# Topic 14: Optical Flow

Dr. V Masilamani

masila@iiitdm.ac.in

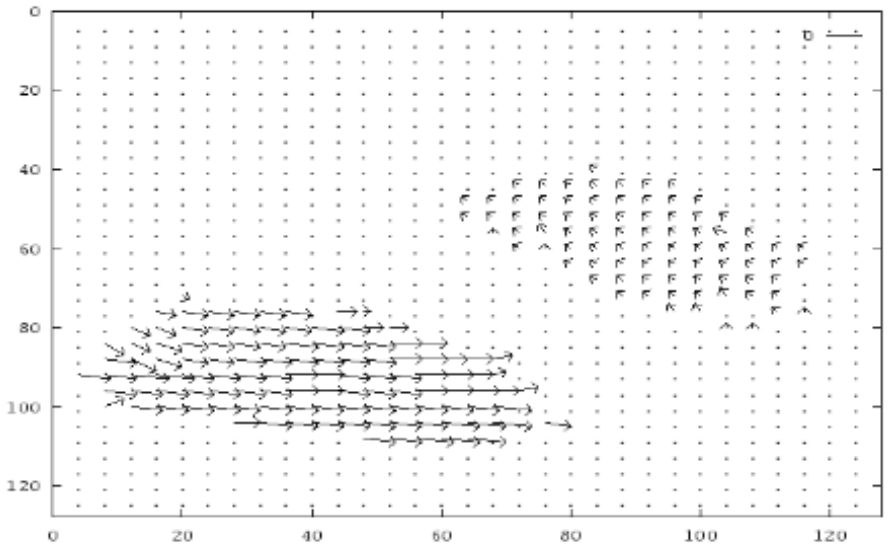
Department of Computer Science and Engineering
IIITDM Kancheepuram
Chennai-127

# Overview

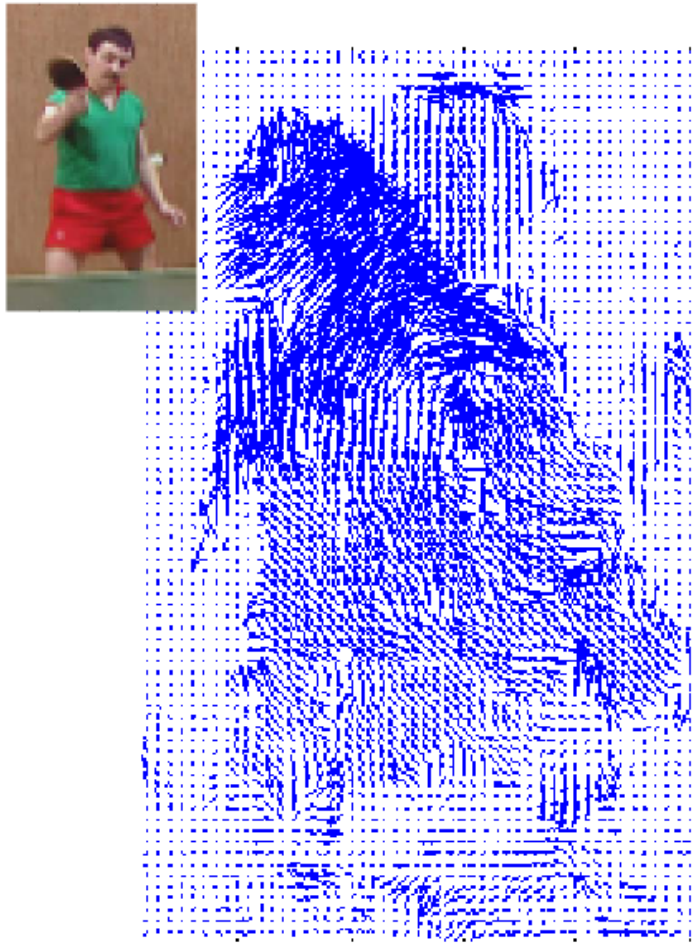
# Optical Flow

- Motion vector (u,v)
  - Image Displacement in x and y directions between two consecutive frames
  - In other words, rate of change of displacement with respect to time
- $u = dx/dt$ , $v = dy/dt$, where $dx$, $dy$ and $dt$ are displacement in x, y and t directions respectively

# Hamburg Taxi seq

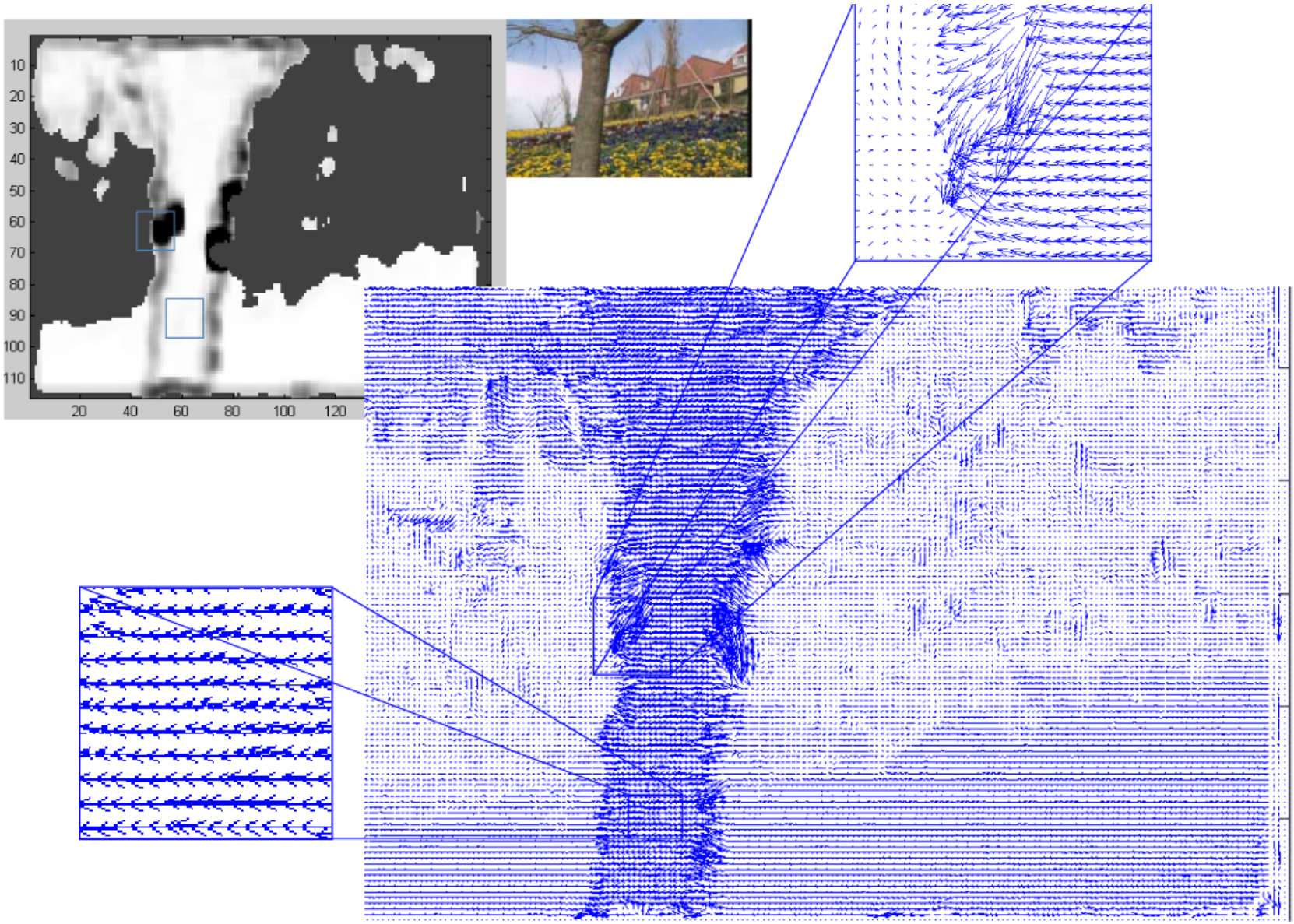


# Examples

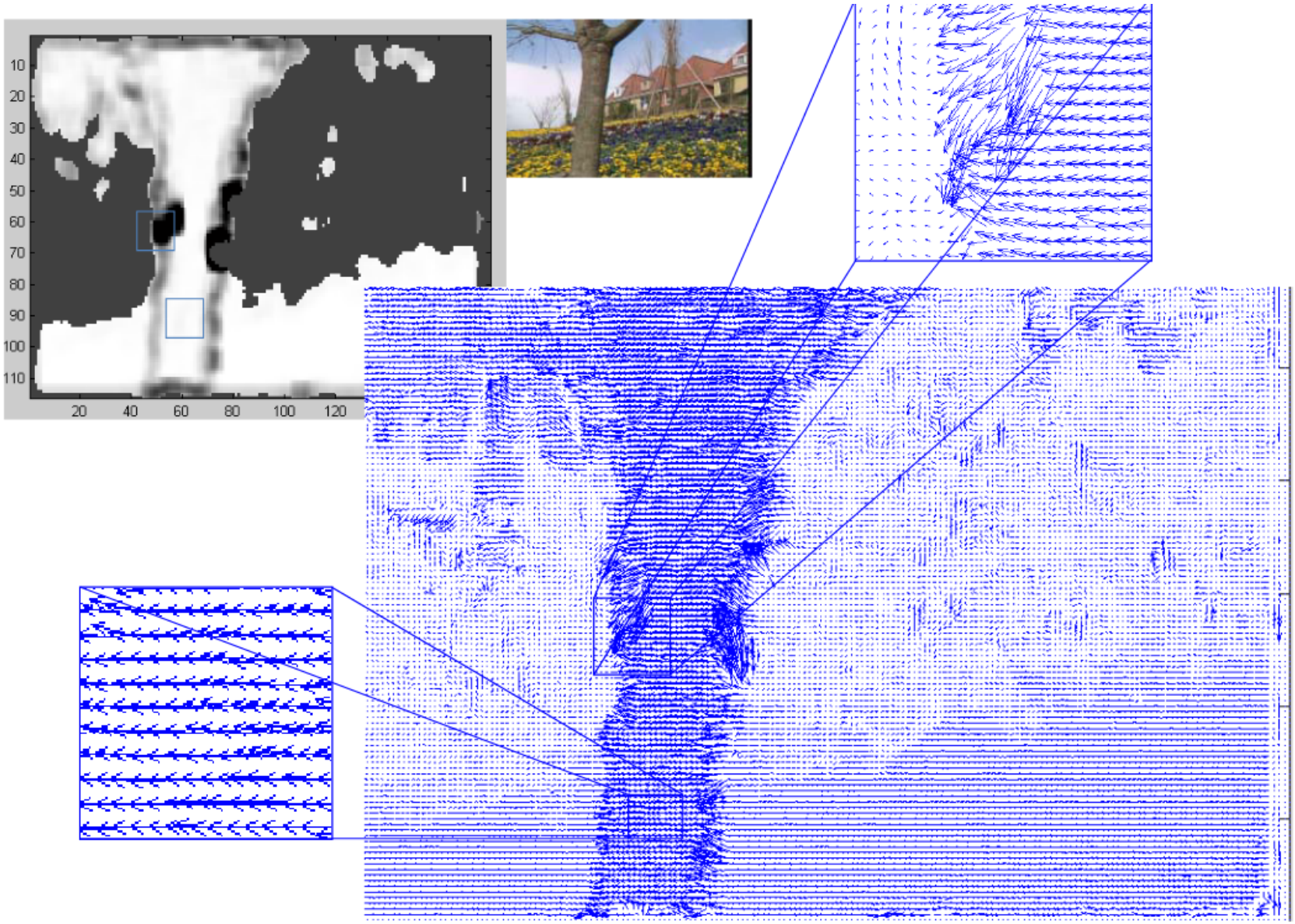


# Examples (cont.)

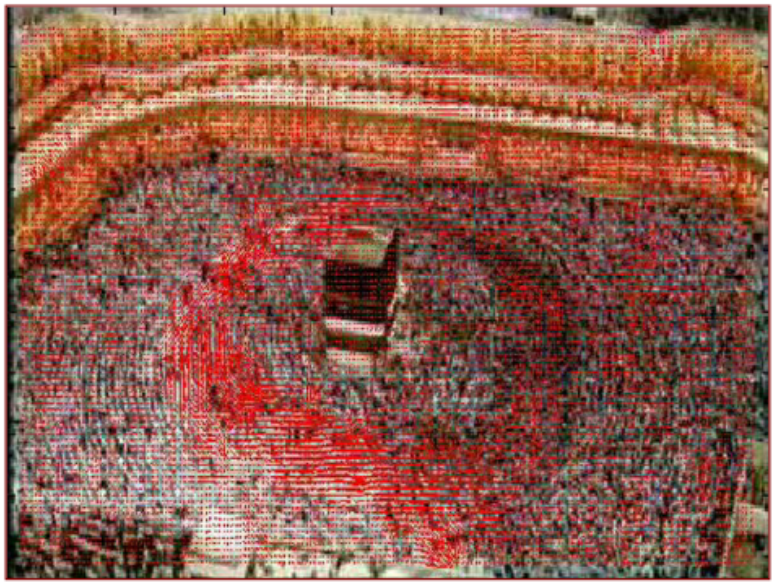
# Examples (cont.)



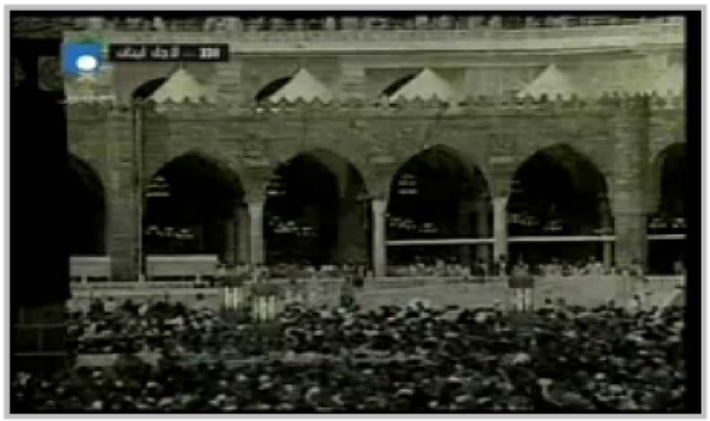
# Examples (cont.)

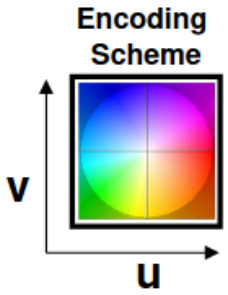


# Examples (cont.)



# Examples (cont.)

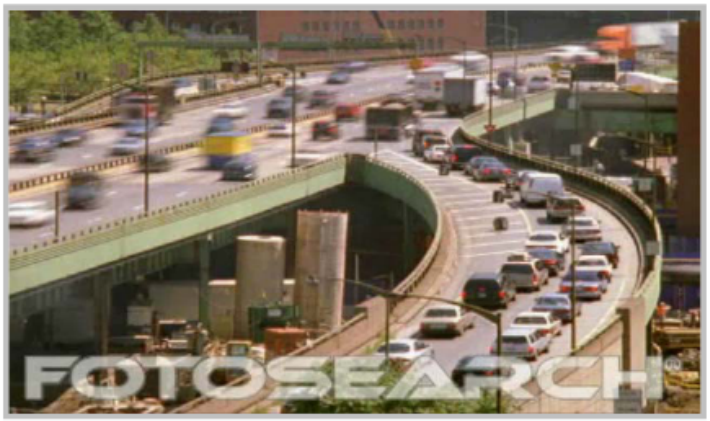


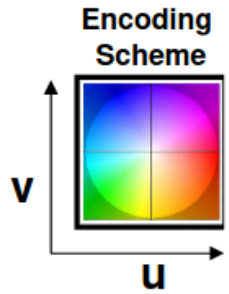



Videos

Color Coded Optical Flows

# Examples (cont.)

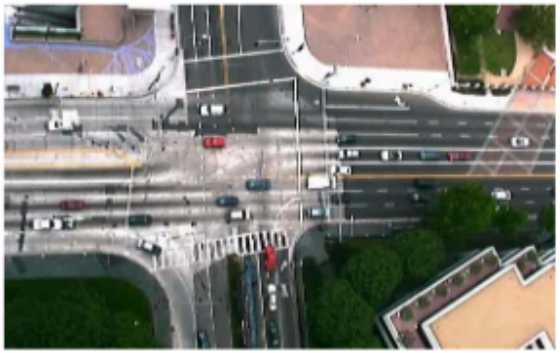


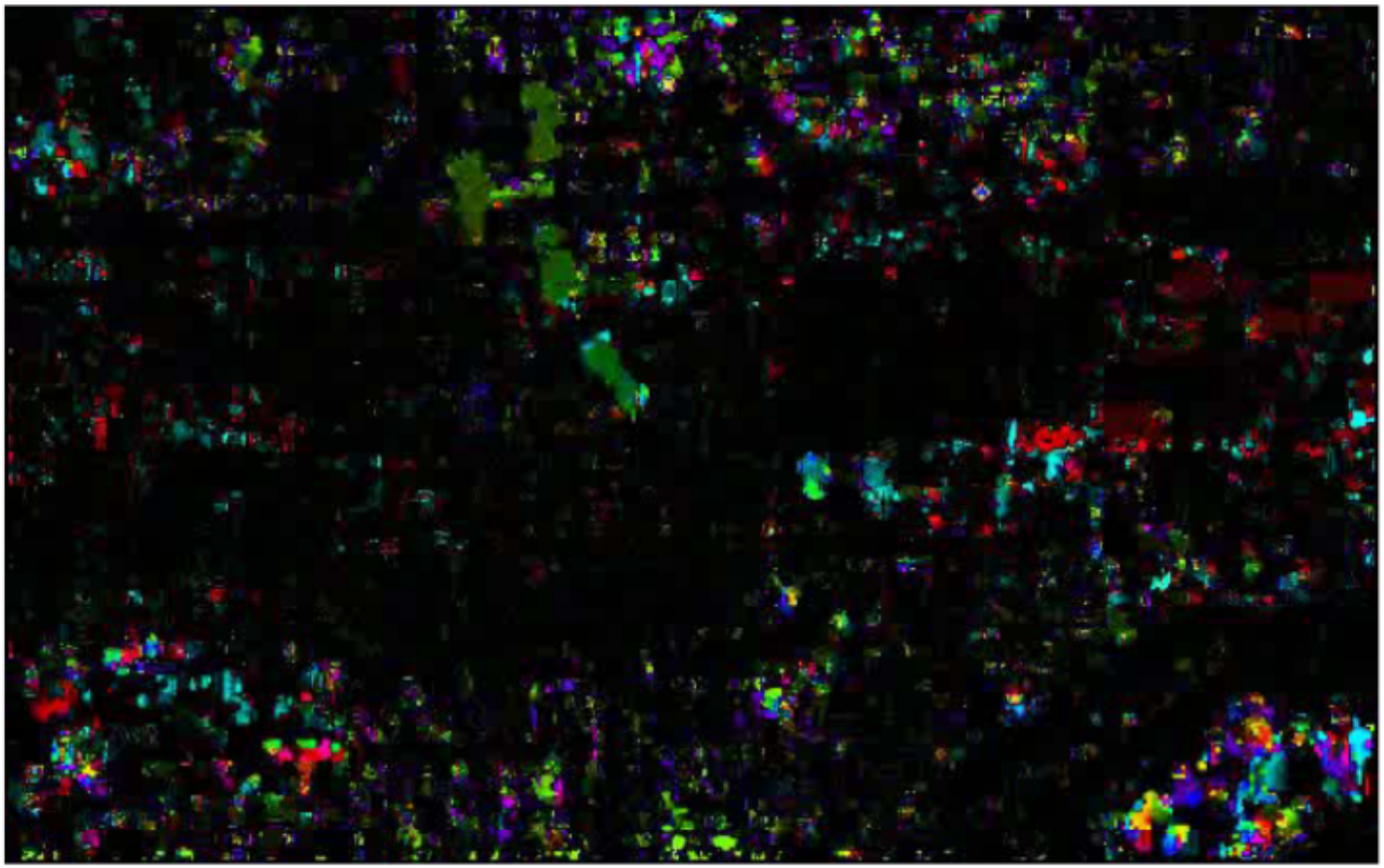




**Videos** **Color Coded Optical Flows**

# Examples (cont.)

Optical Flow





# Optical Flow- Applications

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

# Optical Flow Equation

- To derive optical flow equation the following condition, called Brightness constancy constraint, is assumed
  - $$f(x, y, t) = f(x + dx, y + dy, t + dt) \quad (1)$$
- Apply Taylor series for RHS of above equation
  Eqn (1) becomes
  $f(x, y, t) = f(x, y, t) + \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial t} dt$

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

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

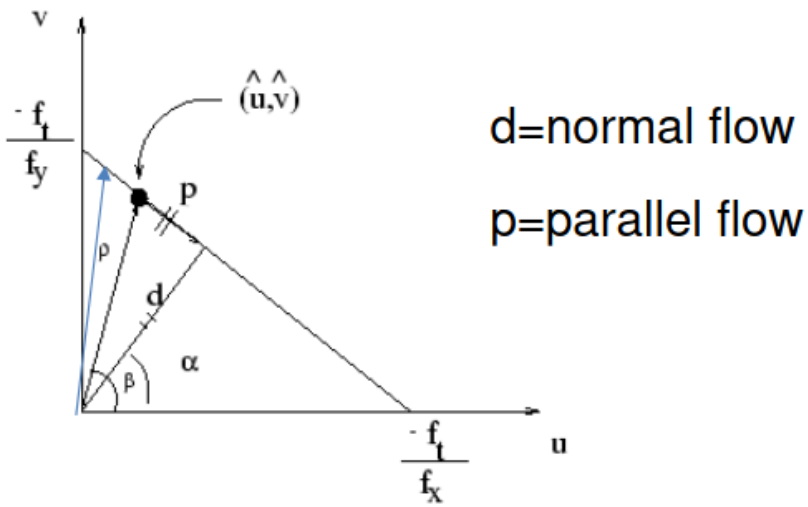
  $0 = f_x u + f_y v + f_t$

# Optical Flow Equation (cont.)

Interpretation of optical flow eq

$f_x u + f_y v + f_t = 0$

$v = -\frac{f_x}{f_y} u - \frac{f_t}{f_y}$ (eq. of st line)




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

# Optical Flow Equation (cont.)

- The Objective is to solve optical flow equation for $(u, v)$
- For each point $(x, y, t)$, we have one equation with two unknowns(u,v)
- To get two or more equations, each optical flow computation algorithm assumes some constraints
  - Horn & Schunk considers smoothness constraint
  - Lucas & Kenade assumes many of neighbours of $(x, y)$ and $(x, y)$ will have the same optical flow

# Horn & Schunck Optical Flow

- Optimization problem formulation by Horn & Schunck: Find $(u, v)$ such that $(u, v)$ satisfy flow equation and also smoothness(change of u and v is slow ) is satisfied
- In other words, find $(u, v)$ such that $E(u, v) = \int\int (f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) dxdy$ is minimum
  - where $(f_x u + f_y v + f_t)^2$ is brightness constancy
  - $u_x^2 + u_y^2 + v_x^2 + v_y^2$ is Smoothness constraint
- By a theorem in calculus, $E(u, v)$ is min iff Considering L as integrant
  - $$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0 \quad (2)$$
  - $$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0 \quad (3)$$

# Horn & Schunck Optical Flow (cont.)

- Calculate the terms in (2) and (3)
  - $\frac{\partial L}{\partial u} = 2(f_x u + f_y v + f_t) f_x$
  - $\frac{\partial L}{\partial v} = 2(f_x u + f_y v + f_t) f_y$
  - $\frac{\partial L}{\partial u_x} = 2u_x$
  - $\frac{\partial L}{\partial u_y} = 2u_y$
  - $\frac{\partial L}{\partial v_x} = 2v_x$
  - $\frac{\partial L}{\partial v_y} = 2v_y$
- By substituting the values in equations (2) and (3), we get
  - $(f_x u + f_y v + f_t) f_x - \lambda(\Delta^2 u) = 0$
  - $(f_x u + f_y v + f_t) f_y - \lambda(\Delta^2 v) = 0$
- Recall that $\Delta^2 u = u_{xx} + u_{yy}$

# Horn & Schunck Optical Flow (cont.)

- Considering discrete version,
  $\Delta^2 u(x,y) = 4(u_{avg}(x,y) - u(x,y)) \approx (u_{avg}(x,y) - u(x,y))$
  where $u_{avg}$ is the average of 4-neighbours of $(x,y)$
- Therefore those two equations become
  - $$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0 \quad (4)$$
    $$(f_x u + f_y v + f_t) f_y + \lambda(v - v_{av}) = 0 \quad (5)$$
- Solving the equations (4) and (5),
  - $u = u_{avg} - f_x \frac{P}{D}$ and $v = v_{avg} - f_y \frac{P}{D}$, where
    - $P = f_x u_{avg} + f_y v_{avg} + f_t$
    - $D = \lambda + f_x^2 + f_y^2$

# Horn & Schunck Optical Flow (cont.)

**How to find $f_x$, $f_y$, $f_t$ and $u_{avg}$**

- Let $f_1$ and $f_2$ be the current and the next frame
- $f_x = (Conv(f_1, h_1) + Conv(f_2, h_2))/2$ where $h_1 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$ and $h_2 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$
- $f_y = (Conv(f_1, h_3) + Conv(f_2, h_4))/2$ where $h_3 = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$ and $h_4 = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$
- $f_t = (Conv(f_1, h_5) + Conv(f_2, h_6))$ where $h_5 = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$ and $h_6 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

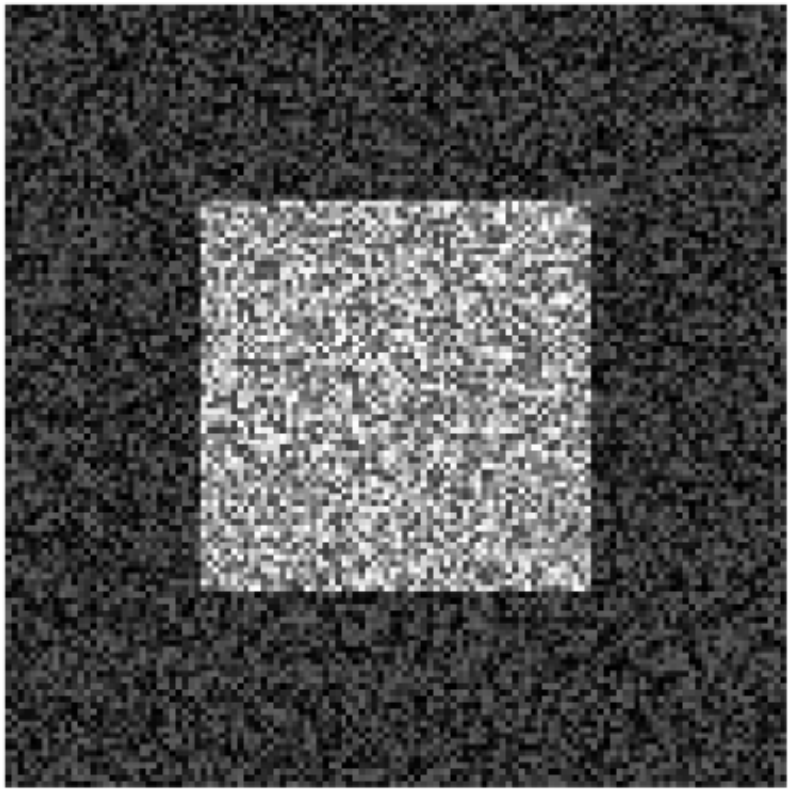
**Horn & Schunck Optical Flow Algorithm**

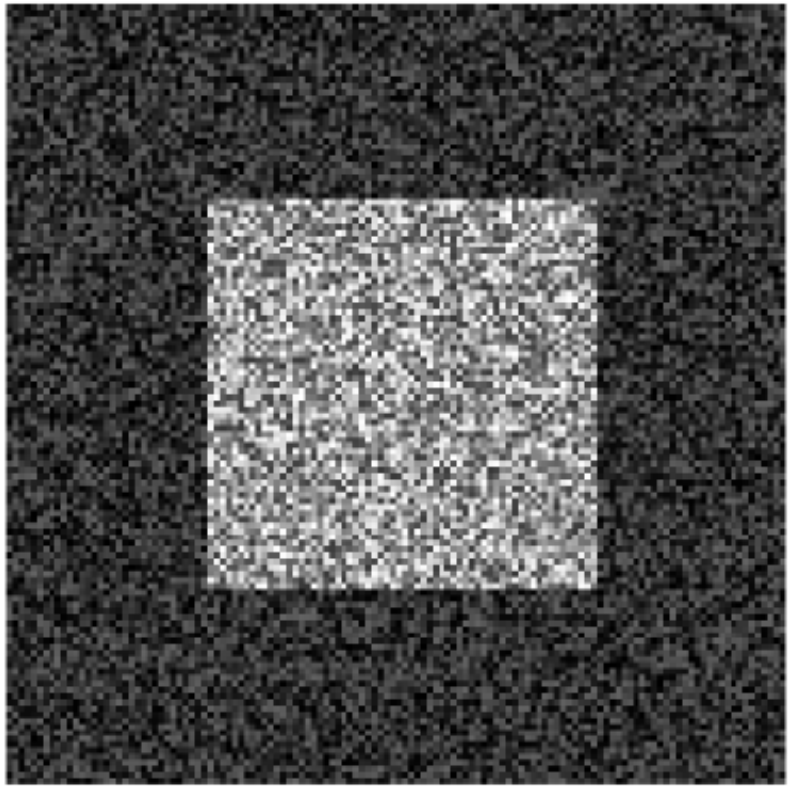
1. Initialize the matrices $u$ and $v$ with zero matrix of size $m \times n$
2. Compute $f_x,\ f_y,\ f_t$
3. Repeat the following until $E(u, v)$ is required minimum
   - $u = u_{avg} - f_x \frac{P}{D}$ and $v = v_{avg} - f_y \frac{P}{D}$
     - where $P = f_x u_{avg} + f_y v_{avg} + f_t$
     - $D = \lambda + f_x^2 + f_y^2$

- Note that the expression for $E(u, v)$ in discrete domain:
  - $E(u, v) = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f_x(x, y)u(x, y) + f_y(x, y)v(x, y) + f_t(x, y))^2 + \lambda(u_x^2(x, y) + u_y^2(x, y) + v_x^2(x, y) + v_y^2(x, y))$

# Horn & Schunck Optical Flow (cont.)

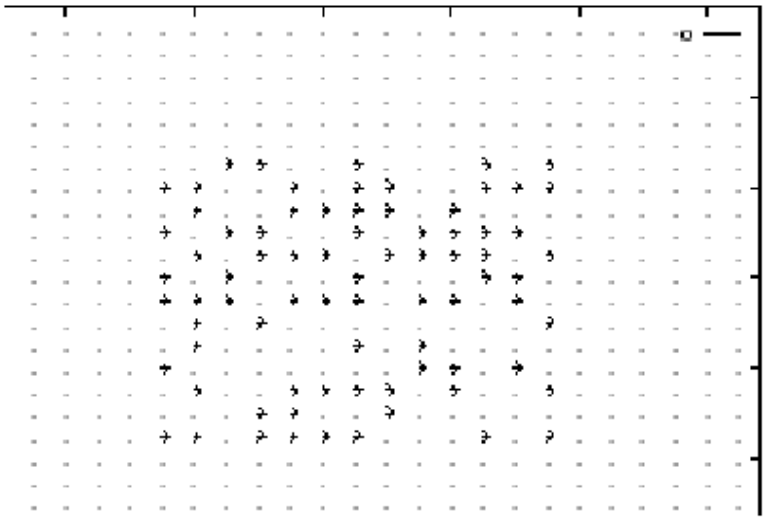
## Synthetic images



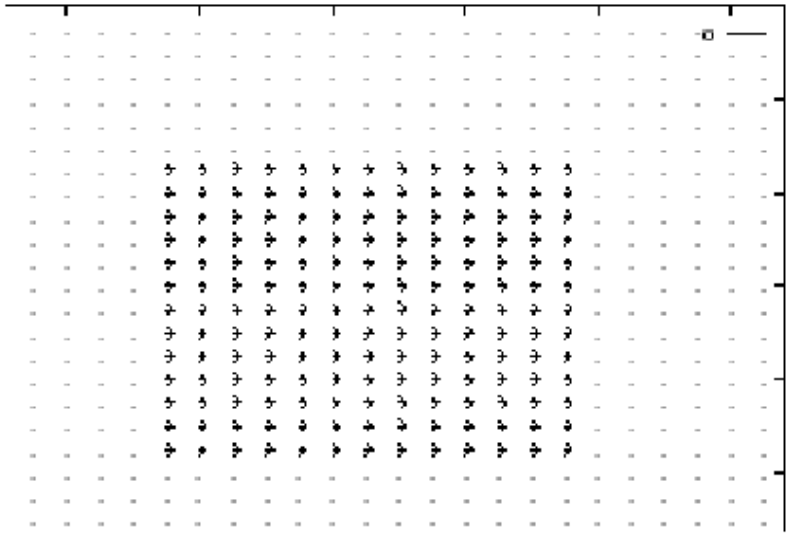


# Horn & Schunck Optical Flow (cont.)

## Horn & Schunck Results



One iteration



10 iterations

## Least Squares

- Optical flow equation:
  - $f_x u + f_y v = -f_t$
- Ill posed Problem: One equation, but two unknowns
- To make it well posed: Include more constraints
- Lucas & Kenade assumes:
  - The optical flow is the same for all the neighbours in $3 \times 3$ window
  - ie. $u(x, y) = u(x + i, y + j)$,
    and $v(x, y) = v(x + i, y + j)$
    where $0 \leq i, j \leq 1$
- For each point $(x + i, y + i)$, where $-1 \leq i, j \leq 1$, one flow equation will be obtained

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

- Hence for 9 points the following 9 equations will be obtained:
  $f_x(x,y)u_0 + f_y(x,y)v_0 = -f_t(x,y)$
  .
  .
  $f_x(x+1,y+1)u_0 + f_y(x+1,y+1)v_0 = -f_t(x+1,y+1)$
- $$\begin{bmatrix} f_x1 & f_y1 \\ . & . \\ . & . \\ f_x9 & f_y9 \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} -f_t1 \\ . \\ . \\ -f_t9 \end{bmatrix}$$
- In the above matrix equation:
  $f_xi$, $f_yi$ for $0 \leq i \leq 9$, represents the coefficient of $u_0, v_0$ respectively in the $i^{th}$ equation

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

**To Solve the matrix equation, pseudo inverse can be used**

- $Au = f_t$
- $A^T Au = A^T f_t$
- $u = (A^T A)^{-1} A^T f_t$

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

**Another method to find (u,v)** Least squares fit: find (u,v) s.t $\sum_i (f_x iu + f_y iv + f_t i)^2$ is min

- Note that the above formulation incorporates
  - Brightness constancy Constraint
  - Optical flow is same for all neighbors

**To solve the optimization problem:**

- $\frac{\partial}{\partial u} \sum_i (f_x iu + f_y iv + f_t i)^2 = 0$
  - $\sum (f_x iu + f_y iv + f_t i) f_x i = 0$
- $\frac{\partial}{\partial v} \sum_i (f_{xi} u + f_{yi} v + f_t)^2 = 0$
  - $\sum (f_{xi} u + f_{yi} v + f_t) f_{yi} = 0$

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

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

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

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

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

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

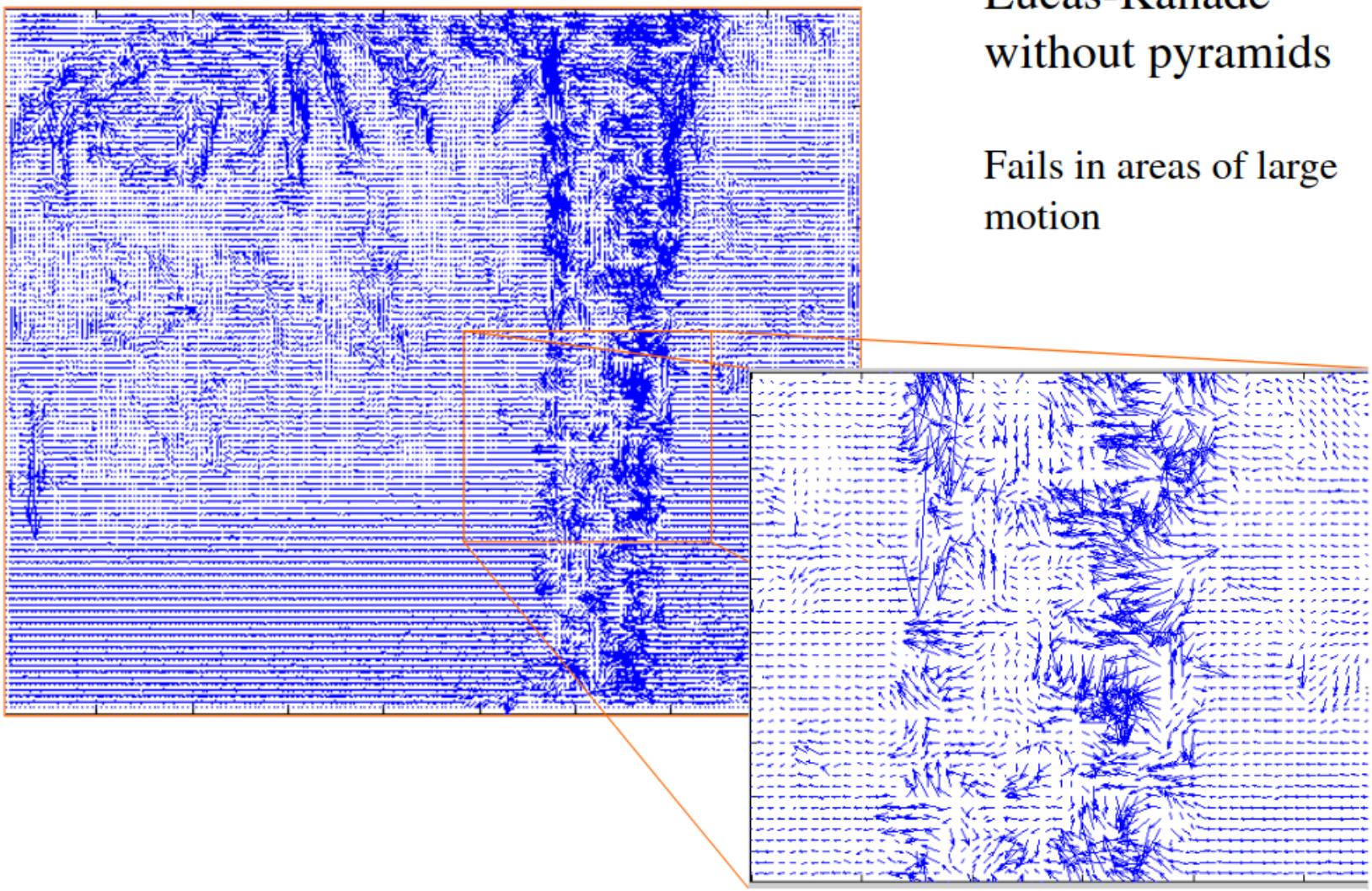
- $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{vi} \\ \sum f_{xi} f_{vi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$
- $\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum f_{xi}^2 \sum f_{yi}^2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

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)

Lucas & Kenade Algorithm:

- Find $fx, f_y, f_t$ for all points in a window
- Find $(u, v)$, where
  - $u = \frac{-\sum f_{yi}^2 \sum f_{xi} f_{ti} + \sum f_{xi} f_{yi} \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$
  - $v = \frac{\sum f_{xi} f_{ti} \sum f_{xi} f_{yi} - \sum f_{xi}^2 \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$

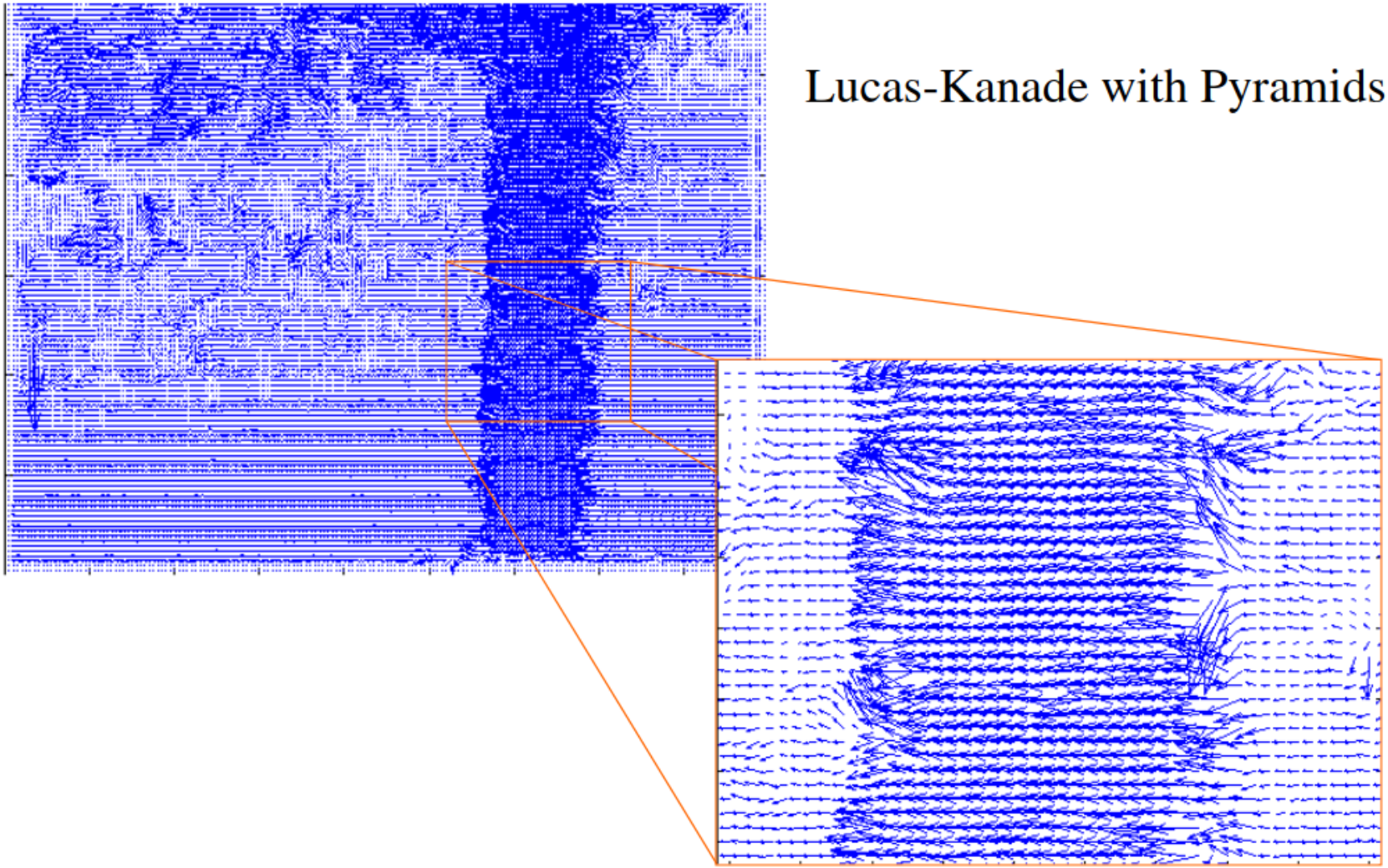
# Lucas & Kanade Optical Flow Computation Algorithm (cont.)



Lucas-Kanade without pyramids

Fails in areas of large motion

# Lucas & Kanade Optical Flow Computation Algorithm (cont.)



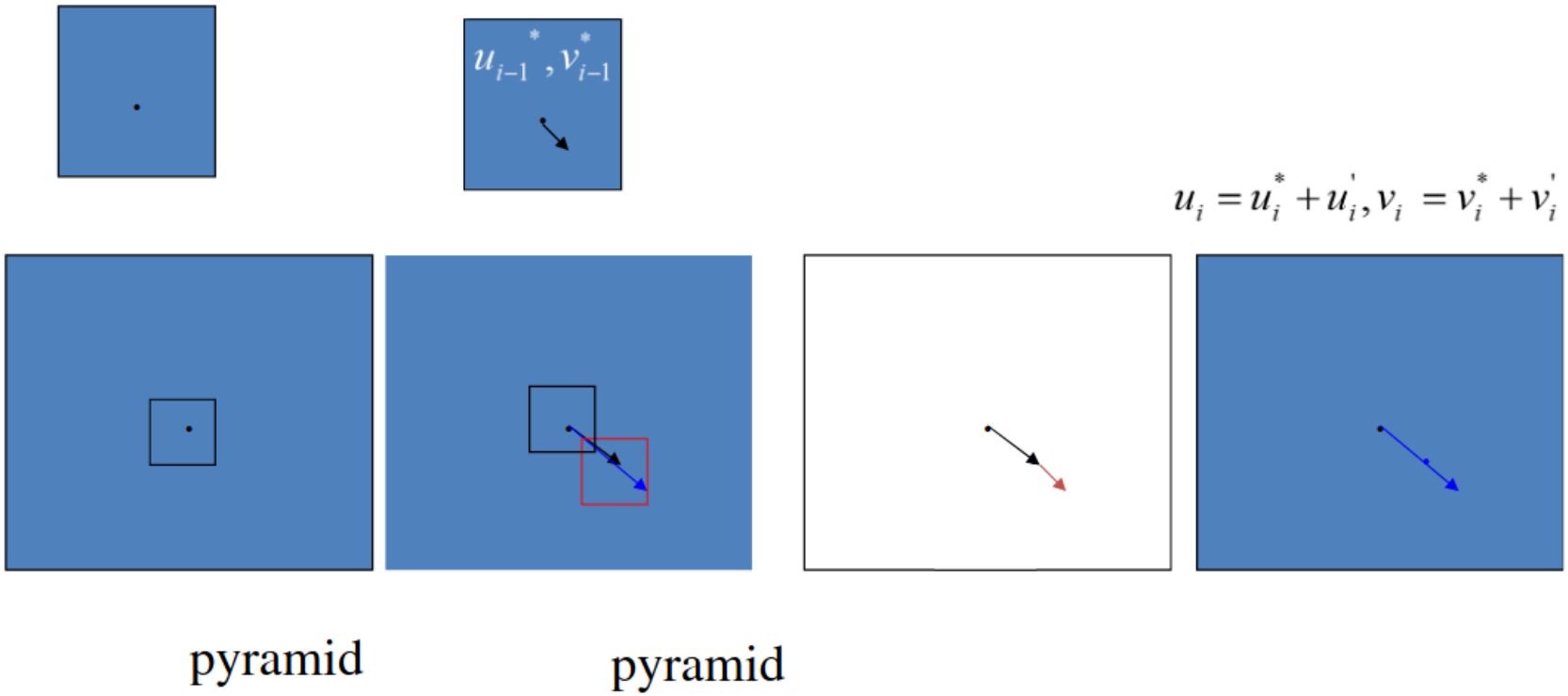
# Comments

- 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

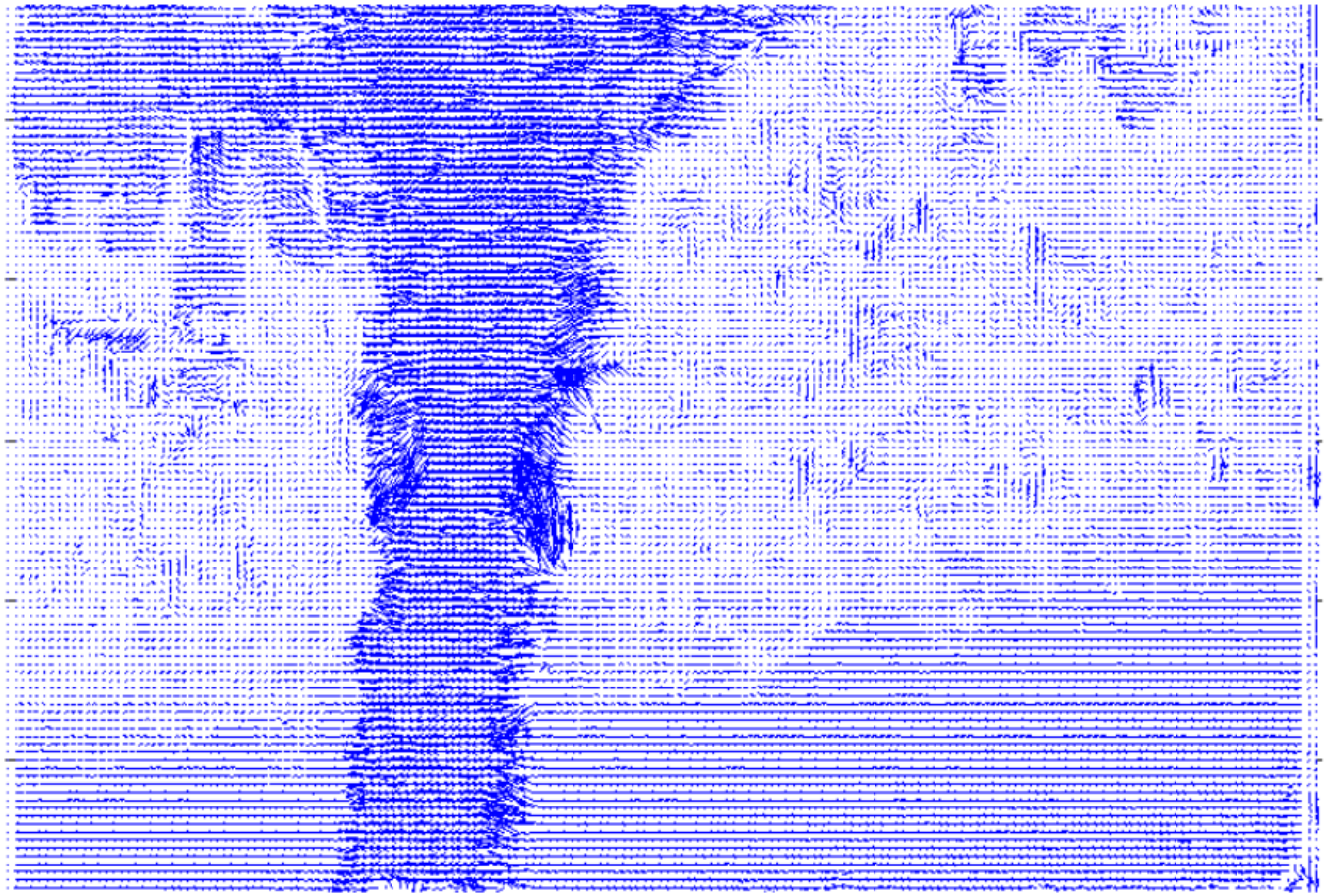
# Lucas Kanade with Pyramids

- 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. $u_i = u_i^* + u_i^{'}, v_i = v_i^* + v_i^{'}$

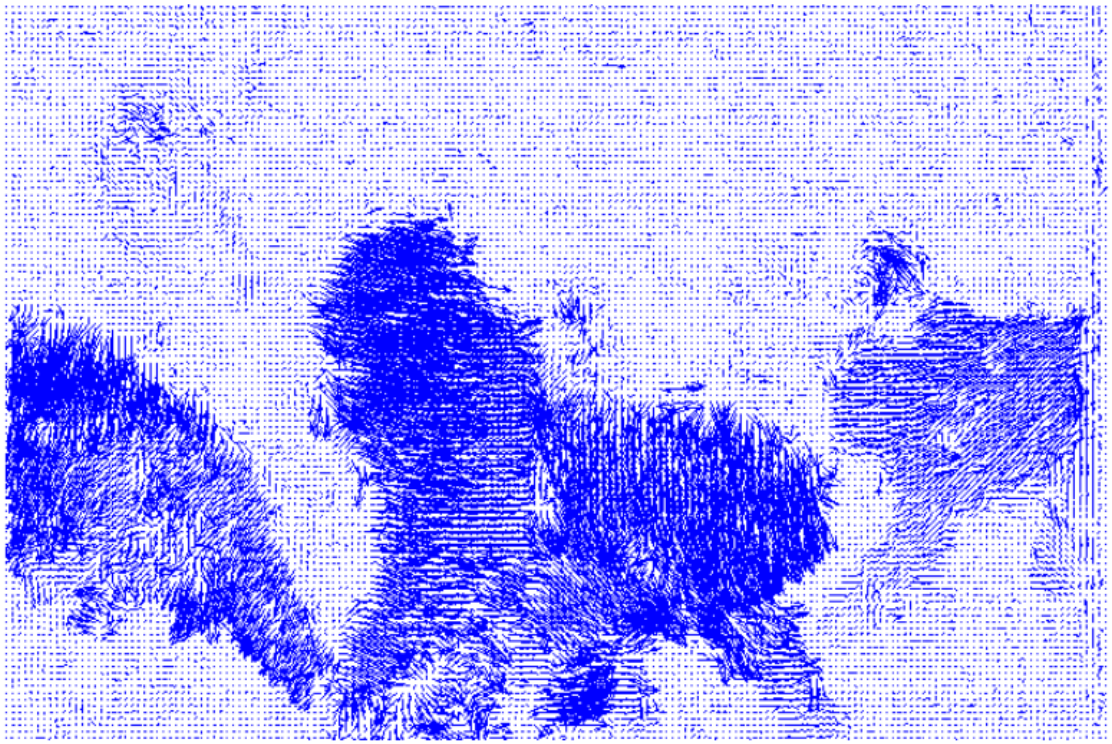
# Pyramids



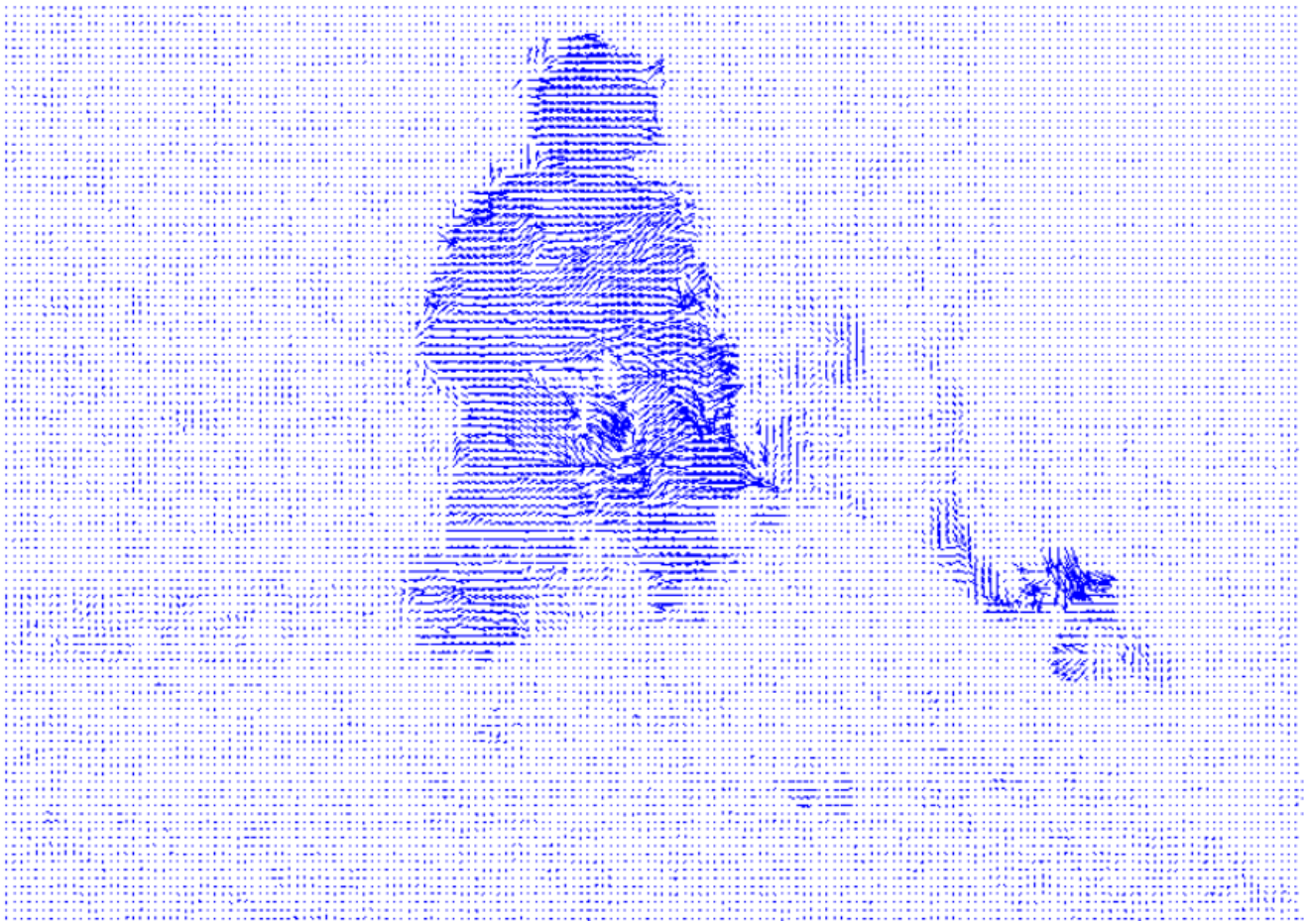
# Examples



# Examples (cont.)



# Examples (cont.)



# Acknowledgements


- The slides are taken from the the lectures by Mubarak shah and other internet sources