Motion and Optical Flow

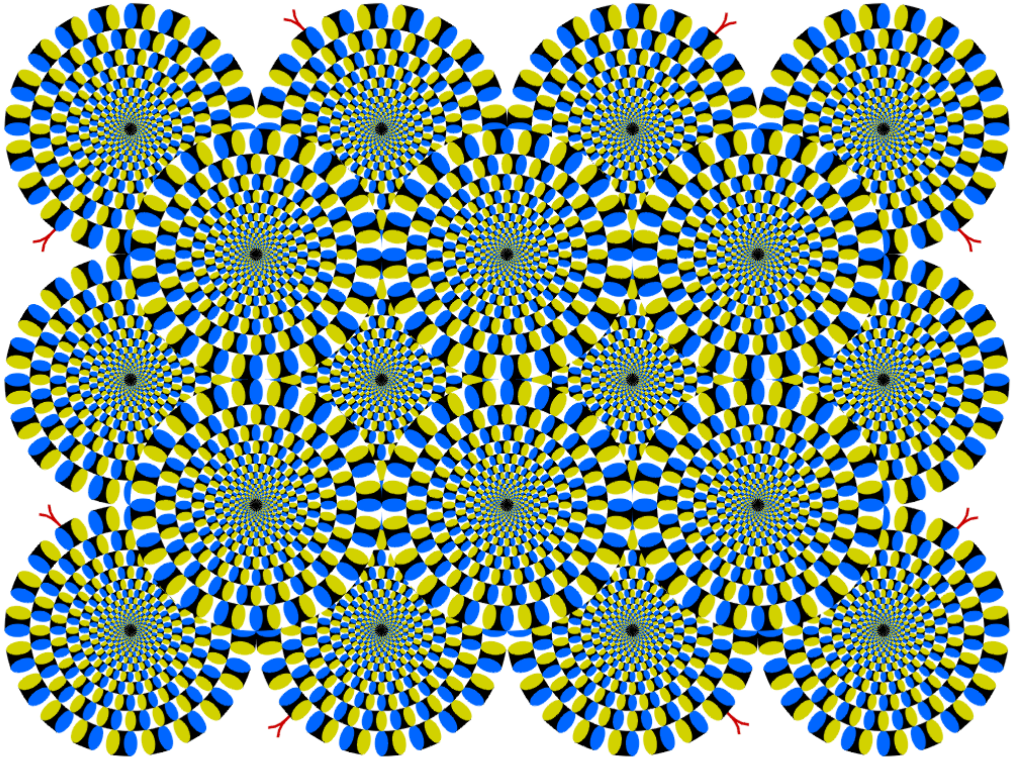
Morteza Rezanejad

(Slides are borrowed from Ali Farhadi)

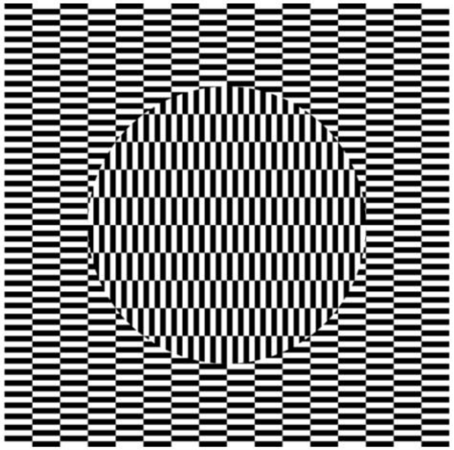
We live in a moving world

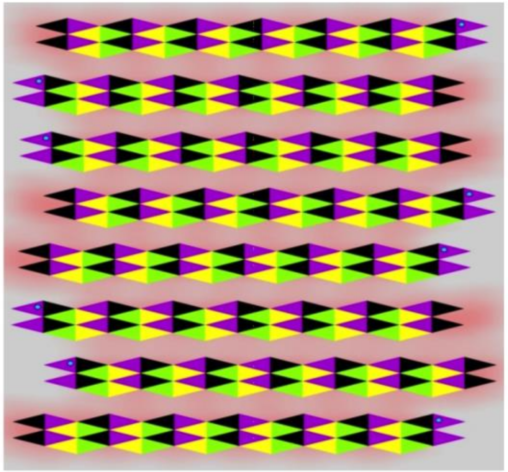
- Perceiving, understanding and predicting motion is an important part of our daily lives

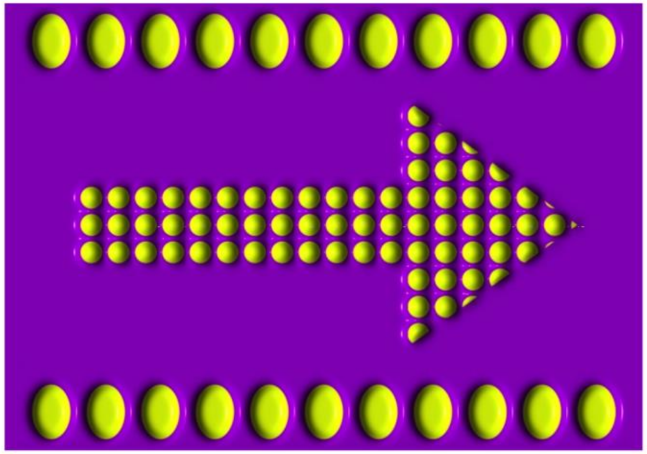
Seeing motion from a static picture?

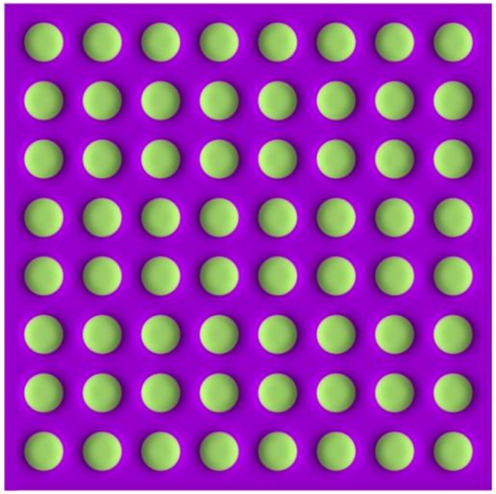


More examples









Motion scenarios (priors)

Static camera, moving scene

Moving camera, static scene

Moving camera, moving scene

Static camera, moving scene, moving light

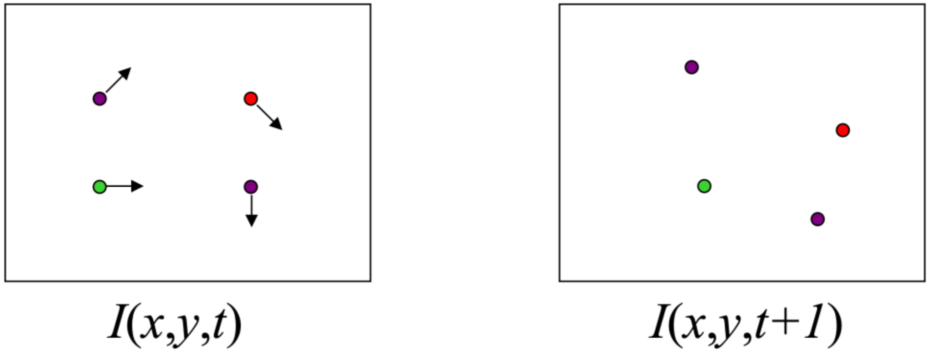
How can we recover motion?

- Extract visual features (corners, textured areas) and "track" them over multiple frames.

- Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow).

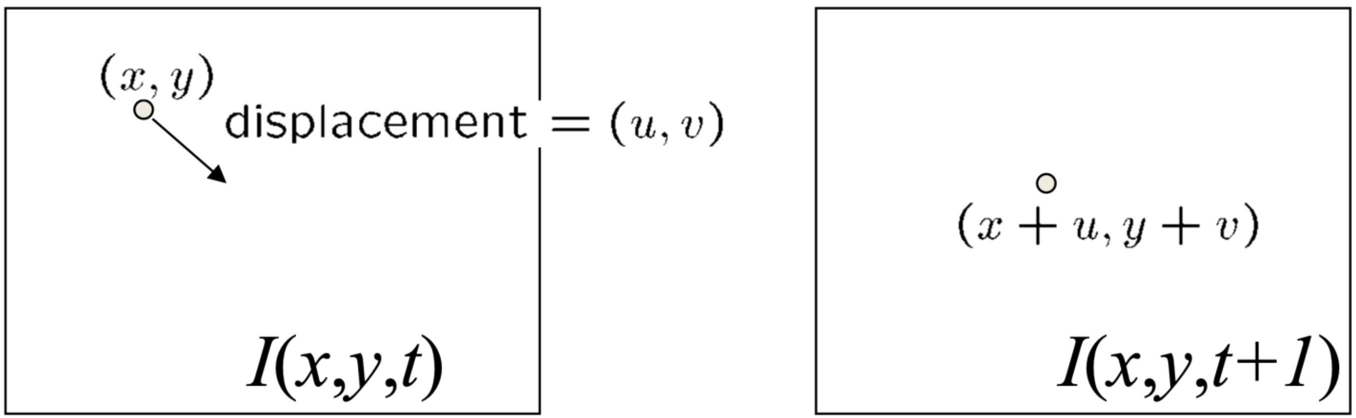
B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679, 1981.

Feature tracking



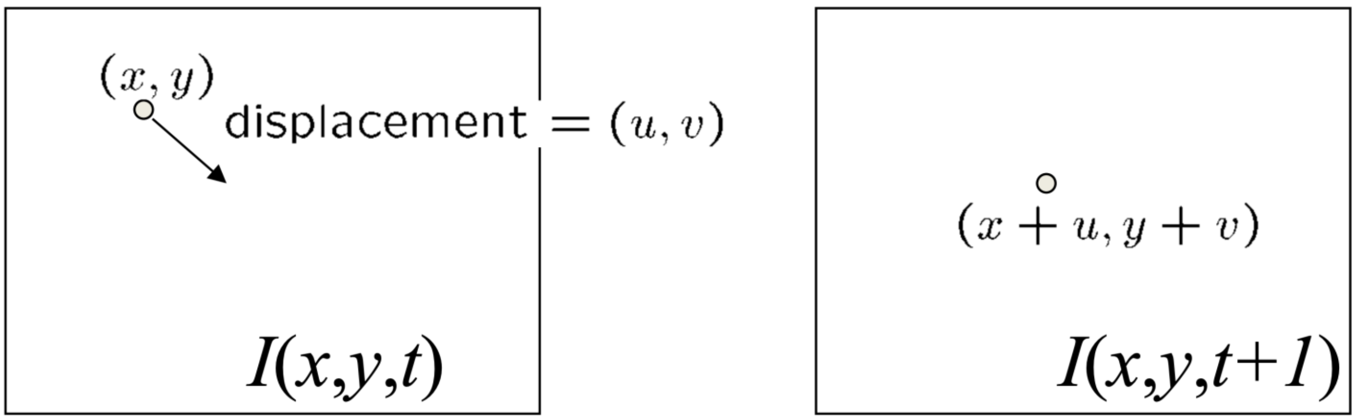
- Given two subsequent frames, estimate the point translation
- Key assumptions:
  - Brightness constancy: projection of the same point looks the same in every frame
  - Small motion: points do not move very far
  - Spatial coherence: points move like their neighbors

The brightness constancy constraint



- Brightness Constancy Equation: $I(x, y, t) = I(x + u\ , y + v, t + 1)$
- Now, take the Taylor expansion of $I(x + u\ , y + v, t + 1)$ at (x, y, t) to linearize the right side

The brightness constancy constraint



$$I(x + u\ , y + v, t + 1)\ \approx I(x, y, t) + I_x . u + I_y . v + I_t$$

$$I(x + u\ , y + v, t + 1) - I(x, y, t) \approx +I_x . u + I_y . v + I_t = \nabla I\ . \begin{bmatrix} u \\ v \end{bmatrix} + I_t$$

Brightness Constancy Equation: $I(x, y, t) = I(x + u\ , y + v, t + 1)$

$$\nabla I\ . \begin{bmatrix} u \\ v \end{bmatrix} + I_t = 0$$

The brightness constancy constraint

- Can we use this equation to recover image motion (u,v) at each pixel?

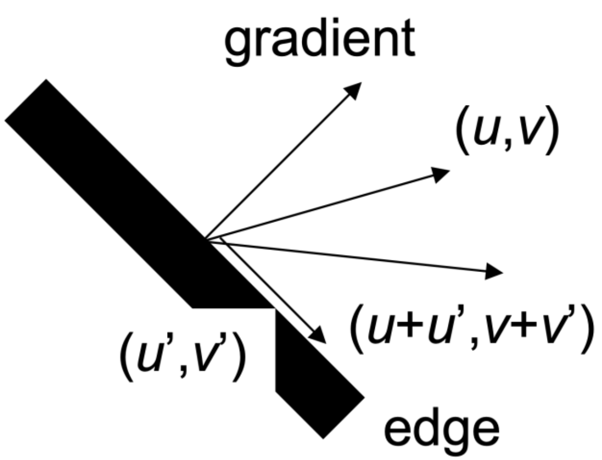
$$\nabla I \,.\begin{bmatrix} u \\ v \end{bmatrix} + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

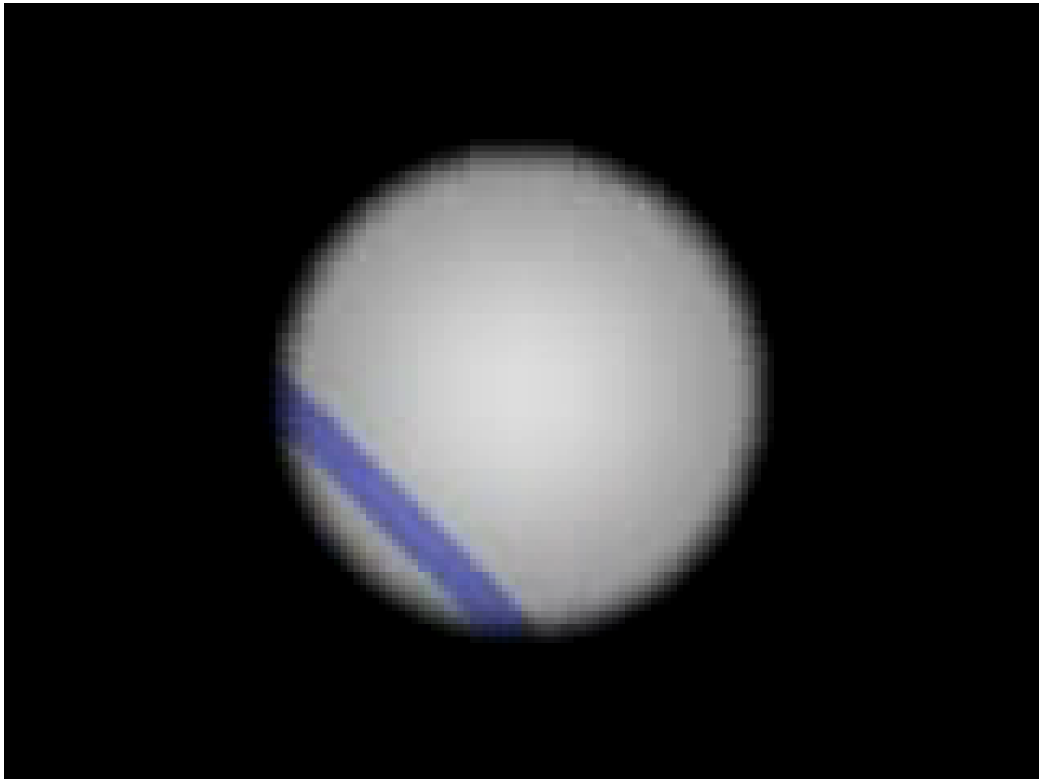
The brightness constancy constraint

- The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured.
  - If (u, v) satisfies the equation, so does $(u + u', v + v')$ if

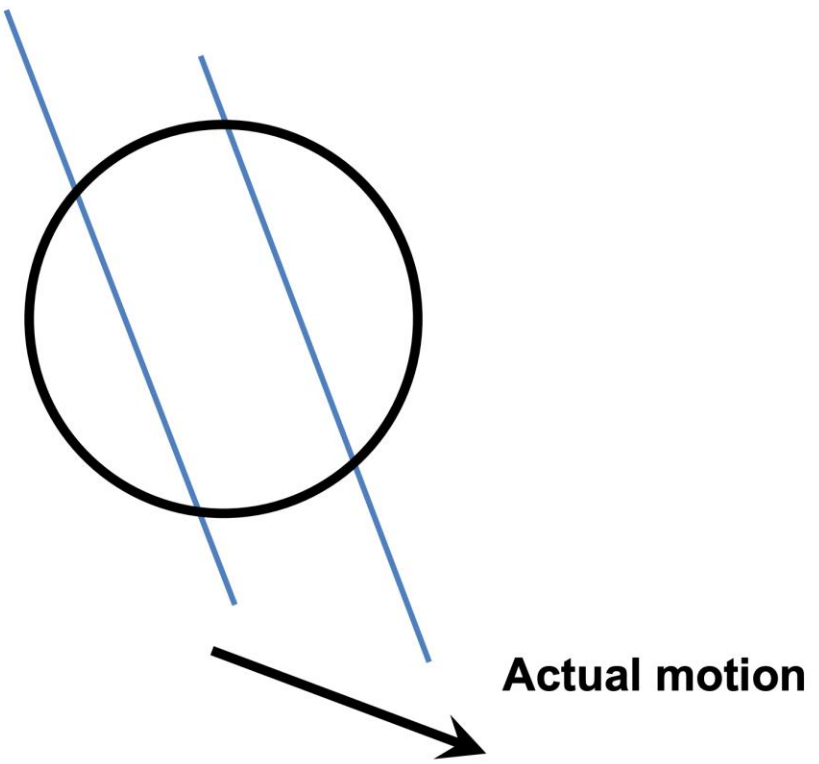
$$\nabla I \,.\begin{bmatrix} u' \\ v' \end{bmatrix} = 0$$



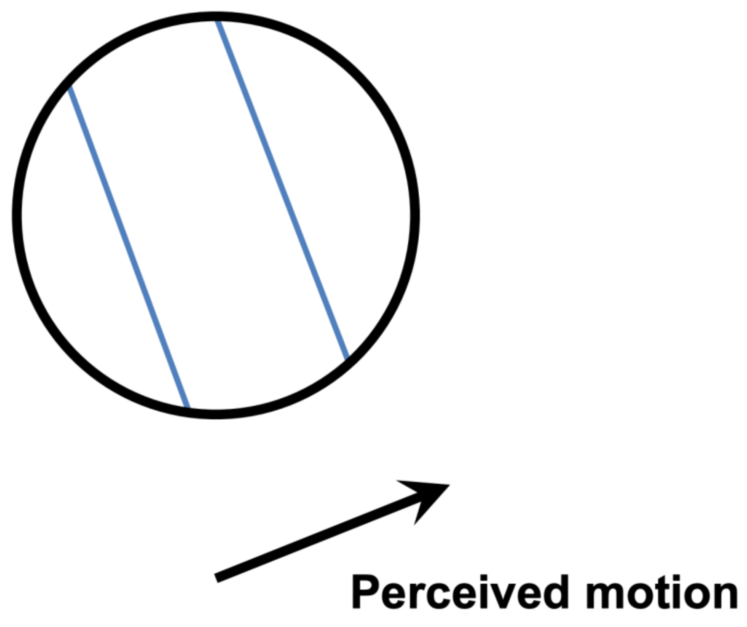
The aperture problem



The aperture problem



The aperture problem



The barber pole illusion

Solving the ambiguity...

- How to get more equations for a pixel?
- Spatial coherence constraint
- Assume the pixel's neighbors have the same (u, v)
  - If we use a 5x5 window, that gives us 25 equations per pixel
  - For $\forall_{p_i} : \nabla I(p_i) . \begin{bmatrix} u \\ v \end{bmatrix} + I_t(p_i) = 0$

Solving the ambiguity...

$$\begin{pmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{pmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{pmatrix} = 0$$

$$\underbrace{\begin{pmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{pmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d} = \underbrace{-\begin{pmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{pmatrix}}_{b}$$

$$A\,d = \ b$$

Solving the ambiguity...

- Least squares solution for d given by

$$A^T A d = A^T b$$

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

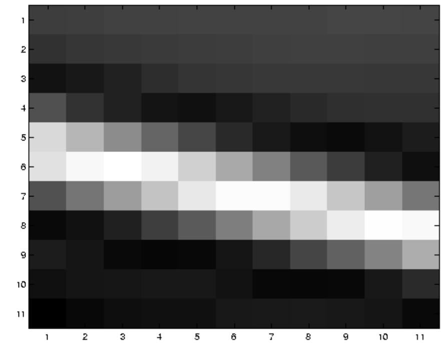
- The summations are over all pixels in the K x K window

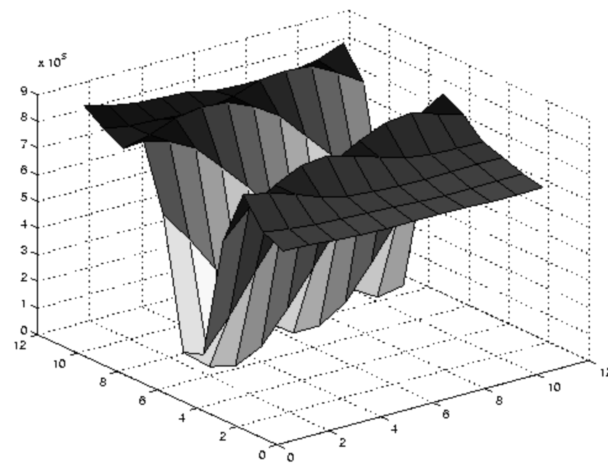
Conditions for solvability

- Optimal (u, v) satisfies Lucas-Kanade equation

- When is this solvable? I.e., what are good points to track?
  - $A^T A$ should be invertible
  - $A^T A$ should not be too small due to noise
    - eigenvalues λ1 and λ 2 of $A^T A$ should not be too small
  - $A^T A$ should be well-conditioned
    - λ1/λ2 should not be too large(λ1=larger eigenvalue)

Edges cause problems



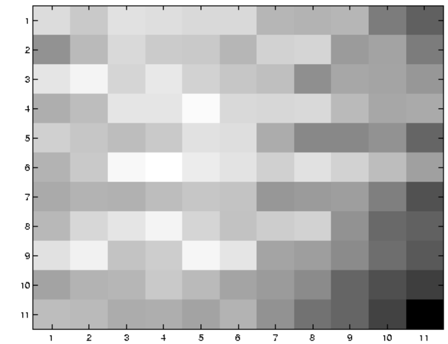


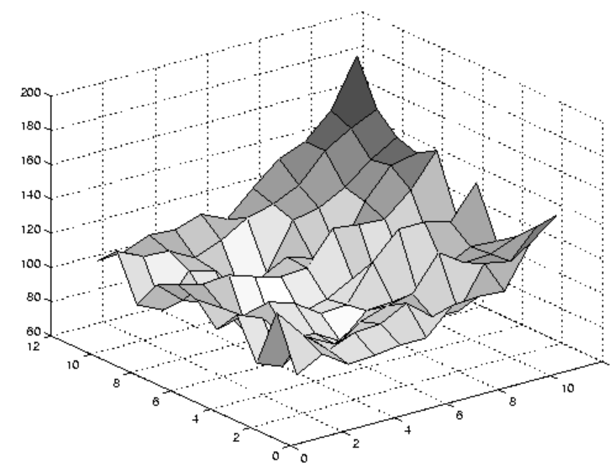
$$\sum \nabla I (\nabla I)^T$$

– large gradients, all the same

– large λ_1, small λ_2

Low texture regions don’t work



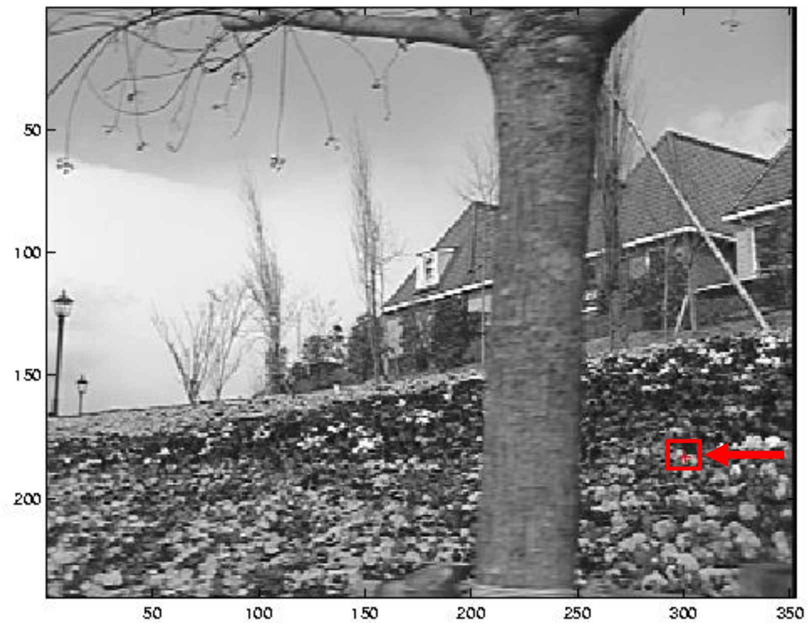


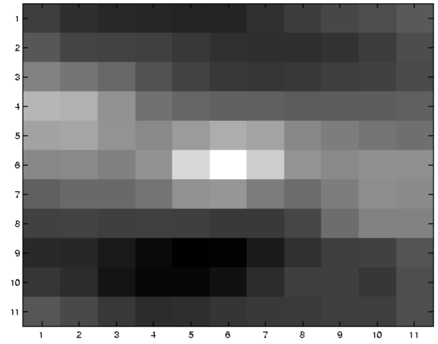
$$\sum \nabla I (\nabla I)^T$$

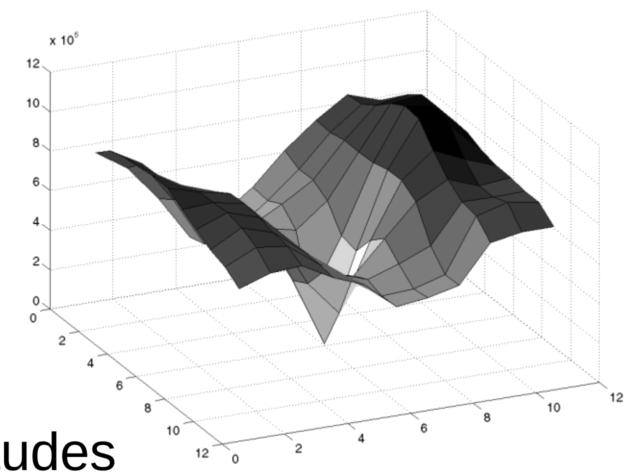
– gradients have small magnitude

– small λ_1, small λ_2

High textured region work best







$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large λ_1, large λ_2

Errors in Lukas-Kanade

- What are the potential causes of errors in this procedure?
  - Suppose A^TA is easily invertible
  - Suppose there is not much 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
  - what is the ideal window size?

Dealing with larger movements: Iterative refinement

Original (x,y) position

$$I_t = I(x', y', t+1) - I(x, y, t)$$

1. Initialize $(x', y') = (x, y)$
2. Compute (u, v) by

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

2nd moment matrix for feature patch in first image

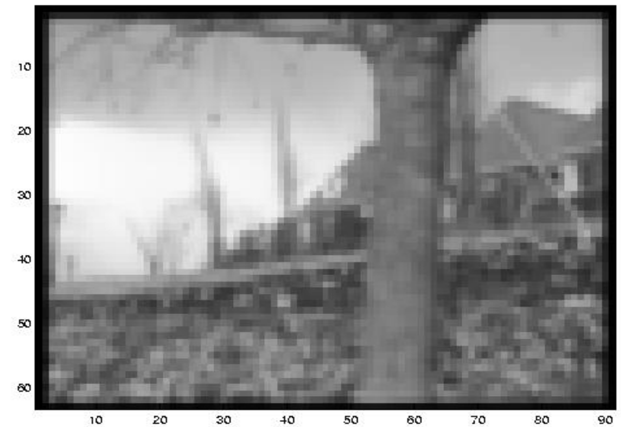
displacement

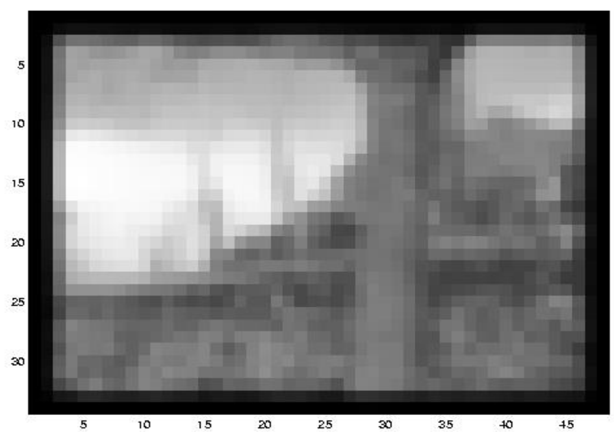
3. Shift window by (u, v): $x' = x' + u$; $y' = y' + v$;
4. Recalculate I_t
5. Repeat steps 2-4 until small change
   - Use interpolation for subpixel values

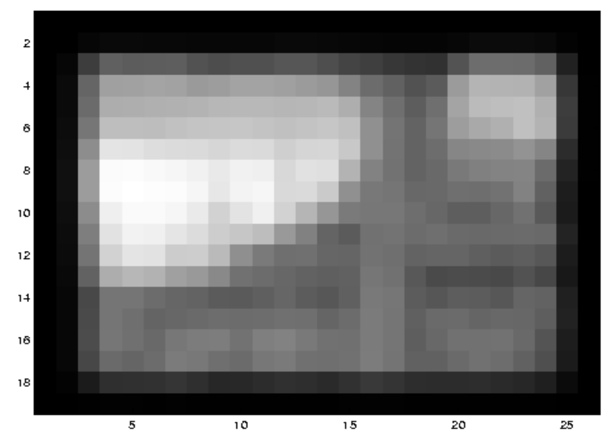
Revisiting the small motion assumption

- Is this motion small enough?
  - Probably not – it's much larger than one pixel (2^{nd} order terms dominate)
  - How might we solve this problem?

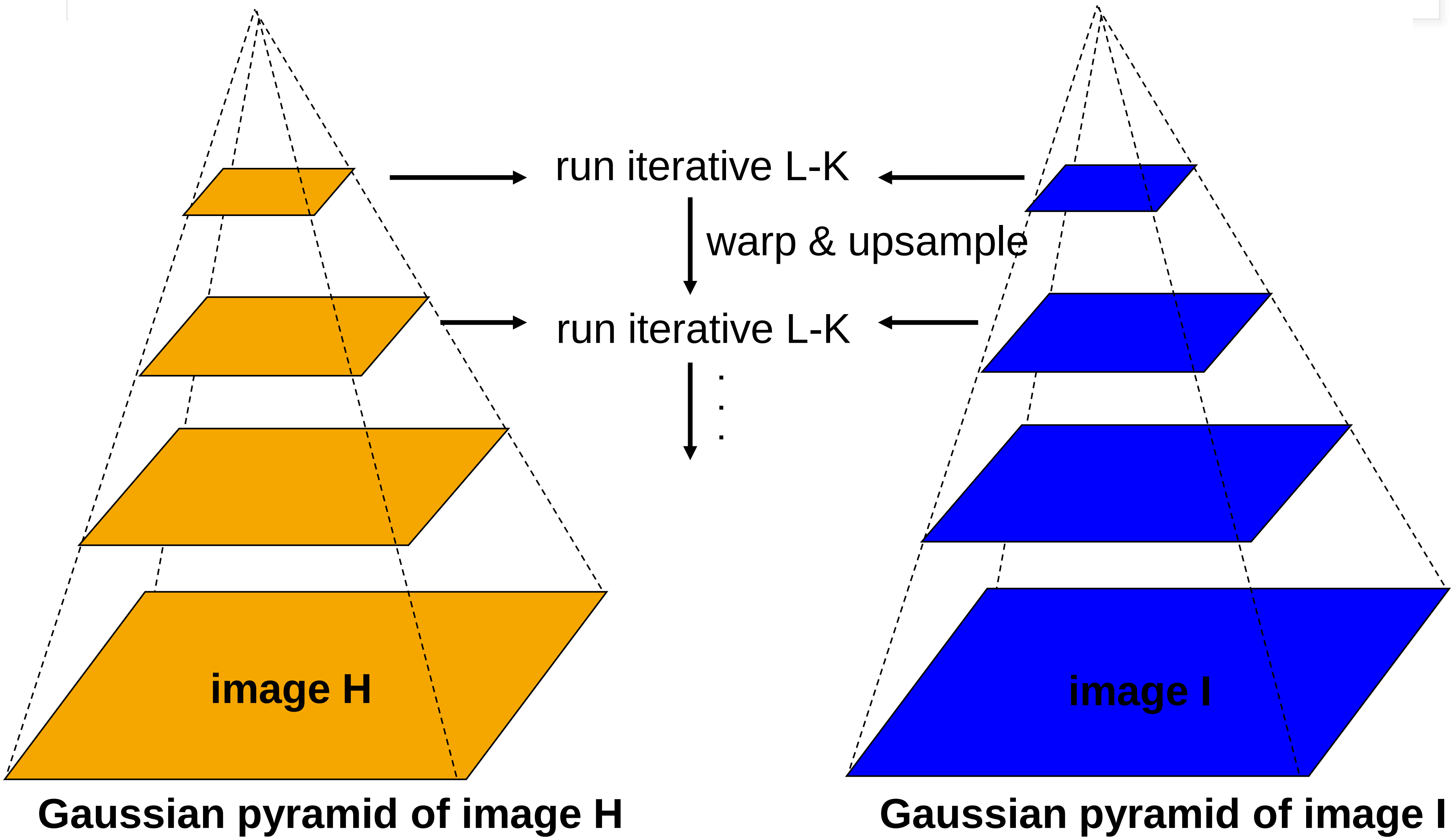
Reduce the resolution!







Coarse-to-fine optical flow estimation

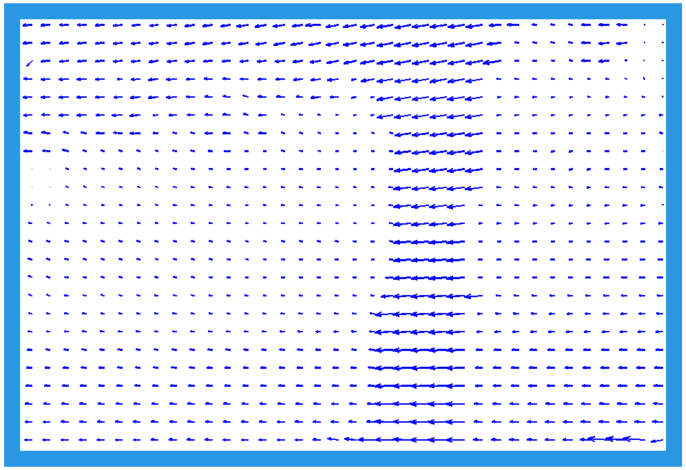


A Few Details

- Top Level
  - Apply L-K to get a flow field representing the flow from the first frame to the second frame.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K on the new warped image to get a flow field from it to the second frame.
  - Repeat till convergence.
- Next Level
  - Upsample the flow field to the next level as the first guess of the flow at that level.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K and warping till convergence as above.
- Etc.

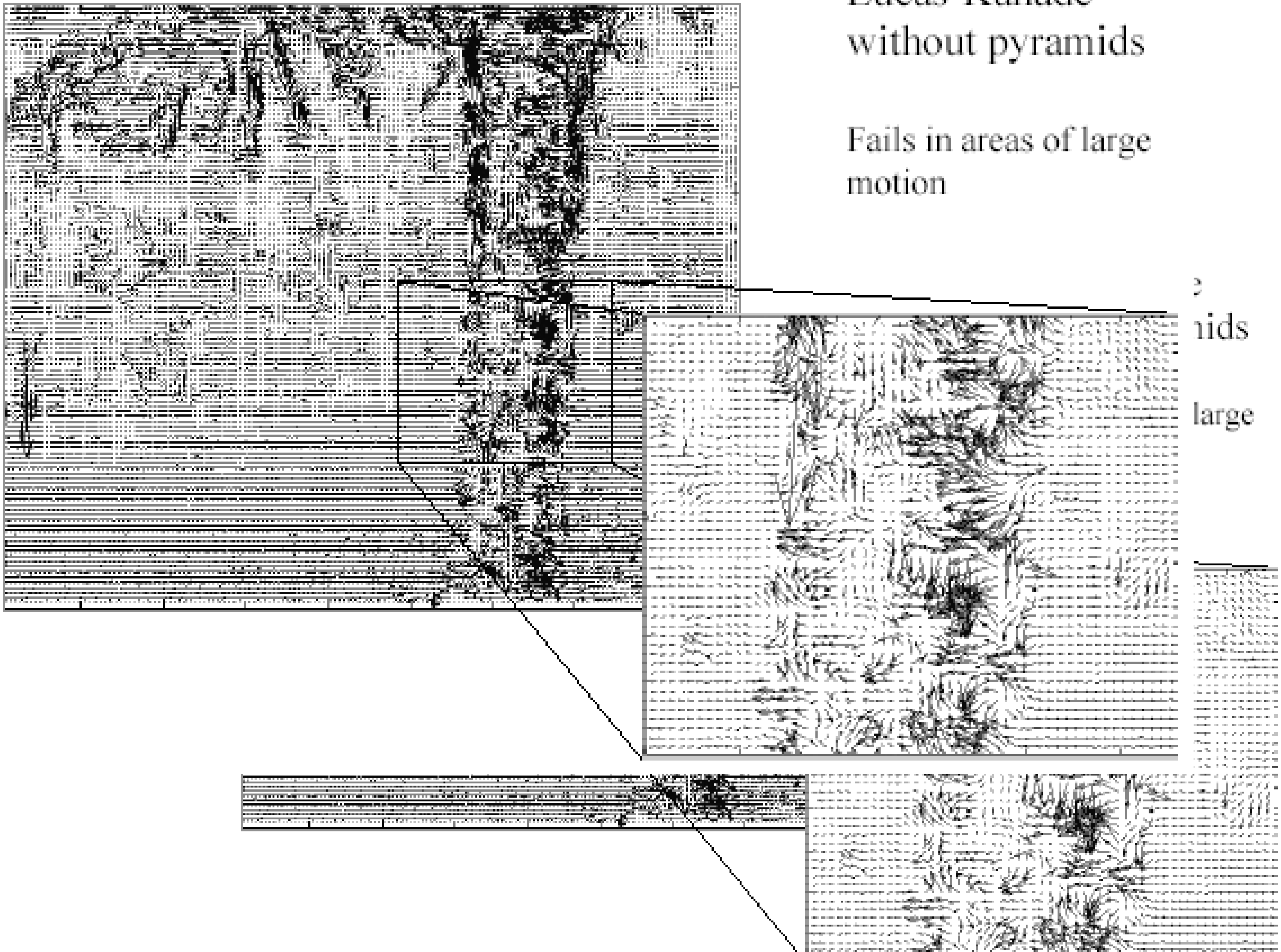
The Flower Garden Video

- What should the
- optical flow be?

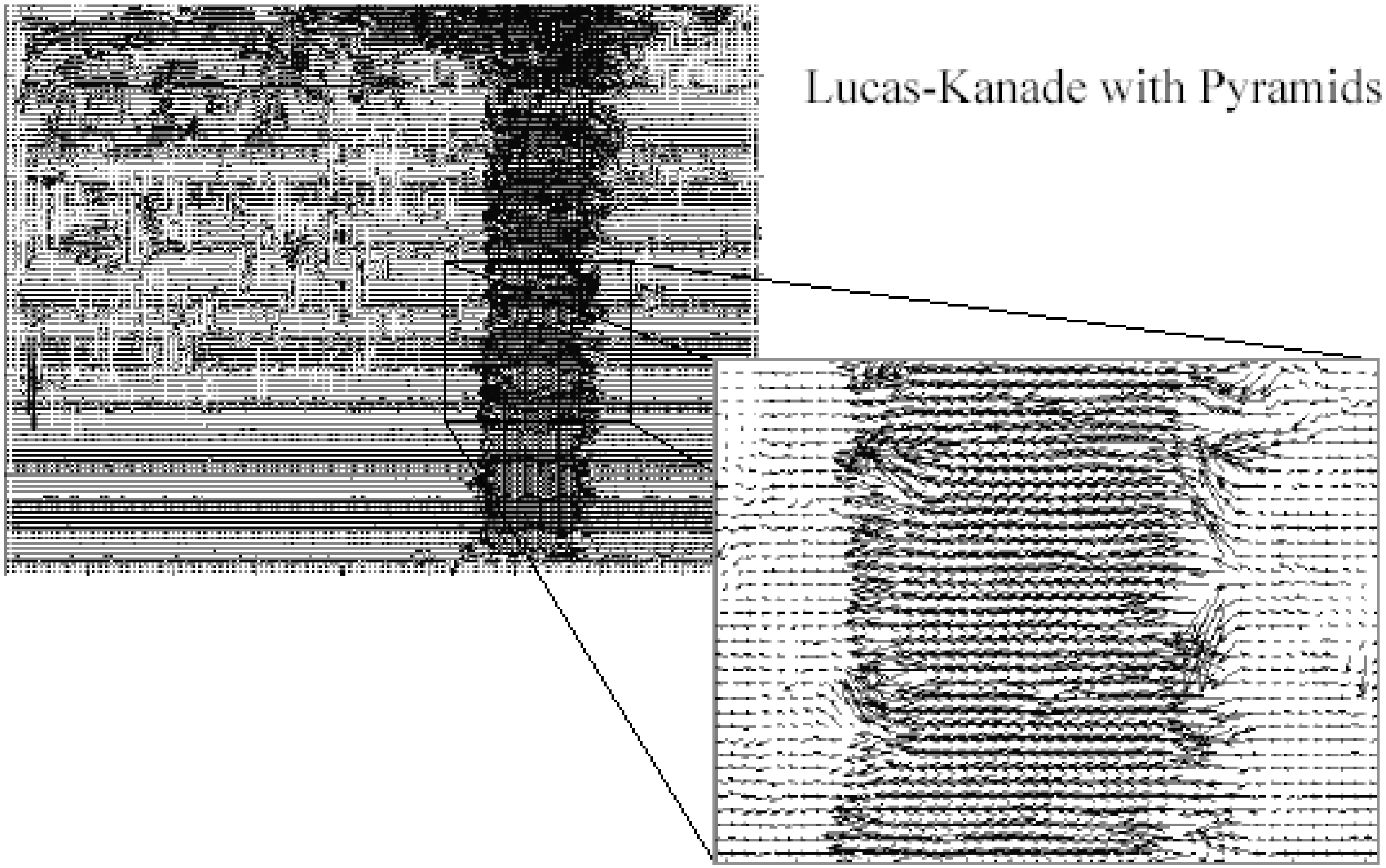


Lucas-Kanade without pyramids

Fails in areas of large motion



Optical Flow Results



Lucas-Kanade with Pyramids