EE 584
MACHINE VISION

Motion Field and Optical Flow
- Motion Field vs Optical Flow
- Solving for Optical Flow
  - Gradient-based methods
  - Kanade Lucas Tracker
  - Parametric methods
  - Frequency-based methods

Motion Field vs Optical Flow

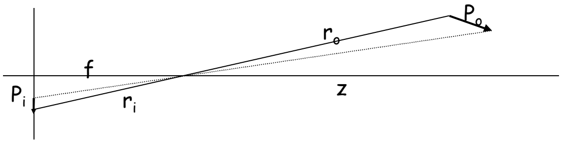
- The apparent motion of brightness patterns observed when a camera is moving relative to the objects being imaged is called the *optical flow*
- Optical flow can be totally different from motion field, which depends on the projection of moving objects on the image plane
  - e.g. barber's pole

- When
  - an object moves in front of a camera or
  - a camera moves through a stationary scene,

  some intensity changes occur in the image
- It is possible to recover the relative motion or even the shapes of objects from these intensity changes

Motion Field

- *Motion field* assigns a velocity vector to each point in the image by



$$\text{Let } \vec{v}_o = \frac{d\vec{r}_o}{dt} \qquad \vec{v}_i = \frac{d\vec{r}_i}{dt}$$

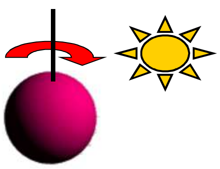
$$\frac{1}{f}\vec{r}_i = \frac{1}{\vec{r}_o \cdot \underbrace{\vec{z}}_{unit\ vec.}}\vec{r}_o \quad \overset{differentiating\ wrt\ t}{\Rightarrow} \quad \frac{1}{f}\vec{v}_i = \frac{(\vec{r}_o.\vec{z})\vec{v}_o - (\vec{v}_o.\vec{z})\vec{r}_o}{(\vec{r}_o.\vec{z})^2} = \frac{(\vec{r}_o \times \vec{v}_o) \times \vec{z}}{(\vec{r}_o.\vec{z})^2}$$

- A vector can be assigned to every image point; these vectors constitute the motion field
- Except the boundaries of the objects, for most of the object motions (rigid, non-rigid), we expect a smooth variation between neighboring points

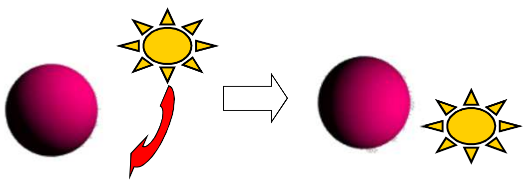
METU EE 584 Lecture Notes by A.Aydin ALATAN © 2012

Optical Flow (1/4)

- *Optical flow* is the apparent motion of brightness pattern
- Ideally, optical flow <-> motion field, but …



Non-zero motion field, zero optical flow



Zero motion field, non-zero optical flow

- Optical flow is the only observation about object motion from an image
- Except for special situations, assume optical flow is approximately equal to the motion field

Optical Flow (2/4)

- Let $E(x,y,t)$ be the irradiance at time t at the image point (x,y)
  - $u(x,y)$ and $v(x,y)$ are horizontal & vertical components of optical flow field
- We assume the image radiance to be the <u>same</u> at the next time instant for the corresponding point

$$E(x+\underbrace{u(x,y)\delta t}_{hor.displ.},\, y+\underbrace{v(x,y)\delta t}_{ver.displ.},\, t+\delta t) = E(x,y,t)$$

- A single constraint is <u>not sufficient</u> to determine both u and v
- Since motion field is continuous, Taylor expansion can be used

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

$$\Rightarrow \underbrace{\frac{\partial E}{\partial x}}_{E_x}\underbrace{\frac{dx}{dt}}_{u} + \underbrace{\frac{\partial E}{\partial y}}_{E_y}\underbrace{\frac{dy}{dt}}_{v} + \underbrace{\frac{\partial E}{\partial t}}_{E_t} = 0$$

Optical flow constraint equation

$$\Rightarrow \quad E_x u + E_y v + E_t = 0$$

METU EE 584 Lecture Notes by A.Aydin ALATAN © 2012

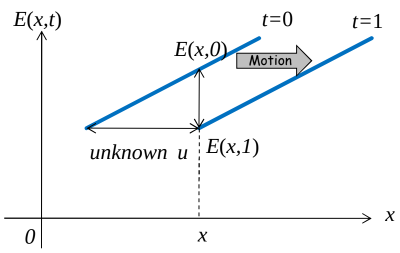
Optical Flow (3/4)

- For simplicity, consider 1-D case, $I(x,t)$,

$$E_t(x,t) \approx -(E(x,0) - E(x,1))$$

$$\Rightarrow E_x(x,t) = \frac{-E_t(x,t)}{u}$$

$$\Rightarrow E_x(x,t)u + E_t(x,t) = 0$$



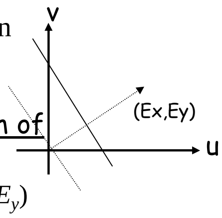
- In 2-D, the relation between displacements and spatio-temporal derivatives becomes

$$E_x u + E_y v + E_t = 0 \quad : \text{optical flow constraint equation}$$

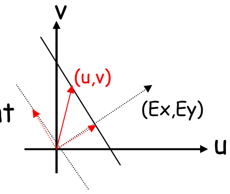
Optical Flow (4/4)

$E_x u + E_y v + E_t = 0$: optical flow constraint equation

Note that $(u\ v).(E_x\ E_y)^t = -E_t$

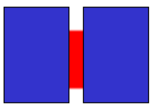


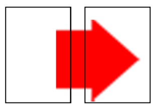
- Optical flow vector (u,v) can be written as the <u>sum of two vectors</u>,
  - one component along brightness gradient, (E_x,E_y)
  - one perpendicular to the brightness gradient



- Component perpendicular to gradient is totally unknown or unobservable (*aperture problem*),
- Component in the direction of brightness gradient

$$(u\ v).\vec{n}_{(Ex,Ey)} = (u\ v).\frac{(E_x\ E_y)^t}{\sqrt{E_x^2 + E_y^2}} = \frac{-E_t}{\sqrt{E_x^2 + E_y^2}}$$





Estimating Motion from Brightness Variation

- Gradient-based methods
  - Images and its spatio-temporal derivatives are used
    - Optical flow solution : minimize optical flow constraint
    - Pel-recursive algorithm :minimize DFD to have $D^{n+1}=D^n+U(D^n)$
- Correspondence-based (matching-type) methods
  - Finding positions of “features” at consecutive two/more images
    - Block/region-based algorithms :
    - Edge/corner matching :
- Parametric methods
  - Fitting a parametric model to motion vectors in a region
    - Affine : $x'=a_0 + a_1x + a_2y$ $\quad y'=a_3 + a_4x + a_5y$
    - Perspective : $x'=(a_0 + a_1x + a_2y)/(1 + a_7x + a_8y)$
      $y'=(a_3 + a_4x + a_5y)/(1 + a_7x + a_8y)$
- Frequency-based Methods

Smoothness of Optical Flow

- In order to solve optical flow equation, we should introduce an extra constraint, as
  - Rigid body assumption (to be analyzed)
  - Smoothness of neighboring motion vectors

$$e_s = \iint \left(\left(u_x^2 + u_y^2 \right) + \left(v_x^2 + v_y^2 \right) \right) dxdy$$

- Error in optical flow equation can also be written as

$$e_c = \iint \left(E_x u + E_y v + E_t \right)^2 dxdy$$

- The problem can be formulated as minimization of e

$$e = e_s + \lambda e_c$$

- Minimization these equations can be achieved by using a discrete version of this integral equation

Estimating Optical Flow : Gradient-based

- Instead of solving the continuous formulation in discrete case, discretize the problem as

$$e = \sum_i \sum_j s_{ij} + \lambda c_{ij} \quad where \quad \begin{aligned} c_{ij} &= \left(E_x u_{ij} + E_y v_{ij} + E_t \right)^2 \\ s_{ij} &= \frac{1}{4} \left(\left(u_{i+1j} - u_{ij} \right)^2 + \left(u_{ij+1} - u_{ij} \right)^2 + \left(v_{i+1j} - v_{ij} \right)^2 + \left(v_{ij+1} - v_{ij} \right)^2 \right) \end{aligned}$$

- Differentiating constraint e wrt u_{kl} and v_{kl}, we obtain :

$$\frac{\partial e}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(E_x u_{kl} + E_y v_{kl} + E_t)E_x \quad (\bar{u}_{kl} : \text{local average of u})$$

$$\frac{\partial e}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(E_x u_{kl} + E_y v_{kl} + E_t)E_y \quad (\bar{v}_{kl} : \text{local average of v})$$

- Equating these derivatives to zero,

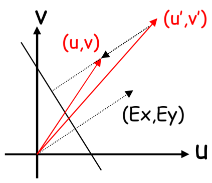
$$(1 + \lambda E_x^2) u_{kl} + \lambda E_x E_y \quad v_{kl} = \bar{u}_{kl} - \lambda E_x E_t$$

$$\lambda E_x E_y \quad u_{kl} + (1 + \lambda E_y^2) v_{kl} = \bar{v}_{kl} - \lambda E_y E_t$$

Estimating Optical Flow : Gradient-based

- Solving these equations for u and v [Horn and Schunck 81] :

$$u_{kl}^{n+1} = \bar{u}_{kl}^{n} - \underbrace{\frac{\left(E_x \bar{u}_{kl}^{n} + E_y \bar{v}_{kl}^{n} + E_t\right)}{(1+\lambda(E_x^2 + E_y^2))}}_{adjustment\ factor} E_x \qquad v_{kl}^{n+1} = \bar{v}_{kl}^{n} - \underbrace{\frac{\left(E_x \bar{u}_{kl}^{n} + E_y \bar{v}_{kl}^{n} + E_t\right)}{(1+\lambda(E_x^2 + E_y^2))}}_{adjustment\ factor} E_y$$



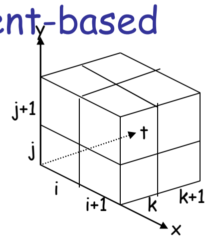
- Note that the adjustment factor at each iteration is in the direction of (E_x, E_y)

- In order to implement this algorithm, the only missing elements are spatial and time derivatives of the brightness function

METU EE 584 Lecture Notes by A.Aydin ALATAN © 2012

Estimating Optical Flow : Gradient-based

- Three first partial derivatives can be obtained as :



$$\frac{\partial E(x,y,t)}{\partial x} = E_x \approx \frac{1}{4\varepsilon}\{(E_{i+1,j+1,k} - E_{i,j+1,k}) + (E_{i+1,j,k} - E_{i,j,k}) + (E_{i+1,j+1,k+1} - E_{i,j+1,k+1}) + (E_{i+1,j,k+1} - E_{i,j,k+1})\}$$

$$\frac{\partial E(x,y,t)}{\partial y} = E_y \approx \frac{1}{4\varepsilon}\{(E_{i+1,j+1,k} - E_{i+1,j,k}) + (E_{i,j+1,k} - E_{i,j,k}) + (E_{i+1,j+1,k+1} - E_{i+1,j,k+1}) + (E_{i,j+1,k+1} - E_{i,j,k+1})\}$$

$$\frac{\partial E(x,y,t)}{\partial t} = E_t \approx \frac{1}{4\varepsilon}\{(E_{i,j+1,k+1} - E_{i,j+1,k}) + (E_{i,j,k+1} - E_{i,j,k}) + (E_{i+1,j+1,k+1} - E_{i+1,j+1,k}) + (E_{i+1,j,k+1} - E_{i+1,j,k})\}$$

- Another way of finding these partial derivatives is fitting a surface to the intensities locally and taking partial derivative analytically (results with better immunity against noise); e.g.

$$E(x,y,t) \cong a_0 + a_1 x + a_2 y + a_3 t + a_4 x^2 + a_5 y^2 + a_6 xy + a_7 xt + a_8 yt$$

Discontinuities in Optical Flow

- Except for the boundaries, smoothness assumption is
  - acceptable for translating rigid bodies
  - merely acceptable for rotating rigid bodies
  - acceptable for elastic (non-rigid) objects
- Solving optical flow equation based on smoothness assumption usually fails at moving object boundaries
- If these boundaries are known, optical flow estimate can be obtained with much better reliability; but the only way to obtain these boundaries based on motion information
  ➔ "chicken-egg problem" for moving object segmentation

Estimating Optical Flow: Kanade-Lucas Tracker

- An approach to overcome the aperture problem is to assume motion vector (u,v) remaining unchanged over the whole block, B [Lucas and Kanade 81]
- Minimize the optical flow equation within this block wrt u and v

$$\min_{u,v} \sum_{(i,j)\in B} \left(E_x^{ij} u + E^{ij}{}_y v + E^{ij}{}_t\right)^2$$

- Differentiate wrt u and v ; then, equate to zero

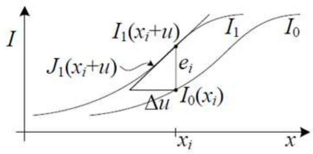
$$\sum_{(i,j)\in B} \left(E_x^{ij} u + E^{ij}{}_y v + E^{ij}{}_t\right) E_x^{ij} = 0 \qquad \sum_{(i,j)\in B} \left(E_x^{ij} u + E^{ij}{}_y v + E^{ij}{}_t\right) E_y^{ij} = 0$$

- Estimate for the optical flow for the block B is obtained as

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} \sum_{(i,j)\in B} E_x^{ij} E_x^{ij} & \sum_{(i,j)\in B} E_y^{ij} E_x^{ij} \\ \sum_{(i,j)\in B} E_x^{ij} E_y^{ij} & \sum_{(i,j)\in B} E_y^{ij} E_y^{ij} \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{(i,j)\in B} E^{ij}{}_t E_x^{ij} \\ -\sum_{(i,j)\in B} E^{ij}{}_t E_y^{ij} \end{bmatrix}$$

Estimating Optical Flow: Kanade-Lucas Tracker

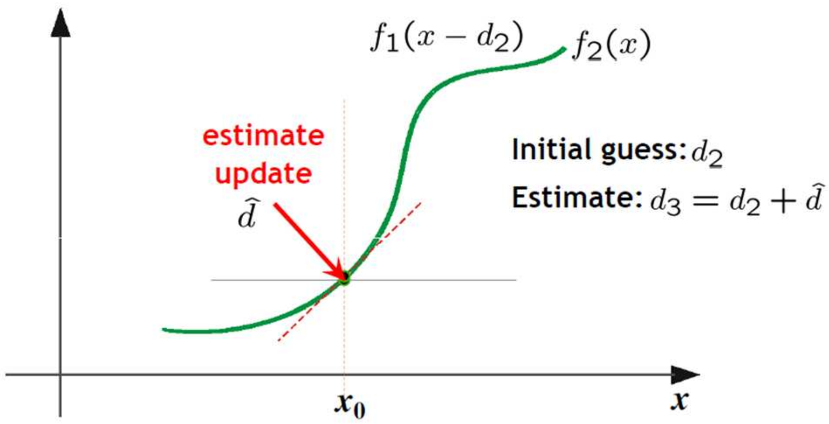
- Iterative (incremental) refinement for the optical flow is possible
- Estimate optical flow vector at each pixel using one iteration of Kanade-Lucas
- Warp the image toward the other using the estimated optical flow vector
- Refine estimate by repeating these steps



- Since linearization of constant brightness constraint is valid for small displacements, apply estimation coarse-to-fine
  - typical less than 1 pixel displacements for better approx.
- Construct image pyramids for two frames (multi-resolution)
- Estimate at the coarser resolution and propagate results to finer resolutions.

Estimating Optical Flow: Kanade-Lucas Tracker

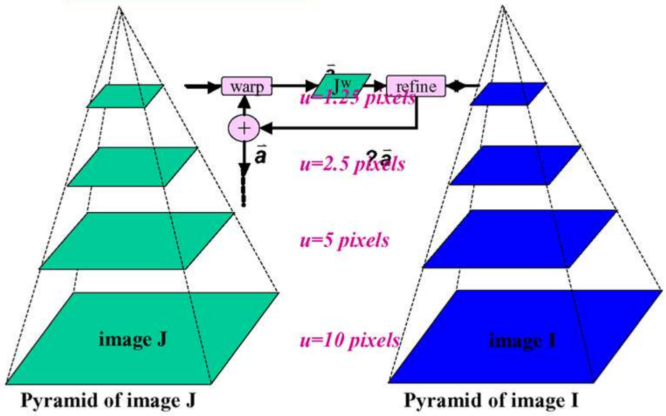
- Iterative (incremental) refinement



Estimating Optical Flow: Kanade-Lucas Tracker

Coarse-to-Fine Estimation

$I_x \cdot u + I_y \cdot v + I_t \approx 0$ ==> small *u* and *v* ...



Slides modified from M. Pollefeys Lecture Notes on Computer Vision

Estimating Optical Flow: Kanade-Lucas Tracker

Iterative and multi-resolution version of KLT

- Inner loop calculates E^{ij}_{t} iteratively, while moving on the trajectory
- Outer loop, calculates and passes motion vector estimates between resolutions

for $L = L_m$ **down to** 0 **with step of** -1

- *Location of point* **u** *on image* I^L: $\mathbf{u}^L = [p_x \ \ p_y]^T = \mathbf{u}/2^L$
- *Derivative of* I^L *with respect to x*: $I_x(x,y) = \frac{I^L(x+1,y) - I^L(x-1,y)}{2}$
- *Derivative of* I^L *with respect to y*: $I_y(x,y) = \frac{I^L(x,y+1) - I^L(x,y-1)}{2}$
- *Spatial gradient matrix*: $G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\,I_y(x,y) \\ I_x(x,y)\,I_y(x,y) & I_y^2(x,y) \end{bmatrix}$
- *Initialization of iterative L-K*: $\overline{\nu}^0 = [0 \ \ 0]^T$
- **for** $k = 1$ **to** K **with step of** 1 (or until $\|\overline{\eta}^k\| <$ accuracy threshold)
  - *Image difference*: $\delta I_k(x,y) = I^L(x,y) - J^L(x + g_x^L + \nu_x^{k-1}, y + g_y^L + \nu_y^{k-1})$
  - *Image mismatch vector*: $\overline{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x,y)\,I_x(x,y) \\ \delta I_k(x,y)\,I_y(x,y) \end{bmatrix}$
  - *Optical flow (Lucas-Kanade)*: $\overline{\eta}^k = G^{-1}\,\overline{b}_k$
  - *Guess for next iteration*: $\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$
- **end of for-loop on** k
- *Final optical flow at level L*: $\mathbf{d}^L = \overline{\nu}^K$
- *Guess for next level* $L-1$: $\mathbf{g}^{L-1} = [g_x^{L-1} \ \ g_y^{L-1}]^T = 2\,(\mathbf{g}^L + \mathbf{d}^L)$

end of for-loop on L

Final optical flow vector: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

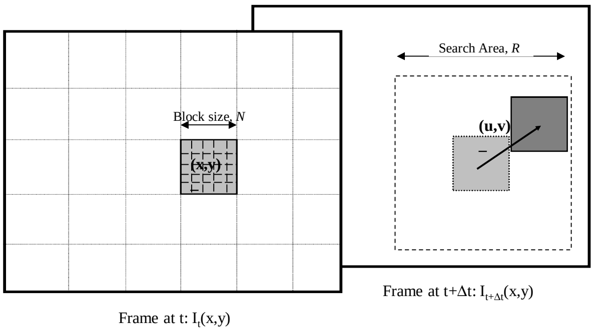
Location of point on J: $\mathbf{v} = \mathbf{u} + \mathbf{d}$

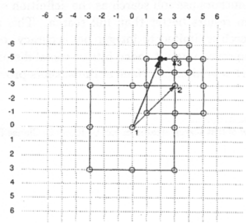
Solution: The corresponding point is at location **v** on image J

Estimating Optical Flow : Block Matching

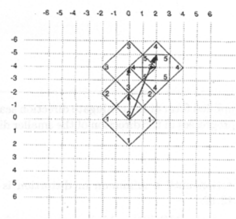
Block-based (match-based) algorithms

- Find intensity differences (or correlation) between blocks in consecutive frames
- Search methodology within the search area differs
- Sub-pixel localization is possible via interpolation





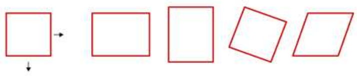
3-step search



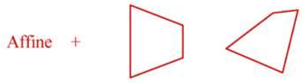
Log search

Estimating Optical Flow : Parametric

- Instead using a displacement model for each point (u,v), fit a parametric model to all motion vectors in a region
  - Affine : $x' = a_0 + a_1x + a_2y$
    $y' = a_3 + a_4x + a_5y$



  - Perspective : $x' = (a_0 + a_1x + a_2y)/(1 + a_7x + a_8y)$
    $y' = (a_3 + a_4x + a_5y)/(1 + a_7x + a_8y)$



- The unknown parameters of each region can be solved
  - Ideally, correct parameters should yield

    $E_x(x'-x) + E_y(y'-y) + E_t = 0$
  - Least squares solution over all pixels within region

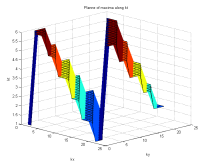
$$\min_a \sum (E_x(x'-x) + E_y(y'-y) + E_t)^2$$

Frequency-based Approaches

- These methods rely on the assumption that the <u>whole</u> observed motion is translational
  - Acceptable for moving cameras
- Spatio-temporal Frequency
  - Spatio-temporal frequency content of a video from a translating camera contains non-zero energies <u>only</u> in a specific planar region

$$f(x,y,t) = f(x-ut, y-vt, 0)$$
$$= f(x,y,0) * \delta(x-ut, y-vt)$$
$$\Leftrightarrow F(w_x, w_y, w_t) = F(w_x, w_y)\,\delta(uw_x + vw_y + w_t)$$

- This method find the plane, whose orientation is determined by the unknown motion parameters (u,v), after taking 3-D FFT of two frames



METU EE 584 Lecture Notes by A.Aydin ALATAN © 2012

Frequency-based Approaches

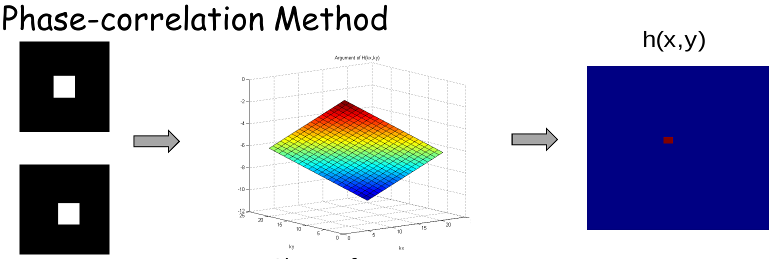
- Phase-correlation Method
  - Relation between 2-D Fourier Transforms of two consecutive frames, having a translational motion in between, has specific properties.

$$f_2(x,y) = f_1(x-ut, y-vt) \Rightarrow F_2(w_x, w_y) = e^{-j\frac{2\pi u w_x}{N_x}} e^{-j\frac{2\pi v w_y}{N_y}} F_1(w_x, w_y)$$
$$\Rightarrow \frac{F_1(w_x,w_y) F_2^*(w_x,w_y)}{\left|F_1(w_x,w_y) F_2^*(w_x,w_y)\right|} = e^{j\frac{2\pi u w_x}{N_x}} e^{j\frac{2\pi v w_y}{N_y}} = H(w_x, w_y)$$
$$\Rightarrow h(x,y) = \delta(x-u, y-v)$$

- Normalized product of 2-D Fourier Transforms of the two consecutive frames should yield a peak (impulse) in the spatial (image) domain.
  - location of this peak is determined by the motion parameters of the camera.
- This algorithm can be applied in
  - sub-pixel resolution by fitting a quadric to the peak and finding its maxima
  - local blocks and the results of each block can be merged robustly

Frequency-based Approaches

- Phase-correlation Method

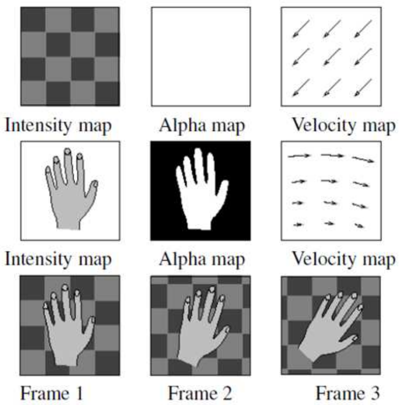


Phase of $H(w_x,w_y)$

- Practical Problems:
  - Non-circular shift effects ➔ Windowing
  - Non-Bandlimited Signals ➔ High-pass filtering
  - Peak Spreading ➔ Selection of local maxima
  - Independently Moving Objects ➔ Select correct peak after reg.
  - Subpixel Resolution ➔ Surface fit to $h(x,y)$ & find its peak
  - Local Partitioning ➔ Ability to fit local deformations

Layered Motion Estimation

- It is possible to estimate motion better, if pixels are grouped into objects (layers)
- At each layer an alpha map defines the boundaries, while parametric motion is assumed within layers



Layered Motion Estimation

- Algorithms alternate between estimating optical flow and segmenting them into coherent motion
- Estimate parametric motion at disjoint patches and cluster based on motion parameters [Wang&Adelson 1994]