constraint)

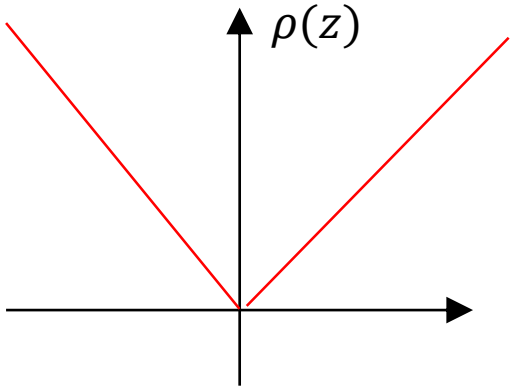
I don't want the green pixels to have the same flow vector. But, the red pixels can have the same flow vectors.

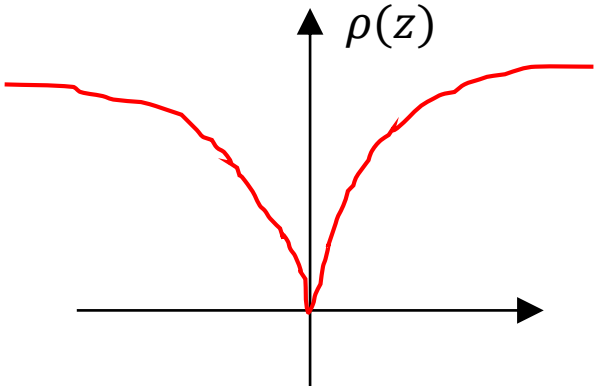


Refinements and extensions

- Robust cost function
  - Basic cost function based on $\left(I_2(x+dx, y+dy) - I_1(x,y)\right)^2$
  - Outliers contribute overmuch to the cost







Many examples in the literature: Lorentzian, Charbonnier, Generalized Charbonnier, etc…

With $z = I_2(x+dx, y+dy) - I_1(x,y)$

Refinements and extensions

- To go further…

Secrets of Optical Flow Estimation and Their Principles

Deqing Sun
Brown University

Stefan Roth
TU Darmstadt

Michael J. Black
Brown University

Abstract

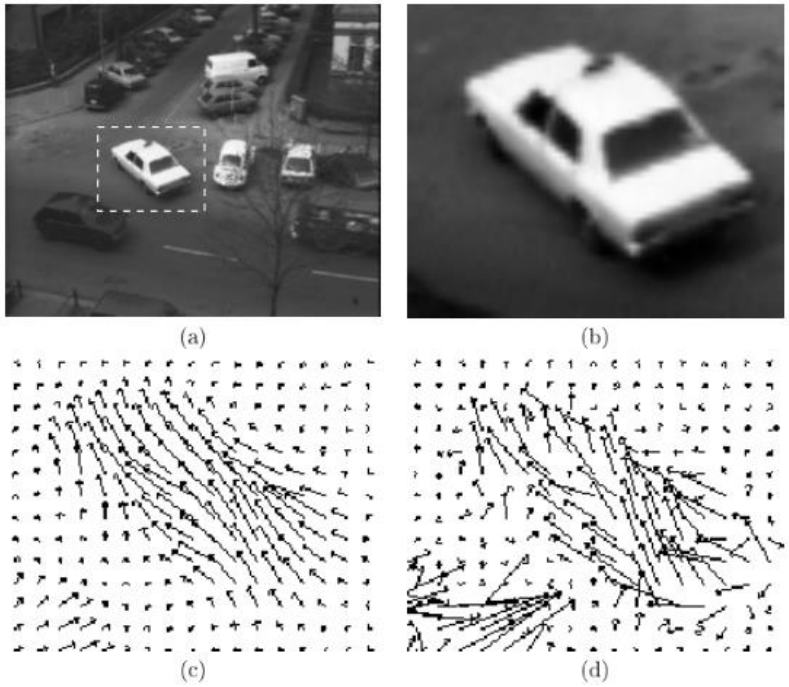
The accuracy of optical flow estimation algorithms has been improving steadily as evidenced by results on the Middlebury optical flow benchmark. The typical formulation, however, has changed little since the work of Horn and Schunck. We attempt to uncover what has made recent advances possible through a thorough analysis of how the objective function, the optimization method, and modern implementation practices influence accuracy. We discover that "classical" flow formulations perform surprisingly well when combined with modern optimization and implementation techniques. Moreover, we find that while median filtering of intermediate flow fields during optimization is a key to recent performance gains, it leads to higher energy solutions. To understand the principles behind this

The most accurate methods on the Middlebury flow dataset make different choices about how to model the objective function, how to approximate this model to make it computationally tractable, and how to optimize it. Since most published methods change *all* of these properties at once, it can be difficult to know which choices are most important. To address this, we define a baseline algorithm that is "classical", in that it is a direct descendant of the original HS formulation, and then systematically vary the model and method using different techniques from the art. The results are surprising. We find that only a small number of key choices produce statistically significant improvements and that they can be combined into a very simple method that achieves accuracies near the state of the art. More importantly, our analysis reveals what makes current flow methods work so well.

CVPR 2010

3. Evaluation

Experimental results



(a) Hamburg sequence (b) Zoom on the taxi for which optical flow is observed (c) Horn and Schunk (d) Lucas and Kanade(*)

(*)G. Aubert, R. Deriche and P. Kornprobst. Computing optical flow via variational techniques. SIAM Journal of Applied Mathematics, pp 156-182, 1999.

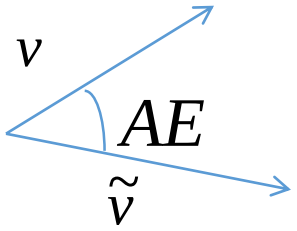
Experimental results

- How to evaluate the optical flow estimation?
  - Synthetic Image sequences
  - Real Image sequences (manual annotation)

Error measurement

- Angular Error
  - The distance between a measured velocity $\tilde{v} = (\tilde{v}_1, \tilde{v}_2)$ and a correct velocity $v = (v_1, v_2)$ can be calculated by :

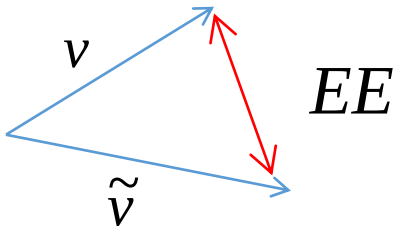
$$AE = \cos^{-1}\left(\frac{\tilde{v}_1 v_1 + \tilde{v}_2 v_2 + 1}{\sqrt{\tilde{v}_1^2 + \tilde{v}_2^2 + 1}\sqrt{v_1^2 + v_2^2 + 1}}\right)$$



Error measurement

- Endpoint Error
  - The distance between a measured velocity $\tilde{v} = (\tilde{v}_1, \tilde{v}_2)$ and a correct velocity $v = (v_1, v_2)$ can be calculated by :

$$EE = \sqrt{(v_1 - \tilde{v}_1)^2 + (v_2 - \tilde{v}_2)^2}$$



Experimental results

- How to evaluate the optical flow estimation?
  - Synthetic Image sequences
  - Real Image sequences

http://vision.middlebury.edu/flow/