Motion

Computer Vision (CS0029)

Motion

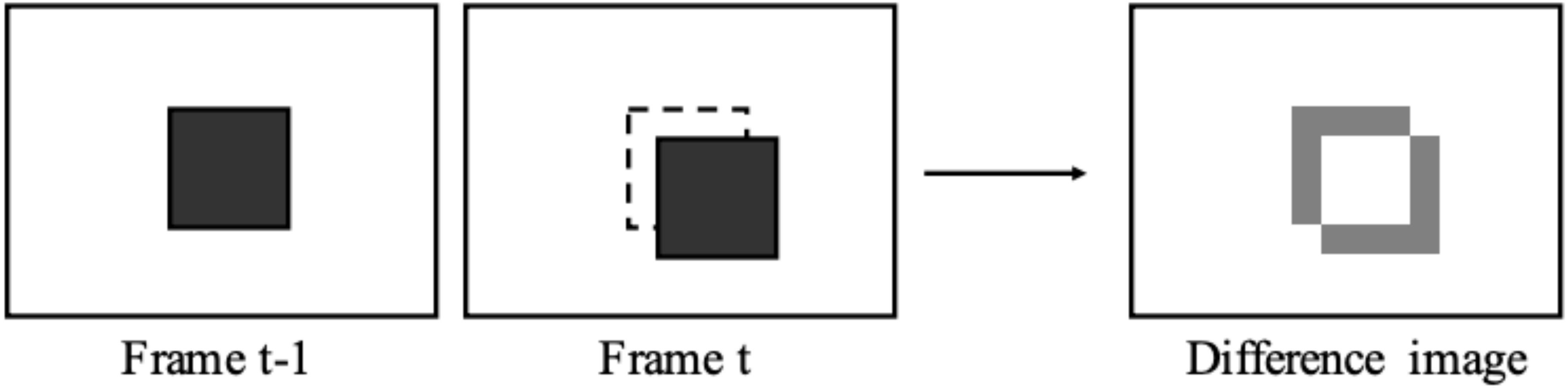
- Changing scene may be observed in a sequence of images
- Changing pixels in image sequence provide important features for object detection and activity recognition

General Cases of Motion

- Still camera, single moving object, constant background
  - Simplest case, motion sensors for security
- Still camera, several moving objects, constant background
  - Tracking, multi-person event analysis
- Moving camera, relatively constant scene
  - Egomotion, panning to provide wider panoramic view
- Moving camera, several moving objects
  - Most difficult, robot navigating through heavy traffic

Image Differencing

- Simplest means of motion detection
- Detect absolute value of difference between frame t-1 and t
  - Threshold result
- Only motion “presence”, no direction



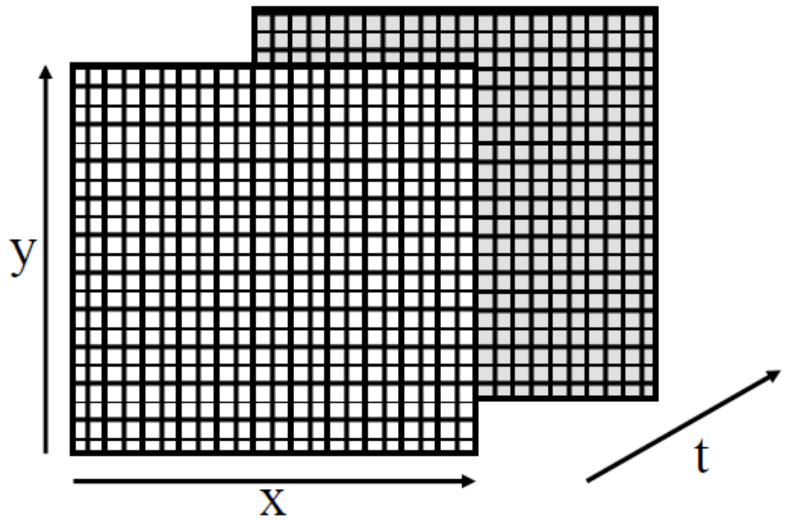
Single, Constant Threshold?

- Image differencing:
  - $\triangle I = \begin{cases} 1 & |I_t - I_{t-1}| > \tau \\ 0 & else \end{cases}$

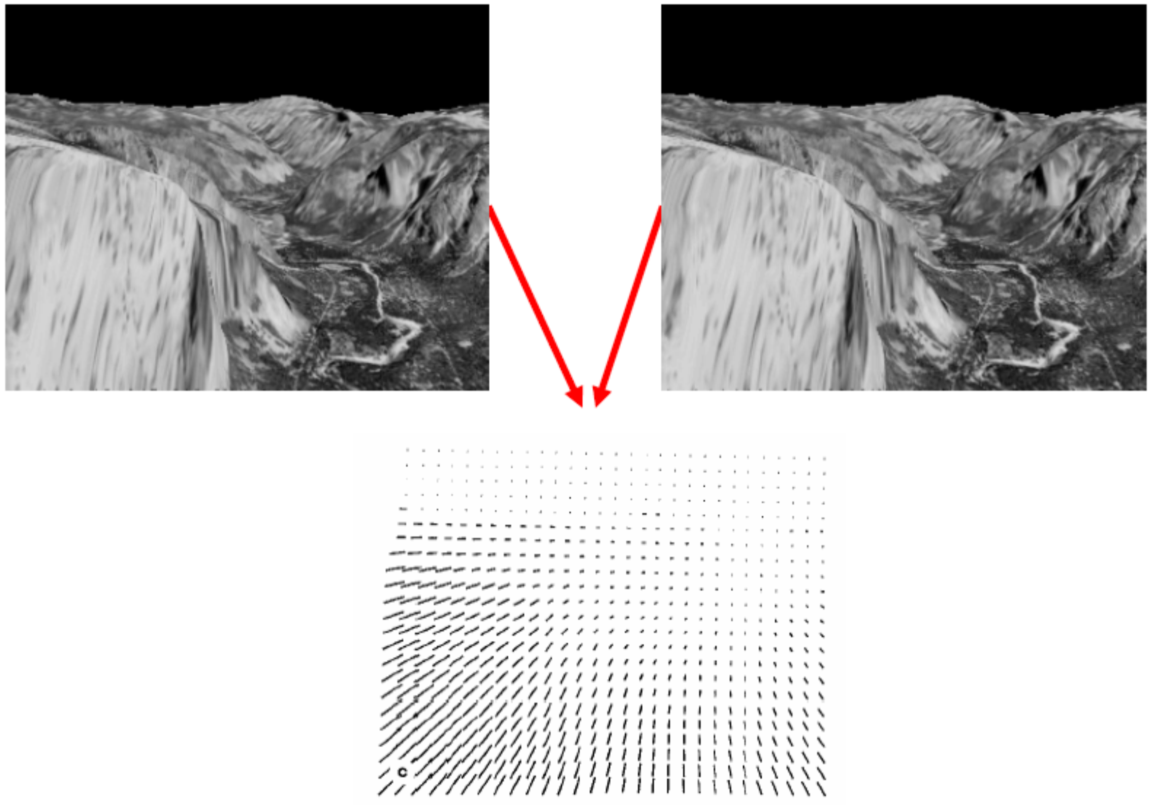
  - Appropriate threshold perhaps not a constant value
    - Set too low and noise will appear in the difference image
    - Set too high and will not work in all situations

Optic Flow

- Observed 2D displacements of brightness patterns in the image
  - Optic flow is what we can estimate from images
- Image sequence $\mathrm{I}(x, y, t)$
- Assume with small motion change (of patch), no change in gray levels (brightness constancy constraint)



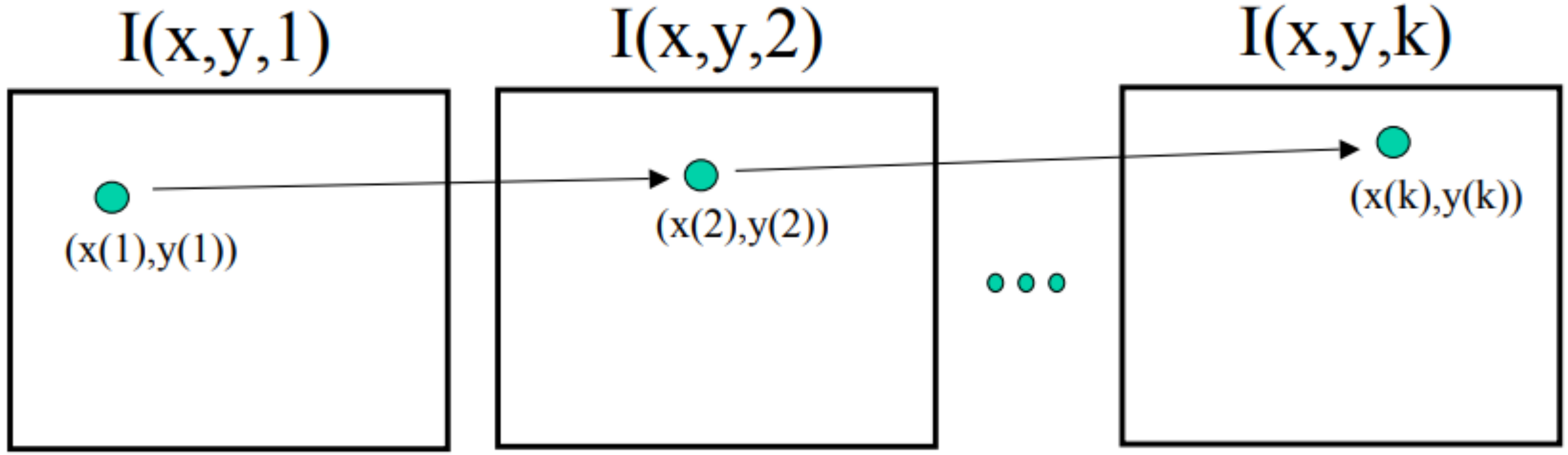
Optical Flow



- Optical flow is apparent motion of objects or surfaces

Brightness Constant Equation

- Consider a scene point moving through an image sequence



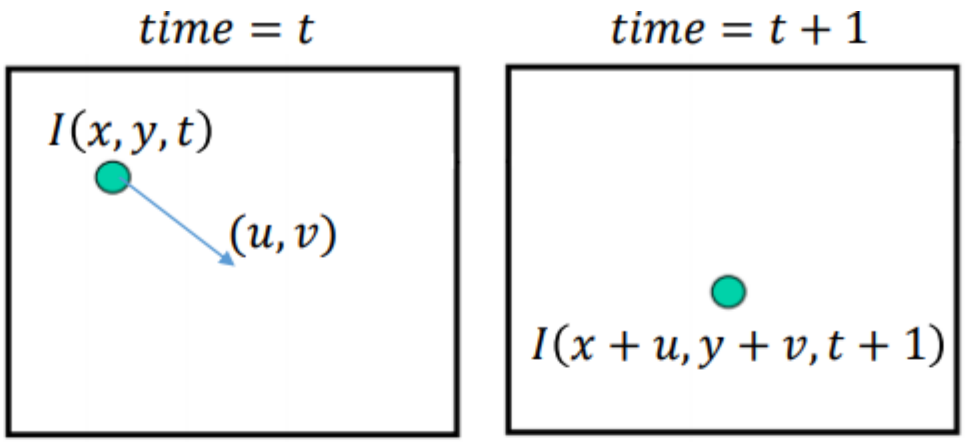
- Claim: its brightness/color will remain the same
- $I(x(t), y(t), t)$ is constant

Brightness Constant Equation

- First Assumption: brightness of an object point is constant
  - $I(x, y, t) = I(x + u, y + v, t + 1)$
  - $0 = I(x + u, y + v, t + 1) - I(x, y, t)$
- Second assumption: (u, v) is less than one pixel or smoothly
  - The gradient (moving direction) of a object point at t and t+1 are the same
- Taylor series expansion of I:
  - $I(x + u, y + v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + [higher\ order\ terms]$
  - $I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$
- $0 = I(x + u, y + v, t + 1) - I(x, y, t)$

  $\approx I(x, y, t + 1) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v - I(x, y, t)$

  $\approx \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + [I(x, y, t + 1) - I(x, y, t)]$
- $0 \approx \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + [I(x, y, t + 1) - I(x, y, t)] = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}$



Brightness Constant Equation

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

- $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}$: spatial gradient, we can compute it by sobel or prewitt filters
- $\frac{dx}{dt}, \frac{dy}{dt}$: optical flow, what we want to find!!!
- $\frac{\partial I}{\partial t}$: derivative across image frames. Known
  - Brightness(intensity) change of a small local spatial region along the temporal(t) direction

Spatial gradient

- $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$
- Gradient can be computed from images
  - $\frac{\partial I}{\partial x} = I_x \rightarrow \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ 1 & 0 & 1 \end{bmatrix} / 8$ $\qquad$ $\frac{\partial I}{\partial y} = I_y \rightarrow \begin{bmatrix} -1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} / 8$
  - Compute I_x and I_y in I_t

- $\frac{\partial I}{\partial t}$=$I_t \rightarrow F(I_t - I_{t-1})$
  - F(x) could be take 3x3 smoothing filter applied to each image before taking the diff at 9 center pixel location

Spatial Gradient

******For motion <u>from</u> I_{t-1} <u>to</u> I_t**

Image (I_t)

0	0	0	0	1
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
1	1	1	1	1

$$I_x = [0(-1) + 0(0) + 1(1) + 0(-2) + 1(0) + 1(2) + 1(-1) + 1(0) + 1(1)]/8 = 3/8$$

$$I_y = [0(-1) + 0(-2) + 1(-1) + 0(0) + 1(0) + 1(0) + 1(1) + 1(2) + 1(1)]/8 = 3/8$$

$$I_t = F(I_t - I_{t-1})$$

or

$$I_t = F(smooth(I_t) - smooth(I_{t-1}))$$

Optical Flow

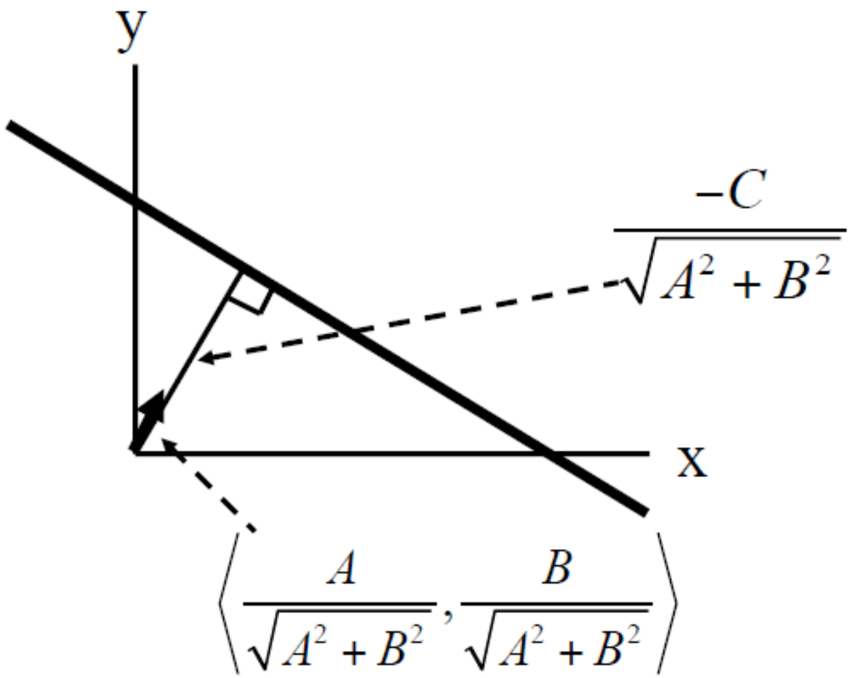
- $\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$

- Optical flow: $\frac{dx}{dt}, \frac{dy}{dt}$
  - One equation, two unknown?

Normal Optical Flow

- Consider normal form of straight line equation:

$$\frac{A}{\sqrt{A^2+B^2}}x+\frac{B}{\sqrt{A^2+B^2}}y+\frac{C}{\sqrt{A^2+B^2}}=0$$



Normal Optical Flow

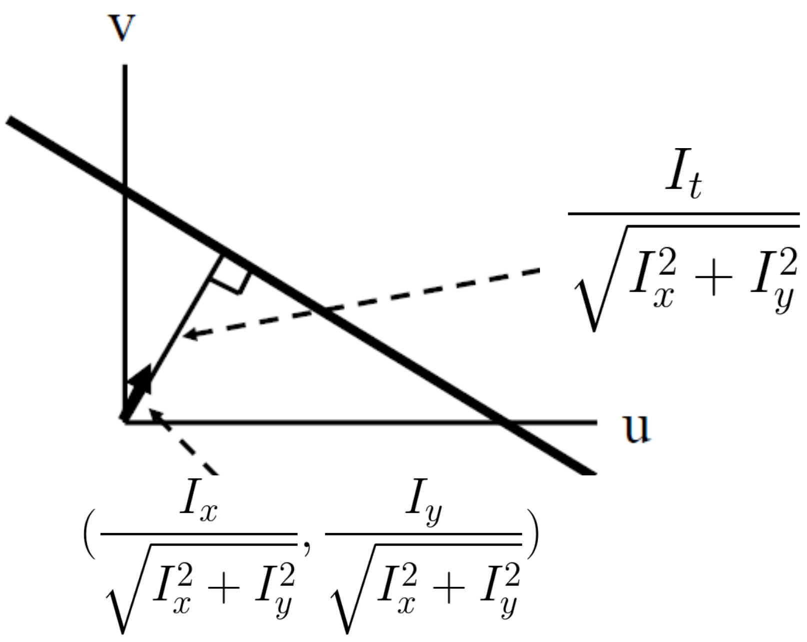
Rewrite $\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$ to $I_x u + I_y v + I_t = 0$, then divided by $\sqrt{I_x^2 + I_y^2}$

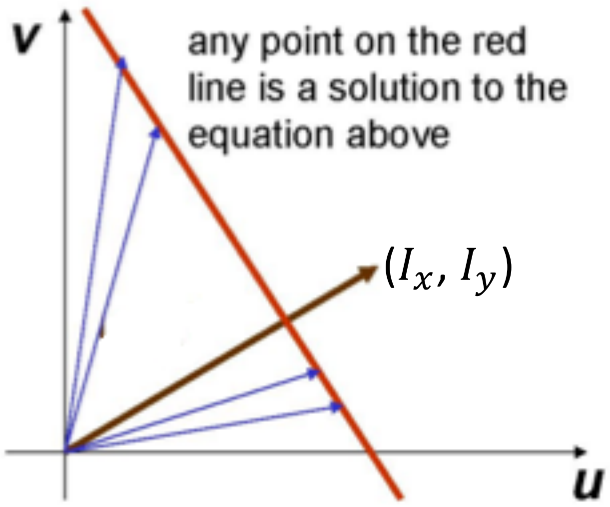
$$\frac{I_x}{\sqrt{I_x^2 + I_y^2}} u + \frac{I_y}{\sqrt{I_x^2 + I_y^2}} v + \frac{I_t}{\sqrt{I_x^2 + I_y^2}} = 0$$

Can determine only one component of flow:

<u>Magnitude</u> of flow in the <u>direction of the brightness gradient</u> (perpendicular to brightness contour)

"Normal Flow"





Aggregate Optic Flow

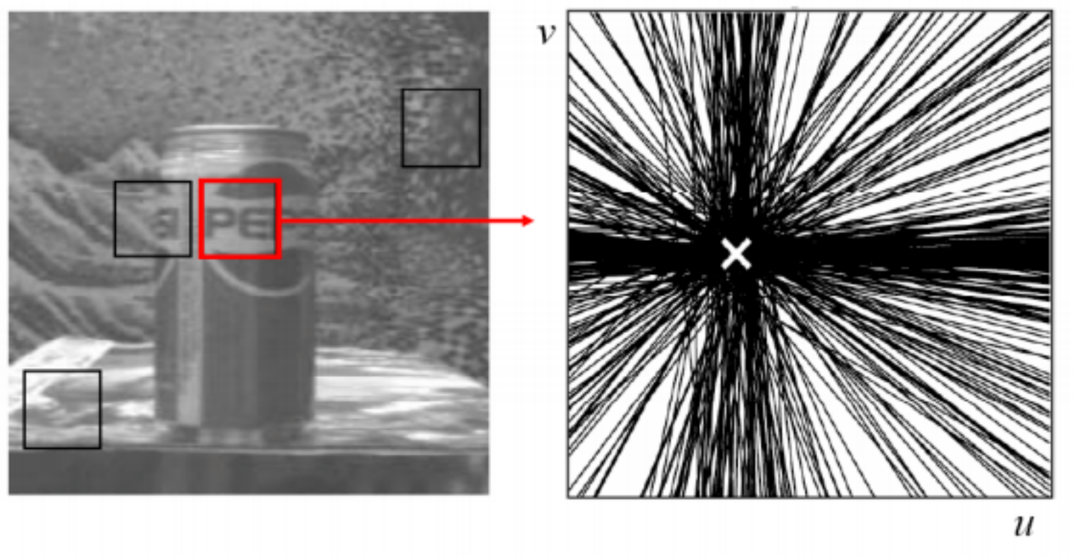
Assume local spatial smoothness

$$\begin{cases} I_x^1 u + I_y^1 v + I_t^1 = 0 \\ I_x^2 u + I_y^2 v + I_t^2 = 0 \\ \dots \\ \dots \\ I_x^N u + I_y^N v + I_t^N = 0 \end{cases} \qquad \begin{cases} I_x^1 u + I_y^1 v = -I_t^1 \\ I_x^2 u + I_y^2 v = -I_t^2 \\ \dots \\ \dots \\ I_x^N u + I_y^N v = -I_t^N \end{cases}$$

$$\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \dots & \dots \\ \dots & \dots \\ I_x^N & I_y^N \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_t^1 \\ -I_t^2 \\ \dots \\ \dots \\ -I_t^N \end{bmatrix}$$

Over-constrain! Solve u, v by least-square

Over-constrain by checking over small "patch" of pixels



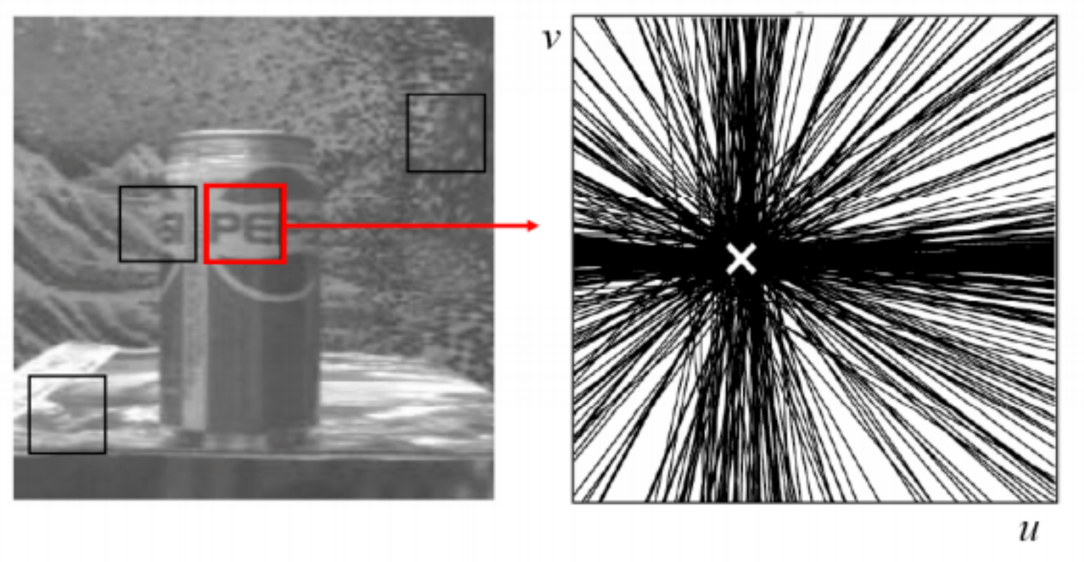
Aggregate Optic Flow

$$\underbrace{\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \cdots & \cdots \\ \cdots & \cdots \\ I_x^N & I_y^N \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{x^*} = \underbrace{\begin{bmatrix} -I_t^1 \\ -I_t^2 \\ \cdots \\ \cdots \\ -I_t^N \end{bmatrix}}_{b}$$

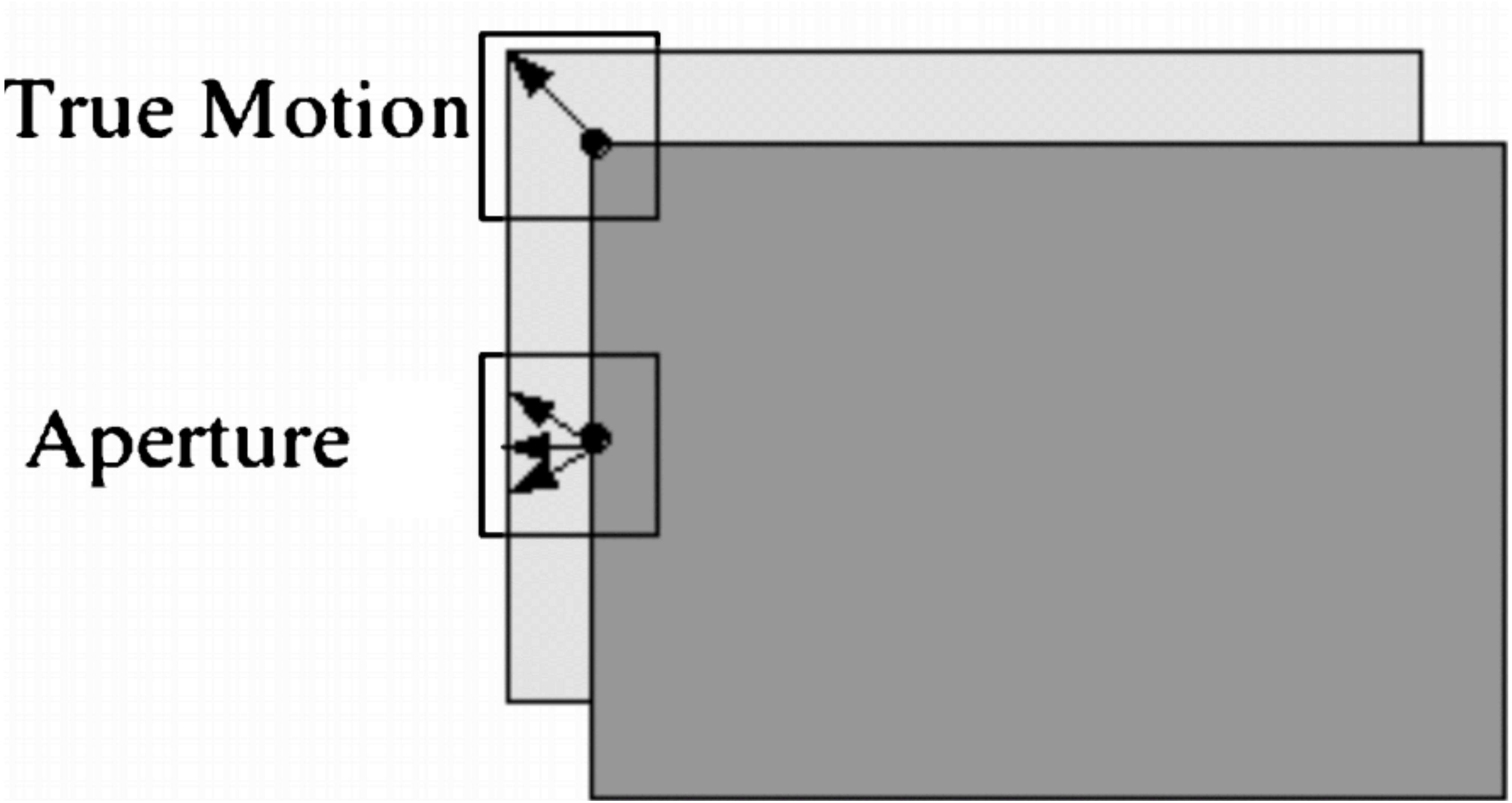
$AA^T x^* = A^T b$ Least-square approximation

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

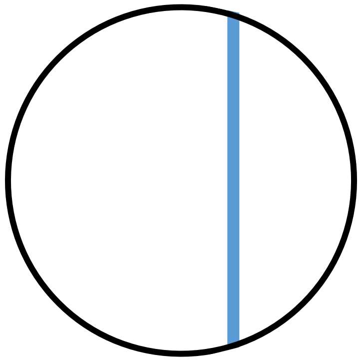
Over-constrain by checking over small "patch" of pixels



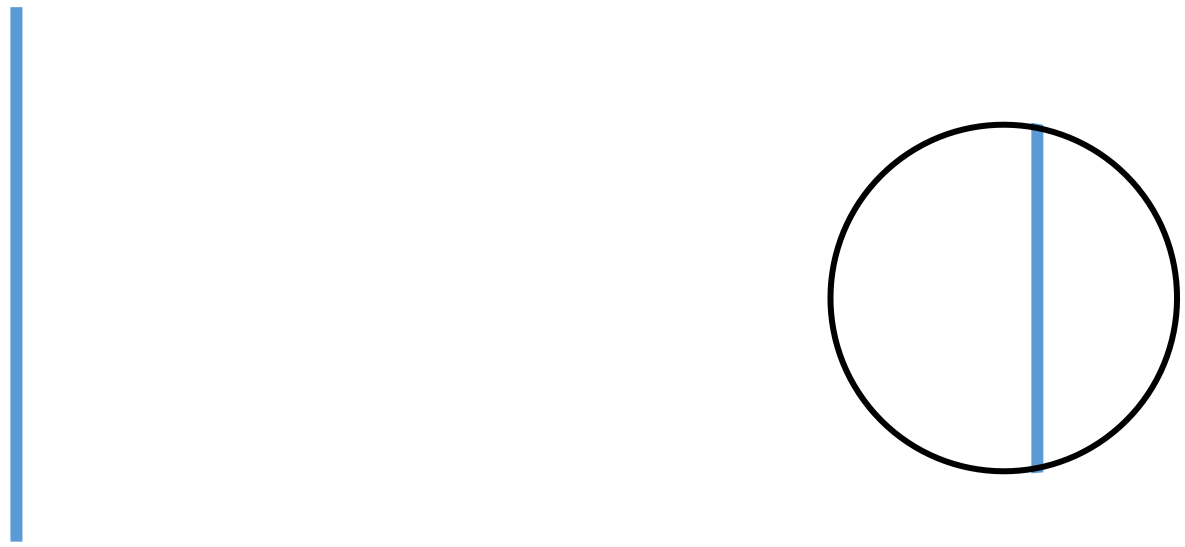
Aperture Problem



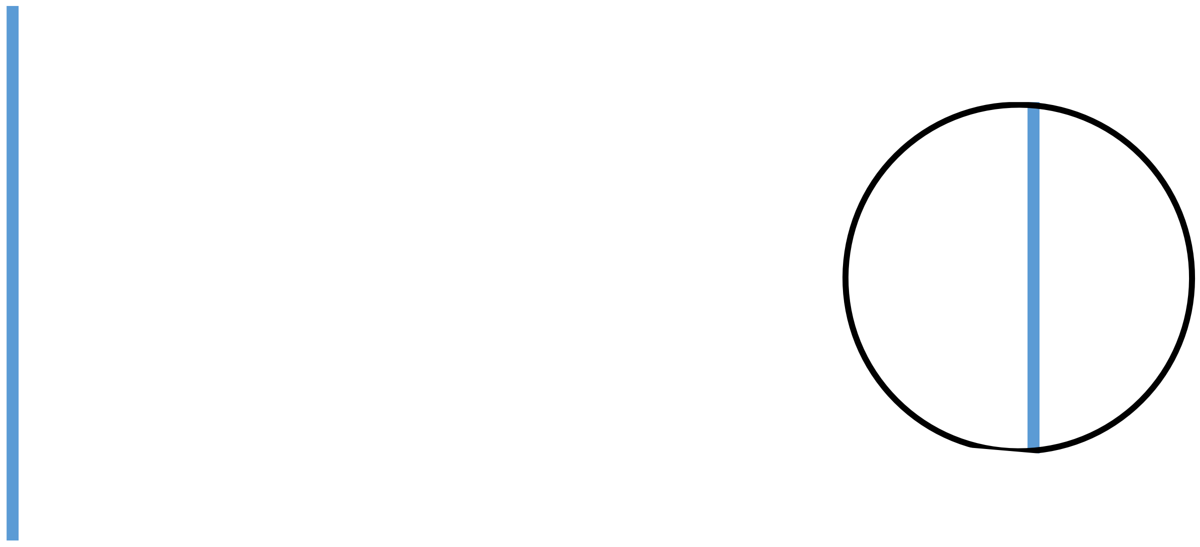
Aperture Problem



Aperture Problem



Aperture Problem



Aperture Problem

Aperture Problem

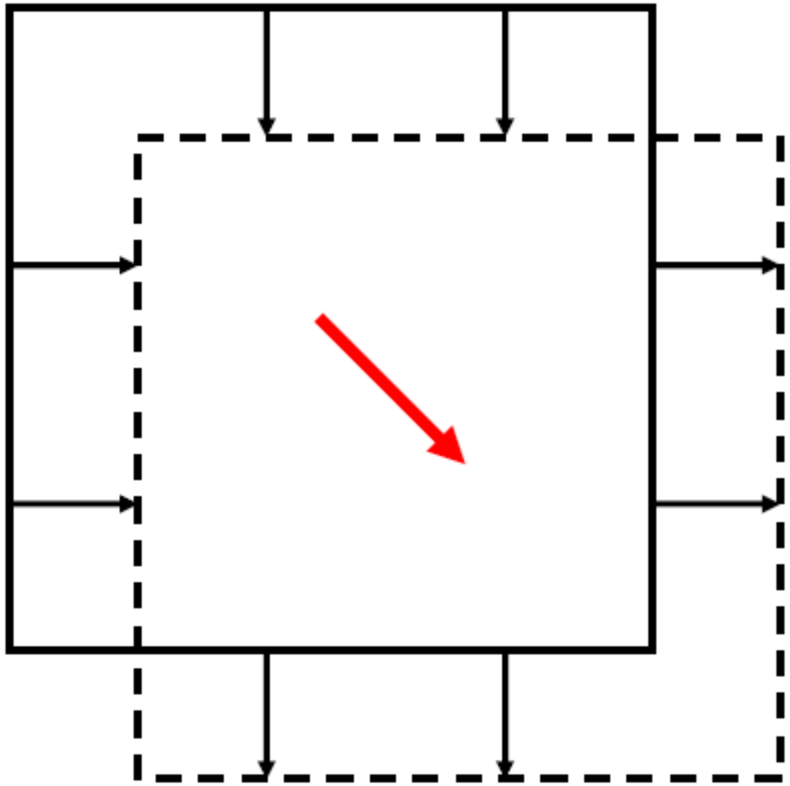
Aperture Problem

Barberpole Illusion

- Ambiguous information about its real direction of movement
  - For a point, the apparent motion direction is toward up
  - But, the true motion is toward left

- Individual motion sensitive neurons have only local information.
  - Usually our visual system prefer the slowest of possible motions

Normal Flow v.s. Desired Optical Flow



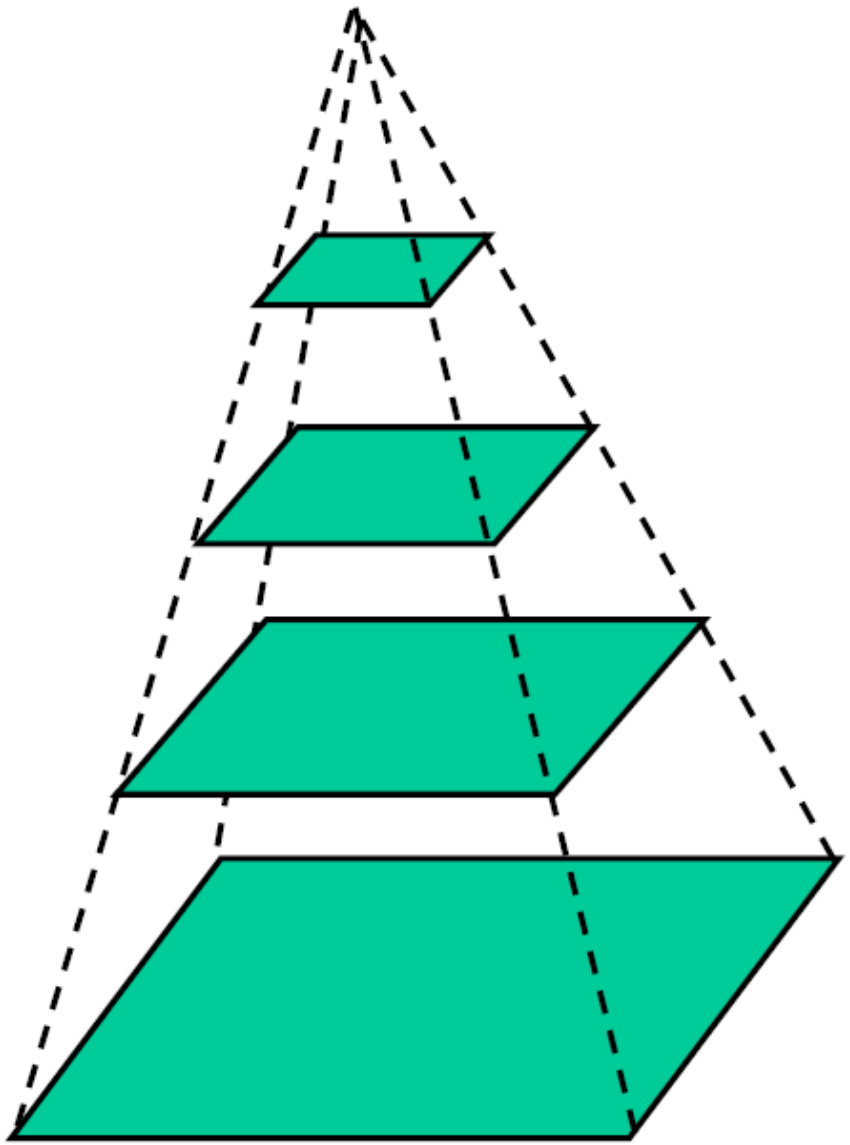
Hierarchical Motion Estimation

- Assumption for computation of optical flow
  - Small motion change
  - No change in gray levels after moving (brightness constancy constraint)
- Not entirely realistic for real-world motions in video
  - Usually larger motion (even at 30 frames/sec)

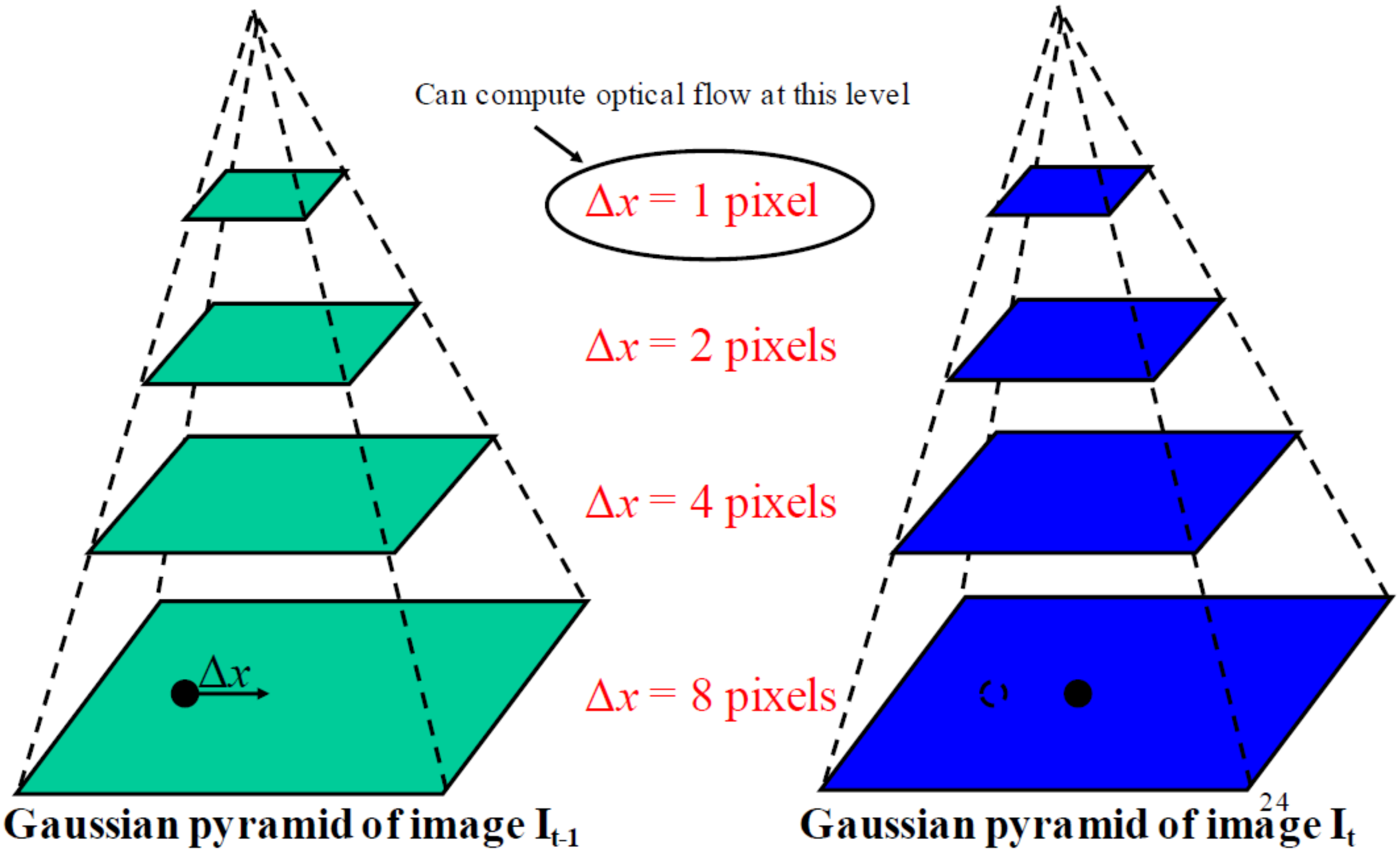
Hierarchical Motion Estimation

- Key framework
  - Coarse-to-fine refinement strategy
    - Local model (used in estimation process)
    - Global model (constrains overall structure of motion)

- Key element is used of multi-resolution image representation to allow optical flow constraint equation even when motion is fairly large



Hierarchical Motion Estimation



Smoothness

- Pyramid construction smoothens out discontinuities to provide better gradient-based estimation
  - Smooths over boundaries, but dealt with in later estimations

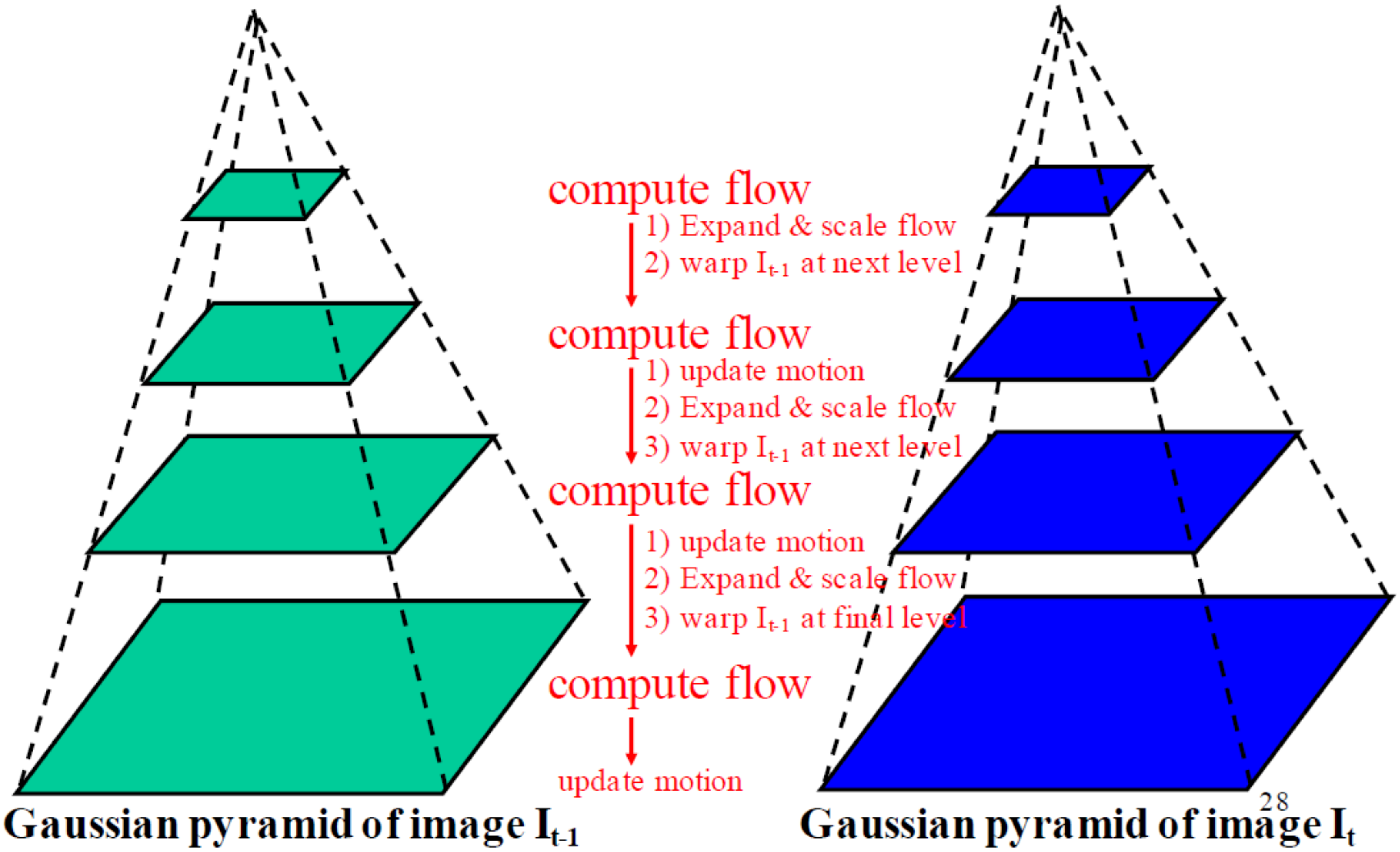
Displacement Reduction in Pyramid

- We can compute optical flow at top level of pyramid, but
  - Low image resolution (reduced size), so not true answer for original sized image
  - Motion may start at lower (larger) levels
- Provides result for initial estimate (has to be refined)

Hierarchical Motion Estimation

- Overview of approach for pair of images (I_{t-1}, I_t)
  - Generate multi-resolution Gaussian pyramid of image
    - Iteratively reduce image
  - Perform local estimation of displacements
    - Compute optical flow
    - Start at smallest pyramid level
  - Expand flow and warp image at next level, then compute local flow estimation
    - Compensates for previously estimated displacement
  - Iteratively refine global optical flow

Course-to-Fine Motion Estimation



Computing Optic Flow

$$\underbrace{\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \cdots & \cdots \\ \cdots & \cdots \\ I_x^N & I_y^N \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{x^*} = \underbrace{\begin{bmatrix} -I_t^1 \\ -I_t^2 \\ \cdots \\ \cdots \\ -I_t^N \end{bmatrix}}_{b}$$

$AA^T x^* = A^T b$ Least-square approximation

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

Expanding & Scaling Motion

- The flow (u,v) is expanded to match the size of the next (bigger) pyramid image
  - Using 2x spatial expand operations (see up-sampling in pyramid lecture)
  - Also must expand the flow magnitude, as the vectors are in pixel units (and image was just doubled in size)
    - $u' = expand(u) * 2$
    - $v' = expand(v) * 2$

Image Warping

- Given image I_{t-1} at next (larger) pyramid level and the corresponding (u', v') motion estimate for that level, use motion vectors to warp I_{t-1} into W

- W and the corresponding level I_t should appear to "be close"
  - Enough to compute valid optical flow

Updating Motion

- Compute new, local motion flow at this level
  - From W to I_t
- Refine previous estimate (that was expanded/scaled and used from previous level for W)

$$u = u' + \text{local\_}u$$
$$v = v' + \text{local\_}v$$

Updated motion estimate at this level

Motion using warped image W and I_t at this level

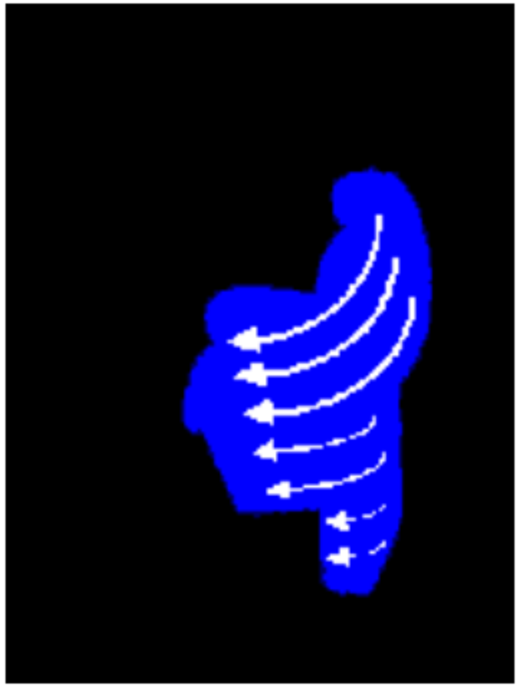
Expanded motion from previous level (used for warping this level into W)

Motion Templates

- Even bad motion data can evoke a strong percept
- Recognize holistic “patterns of motion”
  - No tacking of structure features (hand or elbows)

Representation Theory

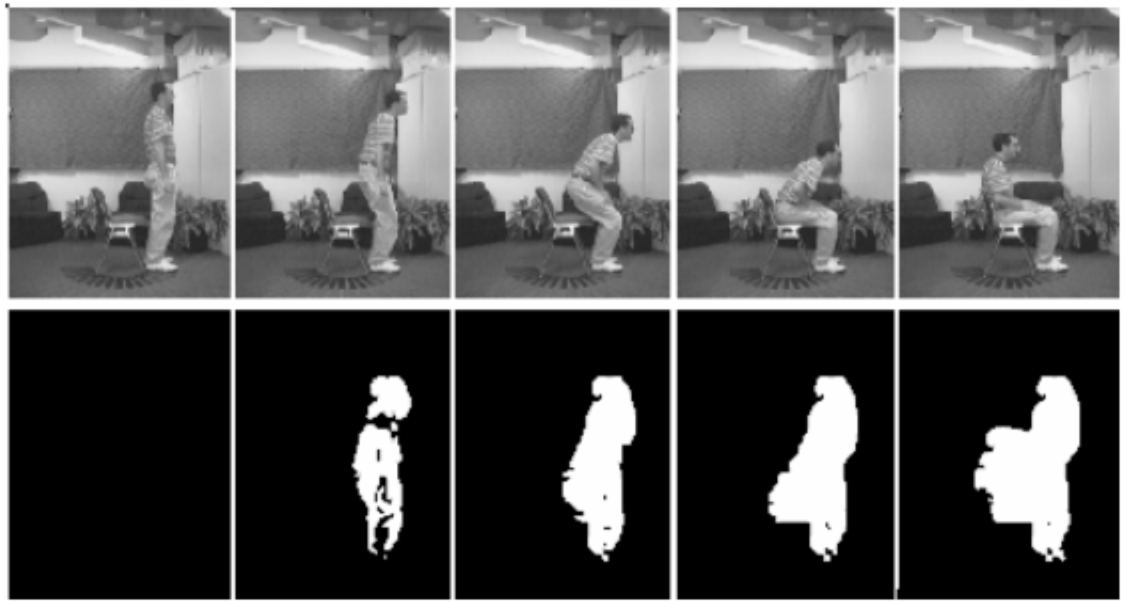
- Decompose motion
- Where
  - Spatial pattern of where motion occurred
  - Motion energy image (MEI)
- How
  - Progression of how the motion is moving
  - Motion history image (MHI)



Temporal Template

- MHI (and MEI) is a static image
  - Value at each pixel is some function of the motion at that pixel
  - MHI: pixel records temporal history
  - MEI: pixel records presence of motion

Cumulative Motion Image



- Cumulative motion presence
  - Image differencing
- Sweeps out particular region
- Shape can be used to suggest action

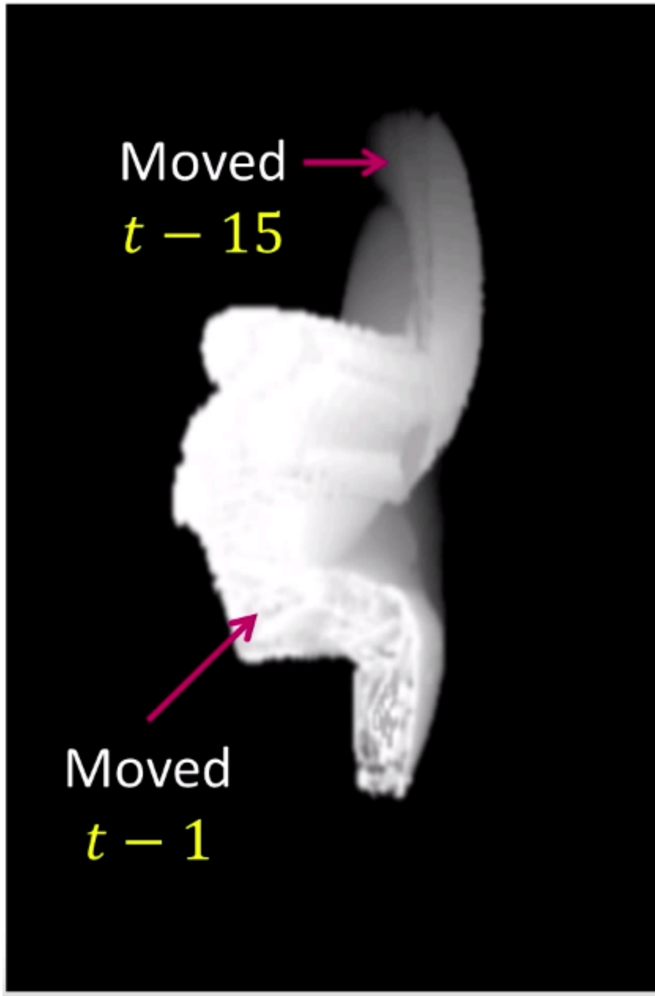
Motion Energy Image (MEI)

- Cumulative motion images
  - $E_\tau(x, y, z) = \bigcup_{i=0}^{\tau-1} D(x, y, t-i)$

- Duration τ defines temporal extent

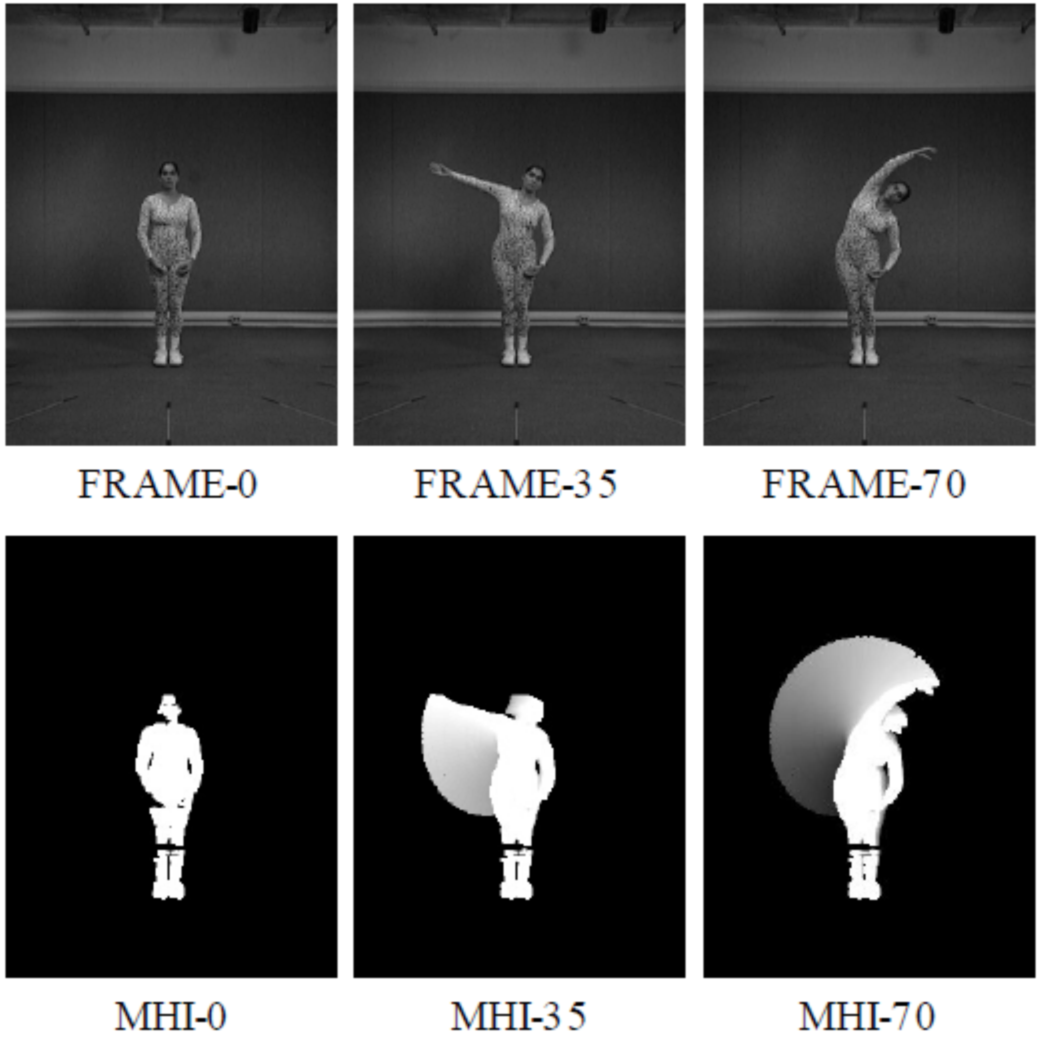
Motion History Image (MHI)

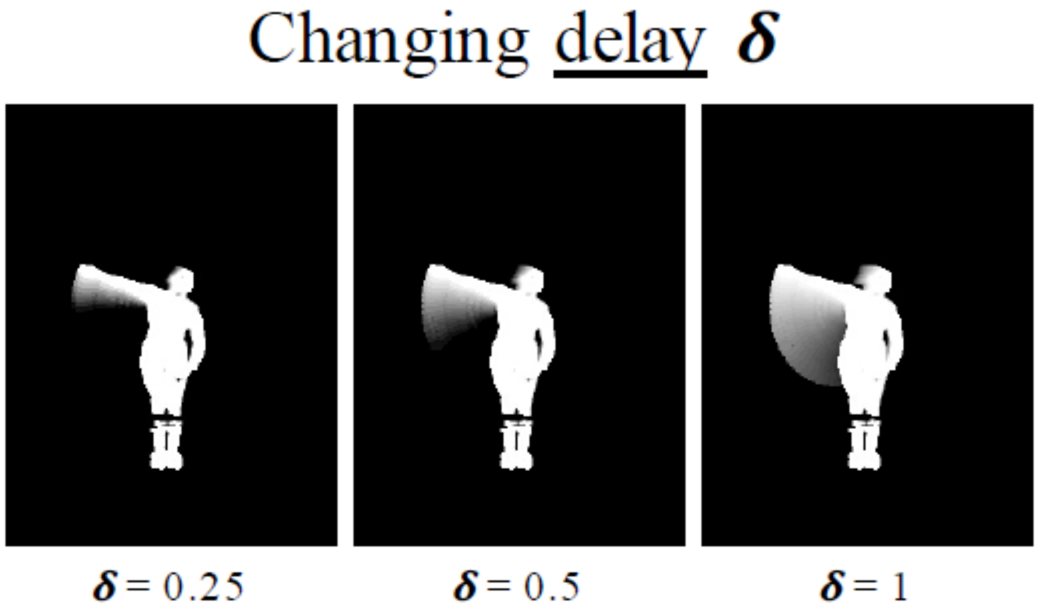
- Represent how motion is moving
- Pixel intensity is function of temporal history at that point
- Simple replace-and-decay operator with timestamp δ



- $MHI_\delta(x,y,t) = \begin{cases} \delta & \mathit{if}\ I(x,y,t) \quad \mathit{is\ moving} \\ \max(MHI_\delta(x,y,t-1)-1, 0) & \mathit{else\ if}\ I(x,y,t)\ \mathit{is\ not\ moving} \end{cases}$
  - δ: delay

Silhouette Differences





Motion History Image

- MHI can be created and implemented in low illumination conditions where the structure cannot be easily detected otherwise.
- The MHI representation is not so sensitive to silhouette noises, holes, shadows, and missing parts.
- The MHI expresses the motion flow or sequence by using the intensity of every pixel in a temporal manner.

MHI (Re)Normalization

- Want “fade to black” for time duration used

$$\max\left(0, \frac{t - [\min(t) - \Delta t]}{\max(t) - [\min(t) - \Delta t]}\right)$$

$$\max(0, \frac{\overbrace{frame\#}^{\text{frame \# at pixel}} - [\min(frame\#) - 1]}{\underbrace{\max(frame\#)}_{\text{The most current frame \# (most recent)}} - [\underbrace{\min(frame\#)}_{\text{The oldest frame \# that still want to keep/display}} - 1]}$$

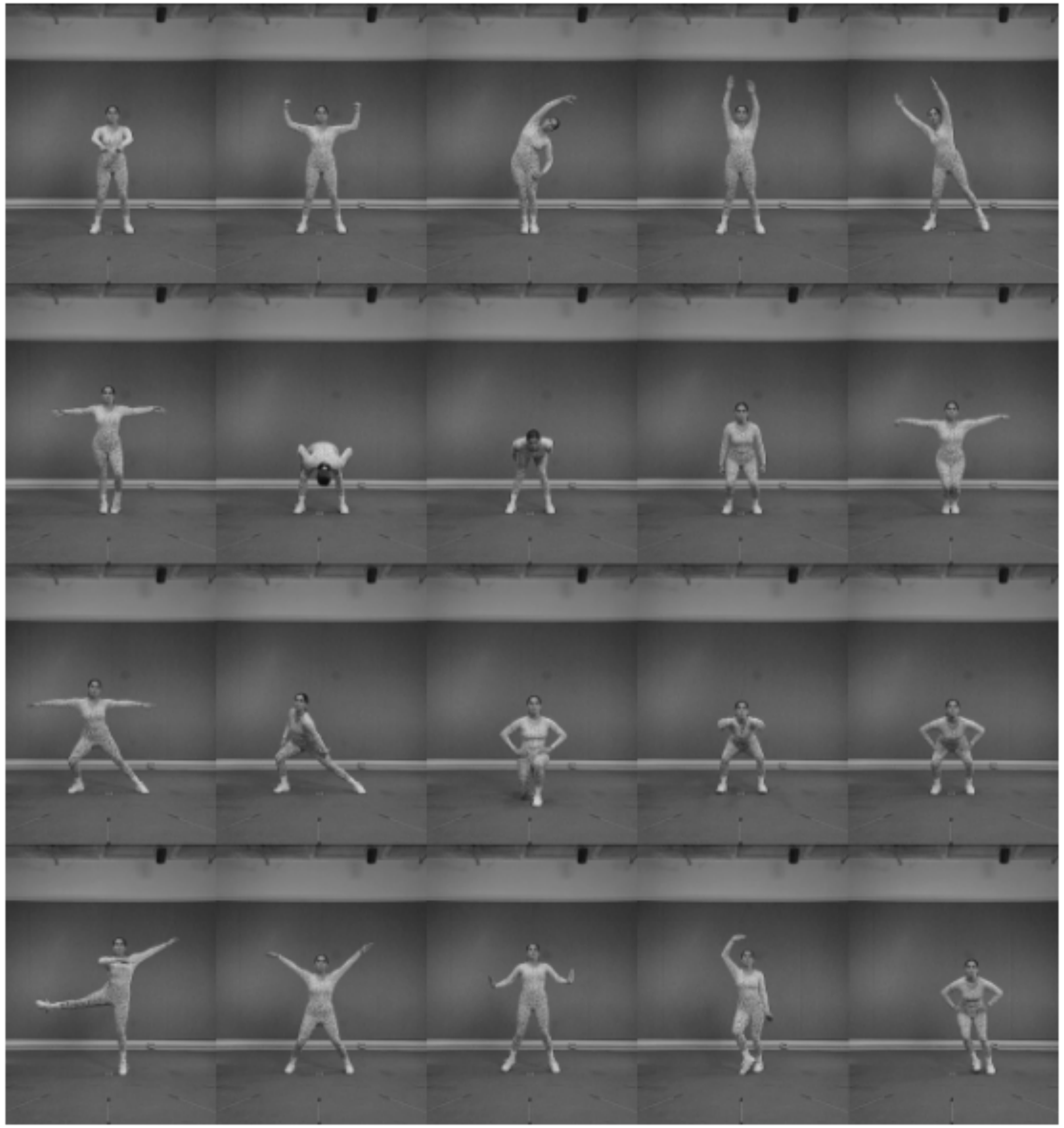
Duration

Long

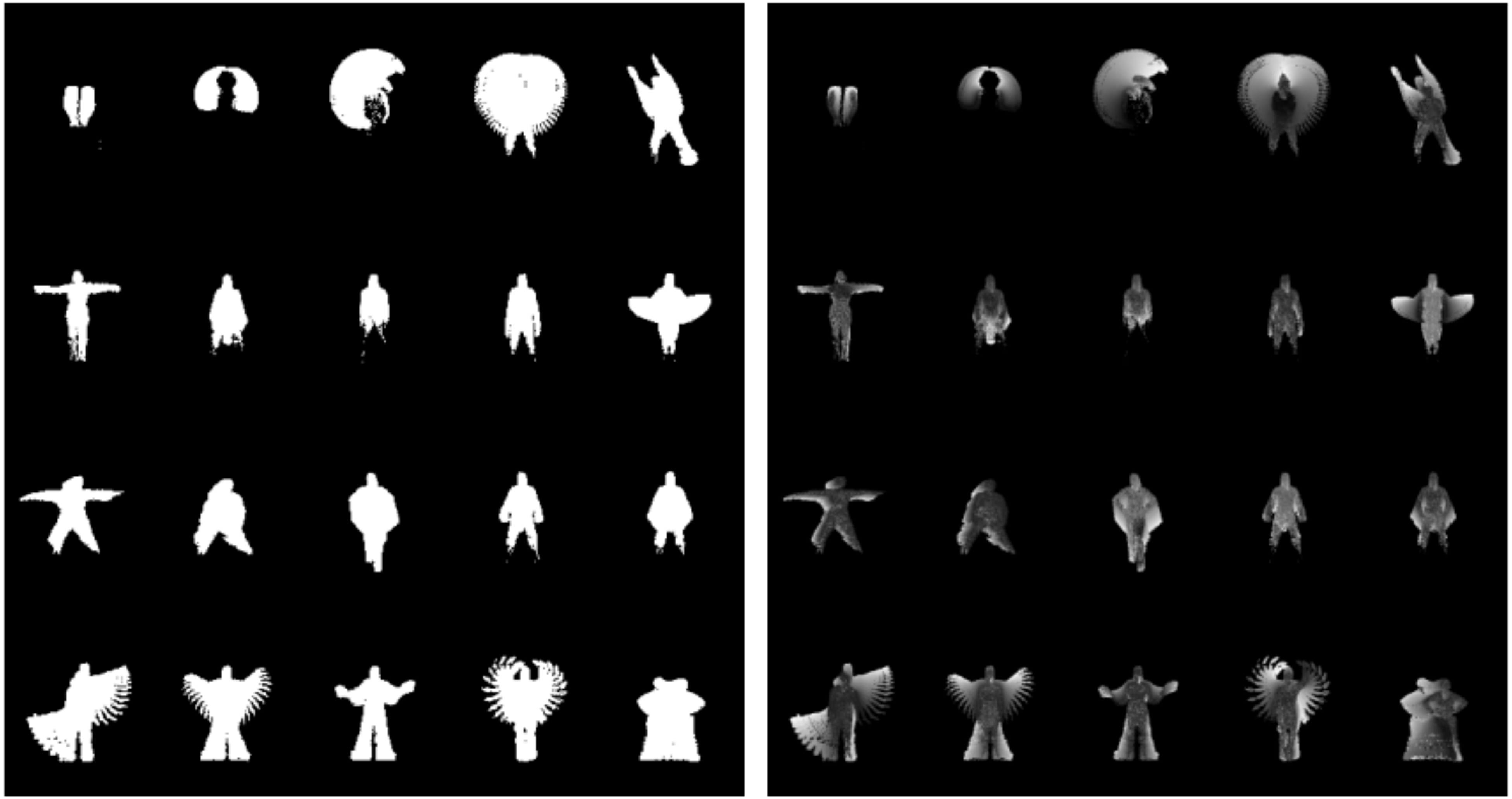
Medium

Short

Aerobics Data



MEIs and MHIs



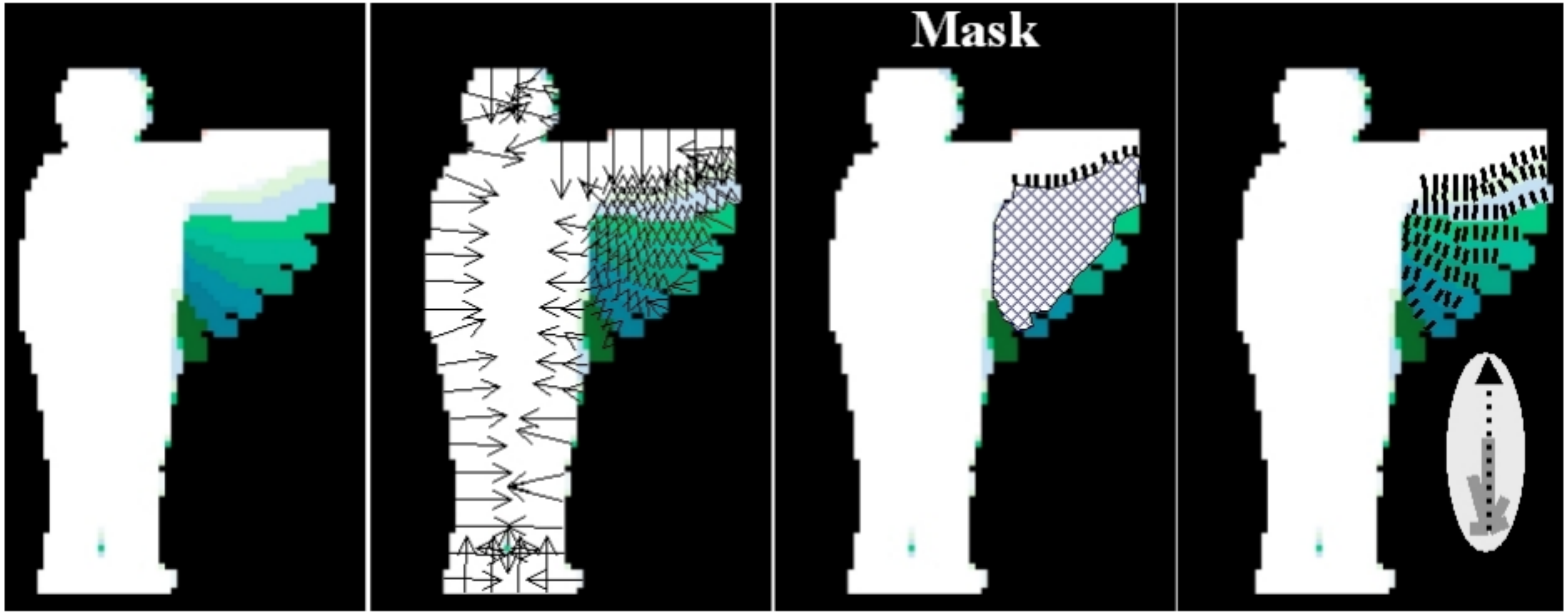
Motion Gradient

- Can perceive "direction of motion" in intensity fading of motion template
  - Downward, backward swipe

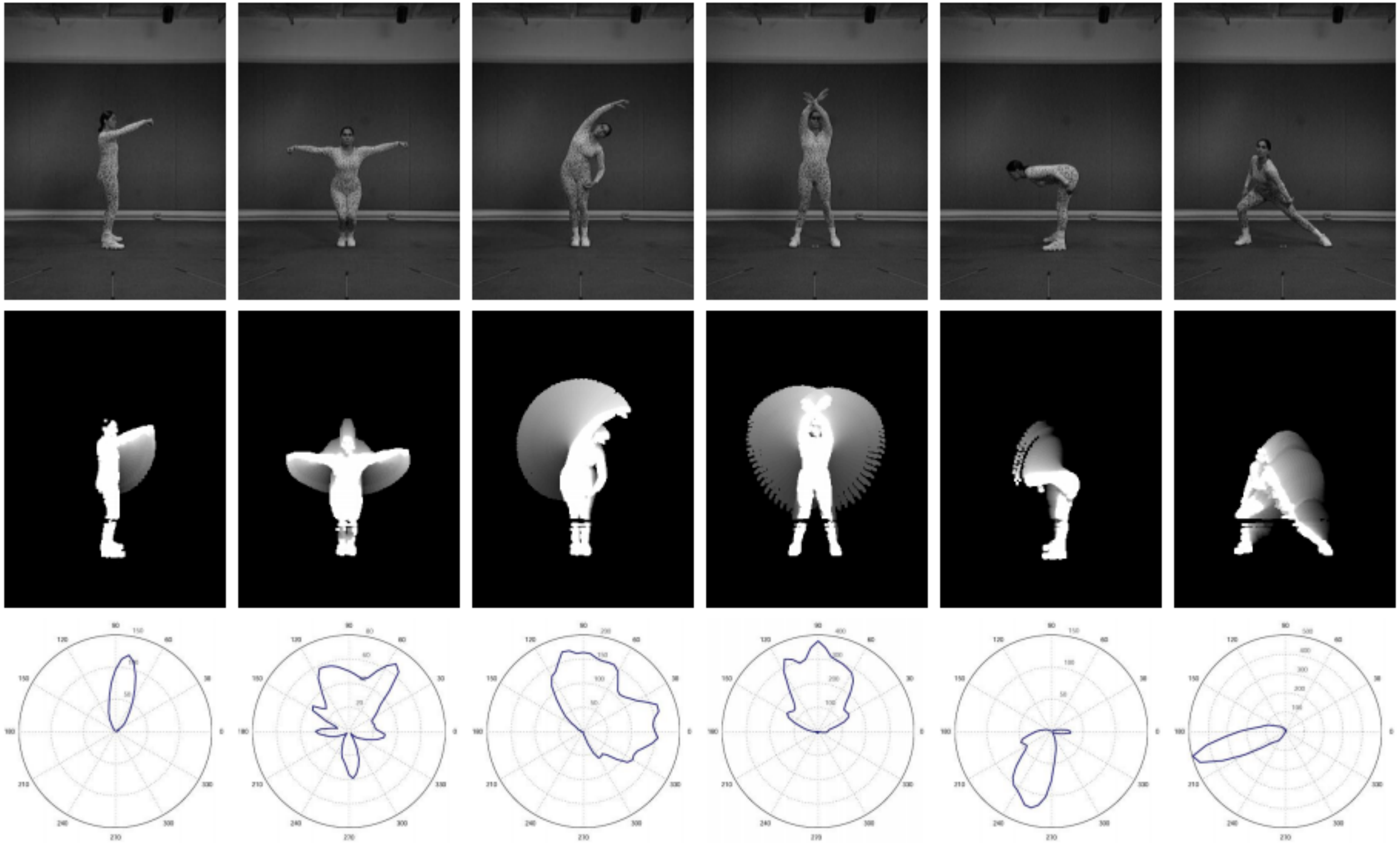
- Convolve gradient masks with template
  - Intensity gradient similar to motion flow (normal flow)

$$F_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad F_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

Motion Gradient



Motion Orientation Histograms



Effect of Occlusion