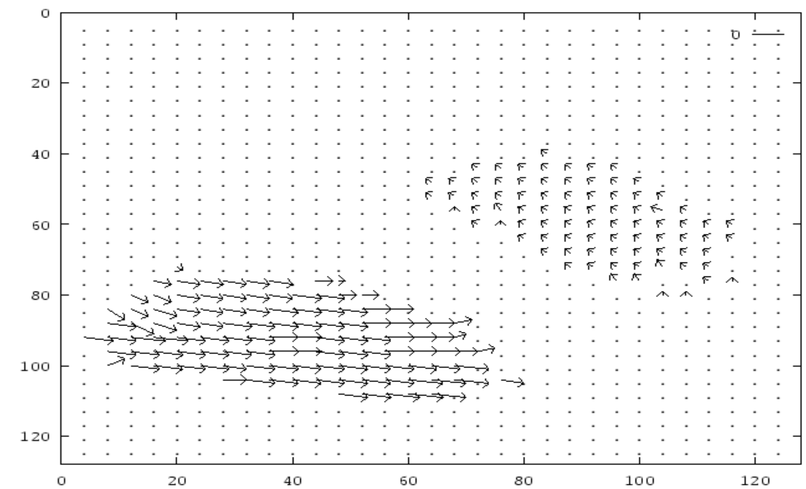
Computing Optical Flow

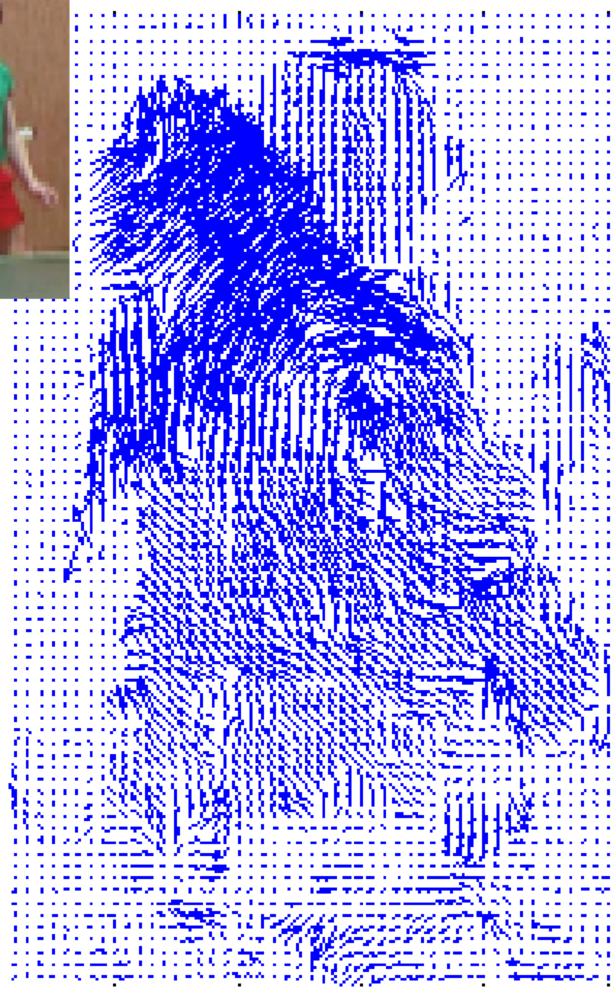
Lecture-7

Optical Flow

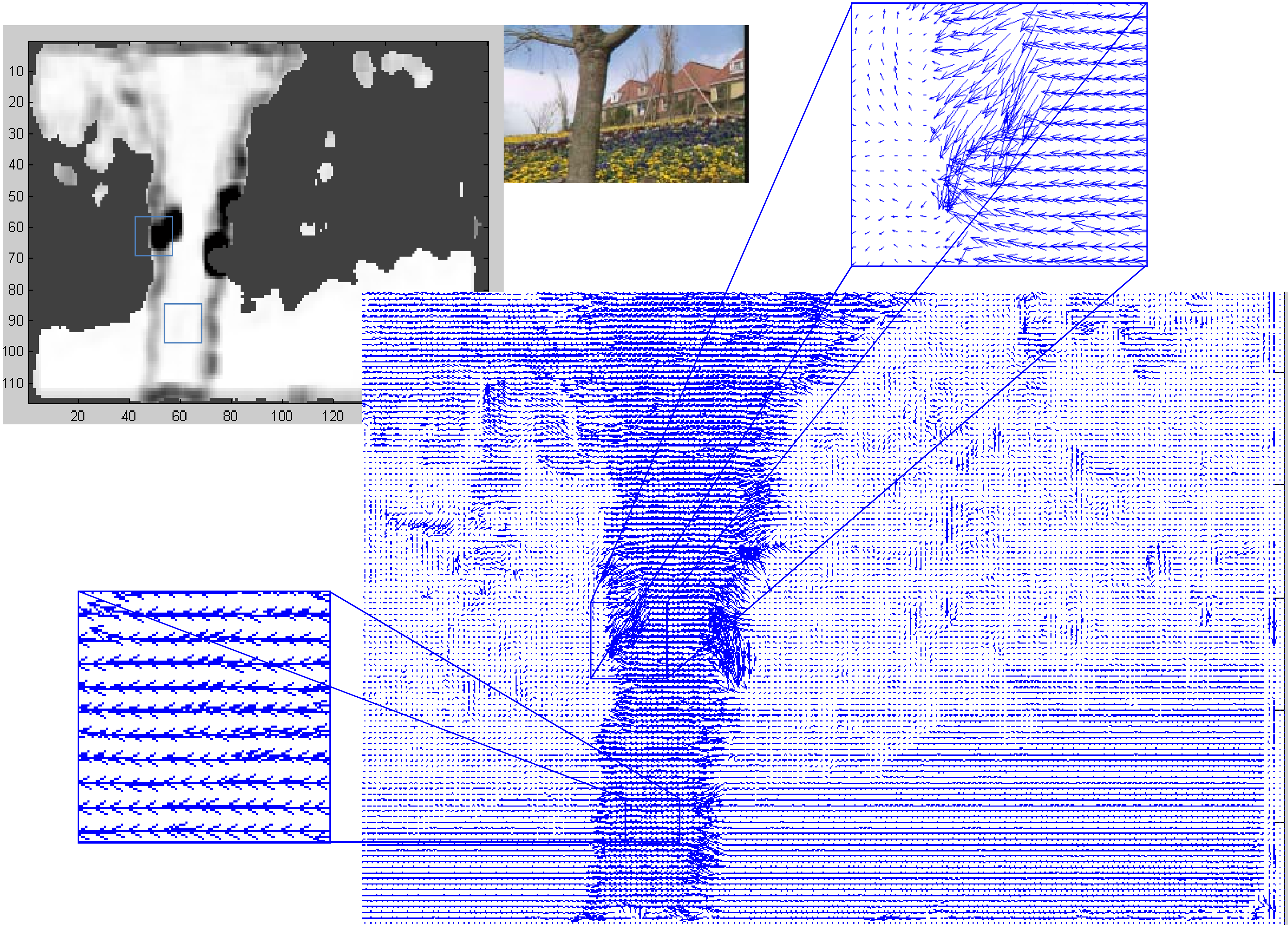
- Motion vector (*u,v)*
  - Image Displacement in *x* and *y* directions between two consecutive frames
- Motion Flow
- Displacement Vector
- Disparity (general case)

Hamburg Taxi seq



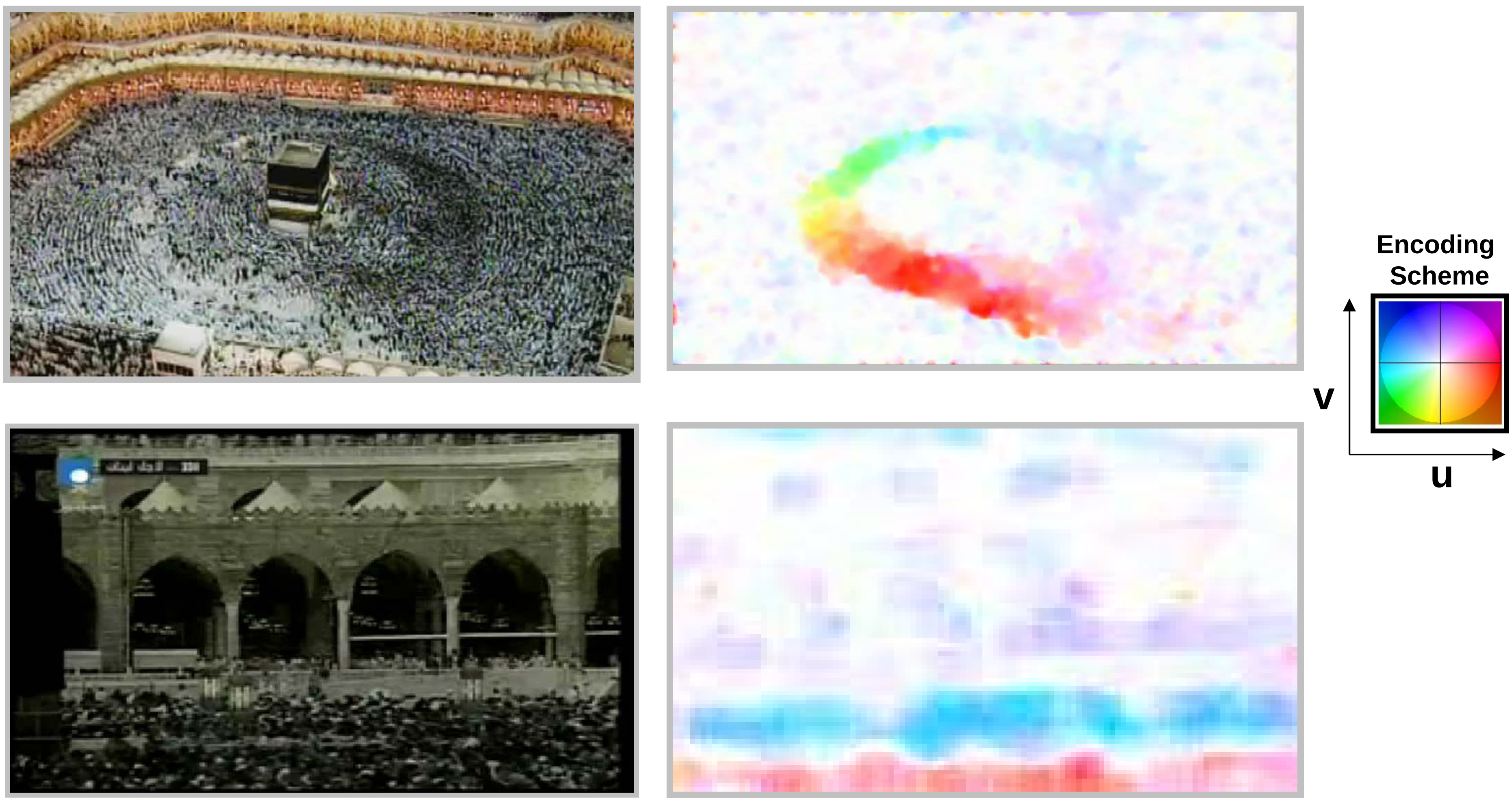

CLAIROL
Lufthans
IBM
ATP Tour
EBEL



10
20
30
40
50
60
70
80
90
100
110
20
40
60
80
100
120

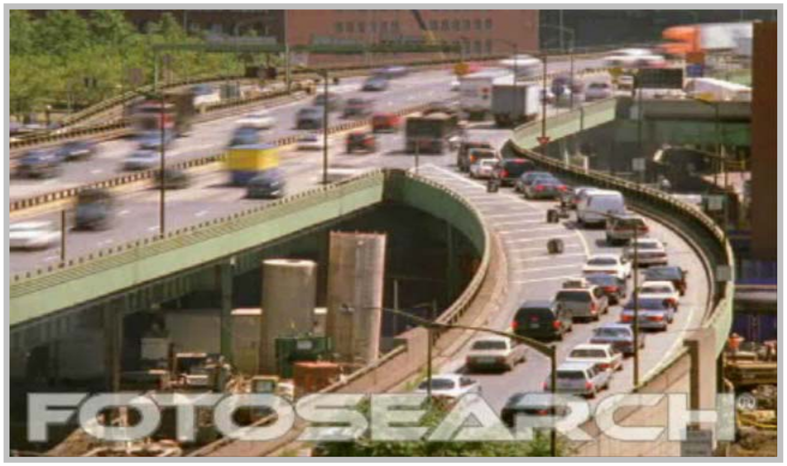
Optical Flow Field Examples

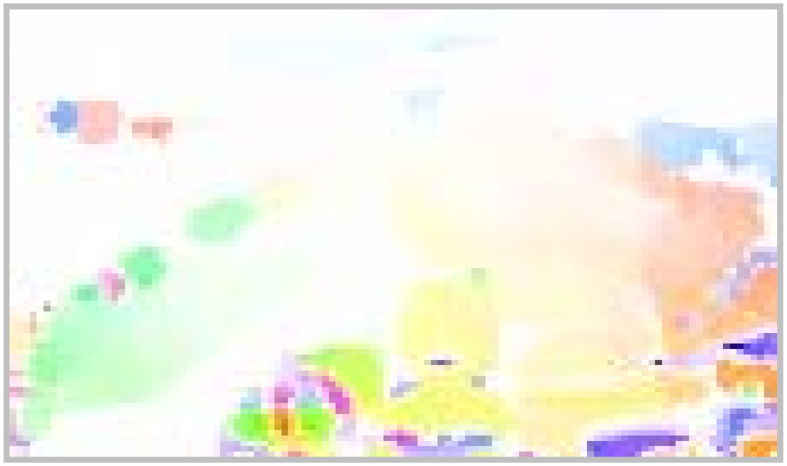
Optical Flow - Examples



Videos **Color Coded Optical Flows**

Optical Flow - Examples

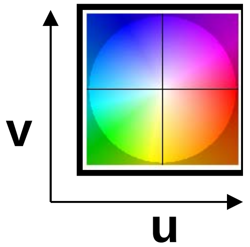




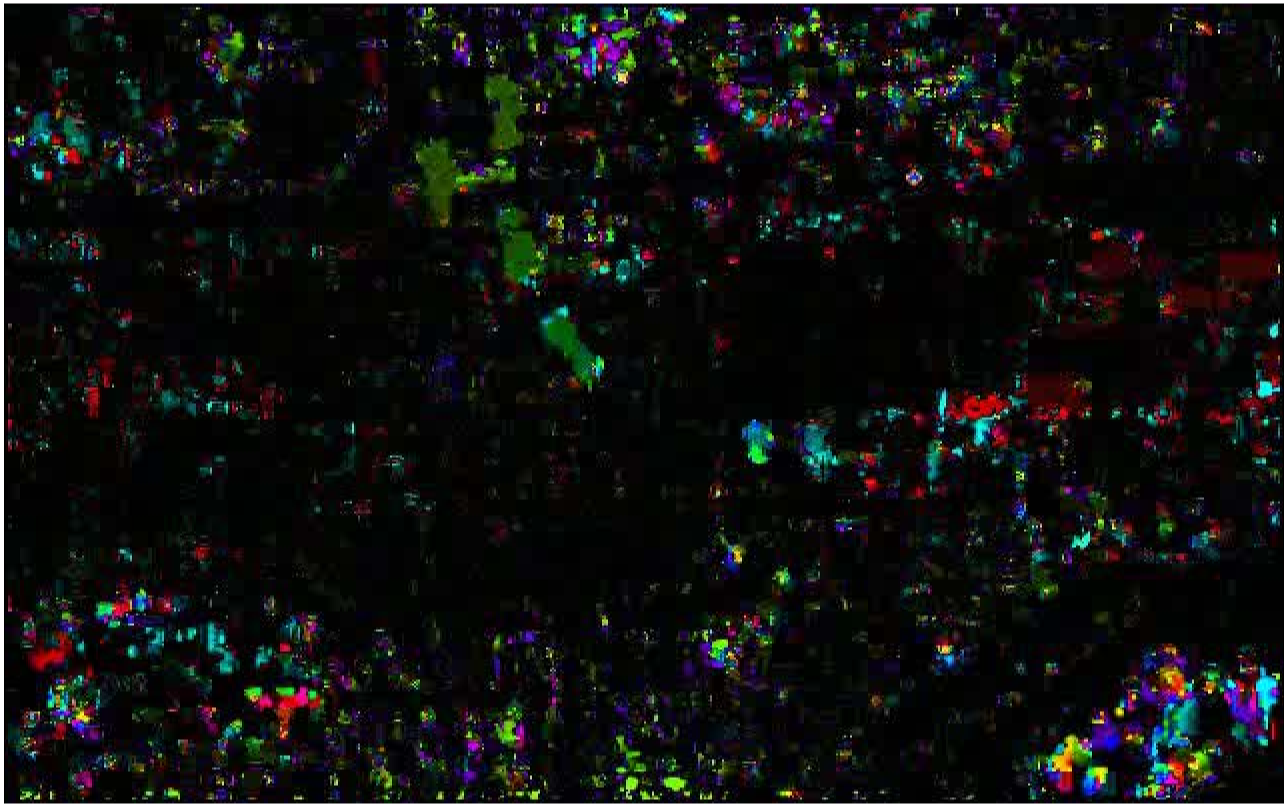
Videos

Color Coded Optical Flows

Encoding Scheme

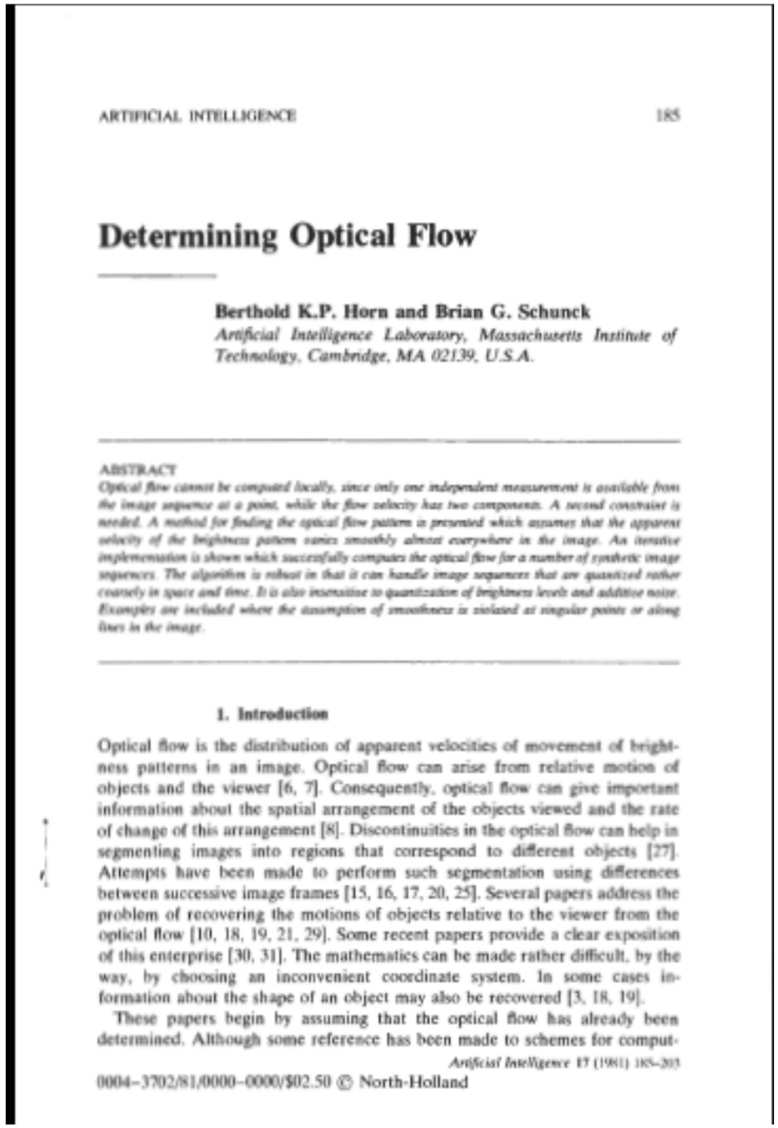


Optical Flow



Optical Flow

- Applications
  - Motion based segmentation
  - Structure from Motion(3D shape and Motion)
  - Alignment (Global motion compensation)
    - Camcorder video stabilization
    - UAV Video Analysis
  - Video Compression



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.

1. Introduction

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer [6, 7]. Consequently, optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement [8]. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects [27]. Attempts have been made to perform such segmentation using differences between successive image frames [15, 16, 17, 20, 25]. Several papers address the problem of recovering the motions of objects relative to the viewer from the optical flow [10, 18, 19, 21, 29]. Some recent papers provide a clear exposition of this enterprise [30, 31]. The mathematics can be made rather difficult, by the way, by choosing an inconvenient coordinate system. In some cases information about the shape of an object may also be recovered [3, 18, 19].

These papers begin by assuming that the optical flow has already been determined. Although some reference has been made to schemes for comput-

Artificial Intelligence 17 (1981) 185–203
0004–3702/81/0000–0000/$02.50 © North-Holland

Horn&Schunck Optical Flow

Brightness constancy assumption

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

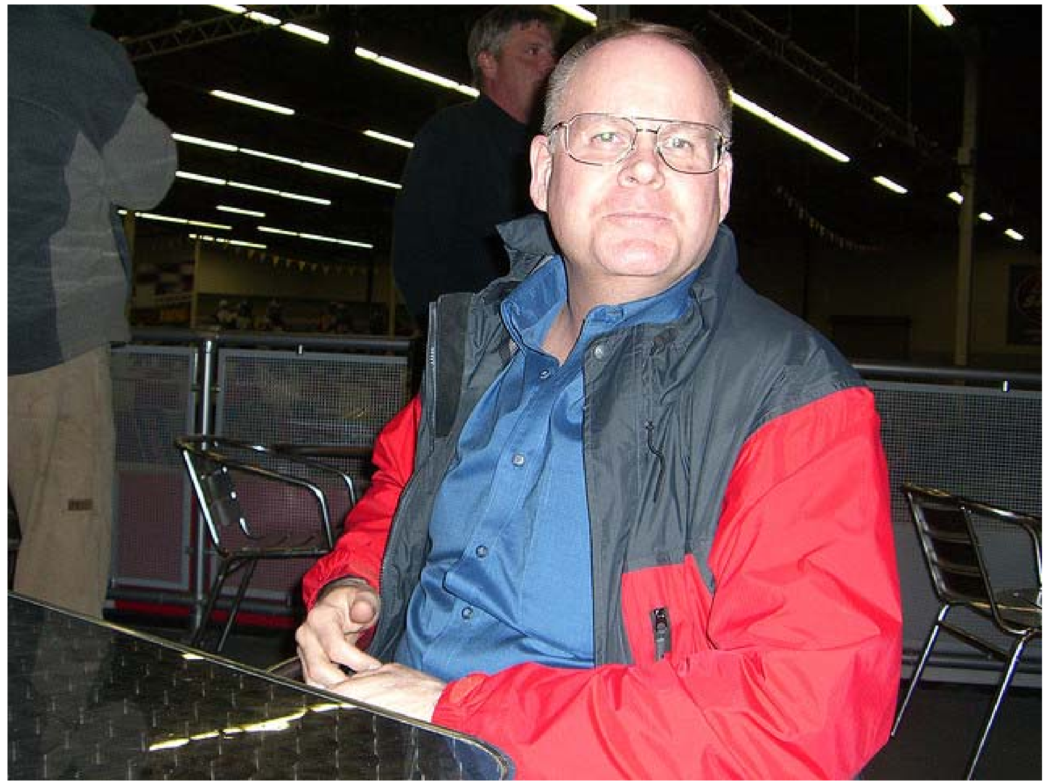
Horn&Schunck Optical Flow

Brightness constancy assumption

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

Taylor Series

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots.$$

Horn&Schunck Optical Flow

Brightness constancy assumption

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

Taylor Series

$$f(x,y,t) = f(x,y,t) + \frac{\partial f}{\partial x}(x+dx-x) + \frac{\partial f}{\partial y}(y+dy-y) + \frac{\partial f}{\partial t}(t+dt-t)$$

$$0 = f_x dx + f_y dy + f_t dt$$

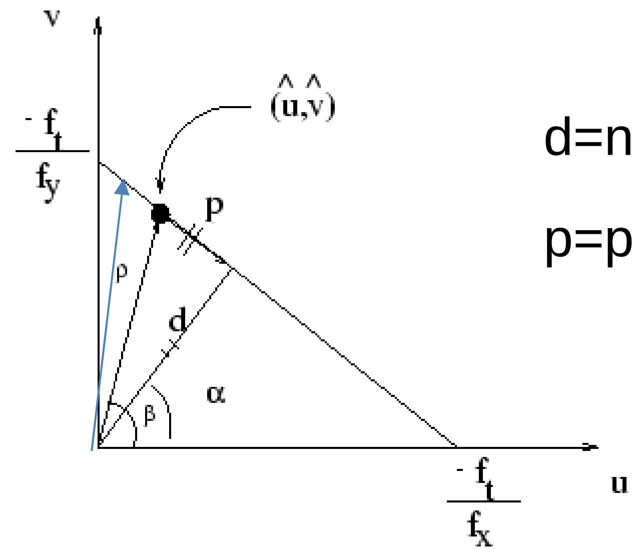
$$0 = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} + f_t \frac{dt}{dt}$$

$$0 = f_x u + f_y v + f_t$$

Interpretation of optical flow eq

$$f_x u + f_y v + f_t = 0$$

$$v = -\frac{f_x}{f_y} u - \frac{f_t}{f_y}$$



d=normal flow

p=parallel flow

$$d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}}$$

Equation of st.line

Horn&Schunck (contd)

$$\int\int\{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\}dxdy$$

Brightness constancy

Smoothness constraint

Min (Variational Calculus

$$(f_x u + f_y v + f_t)f_x + \lambda(\Delta^2 u) = 0$$

$$(f_x u + f_y v + f_t)f_y + \lambda((\Delta^2 v) = 0$$

$$\Delta^2 u = u_{xx} + u_{yy}$$

Derivative Masks (Roberts)

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{first image} \qquad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{first image} \qquad \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \text{first image}$$

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{second image} \qquad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{second image} \qquad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{second image}$$

$$\mathrm{f}_x \qquad\qquad \mathrm{f}_y \qquad\qquad \mathrm{f}_t$$

Apply first mask to 1st image
Second mask to 2nd image
Add the responses to get *f_x*, *f_y*, *f_t*

Laplacian

$$\begin{matrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{matrix}$$

$$f_{xx} + f_{yy}$$

$$f_{xx} + f_{yy} = f - f_{av}$$

Horn&Schunck (contd)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dxdy$$

$\Downarrow$ min

$$(f_x u + f_y v + f_t) f_x + \lambda(\Delta^2 u) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((\Delta^2 v) = 0$$

$\Downarrow$ discrete version

$$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((v - v_{av}) = 0$$

variational calculus

$$u = u_{av} - f_x \frac{P}{D}$$
$$v = v_{av} - f_y \frac{P}{D}$$

$$P = f_x u_{av} + f_y v_{av} + f_t$$
$$D = \lambda + f_x^2 + f_y^2$$

$$\Delta^2 u = u_{xx} + u_{yy}$$

Algorithm-1

- k=0
- Initialize u^K v^K
- Repeat until some error measure is satisfied (converges)

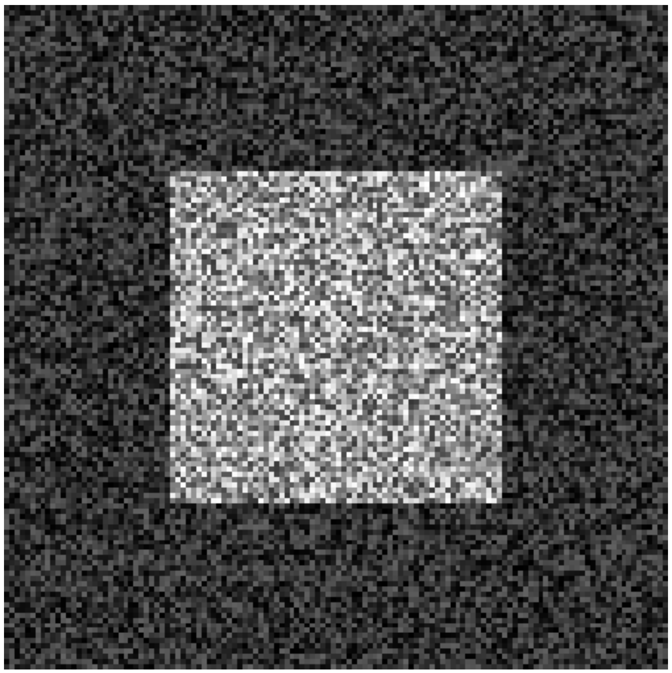
$$u = u_{av} - f_x \frac{P}{D}$$

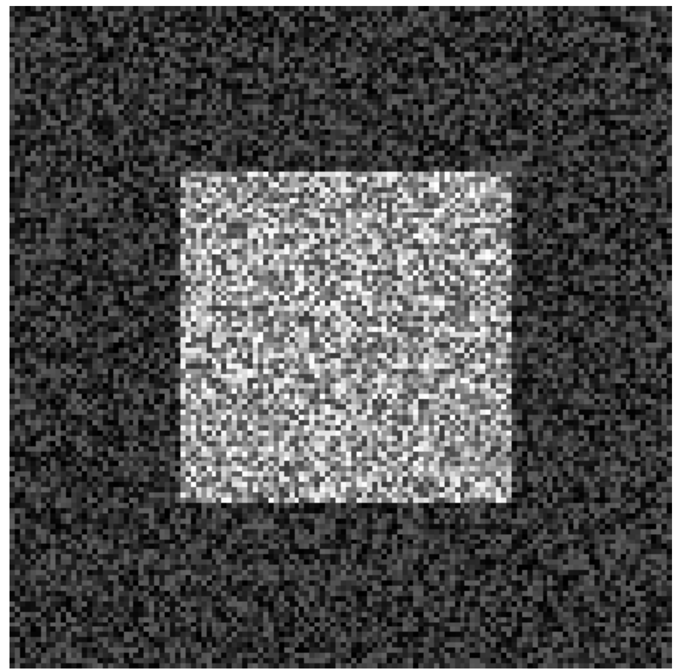
$$v = v_{av} - f_y \frac{P}{D}$$

$$P = f_x u_{av} + f_y v_{av} + f_t$$

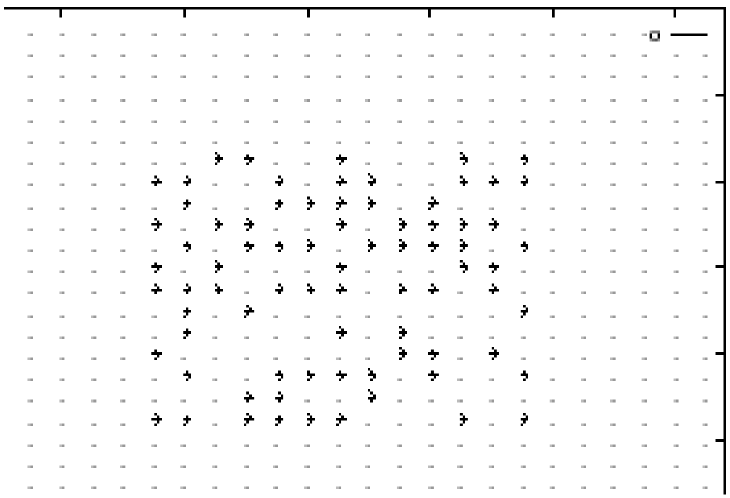
$$D = \lambda + f_x^2 + f_y^2$$

Synthetic Images

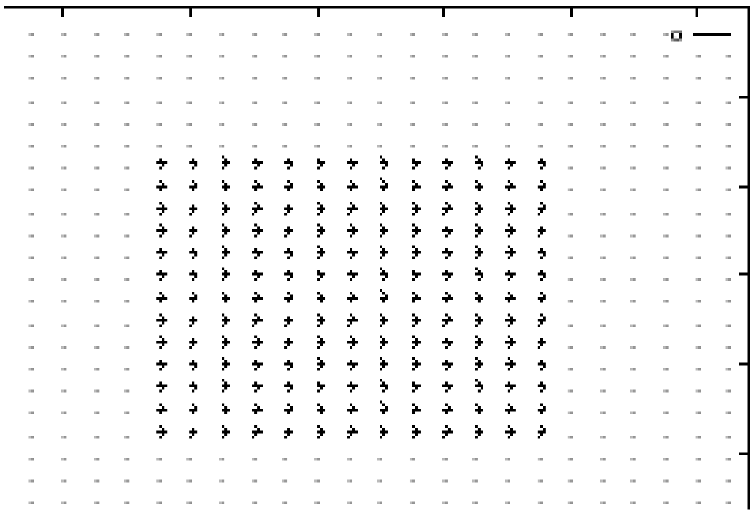




Horn & Schunck Results



One iteration



10 iterations

Lucas & Kanade Method

Proceedings DARPA Image Understanding Workshop, April 1981, pp. 121-130

essentially the same but shorter version of this paper was presented and included in the <u>Proc 7th Intl Joint Conf on Artificial Intelligence (IJCAI) 1981, August 24-28, Vancouver, British Columbia, pp.674-679.</u> when you refer to the work, please refer to the <u>IJCAI paper</u>.

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 images.

In this paper we present a new image registration technique that uses spatial intensity gradient information to direct the search for the position that yields the best match. By taking more information about the images into account, this technique is able to find the best match between two images with far fewer comparisons of images than techniques which examine the possible positions of registration in some fixed order. Our technique takes advantage of the fact that in many applications the two images are already in approximate registration. This technique can be generalized to deal with arbitrary linear distortions of the image, including rotation. We then describe a stereo vision system that uses this registration technique, and suggest some further avenues for research toward making effective use of this method in stereo image understanding.

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 which minimizes some measure of the difference between $F(x+h)$ and $G(x)$, for x in some region of interest R. (See figure 1).

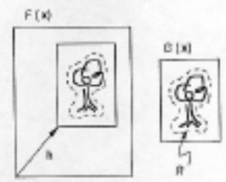


Figure 1: The image registration problem

Typical measures of the difference between $F(x+h)$ and $G(x)$ are:

- L_1 norm $= \sum_{x \in R} |F(x+h) - G(x)|$
- L_2 norm $= \left(\sum_{x \in R} [F(x+h) - G(x)]^2\right)^{1/2}$
- negative of normalized correlation

$$= \frac{-\sum_{x \in R} F(x+h)G(x)}{\left(\sum_{x \in R} F(x+h)^2\right)^{1/2}\left(\sum_{x \in R} G(x)^2\right)^{1/2}}$$

We will propose a more general measure of image difference, of which both the L_2 norm and the correlation are special cases. The L_1 norm is chiefly of interest as an inexpensive approximation to the L_2 norm.

Lucas & Kanade (Least Squares)

- Optical flow eq

$$f_x u + f_y v = -f_t$$

- Consider 3 by 3 window

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ & \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ \\ -f_{t9} \end{bmatrix}$$

$$f_{x1} u + f_{y1} v = -f_{t1}$$

$$\vdots$$

$$f_{x9} u + f_{y9} v = -f_{t9}$$

$$\mathbf{Au} = \mathbf{f_t}$$

Lucas & Kanade

$$\mathbf{Au} = \mathbf{f_t}$$

$$\mathbf{A^T A u} = \mathbf{A^T f_t}$$

$$\mathbf{u} = \boxed{(\mathbf{A^T A})^{-1} \mathbf{A^T}} \mathbf{f_t}$$

Pseudo Inverse

$$\text{find } (u, v) \text{ s.t. } \min \sum_i (f_{xi} u + f_{yi} v + f_t)^2$$

Least Squares Fit

Lucas & Kanade

$$\min \sum_i (f_{xi} u + f_{yi} v + f_t)^2$$

$$\Downarrow$$

$$\frac{\partial}{\partial u} \sum_i (f_{xi} u + f_{yi} v + f_t)^2 = 0 \qquad \sum (f_{xi} u + f_{yi} v + f_{ti}) f_{xi} = 0$$

$$\frac{\partial}{\partial v} \sum_i (f_{xi} u + f_{yi} v + f_t)^2 = 0 \qquad \sum (f_{xi} u + f_{yi} v + f_{ti}) f_{yi} = 0$$

Lucas & Kanade

$$\sum (f_{xi}u + f_{yi}v + f_{ti}) f_{xi} = 0$$

$$\sum (f_{xi}u + f_{yi}v + f_{ti}) f_{yi} = 0$$

$$\sum f_{xi}^2 u + \sum f_{xi} f_{yi} v = -\sum f_{xi} f_{ti}$$
$$\sum f_{xi} f_{yi} u + \sum f_{yi}^2 v = -\sum f_{yi} f_{ti}$$

$$\boxed{\begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix}} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

Lucas & Kanade

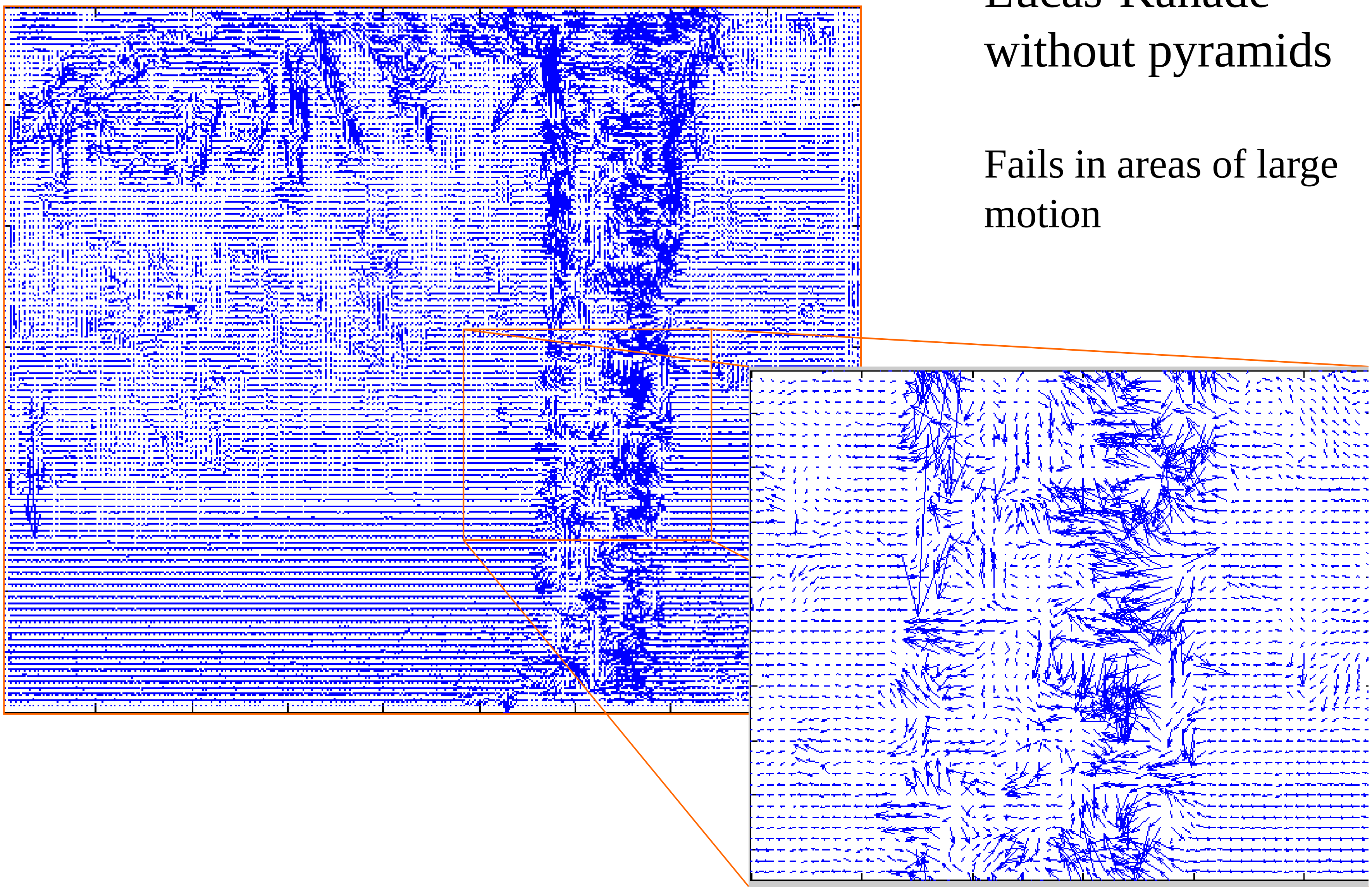
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2} \begin{bmatrix} \sum f_{yi}^2 & -\sum f_{xi} f_{yi} \\ -\sum f_{xi} f_{yi} & \sum f_{xi}^2 \end{bmatrix} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

Lucas & Kanade

$$u = \frac{-\sum f_{yi}^2 \sum f_{xi} f_{ti} + \sum f_{xi} f_{yi} \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$

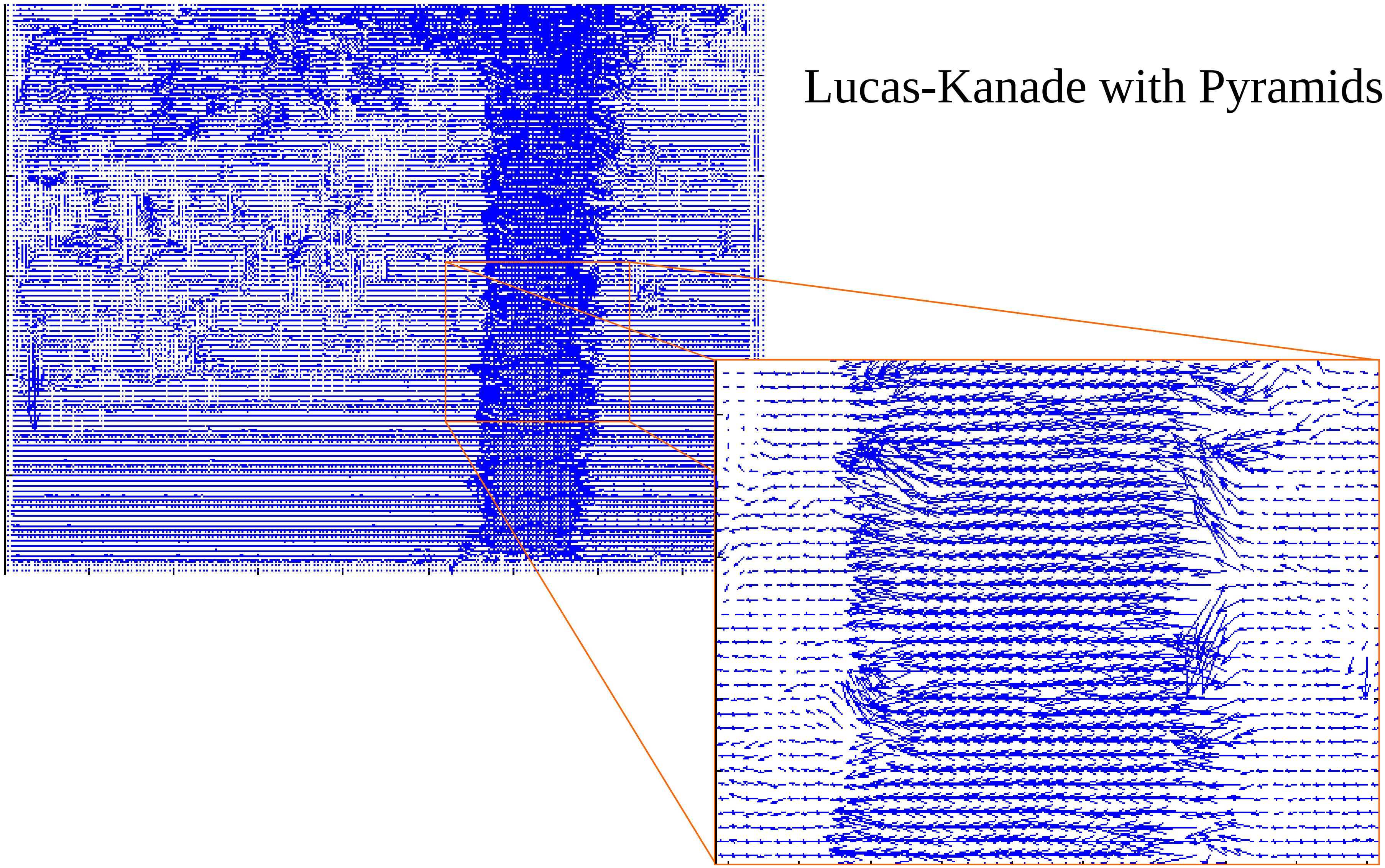
$$v = \frac{\sum f_{xi} f_{ti} \sum f_{xi} f_{yi} - \sum f_{xi}^2 \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$



Lucas-Kanade without pyramids

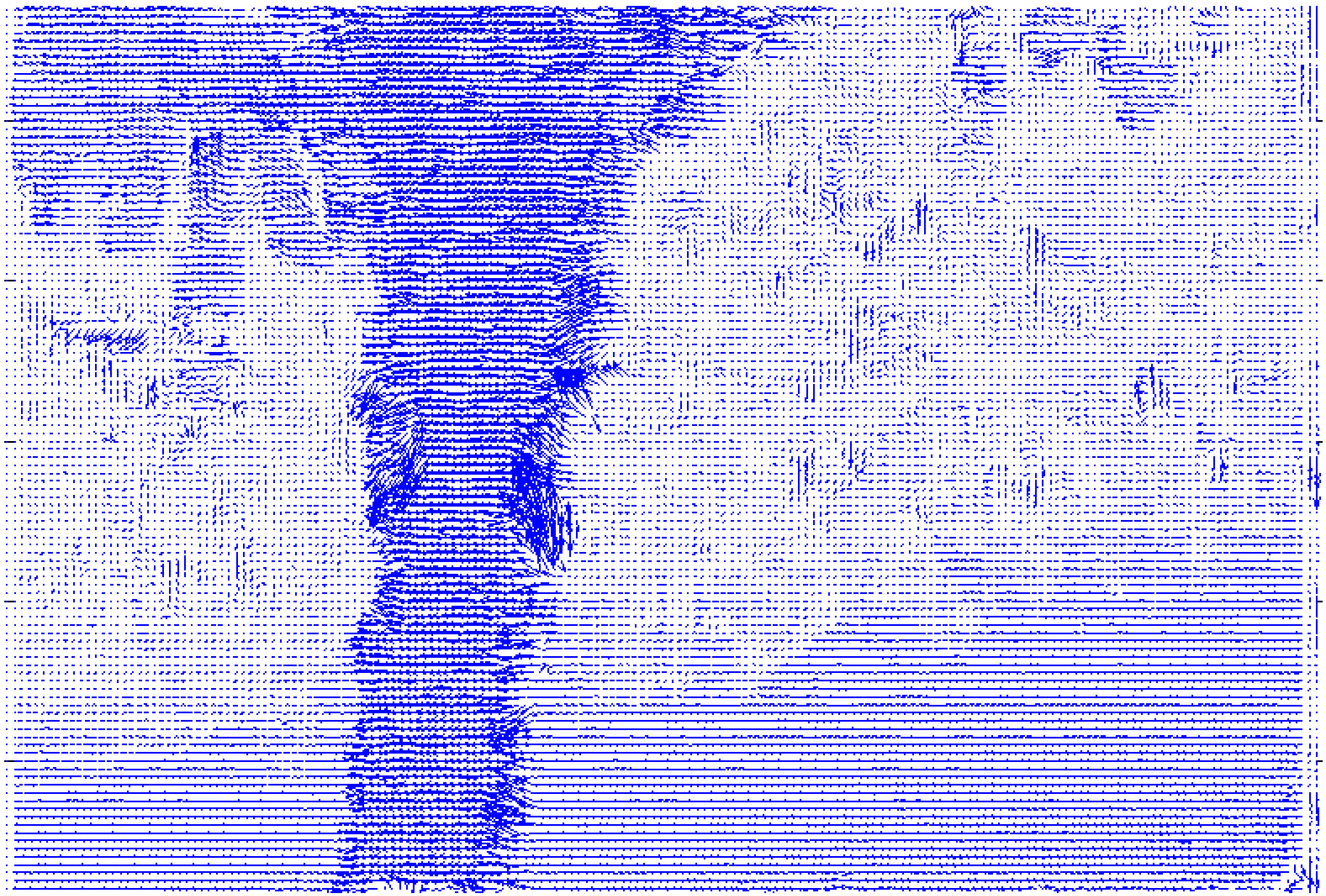
Fails in areas of large motion

Lucas-Kanade with Pyramids

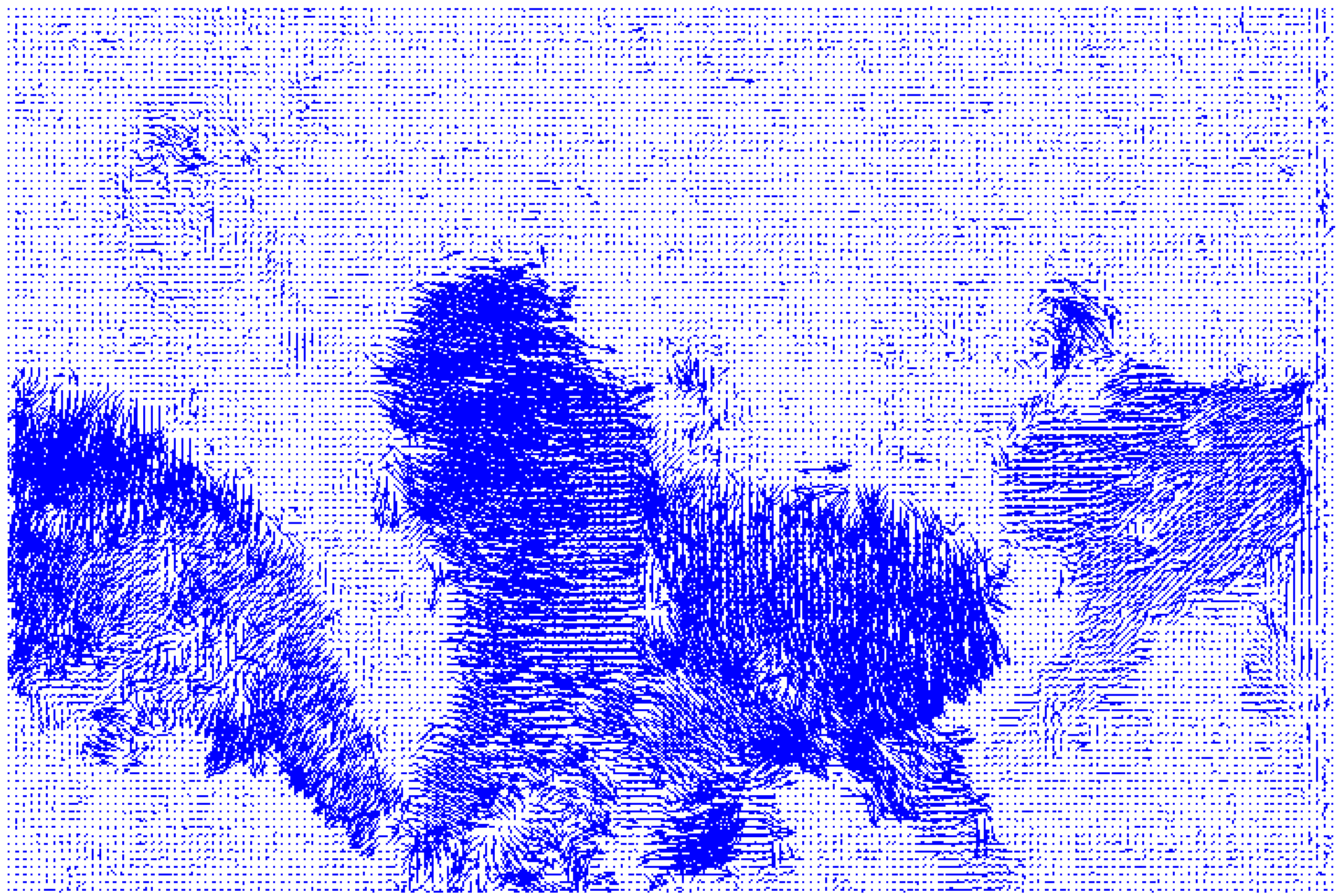


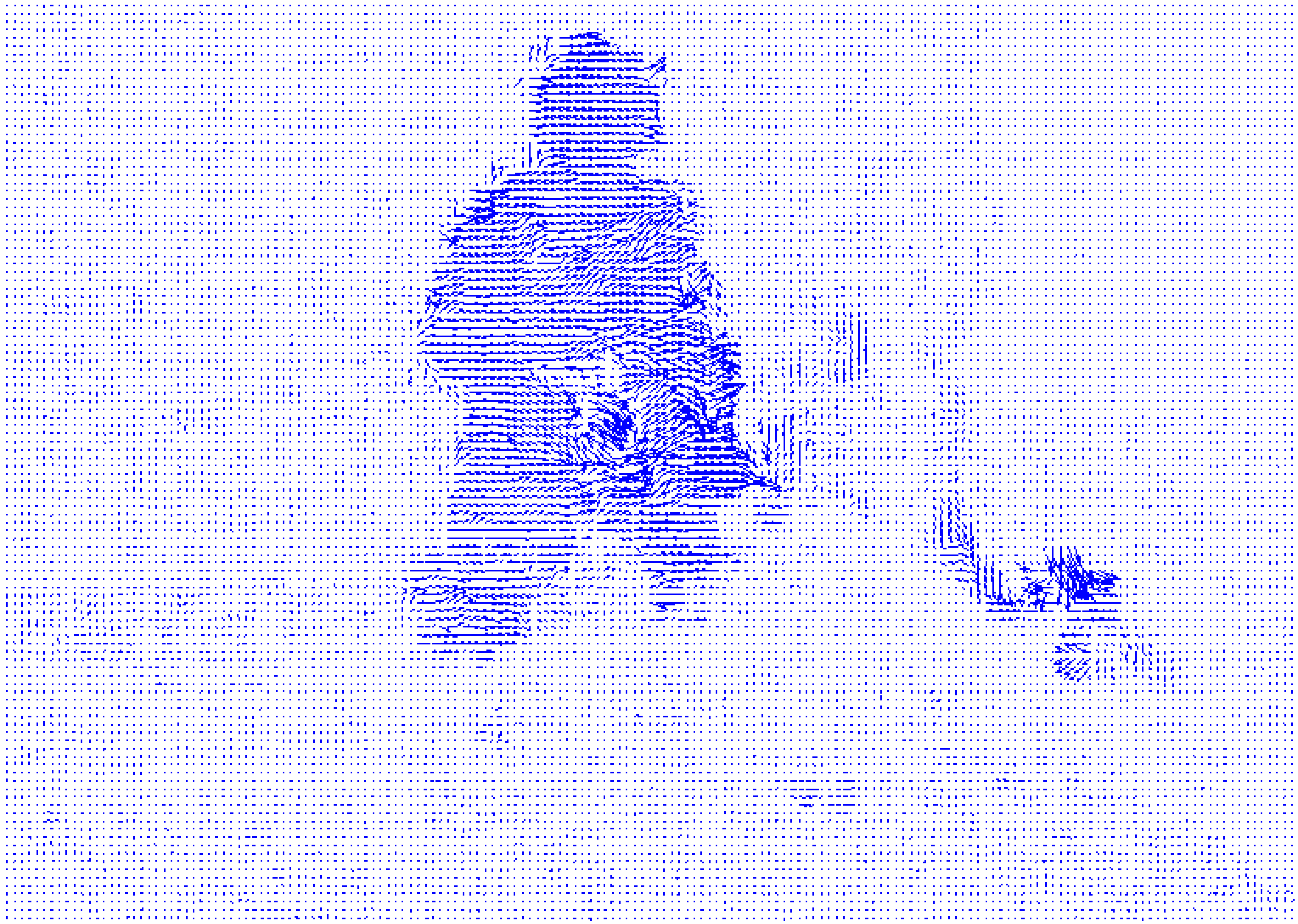
Comments

- Horn-Schunck and Lucas-Kanade optical methods work only for small motion.
- If object moves faster, the brightness changes rapidly,
  - 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.
- Pyramids can be used to compute large optical flow vectors.




87



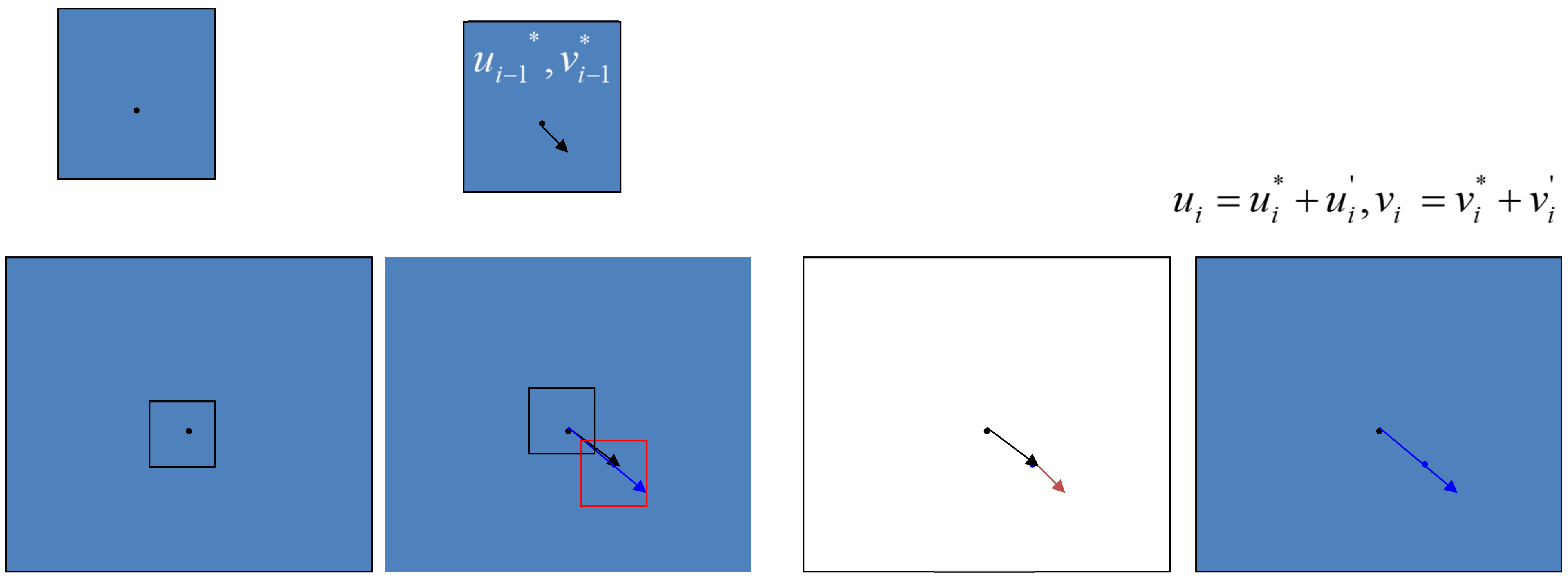


Lucas Kanade with Pyramids

- Compute 'simple' LK optical flow at highest level
- At level *i*
  - Take flow u_{i-1}, v_{i-1} from level *i*-1
  - bilinear interpolate it to create u_i^*, v_i^* matrices of twice resolution for level *i*
  - multiply u_i^*, v_i^* by 2
  - compute f_t ,f_x ,f_y using masks centered at (x,y) and $(x+u_i^*, y+v_i^*)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections u_i' v_i' , *i.e.* $u_i = u_i^* + u_i'$,

  $v_i = v_i^* + v_i'$.

Pyramids



pyramid

pyramid

Interpolation

	0	1	2	3
0	●	●	●	●
$u = 1$	●	●	●	●
2	●	●	●	●
3	●	●	●	●

	0	1	2	3
0	●	●	●	●
$v = 1$	●	●	●	●
2	●	●	●	●
3	●	●	●	●

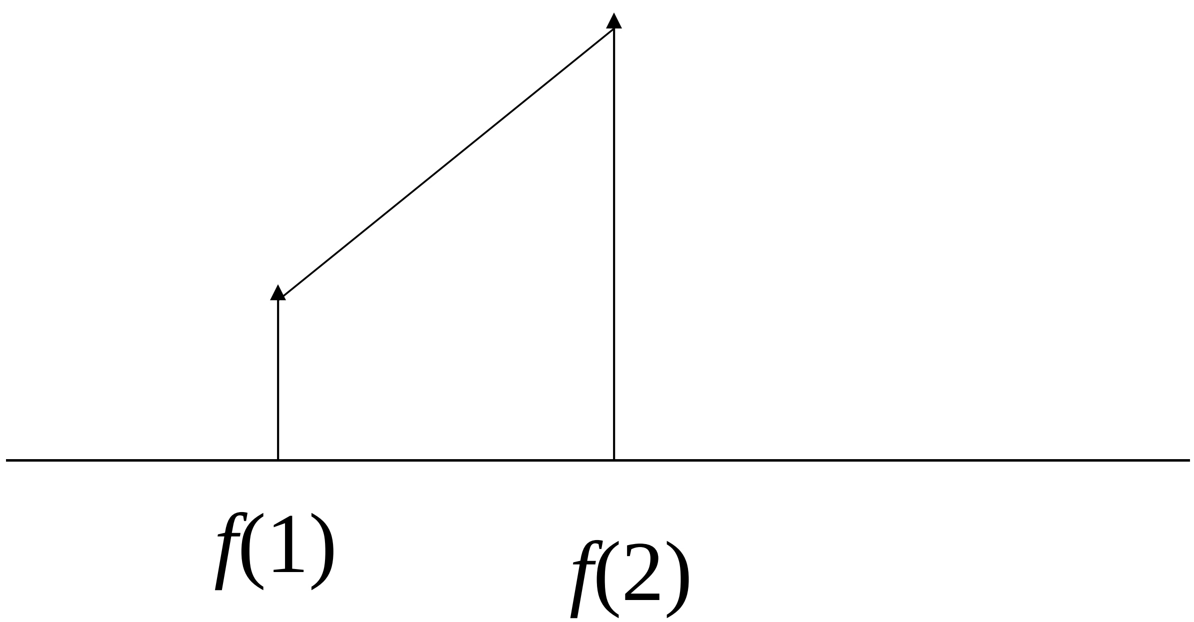
	0	1	2	3	4	5	6	7
0	●	○	●	○	●	○	●	○
1	○	○	○	○	○	○	○	○
2	●	○	●	○	●	○	●	○
$u^* = 3$	○	○	○	○	○	○	○	○
4	●	○	●	○	●	○	●	○
5	○	○	○	○	○	○	○	○
6	●	○	●	○	●	○	●	○
7	○	○	○	○	○	○	○	○

	0	1	2	3	4	5	6	7
0	●	○	●	○	●	○	●	○
1	○	○	○	○	○	○	○	○
2	●	○	●	○	●	○	●	○
$v^* = 3$	○	○	○	○	○	○	○	○
4	●	○	●	○	●	○	●	○
5	○	○	○	○	○	○	○	○
6	●	○	●	○	●	○	●	○
7	○	○	○	○	○	○	○	○

1-D Interpolation

$$y = mx + c$$

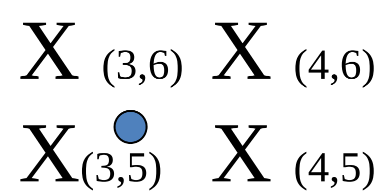
$$f(x) = mx + c$$



2-D Interpolation

$$f(x,y) = a_1 + a_2 x + a_3 y + a_4 xy$$

Bilinear



Bi-linear Interpolation

Four nearest points of (x,y) are:

$$(\underline{x},\underline{y}),(\overline{x},\underline{y}),(\underline{x},\overline{y}),(\overline{x},\overline{y})$$

$$(3,5),(4,5),(3,6),(4,6)$$

$$\underline{x} = \text{int}(x) \qquad 3 \qquad (3.2,5.6)$$

$$\underline{y} = \text{int}(y) \qquad 5$$

$$\overline{x} = \underline{x} + 1 \qquad 4$$

$$\overline{y} = \underline{y} + 1 \qquad 6$$

X (3,6) X (4,6)

X(3,5) X (4,5)

Bi-linear Interpolation

$$f(x,y) = \overline{\varepsilon_x}\,\overline{\varepsilon_y}\, f(\underline{x},\underline{y}) + \underline{\varepsilon_x}\,\overline{\varepsilon_y}\, f(\overline{x},\underline{y}) +$$
$$\overline{\varepsilon_x}\,\underline{\varepsilon_y}\, f(\underline{x},\overline{y}) + \underline{\varepsilon_x}\,\underline{\varepsilon_y}\, f(\overline{x},\overline{y})$$

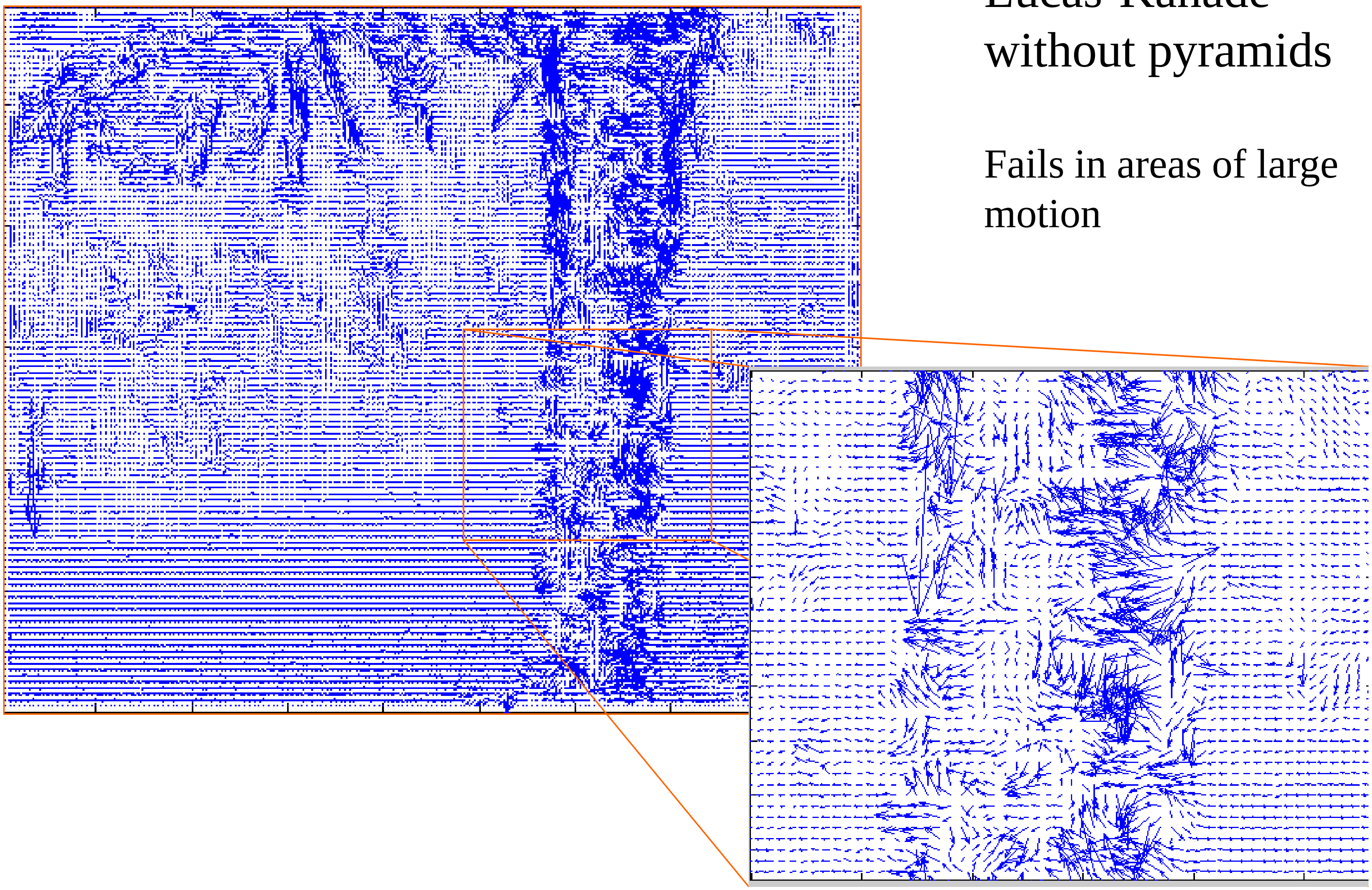
Homework

$$\overline{\varepsilon_x} = \overline{x} - x$$
$$\overline{\varepsilon_y} = \overline{y} - y$$
$$\underline{\varepsilon_x} = x - \underline{x}$$
$$\underline{\varepsilon_y} = y - \underline{y}$$

$$\overline{\varepsilon_x} = \overline{x} - x = 4 - 3.2 = .8$$
$$\overline{\varepsilon_y} = \overline{y} - y = 6 - 5.6 = .4$$
$$\underline{\varepsilon_x} = x - \underline{x} = 3.2 - 2 = .2$$
$$\underline{\varepsilon_y} = y - \underline{y} = 5.6 - 5 = .6$$

Lucas-Kanade without pyramids

Fails in areas of large motion



Lucas-Kanade with Pyramids