CSCI 435/MCS9435
Computer Vision

Dynamic Vision (II)

Lecturer: Wanqing Li, PhD
Room: 3.101
email: wanqing@uow.edu.au
Web: http://www.uow.edu.au/~wanqing

Key problems in DV

- Change detection
  - Most time SCMO
  - Apps- Video surveillance
- Moving object detection and location
  - SCMO or MCSO or MCMO
  - Apps- Cloud tracking, autonomous vehicles, city traffic analysis, target tracking and positioning (military)
- Recovery of 3D object properties
  - MCSO or MCMO

Moving Object Detection & Location

- Background subtraction + size filter
  - SCMO
  - Adaptation of BG
- Color/Intensity segmentation + motion check
  - SCMO or MCMO
  - Region segmentation + correspondence
- Segmentation on motion features
  - Pixel/block based motion

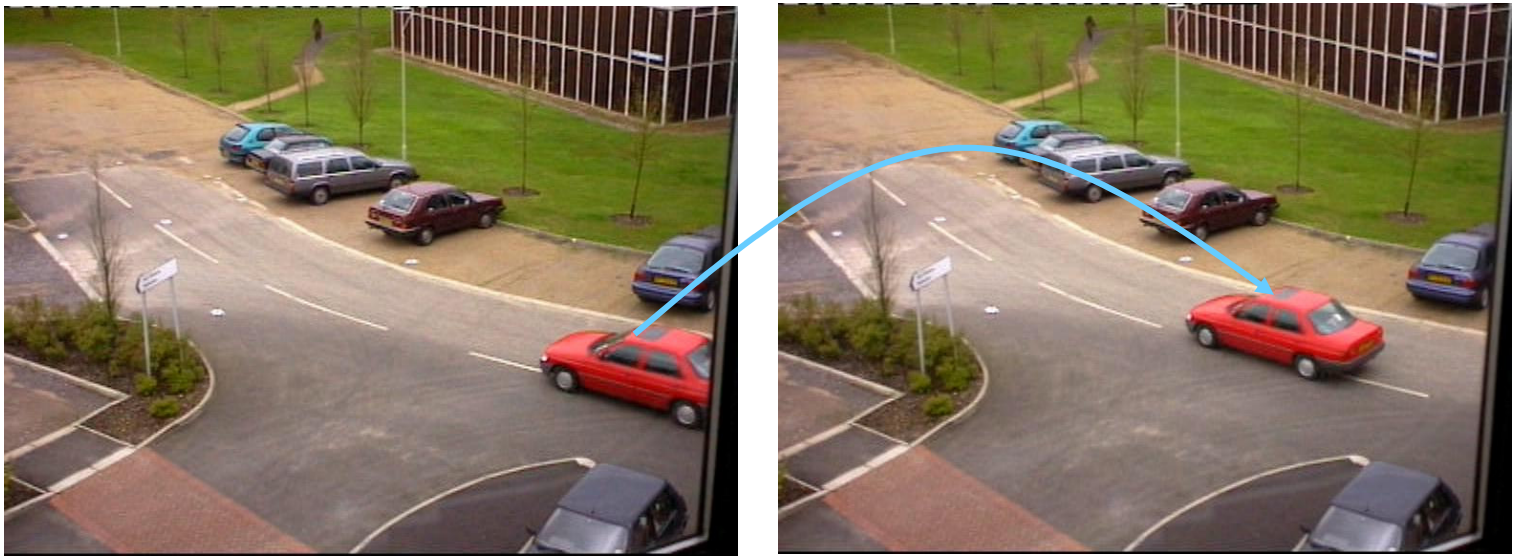
Outline

- Optical Flow
  - What is it?
  - Horn & Schunk method
  - Locas & Kanade method
- Block-based motion fields
  - Efficient methods
- Applications
  - Moving object segmentation

Optical Flow

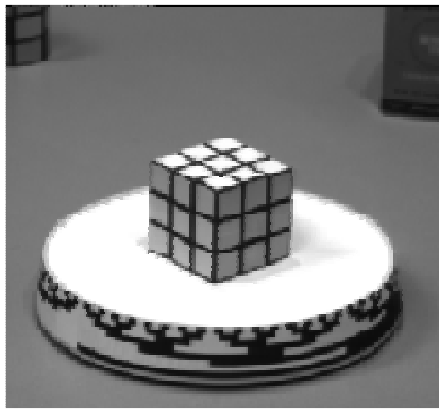
Optical Flow

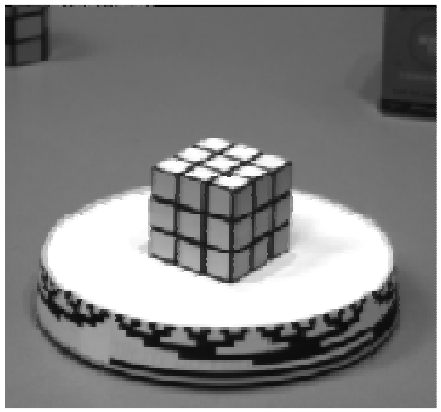
- Optical flow reflects the image changes due to motion during a time interval *dt*, and the optical flow field is the velocity field that represents the 3D motion of object point across a 2D images

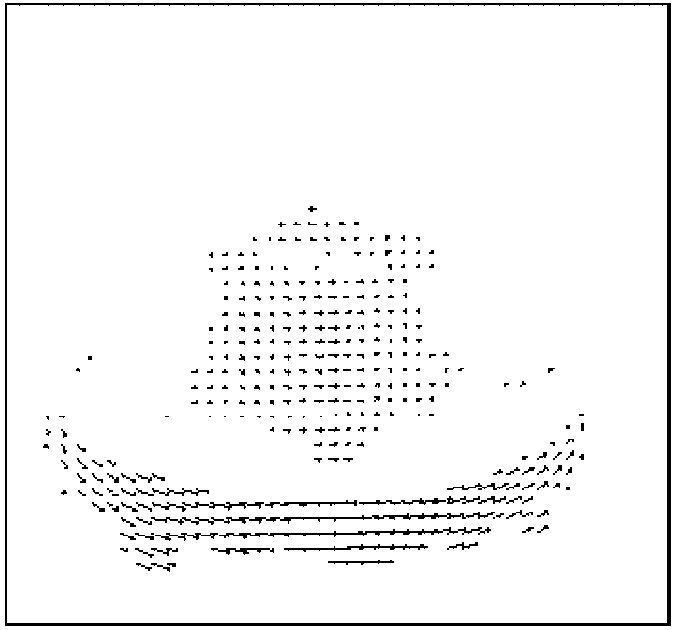


Optical Flow

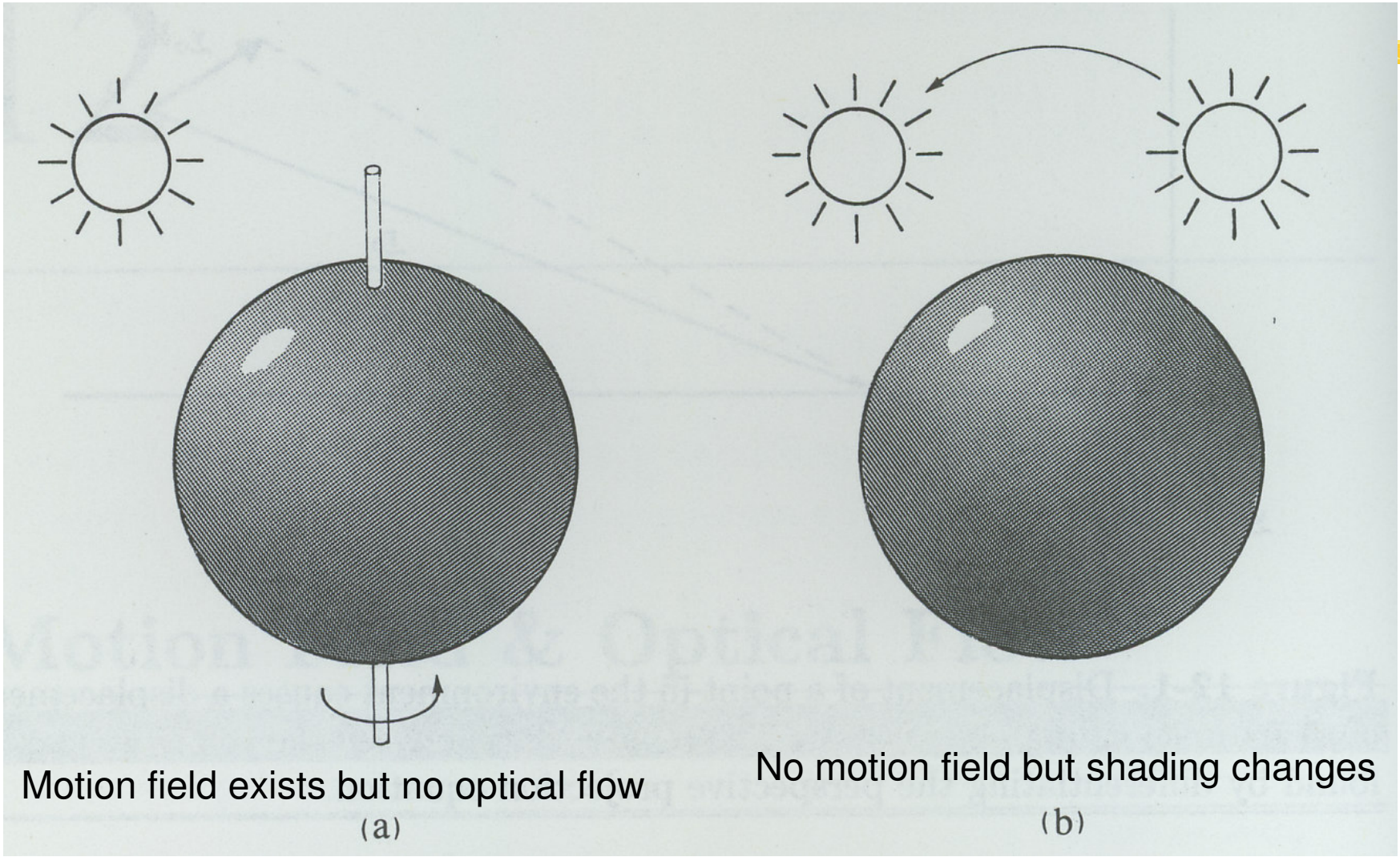
- Motion of brightness pattern in the image at pixel level
- **Ideally** Optical flow = Motion field







Optical Flow ≠ Motion Field



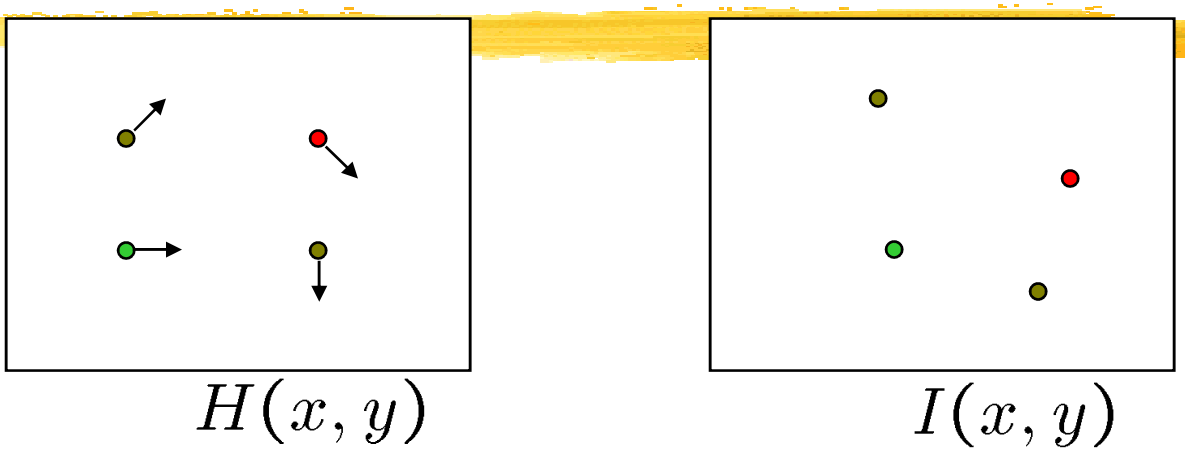
Motion field exists but no optical flow

(a)

No motion field but shading changes

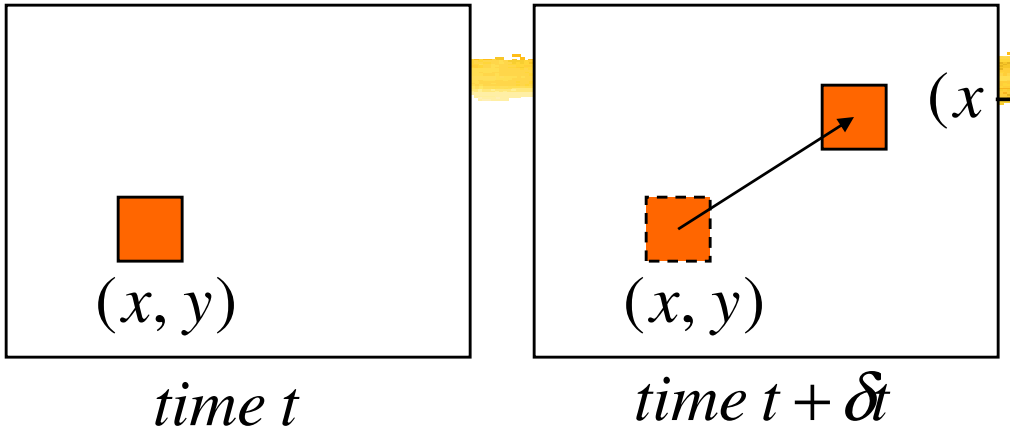
(b)

Horn & Schunck Method (AI'81)



- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - ✓ Given a pixel in H, look for nearby pixels of the same color in I
- Key assumptions
  - **color constancy:** a point in H looks "the same" in image I
    - ✓ For grayscale images, this is **brightness constancy**
  - **small motion**: points do not move very far

Optical Flow Constraint Equation



Optical Flow: Velocities (u, v)

Displacement:

$$(\delta x, \delta y) = (u\,\delta t, v\,\delta t)$$

- Assume brightness of patch remains same in both images:

$$E(x+u\,\delta t, y+v\,\delta t, t+\delta t) = E(x, y, t)$$

- Assume small motion: (Taylor expansion of LHS upto first order)

$$E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} = E(x, y, t)$$

Optical Flow Constraint Equation

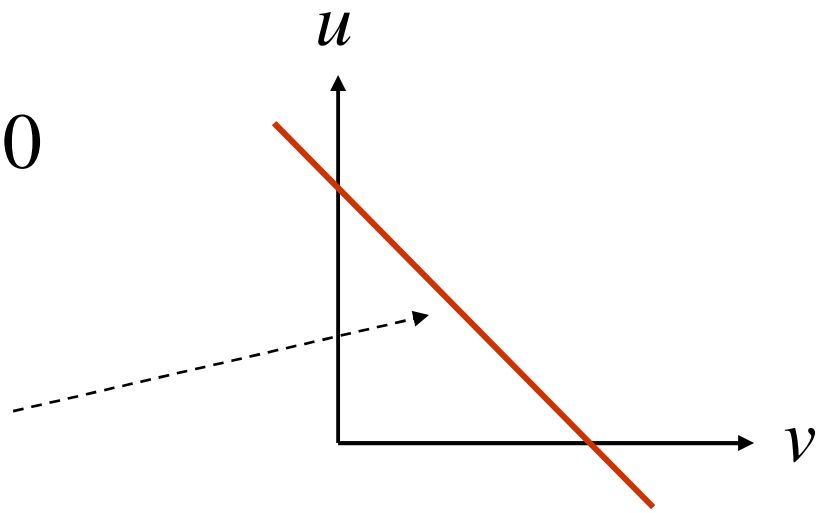
$$\delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} = 0$$

Divide by δt and take the limit $\delta t \to 0$

$$\frac{dx}{dt}\frac{\partial E}{\partial x} + \frac{dy}{dt}\frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0$$

Constraint Equation

$$E_x\, u + E_y\, v + E_t = 0$$

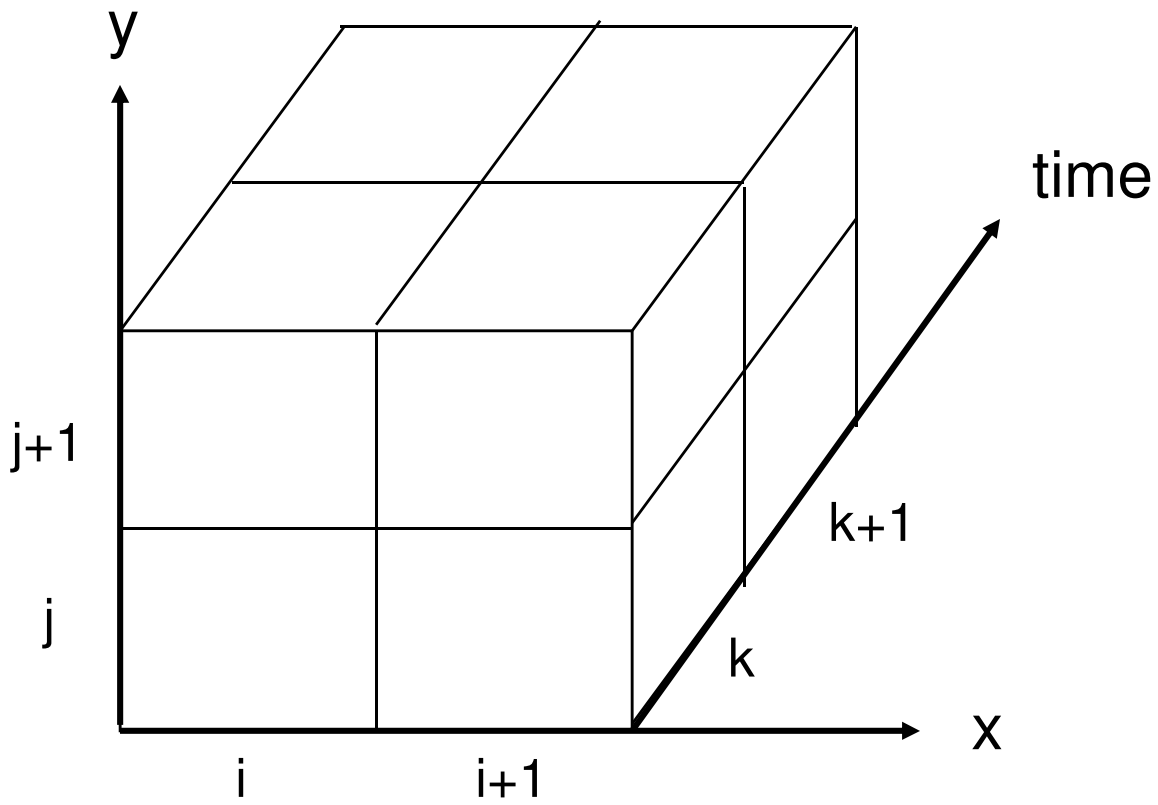


NOTE: (u, v) must lie on a straight line

We can compute E_x , E_y , E_t using gradient operators!

But (u,v) cannot be found uniquely with this constraint!

Finding Gradients in X-Y-T



Computing Gradients

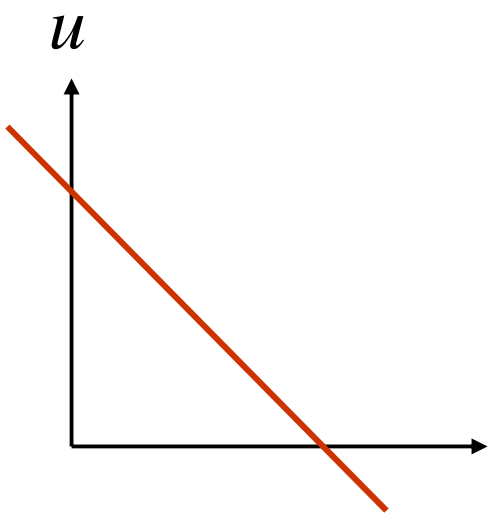
$$E_x \approx \frac{1}{4}(E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} +$$
$$E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1})$$

$$E_y \approx \frac{1}{4}(E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} +$$
$$E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1})$$

$$E_t \approx \frac{1}{4}(E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k} +$$
$$E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k})$$

Optical Flow Constraint

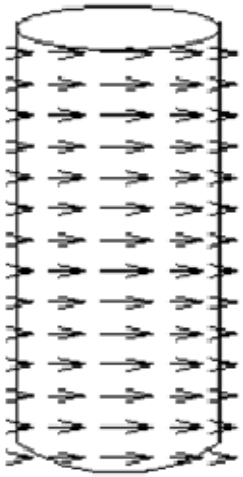
- Intuitively, what does this constraint mean?
  - The component of the flow in the gradient direction is determined
  - The component of the flow parallel to an edge is unknown



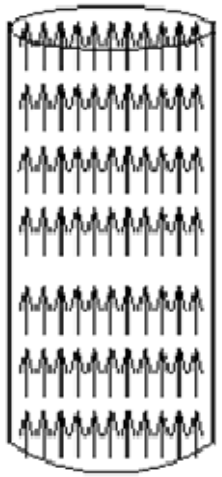
Barber Pole Illusion

Barber pole illusion

Barber's pole

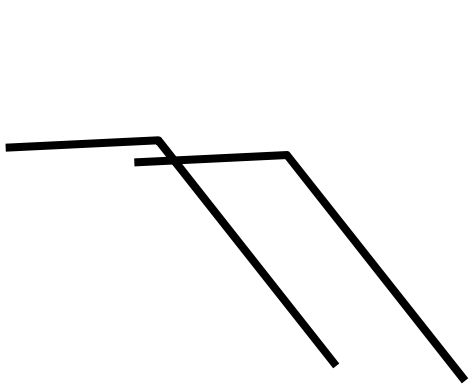


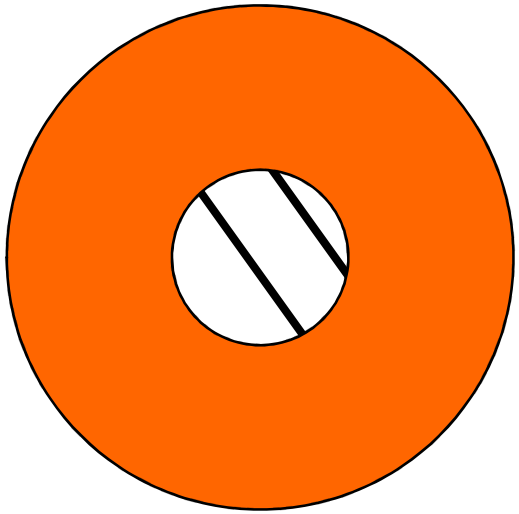
Motion field

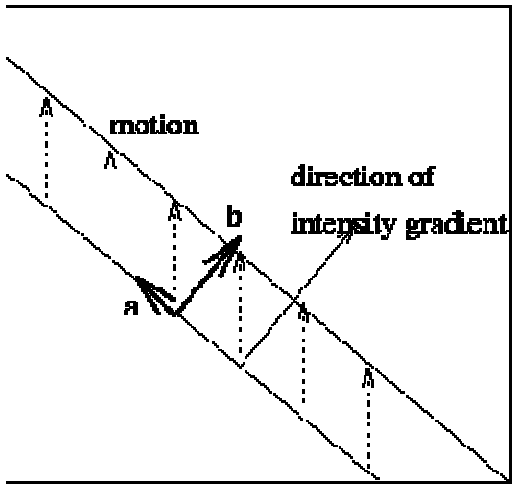


Optical flow

Aperture Problem







Computing Optical Flow

- Formulate Error in Optical Flow Constraint:

$$e_c = \iint_{image} (E_x u + E_y v + E_t)^2 \, dx \, dy$$

- We need additional constraints!
- Smoothness Constraint (as in shape from shading and stereo):

  Usually motion field varies smoothly in the image.
  So, penalize departure from smoothness:

$$e_s = \iint_{image} (u_x^2 + u_y^2) + (v_x^2 + v_y^2) \; dx \, dy$$

- Find (u,v) at each image point that MINIMIZES:

$$e = e_s + \lambda e_c$$

weighting factor

Discrete Optical Flow Algorithm

Consider image pixel (i, j)

- Departure from Smoothness Constraint:

$$s_{ij} = \frac{1}{4}[(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2 + (v_{i+1,j} - v_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2]$$

- Error in Optical Flow constraint equation:

$$c_{ij} = (E^{ij}{}_x\, u_{ij} + E^{ij}{}_y\, v_{ij} + E^{ij}{}_t)^2$$

- We seek the set $\{u_{ij}\}$ & $\{v_{ij}\}$ that minimize:

$$e = \sum_i \sum_j (s_{ij} + \lambda\, c_{ij})$$

NOTE: $\{u_{ij}\}$ & $\{v_{ij}\}$ show up in more than one term

Discrete Optical Flow Algorithm

- Differentiating e w.r.t v_{kl} & u_{kl} and setting to zero:

$$\frac{\partial e}{\partial u_{kl}} = 2\,(u_{kl} - \overline{u_{kl}}) + 2\lambda\,(E_x^{kl} u_{kl} + E_y^{kl} v_{kl} + E_t^{kl}) E_x^{kl} = 0$$

$$\frac{\partial e}{\partial v_{kl}} = 2\,(v_{kl} - \overline{v_{kl}}) + 2\lambda\,(E_x^{kl} u_{kl} + E_y^{kl} v_{kl} + E_t^{kl}) E_y^{kl} = 0$$

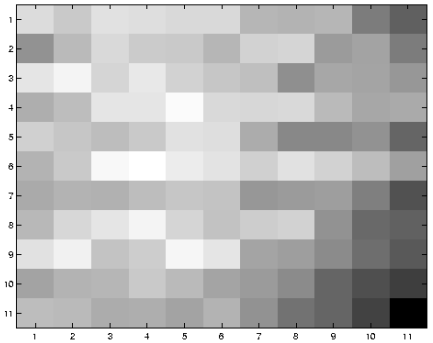
- $\overline{v_{kl}}$ & $\overline{u_{kl}}$ are averages of (u, v) around pixel (k, l)

Update Rule:

$$u_{kl}^{n+1} = \overline{u_{kl}^{n}} - \frac{E_x^{kl}\,\overline{u_{kl}^{n}} + E_y^{kl}\,\overline{v_{kl}^{n}} + E_t^{kl}}{1 + \lambda\,[(E_x^{kl})^2 + (E_y^{kl})^2]} E_x^{kl}$$

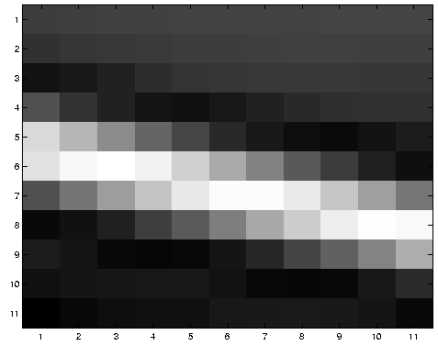
$$v_{kl}^{n+1} = \overline{v_{kl}^{n}} - \frac{E_x^{kl}\,\overline{u_{kl}^{n}} + E_y^{kl}\,\overline{v_{kl}^{n}} + E_t^{kl}}{1 + \lambda\,[(E_x^{kl})^2 + (E_y^{kl})^2]} E_y^{kl}$$

Low Texture Region - Bad



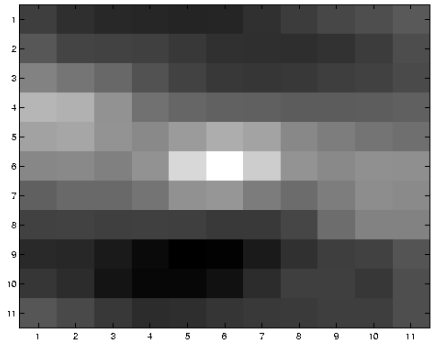
– gradients have small magnitude

Edges – so,so (aperture problem)



– large gradients, all the same

High Textured Region - Good

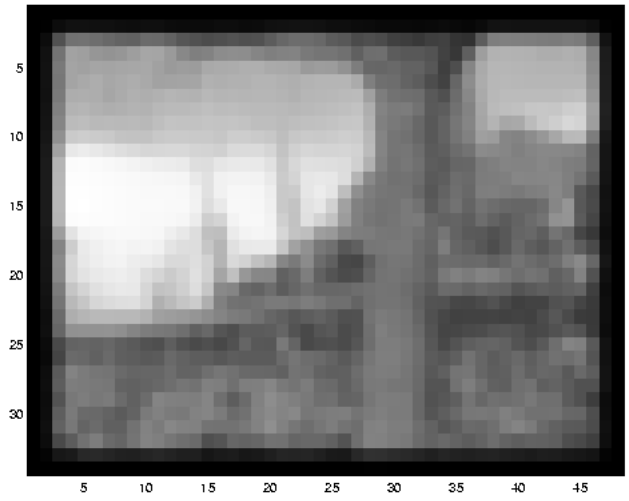


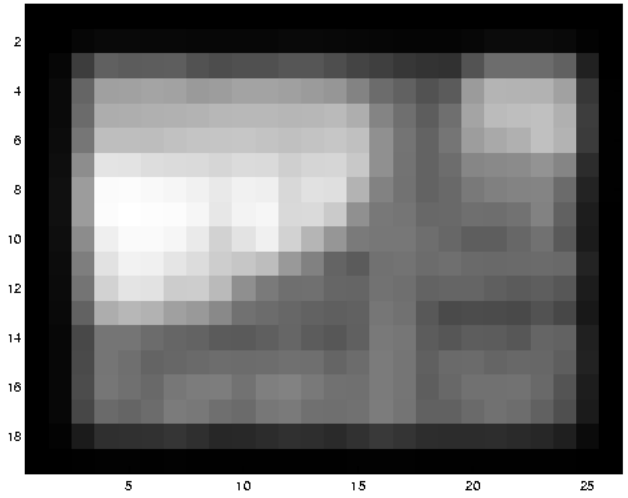
– gradients are different, large magnitudes

Revisiting the Small Motion Assumption

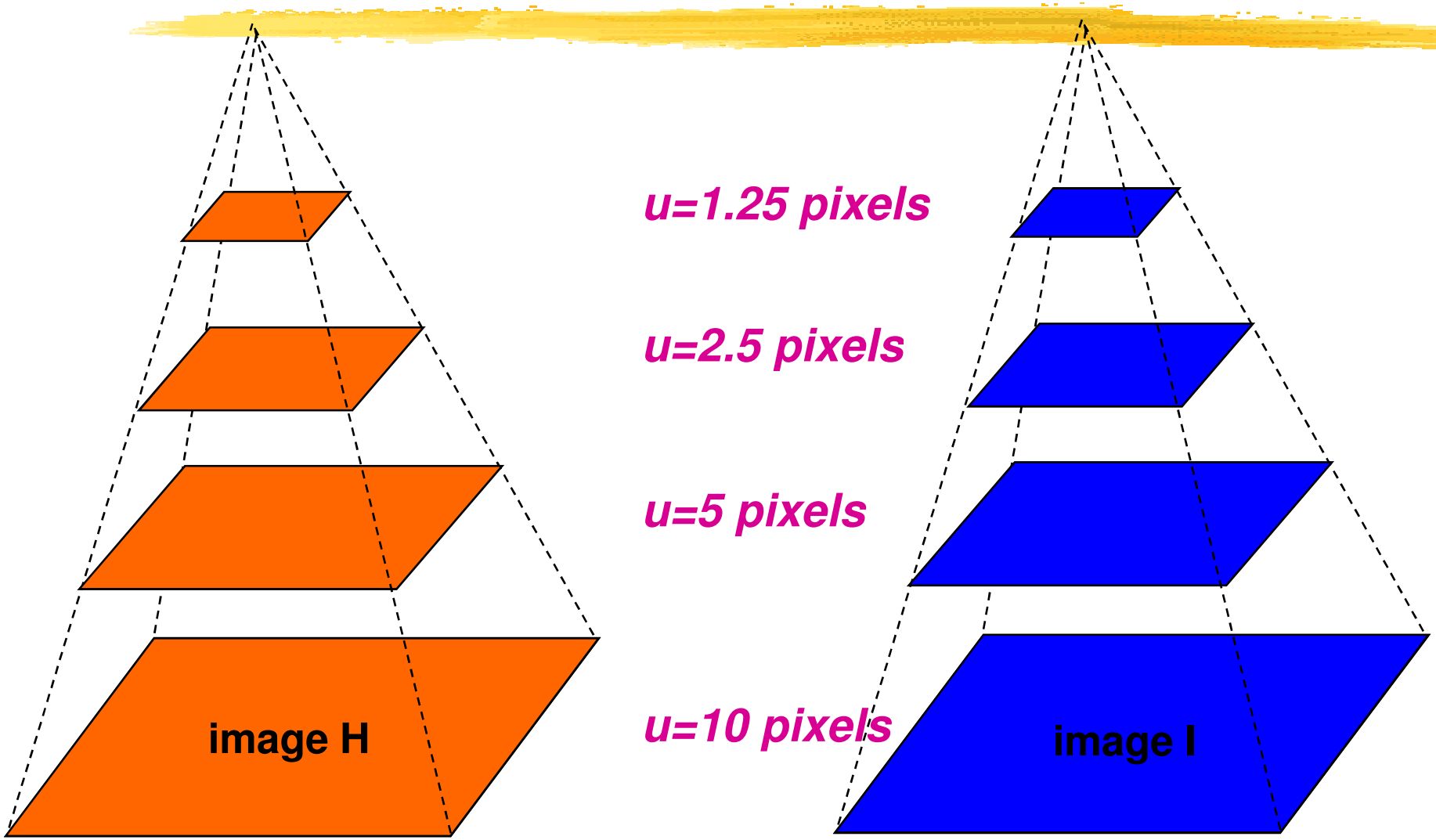
- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2^{nd} order terms dominate)
  - How might we solve this problem?

Reduce the Resolution!





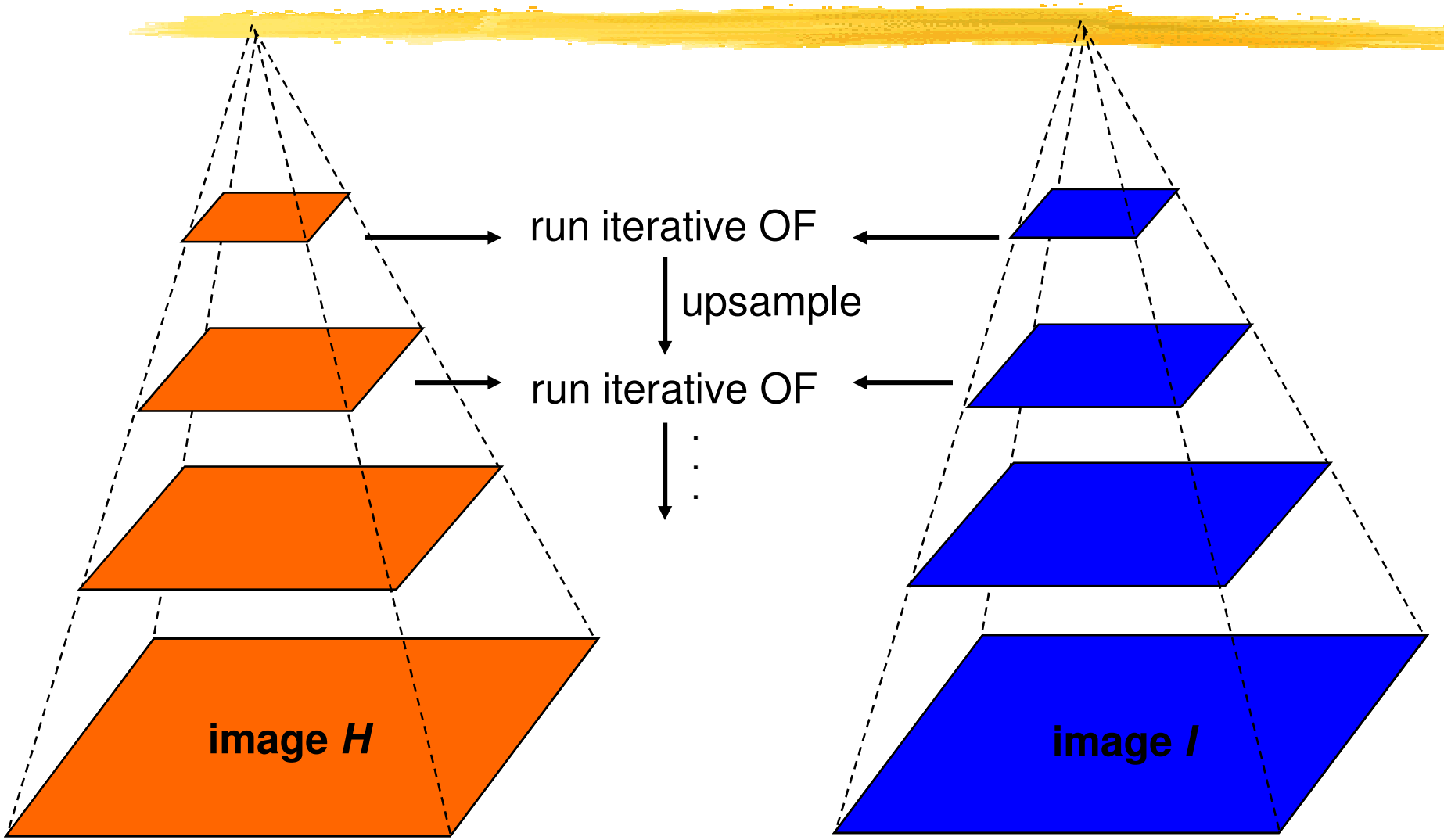
Coarse-to-fine Optical Flow Estimation



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Coarse-to-fine Optical Flow Estimation



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Local Models

- Global vs Local
  - Global methods use a global constraint, usually a smoothing regularization term.
  - Local methods do not have a global term. It need some other techniques to solve the aperture problem.

Lucas & Kanade Method (Ref.2)

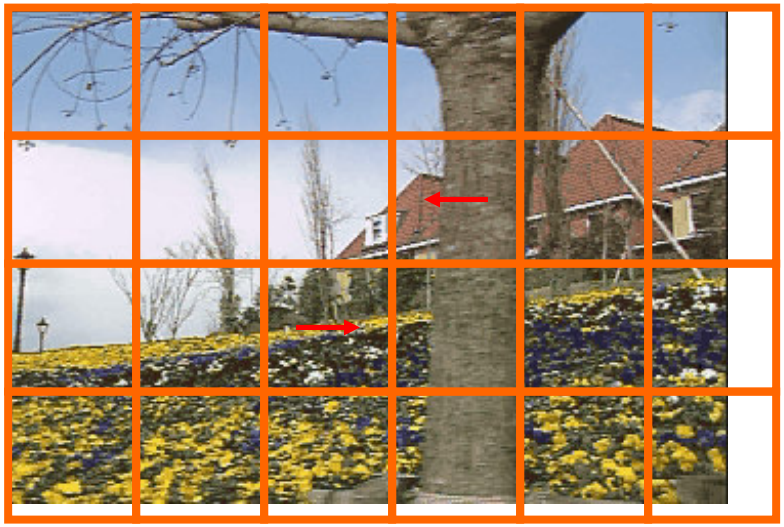
- Horn & Schunk Method
  - Global methods use a global constraint, i.e. a smoothing regularization term.

- Lucas & Kanade Method
  - Local methods do not have a global term. It employs other techniques to solve the aperture problem.

Block-Based Motion Field

Block-Based Motion Estimation

- Break image up into square blocks
- Estimate translation for each block



Motion Estimation (ME)

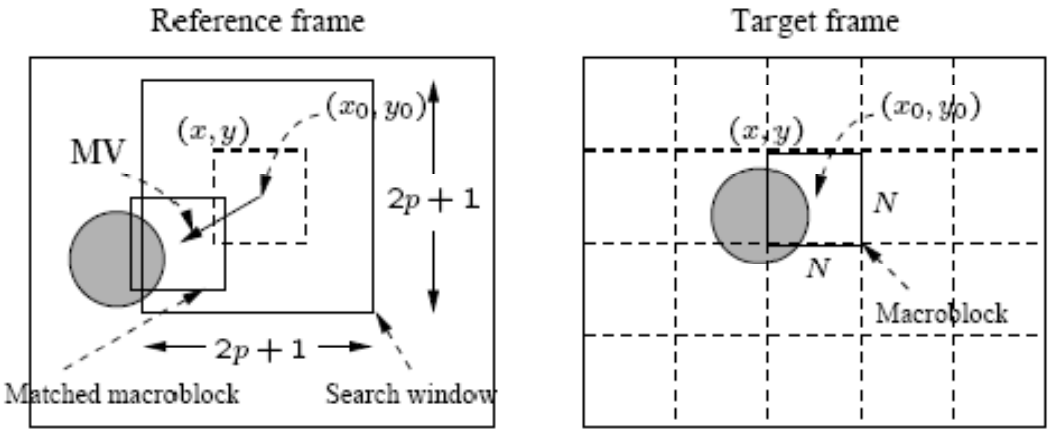


Fig. 10.1: Macroblocks and Motion Vector in Video Compression.

- MV search is usually limited to a small immediate neighborhood — both horizontal and vertical displacements in the range $[-p, p]$.

  This makes a search window of size $(2p+1) \times (2p+1)$.

Typical Measure of Difference (L_1-norm)

- The difference between two macroblocks can then be measured by their *Mean Absolute Difference (MAD)*:

$$MAD(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)| \qquad (10.1)$$

N – size of the macroblock,

k and l – indices for pixels in the macroblock,

i and j – horizontal and vertical displacements,

$C(x+k, y+l)$ – pixels in macroblock in Target frame,

$R(x+i+k, y+j+l)$ – pixels in macroblock in Reference frame.

- The goal of the search is to find a vector (i,j) as the motion vector $\mathrm{MV} = (\mathrm{u}, \mathrm{v})$, such that $MAD(i,j)$ is minimum:

$$(u,v) = [\ (i,j)\ |\ MAD(i,j)\ \text{ is minimum, }\ i \in [-p,p],\ j \in [-p,p]\ ] \qquad (10.2)$$

ME Algorithms

- Full Search – Sequential Search
- 2D logarithmic Search
- Hierarchical Search

Sequential Search

- **Sequential search**: sequentially search the whole $(2p+1) \times (2p+1)$ window in the Reference frame (also referred to as Full search).
  - a macroblock centered at each of the positions within the window is compared to the macroblock in the Target frame pixel by pixel and their respective MAD is then derived using Eq. (10.1).
  - The vector (i, j) that offers the least MAD is designated as the MV (u, v) for the macroblock in the Target frame.
  - sequential search method is very costly — assuming each pixel comparison requires three operations (subtraction, absolute value, addition), the cost for obtaining a motion vector for a single macroblock is $(2p+1) \cdot (2p+1) \cdot N^2 \cdot 3 \Rightarrow O(p^2 N^2)$.

Sequential Search…

PROCEDURE 10.1 Motion-vector:sequential-search

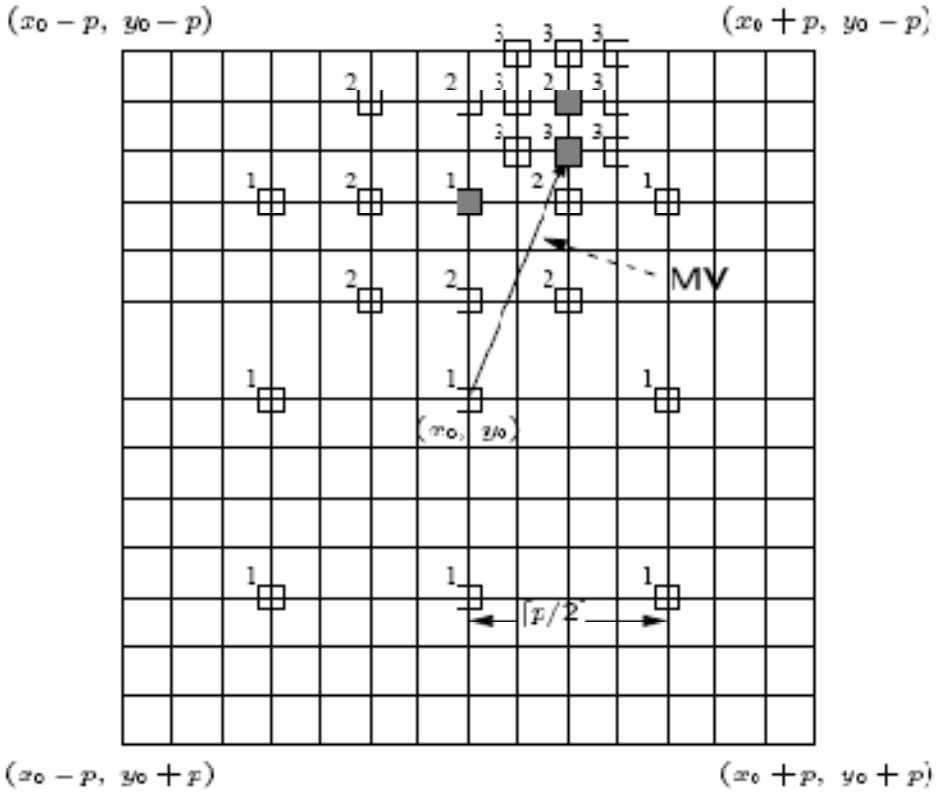
```
begin
  min_MAD = LARGE_NUMBER;        /* Initialization */
  for i = -p to p
    for j = -p to p
      {
        cur_MAD = MAD(i, j);
        if cur_MAD < min_MAD
          {
            min_MAD = cur_MAD;
            u = i;        /* Get the coordinates for MV. */
            v = j;
          }
      }
end
```

2D Logarithmic Search

- **Logarithmic search**: a cheaper version, that is suboptimal but still usually effective.

- The procedure for 2D Logarithmic Search of motion vectors takes several iterations and is akin to a binary search:
  - As illustrated in Fig.10.2, initially only nine locations in the search window are used as seeds for a MAD-based search; they are marked as `1'.
  - After the one that yields the minimum *MAD* is located, the center of the new search region is moved to it and the step-size ("offset") is reduced to half.
  - In the next iteration, the nine new locations are marked as `2', and so on.

2D Logarithmic Search…



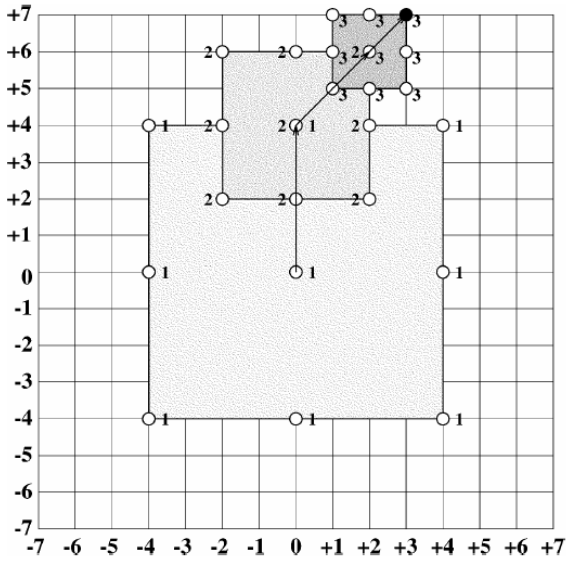


Fig. 10.2: 2D Logarithmic Search for Motion Vectors.

2D Logarithmic Search…

PROCEDURE 10.2 Motion-vector:2D-logarithmic-search

```
begin
  offset = ⌈p/2⌉;
  Specify nine macroblocks within the search window in the Reference frame,
  they are centered at (x0, y0) and separated by offset horizontally and/or
  vertically;
  while last ≠ TRUE
    {
      Find one of the nine specified macroblocks that yields minimum MAD;
      if offset = 1 then last = TRUE;
      offset = ⌈offset/2⌉;
      Form a search region with the new offset and new center found;
    }
end
```

Hierarchical Search

- The search can benefit from a hierarchical (multiresolution) approach in which initial estimation of the motion vector can be obtained from images with a significantly reduced resolution.

- Figure 10.3: a three-level hierarchical search in which the original image is at Level 0, images at Levels 1 and 2 are obtained by down-sampling from the previous levels by a factor of 2, and the initial search is conducted at Level 2.

- Since the size of the macroblock is smaller and p can also be proportionally reduced, the number of operations required is greatly reduced.

Hierarchical Search…

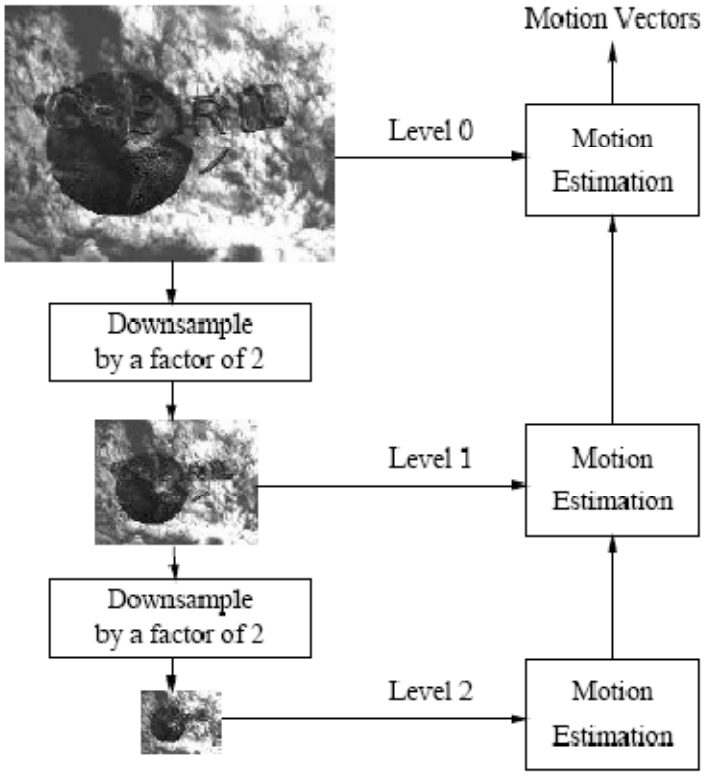


Fig. 10.3: A Three-level Hierarchical Search for Motion Vectors.

Hierarchical Search…

- Given the estimated motion vector (u^k, v^k) at Level k, a 3×3 neighborhood centered at $(2 \cdot u^k, 2 \cdot v^k)$ at Level $k-1$ is searched for the refined motion vector.

- the refinement is such that at Level $k-1$ the motion vector (u^{k-1}, v^{k-1}) satisfies:

$$(2u^k - 1 \leq u^{k-1} \leq 2u^k + 1, \quad 2v^k - 1 \leq v^{k-1} \leq 2v^k + 1)$$

- Let (x_0^k, y_0^k) denote the center of the macroblock at Level k in the Target frame. The procedure for hierarchical motion vector search for the macroblock centered at (x_0^0, y_0^0) in the Target frame can be outlined as follows:

Hierarchical Search…

PROCEDURE 10.3 Motion-vector:hierarchical-search

begin

// Get macroblock center position at the lowest resolution Level k

$x_0^k = x_0^0/2^k; \quad y_0^k = y_0^0/2^k;$

Use Sequential (or 2D Logarithmic) search method to get initial estimated

$\mathrm{MV}(u^k, v^k)$ at Level k;

while last $\neq$ TRUE

{

Find one of the nine macroblocks that yields minimum MAD

at Level $k-1$ centered at

$(2(x_0^k+u^k)-1 \leq x \leq 2(x_0^k+u^k)+1, \quad 2(y_0^k+v^k)-1 \leq y \leq 2(y_0^k+v^k)+1)$;

if $k=1$ then last = TRUE;

$k = k-1$;

Assign (x_0^k, y_0^k) and (u^k, v^k) with the new center location and MV;

}

end

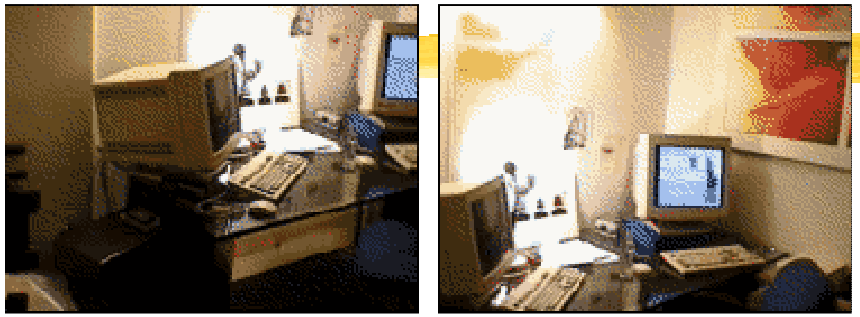
Comparison of Computational Cost

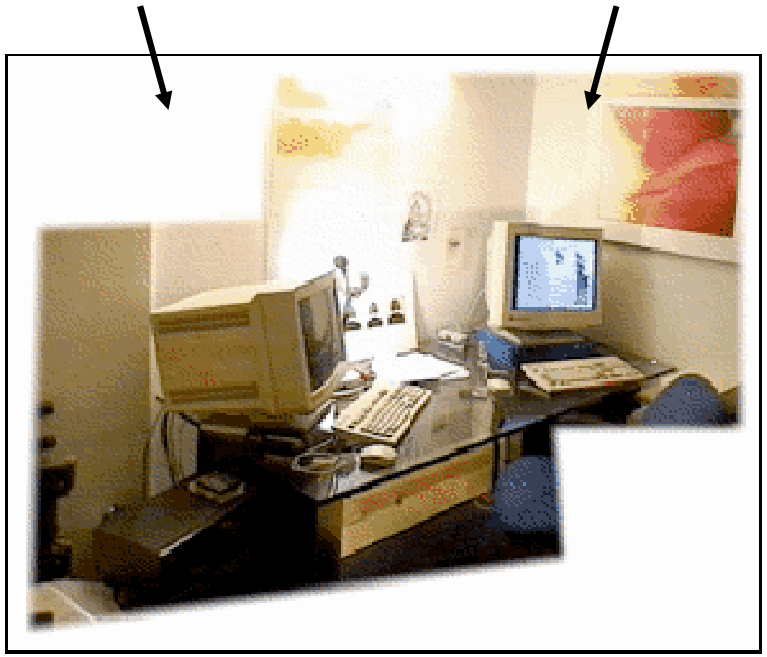
Table 10.1 Comparison of Computational Cost of Motion Vector Search based on examples

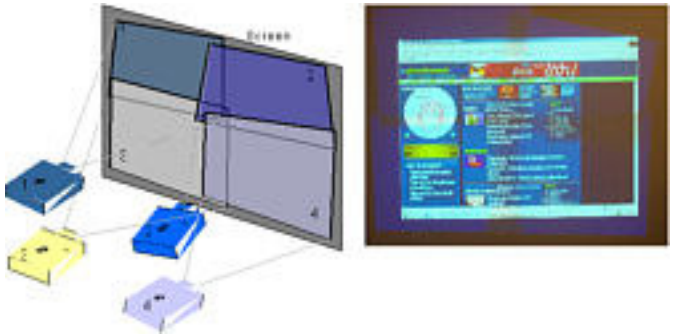
Search Method	OPS_per_second for 720×480 at 30 fps	
	$p = 15$	$p = 7$
Sequential search	29.89×10^9	7.00×10^9
2D Logarithmic search	1.25×10^9	0.78×10^9
3-level Hierarchical search	0.51×10^9	0.40×10^9

Applications

Image Alignment

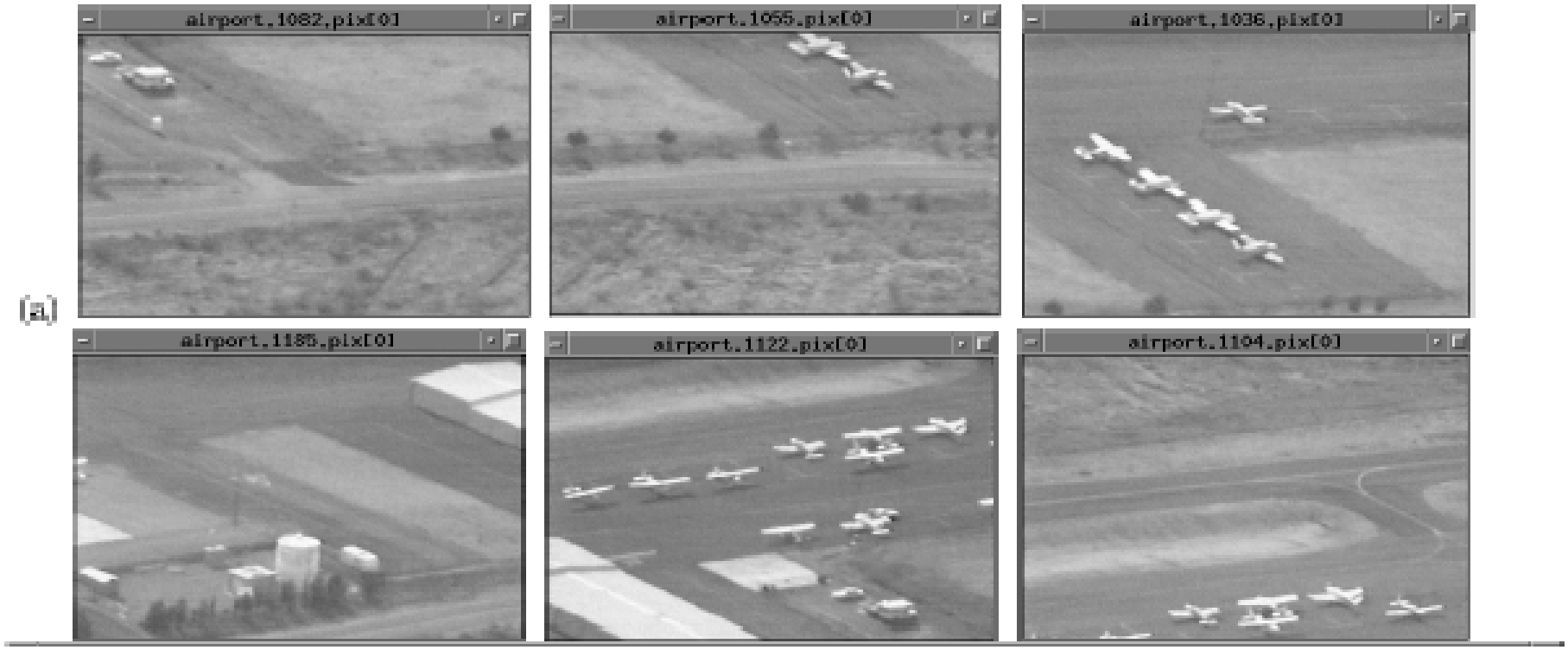






- Goal: Estimate single *(u,v)* translation (transformation) for entire image

Mosaicing



(Michal Irani, Weizmann)

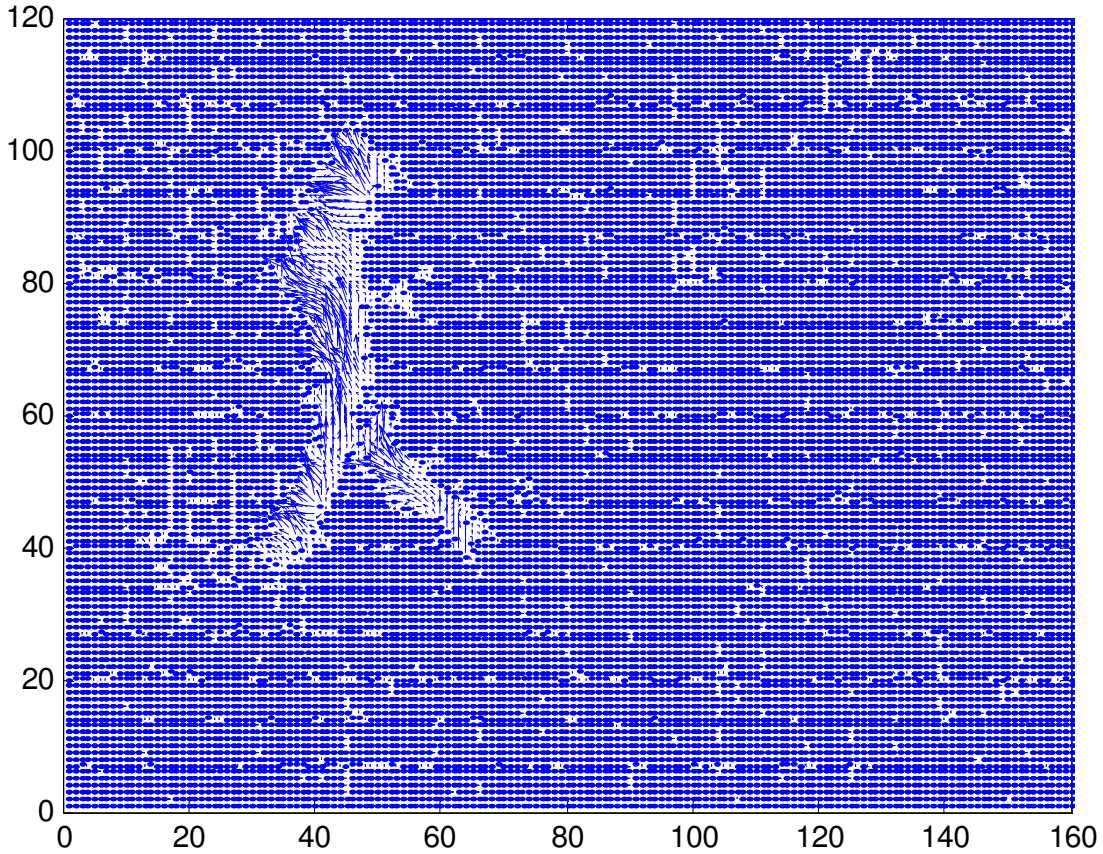
Mosaicing

. 1. Static background mosaic of an airport video clip.
(a) A few representative frames from the minute-long video clip. The video shows an airport being imaged from the air with a moving camera. The scene itself is static (i.e., no moving objects). (b) The static background mosaic image which provides an extended view of the entire scene imaged by the camera in the one-minute video clip.

(Michal Irani, Weizmann)

Moving Object Segmentation

Moving Object Segmentation



References

- B. K. P. Horn, B. G. Schunck, “Determining Optical Flow”, AI, vol.17, 1981, pp.185-203
- B. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, Proceedings DARPA Image Understanding Workshop, Washington, D. C. 1981, pp.121-130