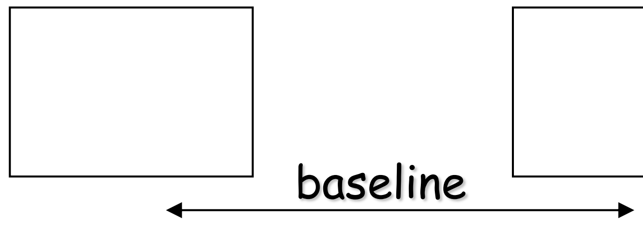
Spatial track: motion modeling

Virginio Cantoni

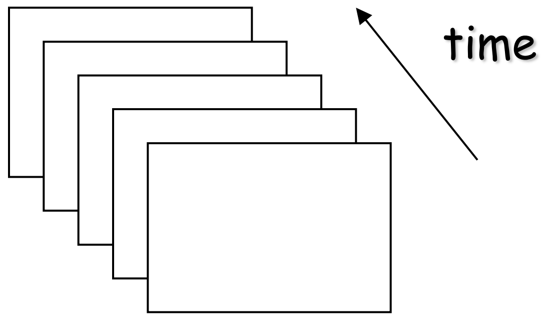
Laboratorio di Visione Artificiale
Università di Pavia
Via A. Ferrata 1, 27100 Pavia
virginio.cantoni@unipv.it
http://vision.unipv.it/va

Comparison between Motion Analysis and Stereo

- Stereo: Two or more frames



- Motion: N frames



•The baseline is usually larger in stereo than in motion:

  •Motion disparities tend to be smaller

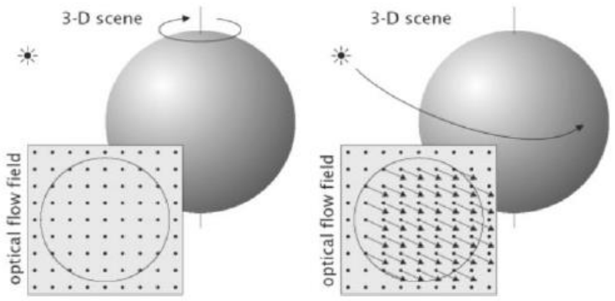
•Stereo images are taken at the same time:

  •Motion disparities can be due to scene motion

  •There can be more than one transformation btw frames

Ill-posed problem

- As an illustrative example, positively using this limitation to attract attention, consider the barber-shop banner usually displayed outdoors.
- Typically, a rotation movement of a 3-coloured striped pattern on a cylinder, perceptually suggests that the whole pattern is translated vertically upwards.
- In some cases, two helices (clockwise and counter-clockwise) appear under rotation to clash in the middle of the cylinder. In fact, a rotational movement is the only one really existing since the cylinder has a fixed size.



Why Multitude of Formulations?

- How is the camera moving?
  - The camera can be stationary
  - execute simple translational motion
  - undergo general motion with both translation and rotation
- How many moving objects are there?
  - The object(s) can be stationary
  - execute simple 2D motion parallel to the image plane
  - undergo general motion with both 3D translation and rotation
    - Which directions are they moving in?
    - How fast are they moving?
    - Can we recognize their type of motion (e.g. walking, running, etc.)?
- The camera motion may be known or unknown
- The shape of the object may be known or unknown
- The motion of the object may be known or unknown
- etc. etc. ...

Classes of Techniques

- Feature-based methods
  - Extract visual features (corners, textured areas) and track them
  - Sparse motion fields, but possibly robust tracking
  - Suitable especially when image motion is large (10s of pixels)
- Direct-methods
  - Directly recover image motion from spatio-temporal image brightness variations
  - Global motion parameters directly recovered without an intermediate feature motion calculation
  - Dense motion fields, but more sensitive to appearance variations
  - Suitable for video and when image motion is small (< 10 pixels)

Global Flow

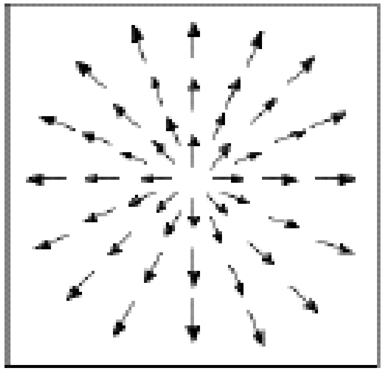
- Dominant Motion in the image
  - Motion of all points in the scene
  - Motion of most of the points in the scene
  - A Component of motion of all points in the scene
- Global Motion is caused by
  - Motion of sensor (Ego Motion)
  - Motion of a rigid scene
- Estimation of Global Motion can be used to
  - Video Mosaics
  - Image Alignment (Registration)
  - Removing Camera Jitter
  - Tracking (By neglecting camera motion)
  - Video Segmentation etc.

Motion Detection and Estimation in Literature

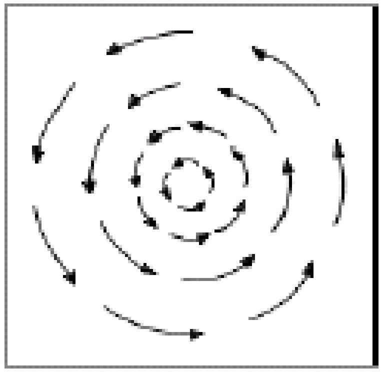
- Image differencing
  - based on the thresholded difference of successive images
  - difficult to reconstruct moving areas
- Background subtraction
  - foreground objects result by calculating the difference between an image in the sequence and the background image (previously obtained)
  - remaining task: determine the movement of these foreground objects between successive frames
- Block motion estimation
  - Calculates the motion vector between frames for sub-blocks of the image
- Optical Flow

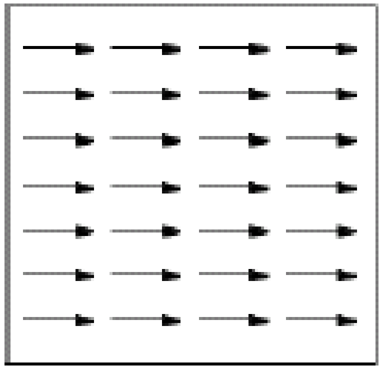
Motion Field (MF)

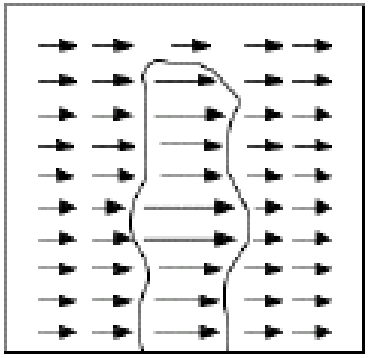
- The MF assigns a velocity vector to each pixel in the image.
- These velocities are INDUCED by the RELATIVE MOTION btw the camera and the 3D scene
- The MF can be thought as the *projection* of the 3D velocities on the image plane.


Forward motion

M


Rotation


Horizontal translation


Closer objects appear to move faster!!

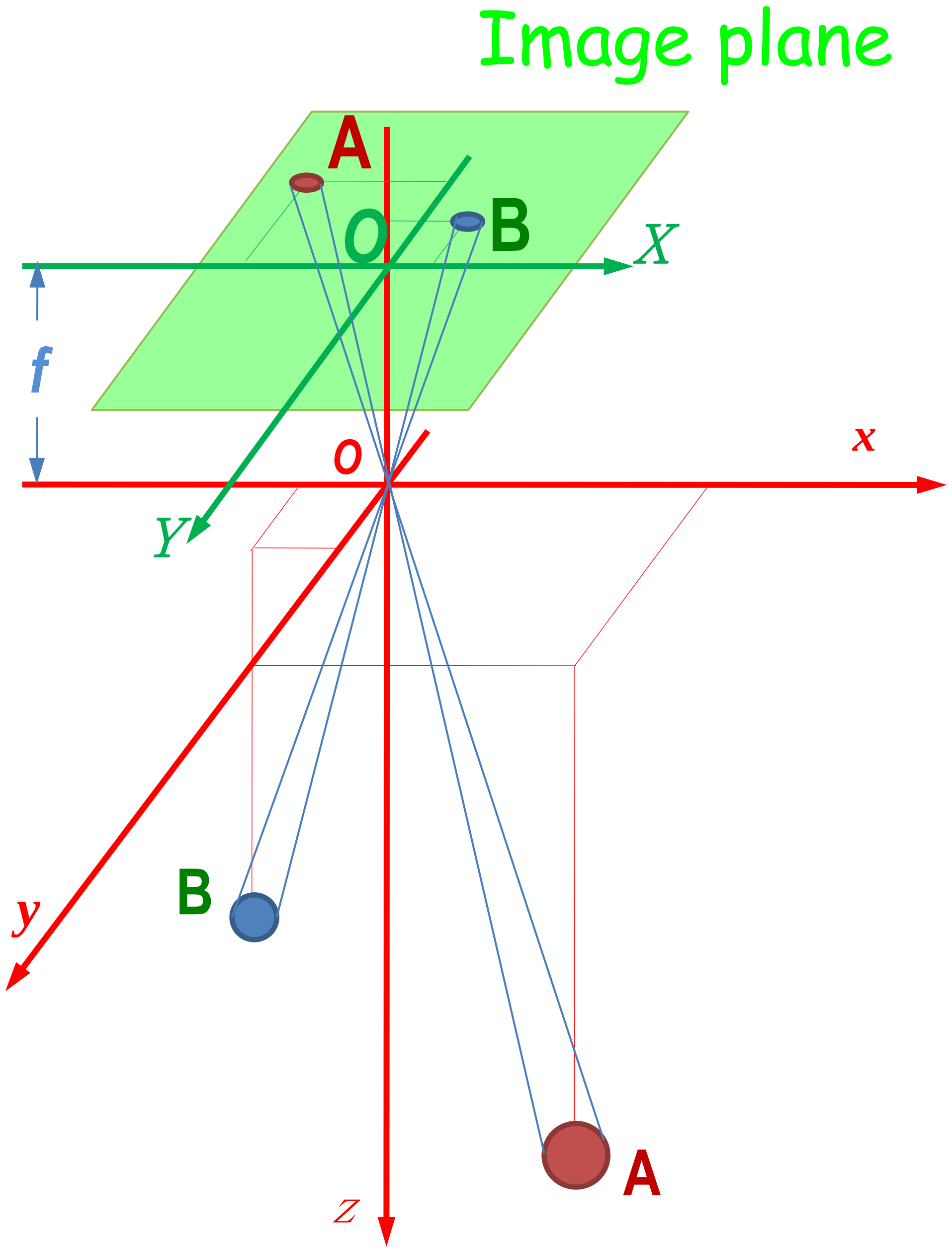
Shape from.....

Egomotion

$$\frac{Y}{y} = -\frac{f}{z} \qquad Y = -\frac{f\,y}{z}$$

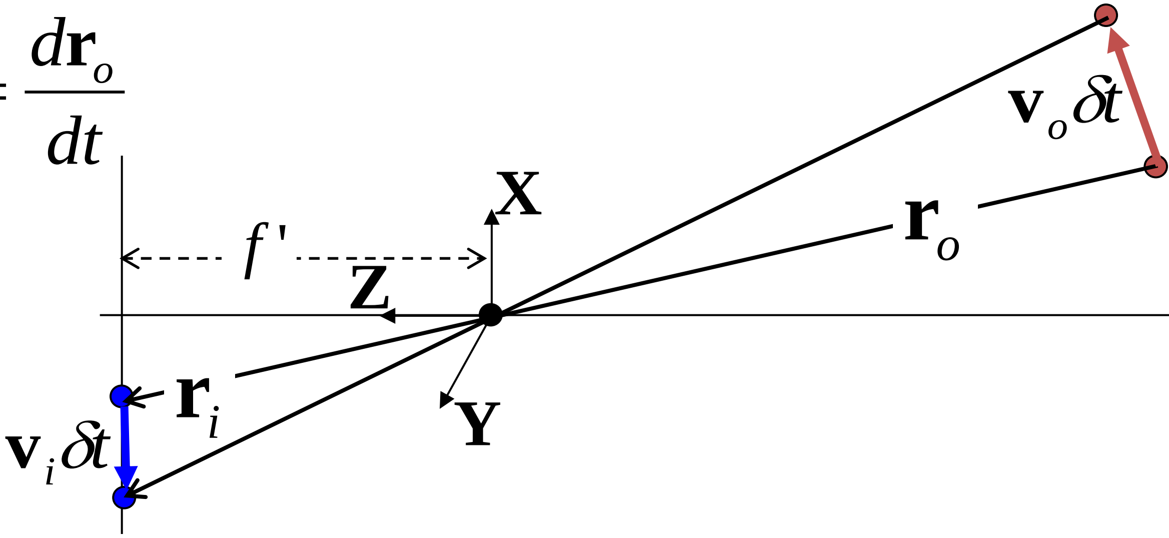
$$\frac{\partial Y}{\partial z} = \frac{yf}{z^2} = -\frac{Y}{z}$$

$$z = -\frac{Y\,\partial z}{\partial Y}$$



Motion Field

- Image velocity of a point moving in the scene
- Scene point velocity: $\mathbf{v}_o = \frac{d\mathbf{r}_o}{dt}$
- Image velocity: $\mathbf{v}_i = \frac{d\mathbf{r}_i}{dt}$



- Perspective projection: $\frac{\mathbf{r}_i}{f'} = \frac{\mathbf{r}_o}{\mathbf{r}_o \cdot \mathbf{Z}}$
- Motion field

$$\mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} = f' \frac{(\mathbf{r}_o \cdot \mathbf{Z})\mathbf{v}_o - (\mathbf{v}_o \cdot \mathbf{Z})\mathbf{r}_o}{(\mathbf{r}_o \cdot \mathbf{Z})^2} = f' \frac{(\mathbf{r}_o \times \mathbf{v}_o) \times \mathbf{Z}}{(\mathbf{r}_o \cdot \mathbf{Z})^2}$$

Quotient rule:
D(f/g) = (g f' – g' f)/g^2

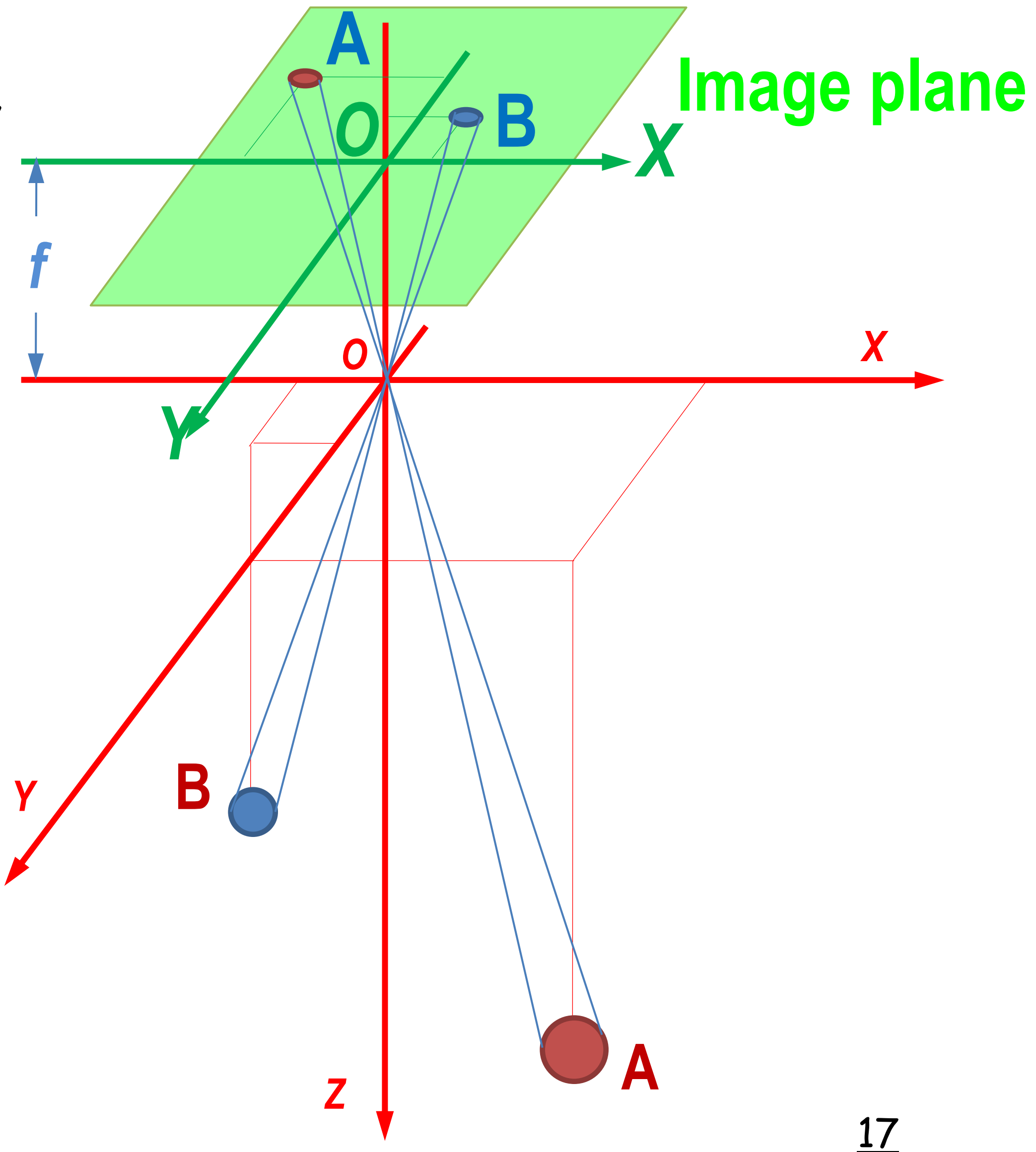
Preliminaries for motion analysis

- If *A* or B are moving objects with velocity components v_x, v_y, v_z in 3D space, the corresponding velocity of the *A* or B image points, may be computed as follows:

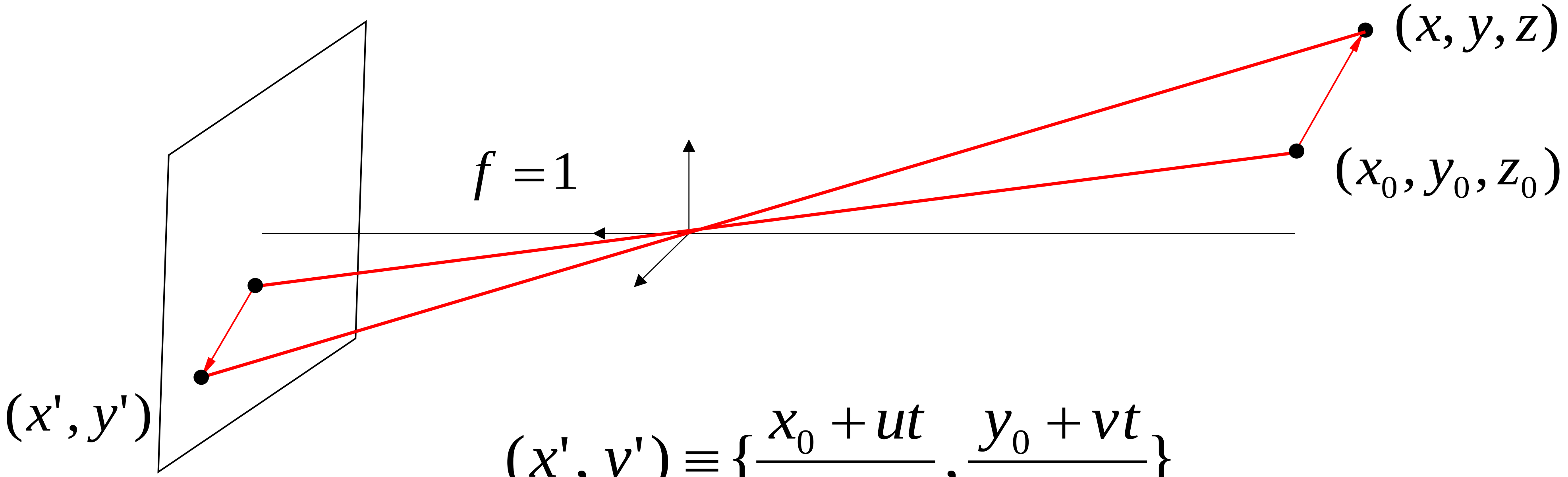
$$v_X = \frac{f}{z}\left(\frac{x}{z}v_z - v_x\right)$$

$$v_Y = \frac{f}{z}\left(\frac{y}{z}v_z - v_y\right)$$

- The object speed in the scene is not known a-priori so that it must be estimated by the detected movement of the object projection on the image.
- Unfortunately, this problem is ill-posed since it is seldom possible to compute the object speed in space only knowing the planar displacement of its projections.



Focus of Expansion-Contraction (FOE-FOC)



$$(x', y') \equiv \{\frac{x_0 + ut}{z_0 + wt}, \frac{y_0 + vt}{z_0 + wt}\}$$

- As t varies the image point moves along a straight line in the image
- Focus of Expansion: Lets backtrack time or $(t \to -\infty)$

$$\boxed{(x', y') \equiv \{\frac{u}{w}, \frac{v}{w}\}}$$

Motion modeling

- In its dynamic mode, space modeling addresses motion. In many applications, a significant feature of the scene to be analyzed is the movement of some objects during a time interval.
- Such apparent movements may be due either to the image sensor, as in an airplane photographic campaign (*egomotion*), or to some scene components, as cars in a road scenario or both.
- First, the camera is assumed to remain still, next, egomotion is introduced and compensated for, leading back to the former analysis.

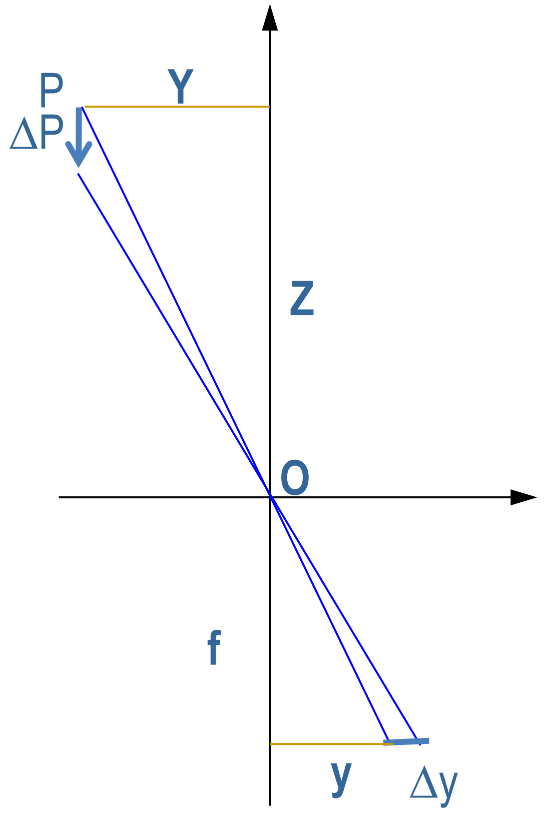
Egomotion

- the optical flow is istrumental at evaluating the shape and position of still components from their apparent motion due to the camera movement (egomotion). The sketch shows a camera downwards shift along the **Z** axis.
- Thanks to the relativity of perception it is equivalent to assume that the camera is still and the scene moves in the opposite direction along the **Z** axis. In this way, while the **P** point belonging to the **ZY** plane moves vertically down by **ΔP**, its corresponding image point P moves along the Y axis by ΔP. Considering the triangle similarity:

  $$Y = f\frac{\mathbf{Y}}{\mathbf{Z}} \quad \text{and} \quad \frac{dY}{Y} = -\frac{d\mathbf{Z}}{\mathbf{Z}}$$

  so that the distance **Z** may be derived considering that d**Z** is the known motion of the camera and (X, dX) is determined from the image:

  $$\mathbf{Z} = -\frac{d\mathbf{Z}}{dY}Y$$



Motion field + camera motion

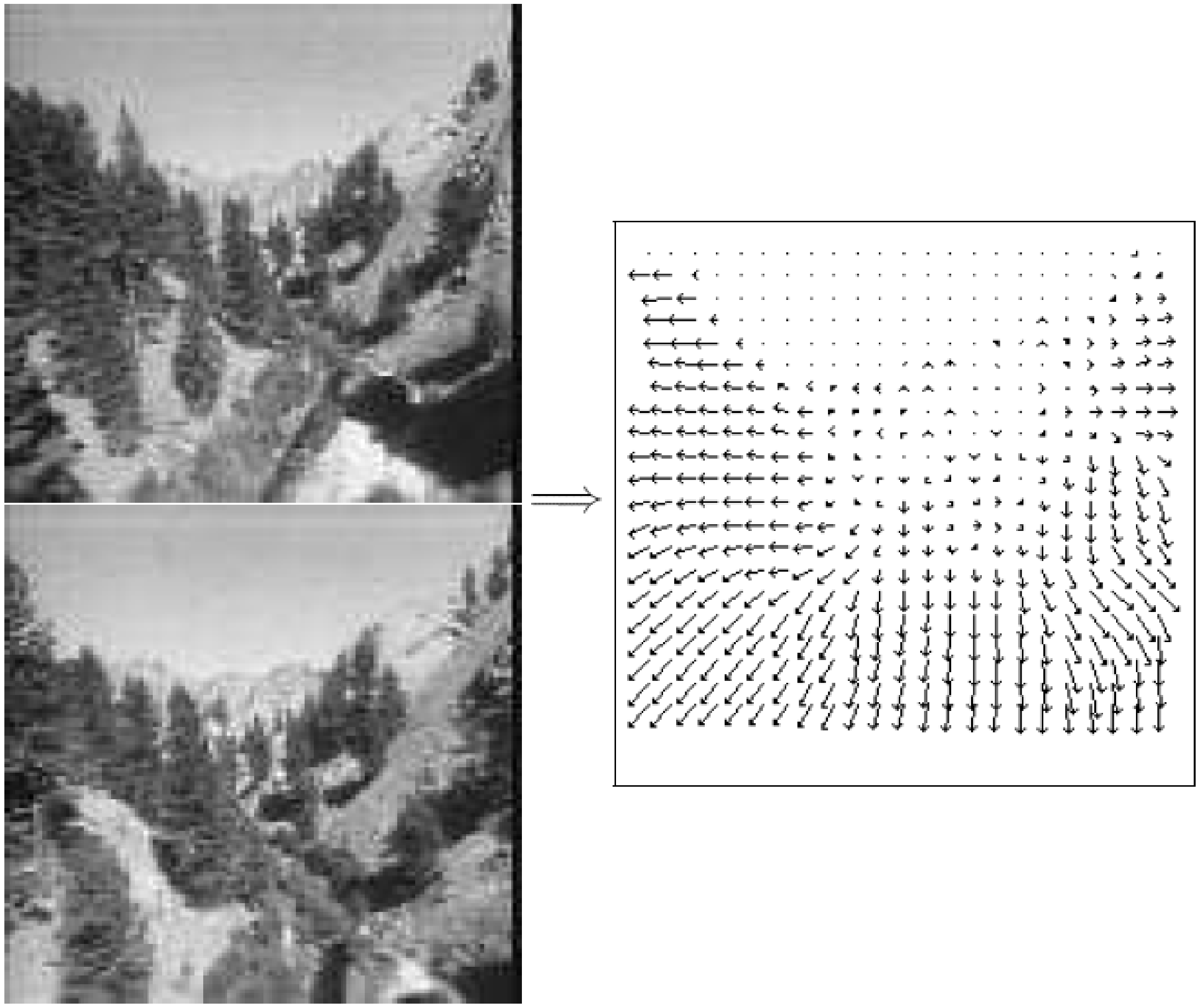


Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

Length of flow vectors inversely proportional to depth Z of 3d point

points closer to the camera move more quickly across the image plane

Figure from Michael Black, Ph.D. Thesis

Egomotion: collision time

- The apparent movements are radial centered on the *focus of expansion* (FOE).
- A collision time (camera/object) could be estimated: the displacement dY with respect to the focus of expansion has the same relationship as the displacement along the **Z** axis with respect to the focal plane.
- For a camera having general velocity with components u, v and w respectively along the **X**, **Y** and **Z** axis, the generic object point **Xo**, **Yo**, **Zo** will be displaced on the image as:

$$X = -f\frac{\boldsymbol{X}_O + ut}{\boldsymbol{Z}_O + wt} \;;\; Y = -f\frac{\boldsymbol{Y}_O + vt}{\boldsymbol{Z}_O + wt}$$

- In order to compute the coordinates of the final/original destination of the moving point we may evaluate these for t= ±∞ so obtaining the focus of expansion/contraction coordinates:

$$X_{FOE} = -f\frac{u}{w} \;; Y_{FOE} = -f\frac{v}{w}$$

and consequently the collision time.

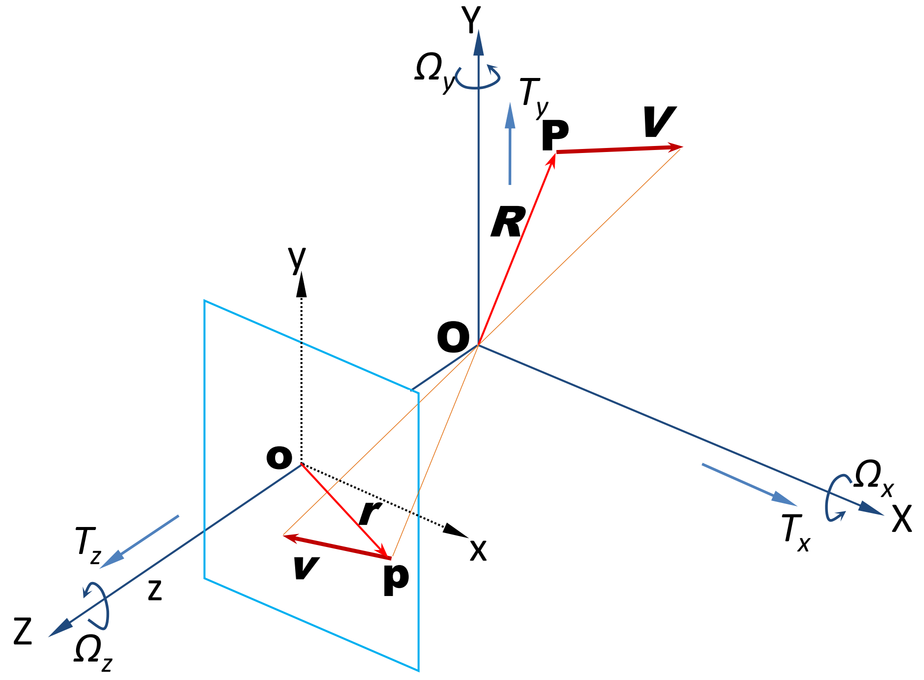
Camera and egomotion

- The egomotion makes all still objects in the scene to verify the same motion model defined by three translations **T** and three rotations **Ω**. Conversely, mobile obstacles pop out as not resorting to the former dominating model.
- Under such assumptions, the following classical equations hold:

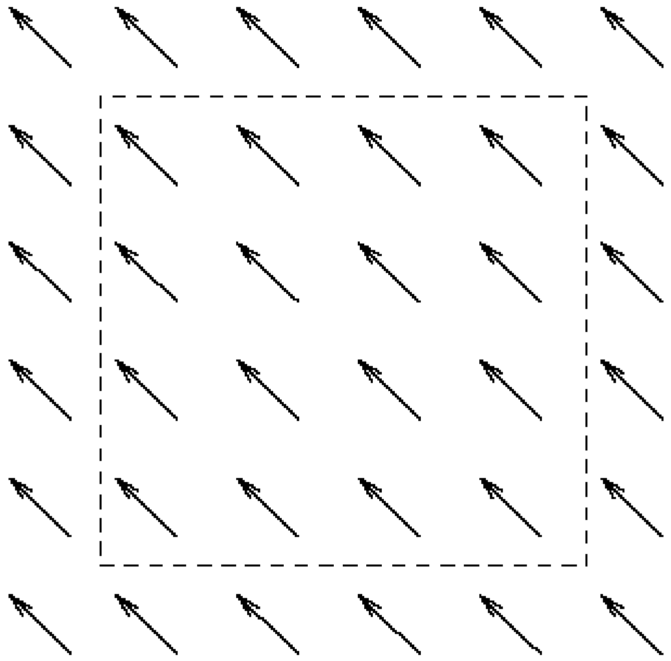
$$u_t = \frac{-fT_X + xT_Z}{Z},\ u_r = \frac{-xy}{f}\Omega_X - \left(\frac{-x^2}{f} + 1\right)\Omega_Y + y\Omega_Z$$

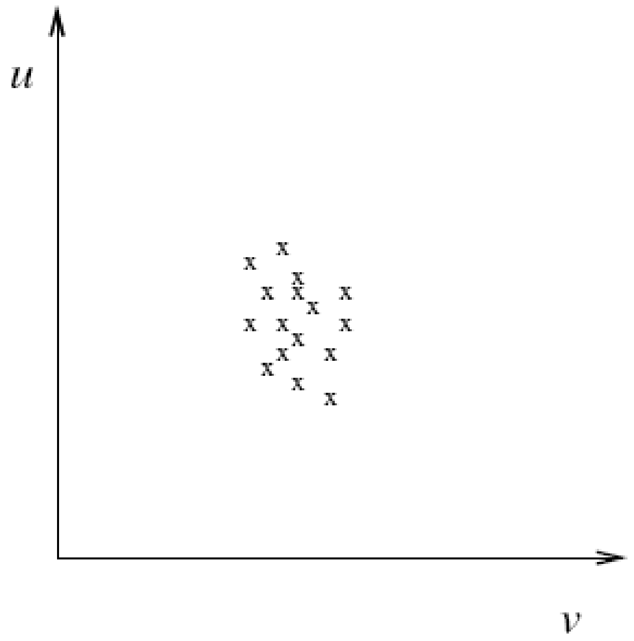
$$v_t = \frac{-fT_Y + yT_Z}{Z}, v_r = \frac{-xy}{f}\Omega_Y - \left(\frac{-y^2}{f} + 1\right)\Omega_X + x\Omega_Z$$

- where $\mathbf{w} = [u, v]^T = [u_t + u_r, v_t + v_r]^T$ stands for the 2-D velocity vector of the pixel under the focal length **f**.



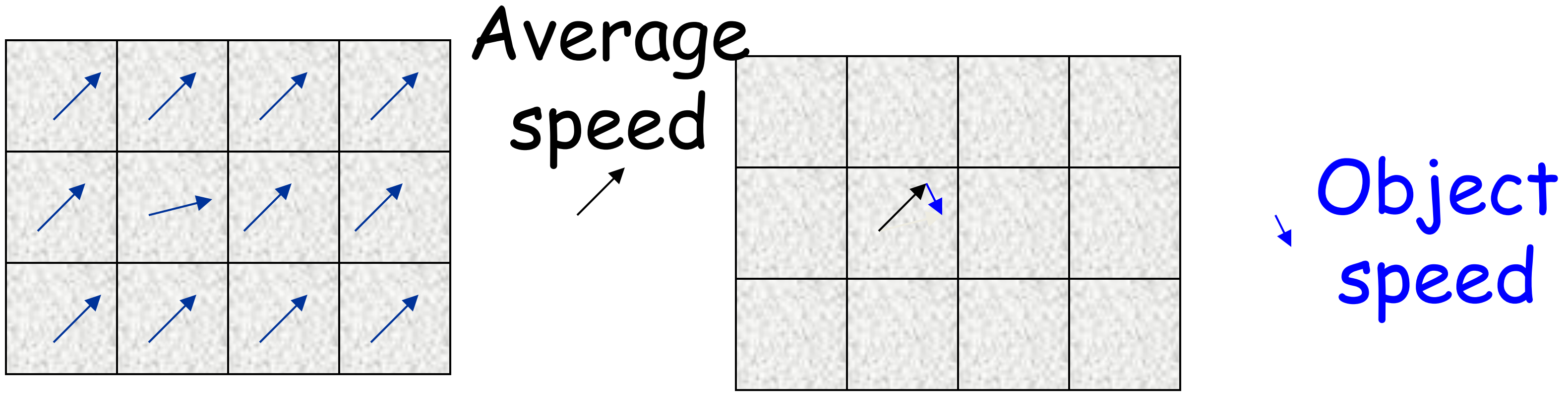
Coherent Motion





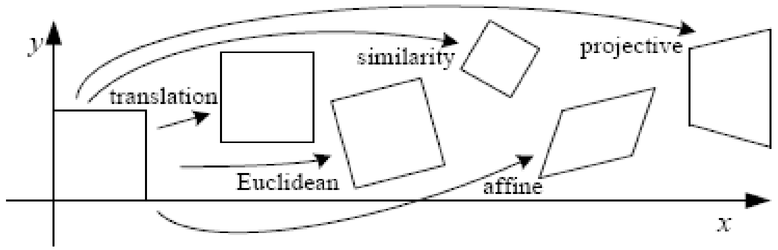
Possibly Gaussian

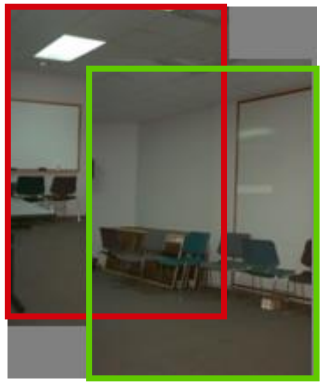
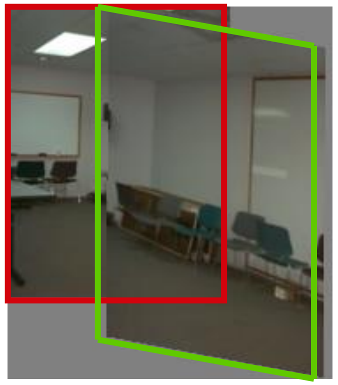
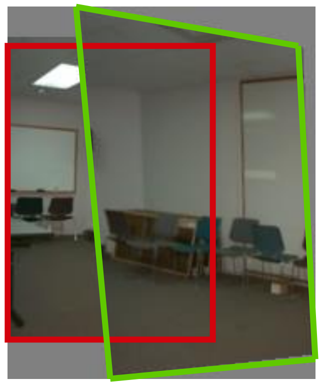
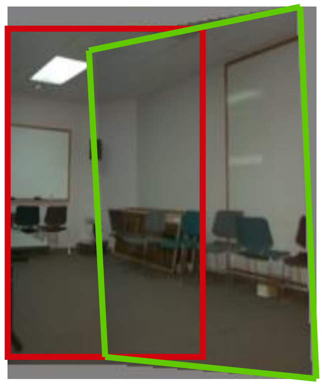
Egomotion



Detected
speeed

Motion models



Translation	Affine	Perspective	3D rotation
			
2 unknowns	6 unknowns	8 unknowns	3 unknowns

Szelisl

- Transformations/warping of image

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{x} + \mathbf{h}) - I_0(\mathbf{x}) \right]^2$$

Translations: $\mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

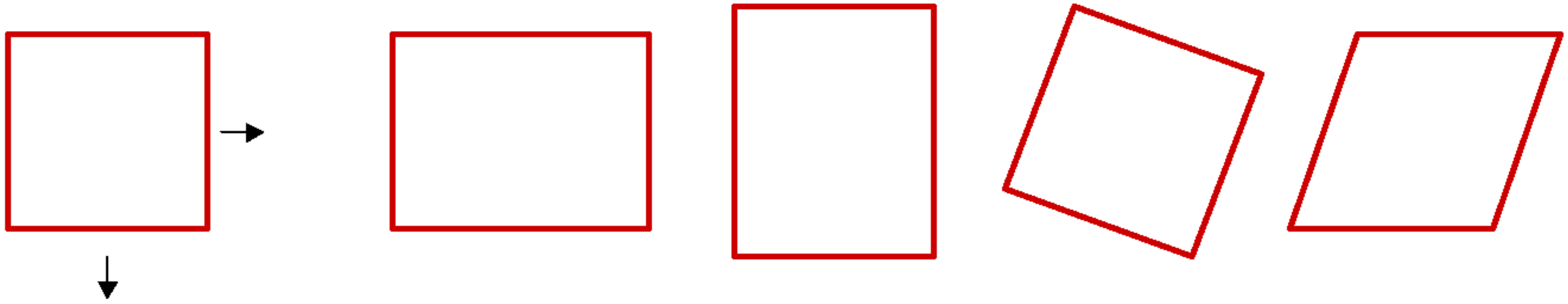
Generalization

- Transformations/warping of image

$$E(\mathbf{A}, \mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{A}\mathbf{x} + \mathbf{h}) - I_0(\mathbf{x}) \right]^2$$

Affine: $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

Generalization



Affine: $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

Example: Affine Motion

$$\boxed{\begin{aligned} u(x,y) &= a_1 + a_2 x + a_3 y \\ v(x,y) &= a_4 + a_5 x + a_6 y \end{aligned}}$$

Substituting into the B.C. Equation:

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

Each pixel provides 1 linear constraint in 6 *global* unknowns
(minimum 6 pixels necessary)

Least Square Minimization (over all pixels):

$$Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2$$

Generalization

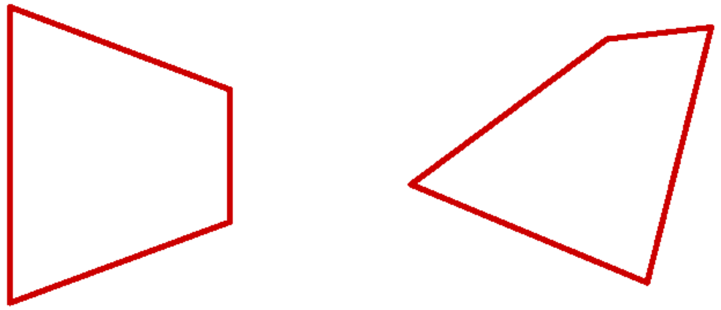
- Transformations/warping of image

$$E(\mathbf{A}) = \sum_{\mathbf{x} \in R} [I(\mathbf{A}\mathbf{x}) - I_0(\mathbf{x})]^2$$

$$\text{Planar perspective: } \mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$$

Generalization

Affine +



Planar perspective: $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$

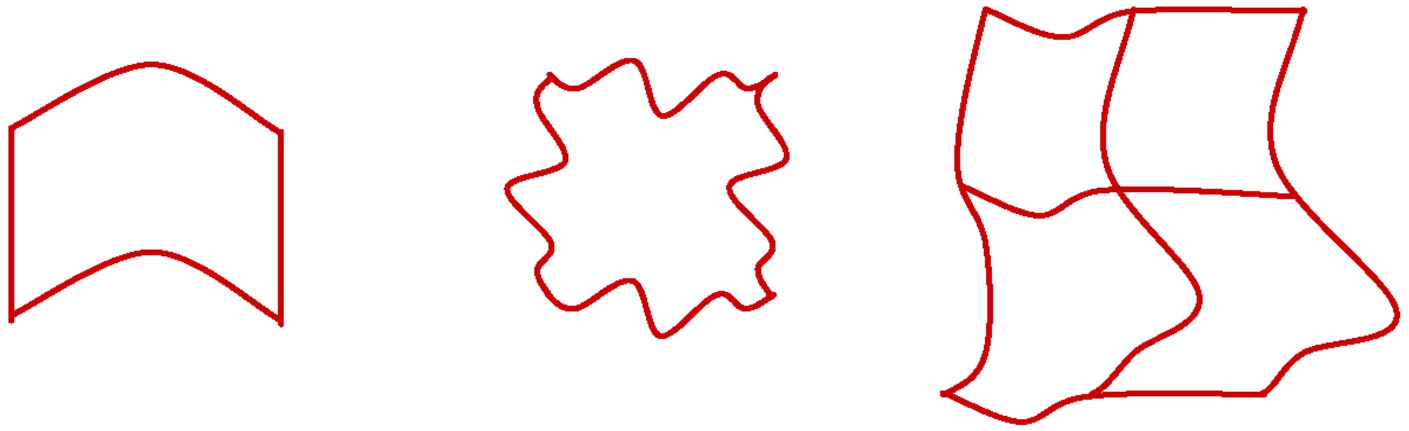
Generalization

- Transformations/warping of image

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{f}(\mathbf{x}, \mathbf{h})) - I_0(\mathbf{x}) \right]^2$$

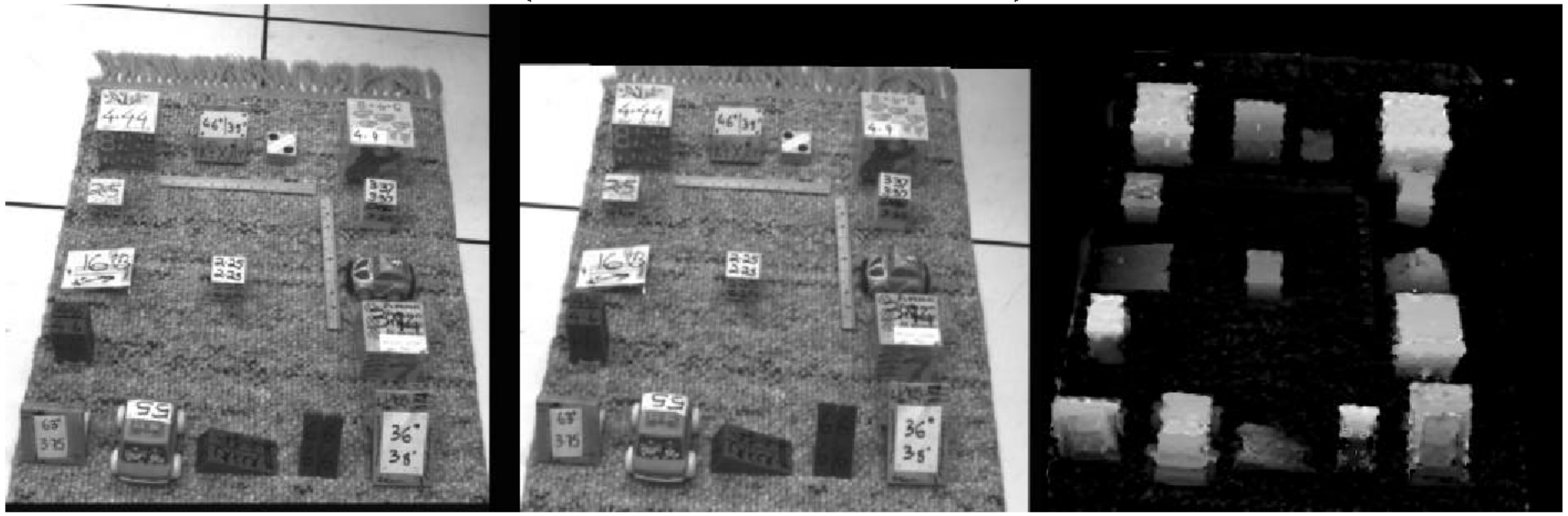
Other parametrized transformations

Generalization



Other parametrized transformations

Residual Planar Parallax Motion (Plane+Parallax)

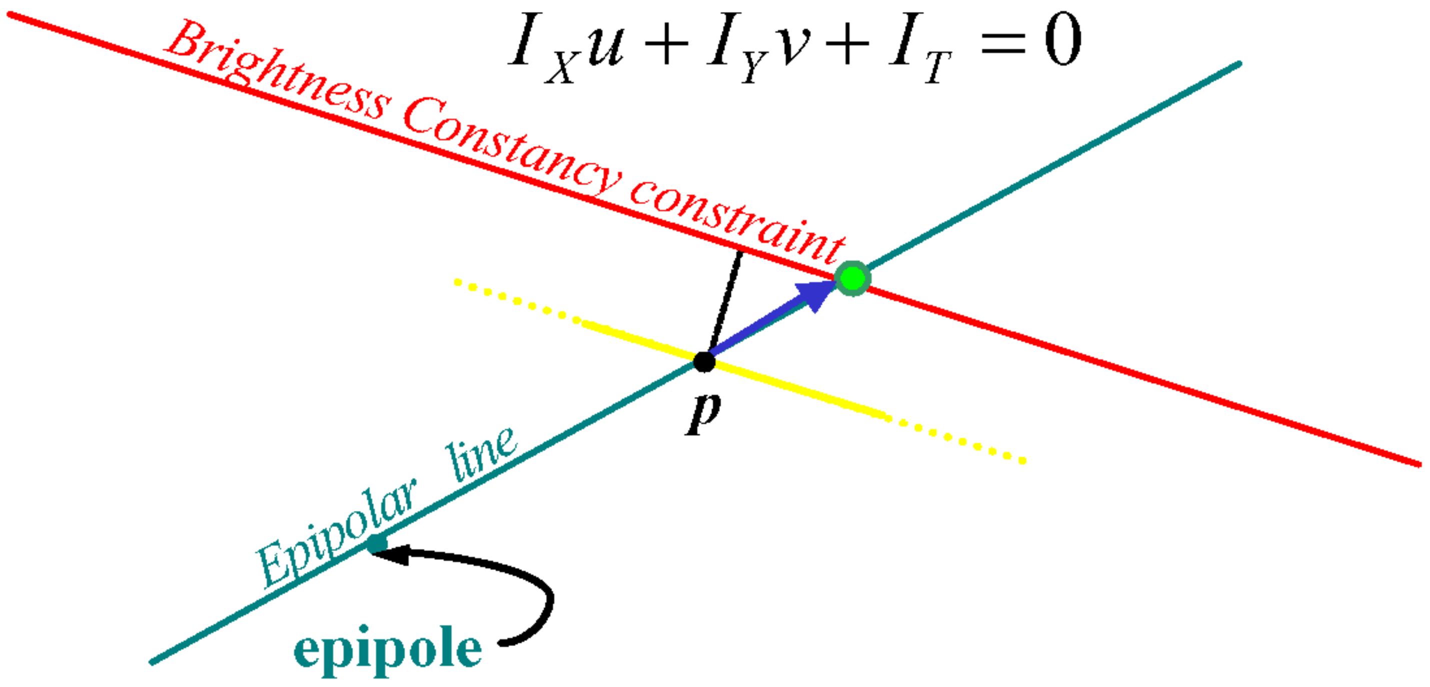


Original sequence **Plane-aligned sequence** **Recovered shape**

Block sequence from [Kumar-Anandan-Hanna'94]

"Given two views where motion of points on a parametric surface has been compensated, the residual parallax is an epipolar field"

Residual Planar Parallax Motion

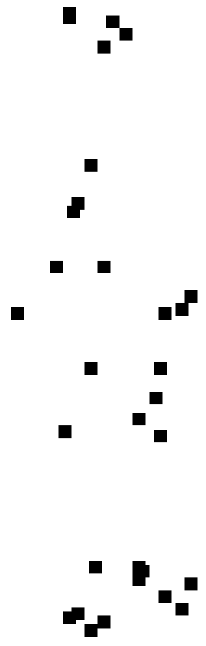


The intersection of the two line constraints <u>uniquely</u> defines the displacement.

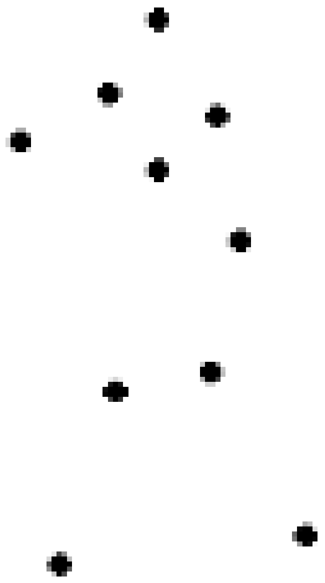
Motion via correspondences

Even "impoverished" motion data can evoke a strong percept

30 points



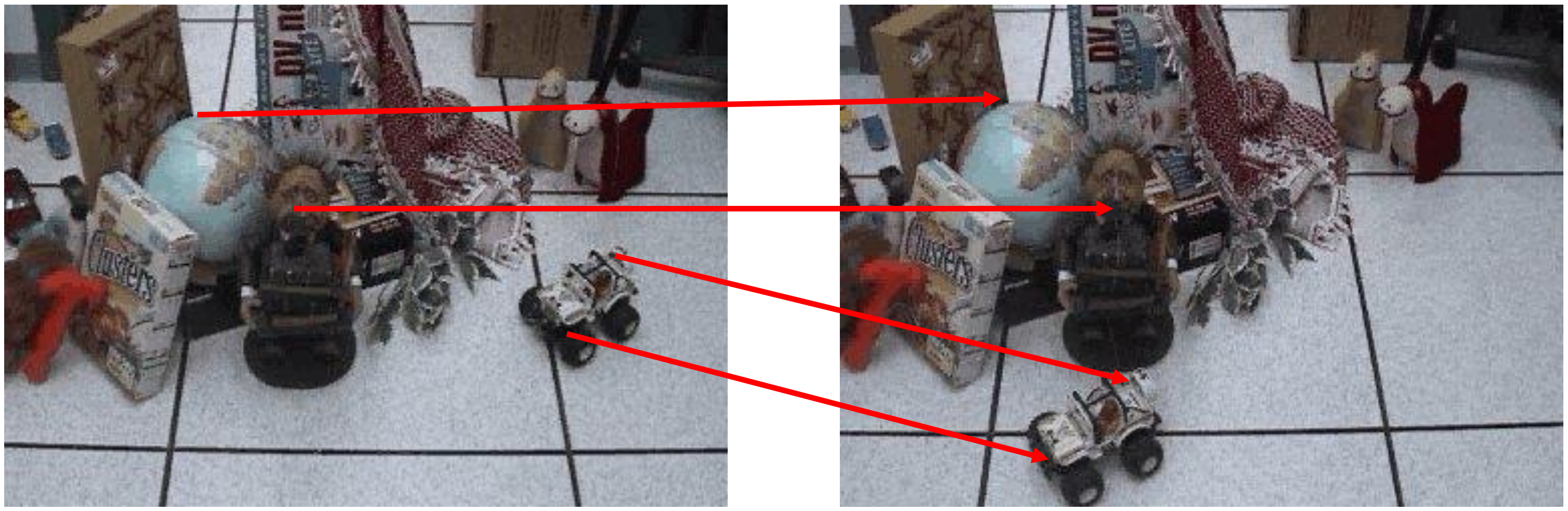
10 points



- Normally, peculiar points on the first image are located so as to search their corresponding points on the second image.
- As in the triangulation for stereovision, there is no guarantee that such corresponding points exist and the new point of view may not include such points, moved out of the field of view.
- The object is first considered as a rigid one and therefore without plastic distortion and the background is regarded as stationary.
- In order to reduce the computational cost, the number of points is limited to the truly characteristic ones.
- Similarly to the epipolar segment for stereovision, the corresponding points are searched in a restricted area determined by a few heuristics.
- Primal sketch: locate the position of a pixel in the current image having similarity and the shortest *Euclidean distance* with respect to a point in the previous frame.

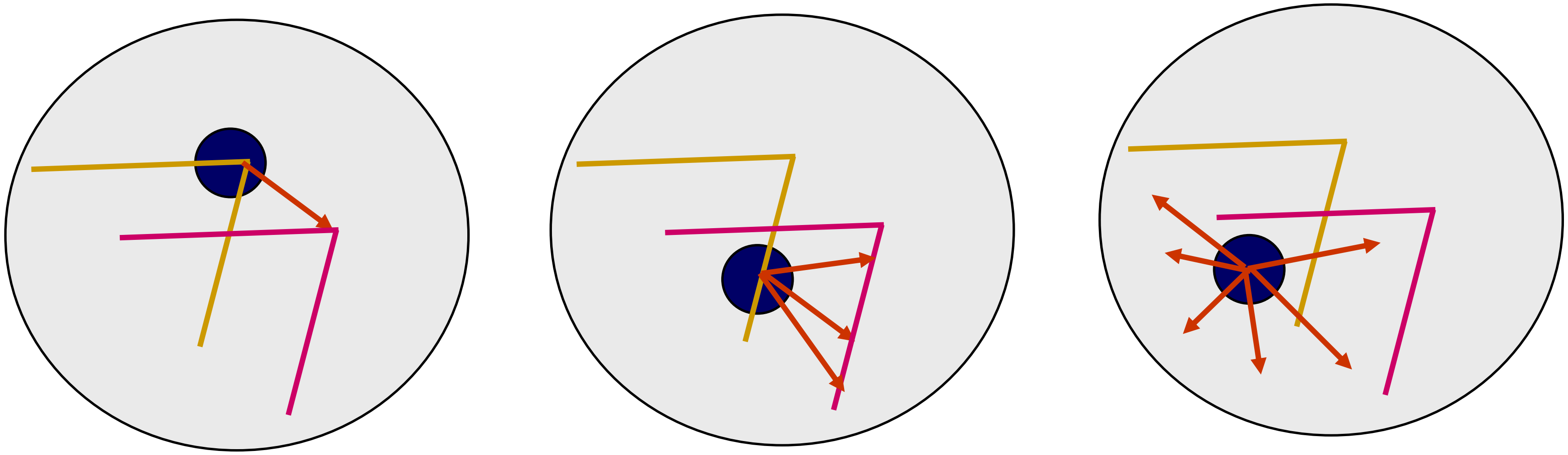
Patch Matching

Where did each pixel in image 1 go to in image 2



Local Patch Analysis

- How *certain* are the motion estimates?

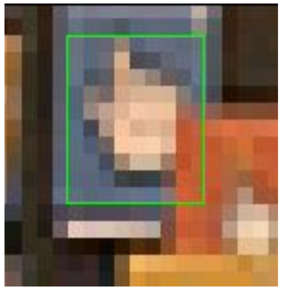
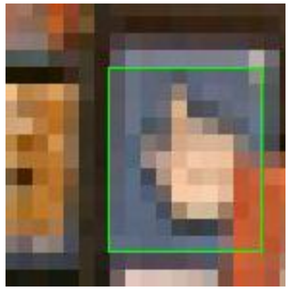


Szelisl

Patch matching (revisited)

- How do we determine correspondences?
  - *block matching* or *SSD* (sum squared differences)

$$E(x, y; d) = \sum_{(x',y')\in N(x,y)} [I_L(x'+d, y') - I_R(x', y')]^2$$

Correlation Window Size

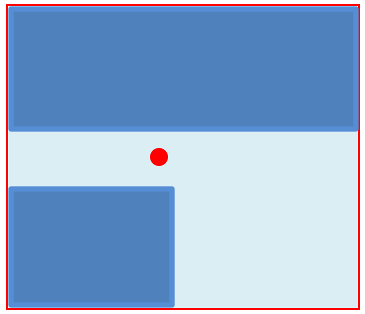
- Small windows lead to more false matches
- Large windows are better this way, but…
  - Neighboring flow vectors will be more correlated (since the template windows have more in common)
  - Flow resolution also lower (same reason)
  - More expensive to compute
- Small windows are good for local search: more detailed and less smooth (noisy?)
- Large windows good for global search: less detailed and smoother

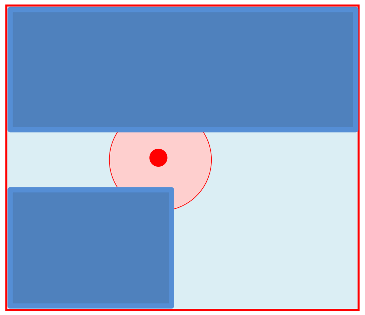
Maximum velocity

- A generic central object point can be located in the successive frame within a circle with a radius equal to V_{max} Δt, where V_{max} is the highest possible velocity of such point:

Obstacles

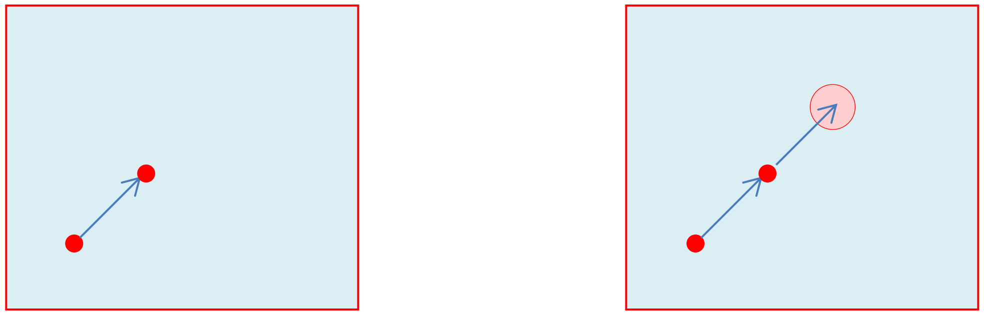
- The previous circular field is also limited by existing obstacles and physical boundaries contained in the scene





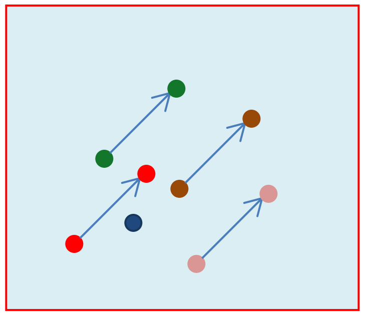
Maximum acceleration

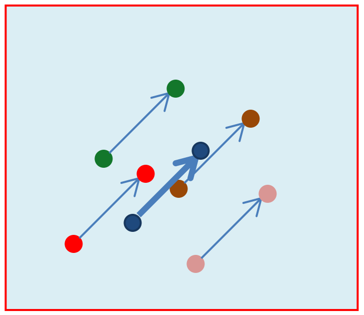
- An extrapolation can enable *TRACKING* the object point in successive frames.
- The velocity detected in the two previous frames may be exploited to foresee the future position of the object point (*time filtering*). Same as before, a displacement will be inside a circle of radius equal to ½ A_{max} Dt^2 where A_{max} is the maximum acceleration;



Consistent matching

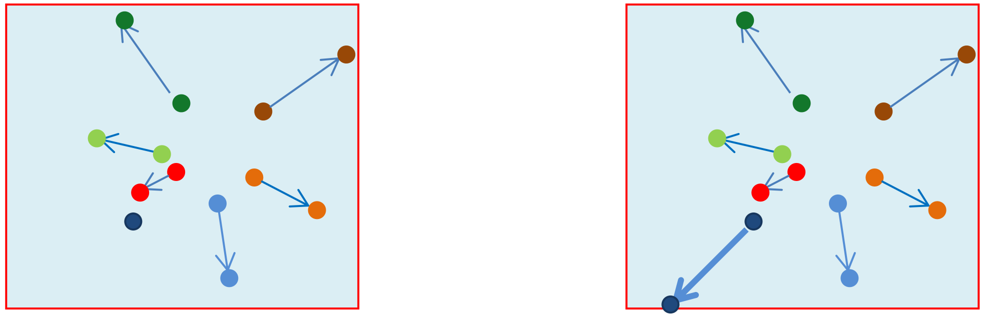
- Object points do not likely coalesce into one single point of the following frame, leading to the so-called *consistent matching* criterion. The picture shows four identified points that force the correspondence of the fifth dark one





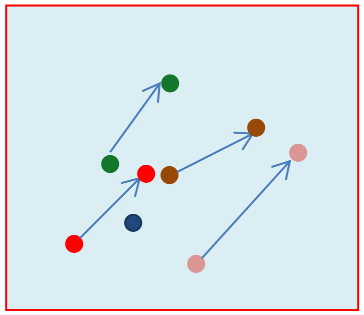
Common motion

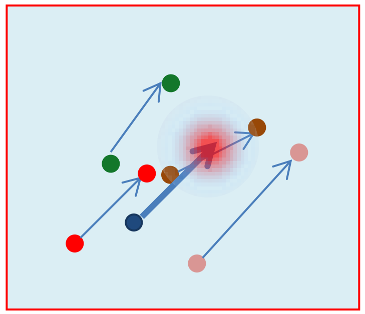
- *Common motion* situation: once the motion of the neighbors has been identified, the dark point necessarily maps into a congruent position (the depicted case is an expansion centered in the figure window)



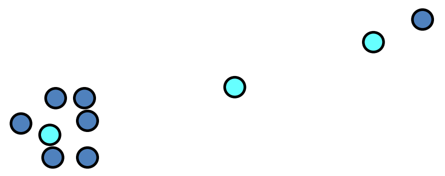
Flexible motion model

- motion model for a 'herd' of points suggesting the most plausible displacement of the dark object point.





Robust estimation

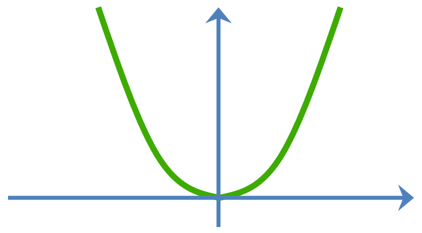


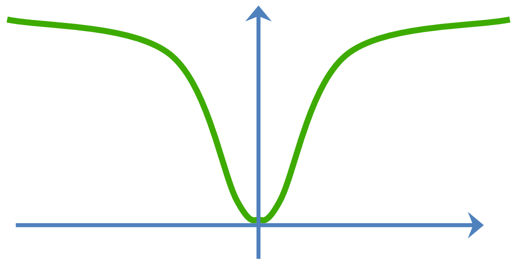
- $$E(m) = \sum_i \rho(x_i)$$

  $$\rho(x_i) = (x_i - m)^2$$

  $$\text{Influence } \psi(x) = \frac{\partial \rho}{\partial x} = 2(x_i - m)$$

influence function error penalty function



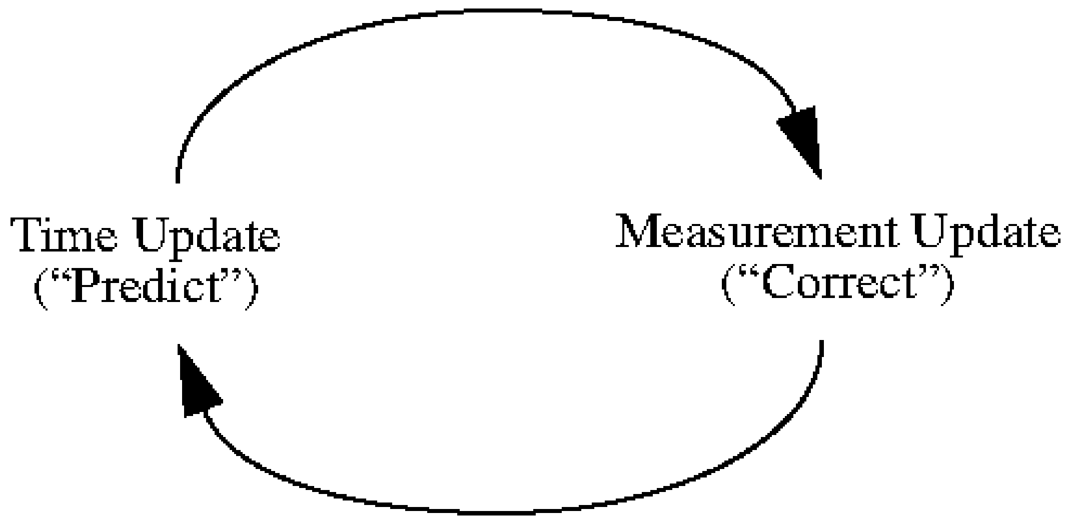


✓ Problem: Least-squares estimators penalize deviations between data & model with quadratic error fⁿ (extremely sensitive to outliers)

✓ Redescending error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

Tracking as induction

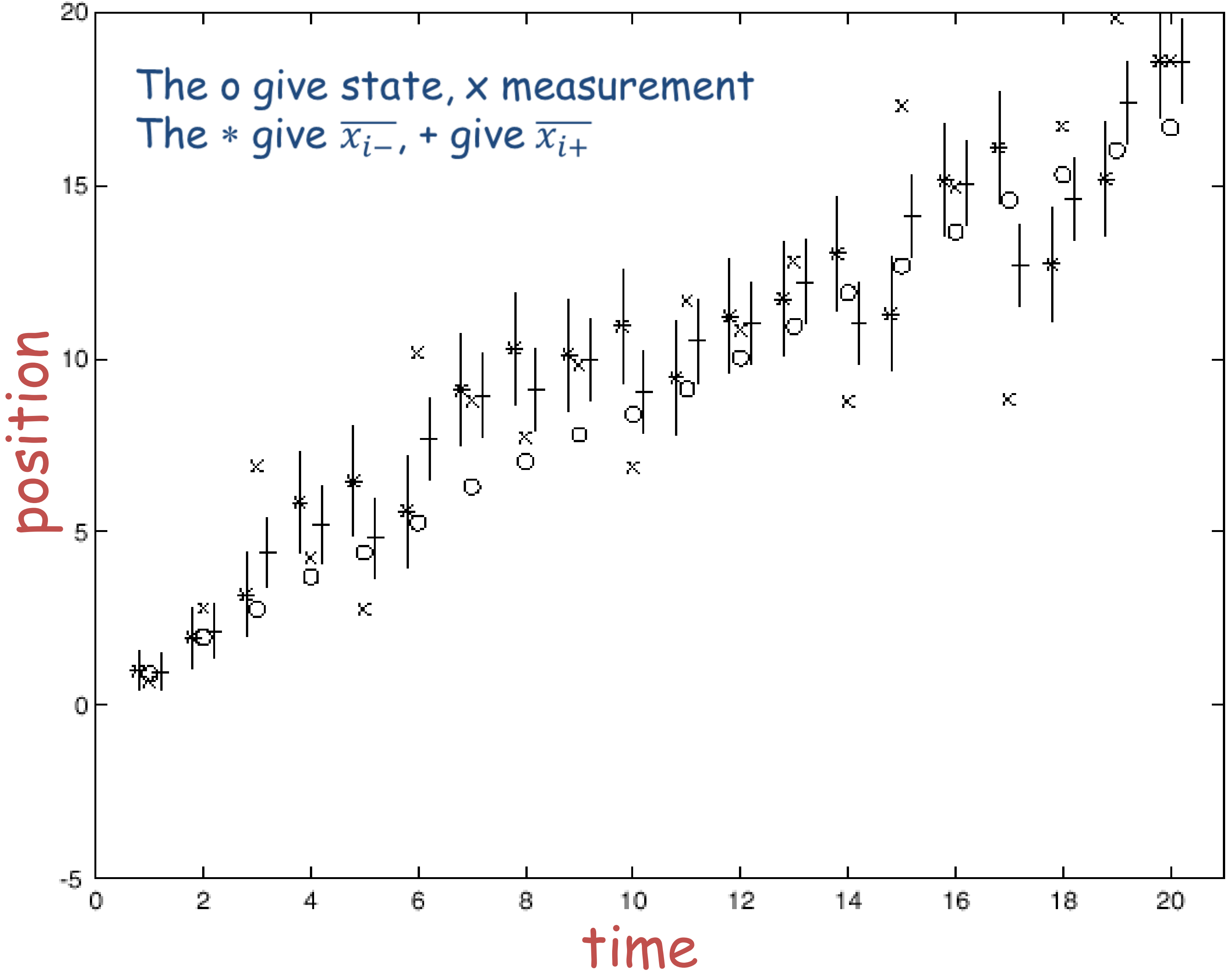
- Make a measurement starting in the 0th frame
- Then: assume you have an estimate at the ith frame, after the measurement step.
- Show that you can do prediction for the i+1th frame, and measurement for the i+1th frame.



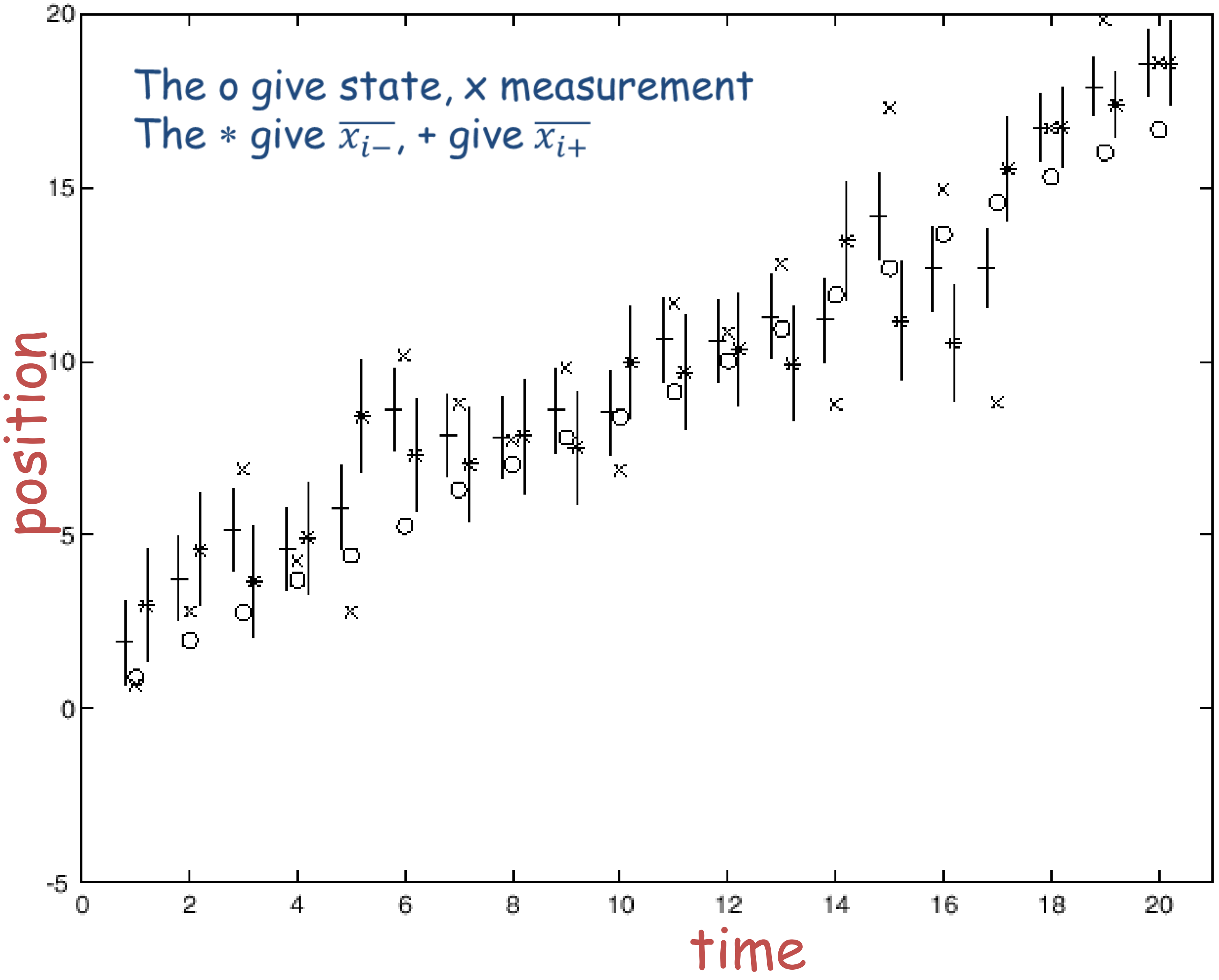
Tracking

- Idea
  - We don't have the best estimate of state - what about the future?
  - Run two filters, one moving forward, the other backward in time.
  - Now combine state estimates
    - The crucial point here is that we can obtain a smoothed estimate by viewing the backward filter's prediction as yet another measurement for the forward filter

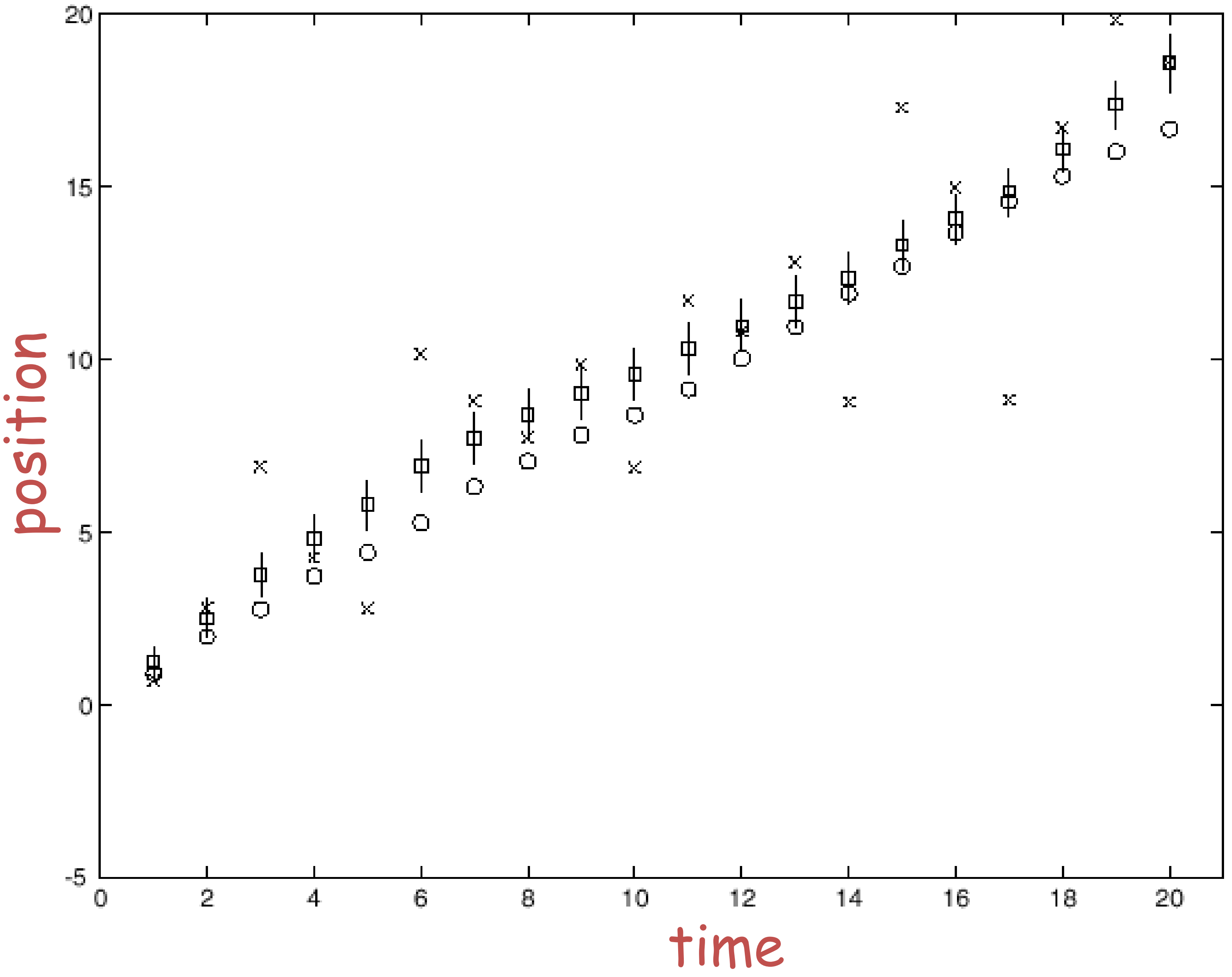
Forward estimates



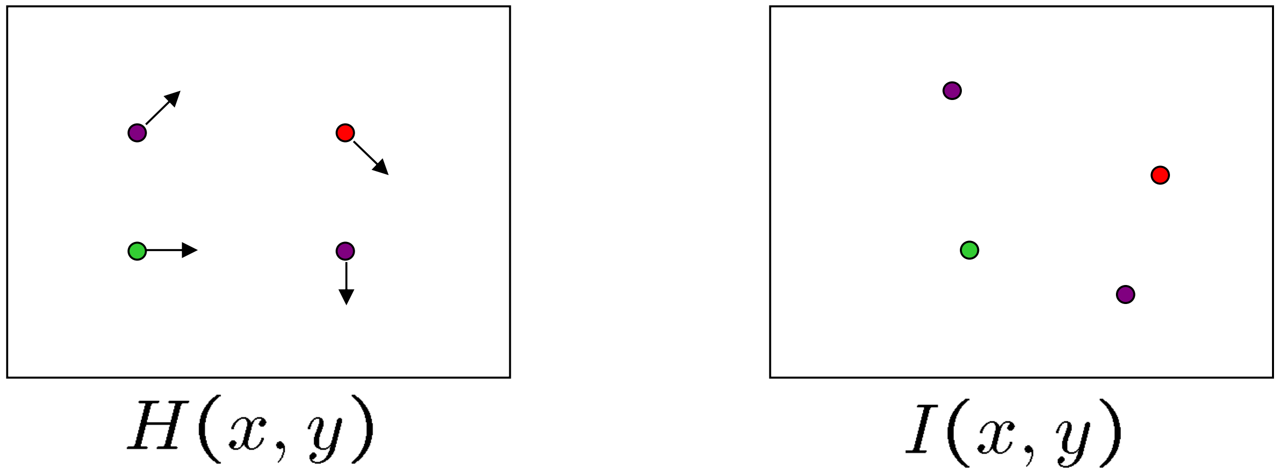
Backward estimates



Combined forward-backward estimates



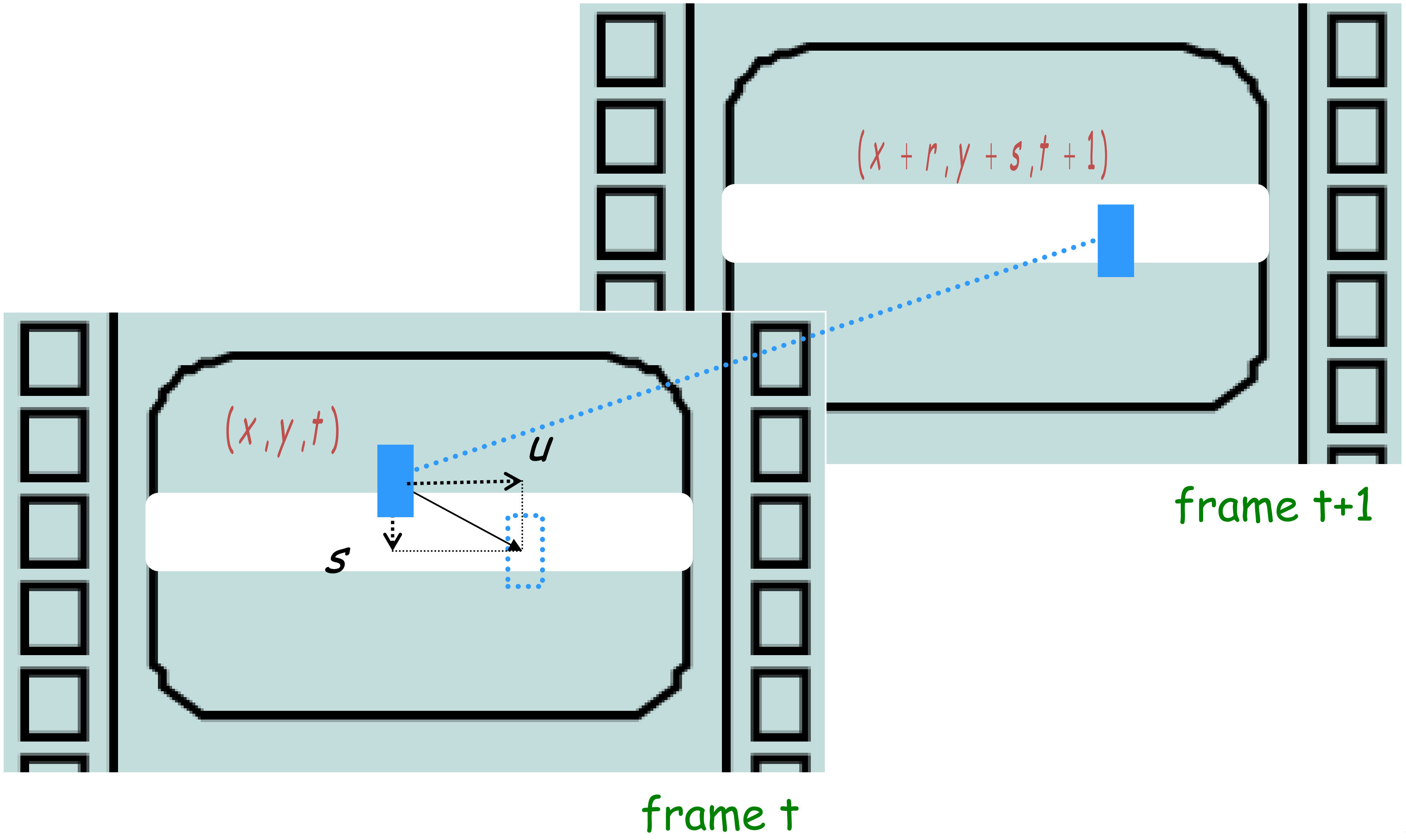
Problem definition: optical flow



- How to estimate pixel motion from image H to image I?
  - Solve pixel correspondence problem
    - given a pixel in H, look for $\boxed{\text{nearby}}$ pixels of the $\boxed{\text{same color}}$ in I
- Key assumptions
  - color constancy: a point in H looks the same in I
    - For grayscale images, this is brightness constancy
  - small motion: points do not move very far
- This is called the optical flow problem

Brightness Constancy

$$I(x+r, y+s, t+1) - I(x, y, t) = 0$$



Optical Flow Constraints

- Optical flow is a constrained problem that only guarantees results under certain conditions.
  - The Smoothness Constraint requires that neighboring points in the image have similar velocities in the velocity field.
    - This means discontinuities represent object occlusion.
  - The Minimization Constraint requires that as our expected noise trends toward zero our error rate for our derivative must be tend towards zero as well
    - This means that our derivative estimates have to be accurate. Well that sucks.

Motion via local change

- Differently from the previous approach, the computation now is performed point-wise, evaluating the motion based on the grey level variations of pixels.
- Let be P_n and P_{n+1} the pixels respectively corresponding to the same 3D point in successive frames n and n+1, under the assumption that this object point remains visible and that illumination conditions do not change, it is assumed that the grey value f of P_n and P_{n+1} is constant (*optical flow constraint*):

$$f(x+\partial x, y+\partial y, t+\partial t) = f(x,y,t)$$

- where ∂x, ∂y represents the pixel displacement between images and ∂t is the time interval. By differentiating, one can write: $f_x \frac{\partial x}{\partial t} + f_y \frac{\partial y}{\partial t} = -f_t$ or using $\bullet$ for the scalar product and ∇ for the gradient $\nabla f \cdot \bar{V} = f_t$
- Under the hypothesis that $\nabla f \neq 0$ it writes: $\bar{V}_\perp = \frac{-f_t}{\sqrt{f_x^2 + f_y^2}}$ giving the velocity along the gradient direction.

Mathematical formulation

Brightness constancy assumption:

$$I(x+\frac{dx}{dt}\delta t, y+\frac{dy}{dt}\delta t, t+\delta t) = I(x,y,t)$$

Optical flow constraint equation :

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

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt} \qquad I_x = \frac{\partial I}{\partial y}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t}$$

$$\boxed{I_x u + I_y v + I_t = 0}$$

1 equation in 2 unknowns

The brightness constancy constraint

- Can we use this equation to recover image motion $\vec{v}$=[u,v] at each pixel?

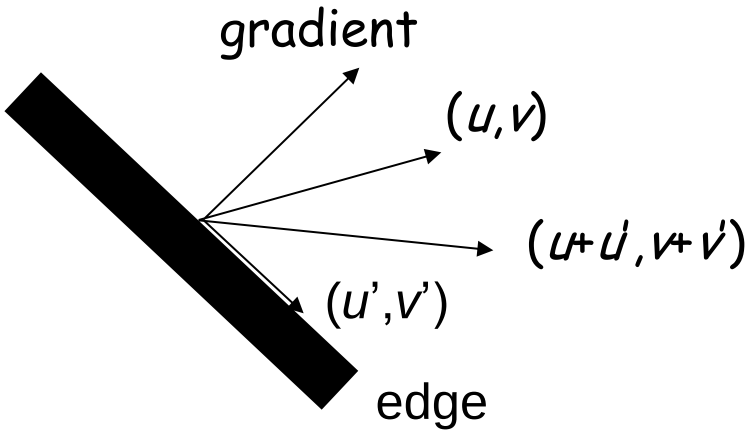
$$\nabla I \cdot [u \;\; v]^T + I_t = 0$$

- One equation (this is a scalar equation!), two unknown $\vec{v} = [u, v]$

- The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

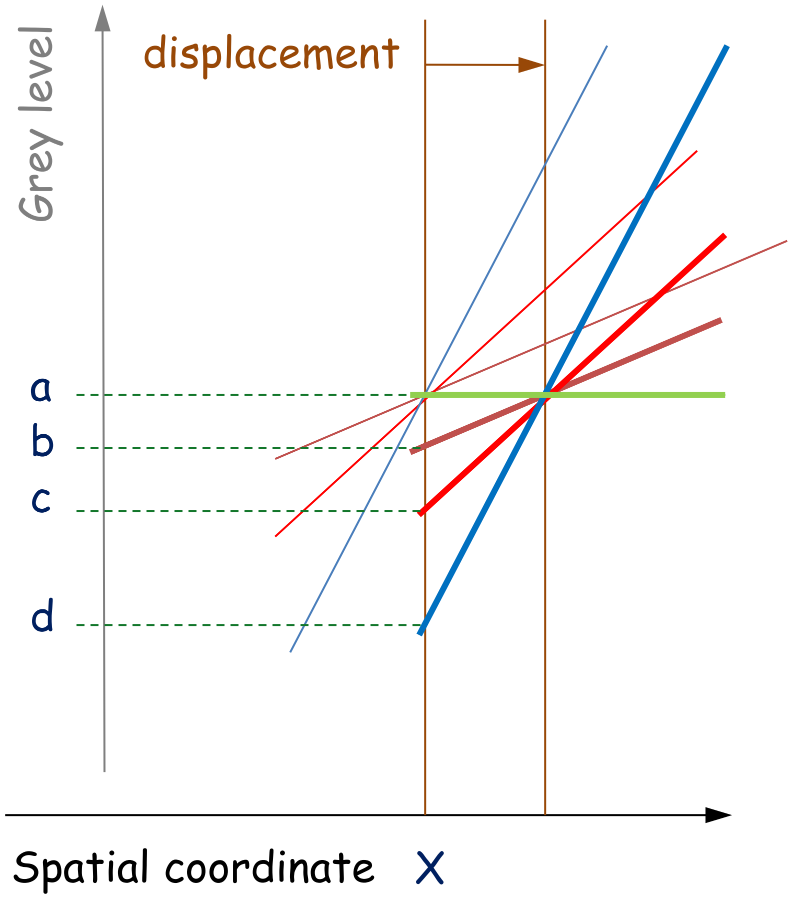
If [u,v] satisfies the equation, so does [u+u', v+v'] if

$$\nabla I \cdot [u' \;\; v']^T = 0$$



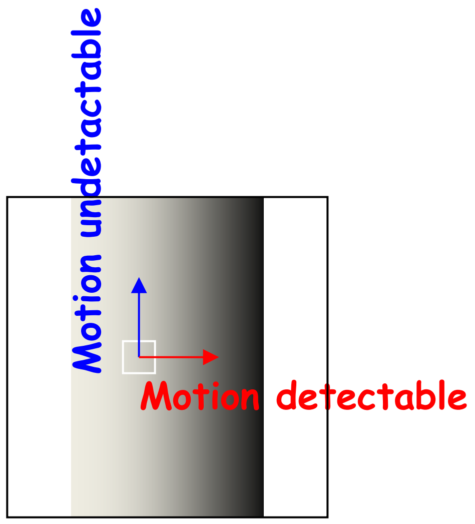
Optical flow

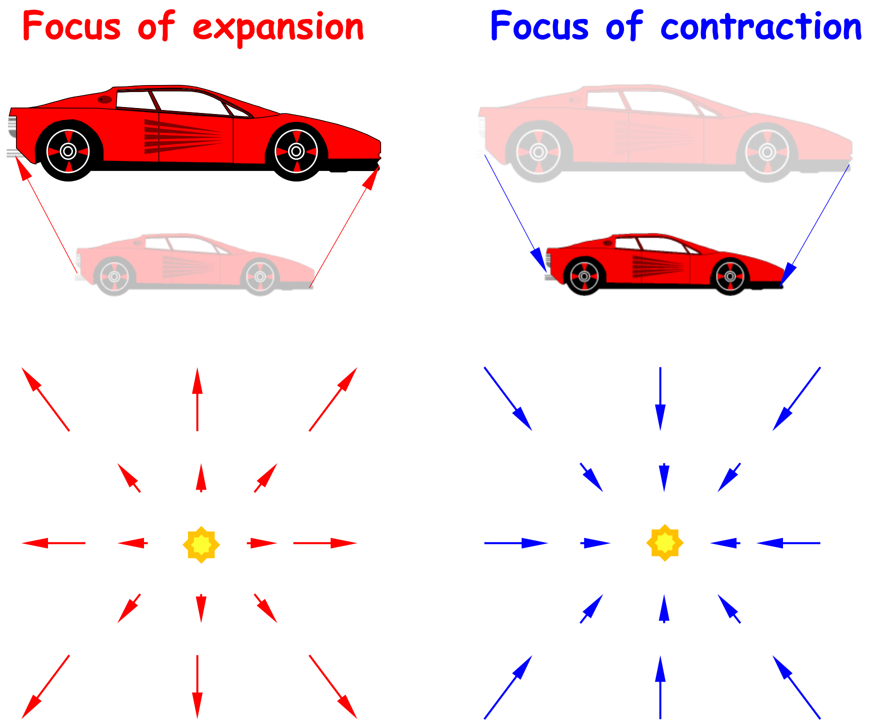
- A geometric representation of grey level variations due to the movement of the object points are illustrated in figure . X corresponds to the location examined along a given spatial coordinate while four straight lines materialize four potential grey level variations (*linearized*) on the object. Bold lines show the grey level pattern due to the object displacement.
- If the object point moves along a direction having constant grey level, no variation can be detected. Conversely, the higher the gradient value the greater the grey level variation due to motion, so that the movement along the gradient direction be evaluated easily in accordance through it: the apparent movement is inversely weighted with the gradient intensity.
- The information obtained via this approach only refers to the orthogonal direction with respect to the contour and a number of algorithms along the years, have been given to provide a more detailed movement information..



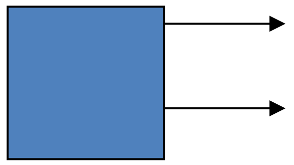
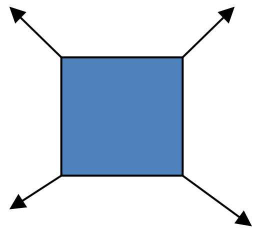
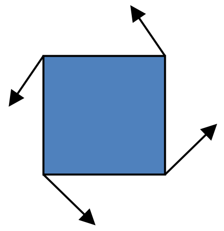
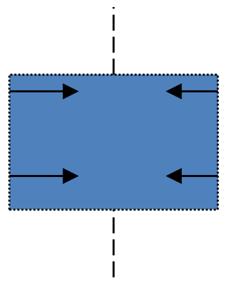
Optical flow

- The optical flow is defined as the vectorial field of luminosity variation.
- This variation depends on the motion and on the grey level gradient
- Examples:

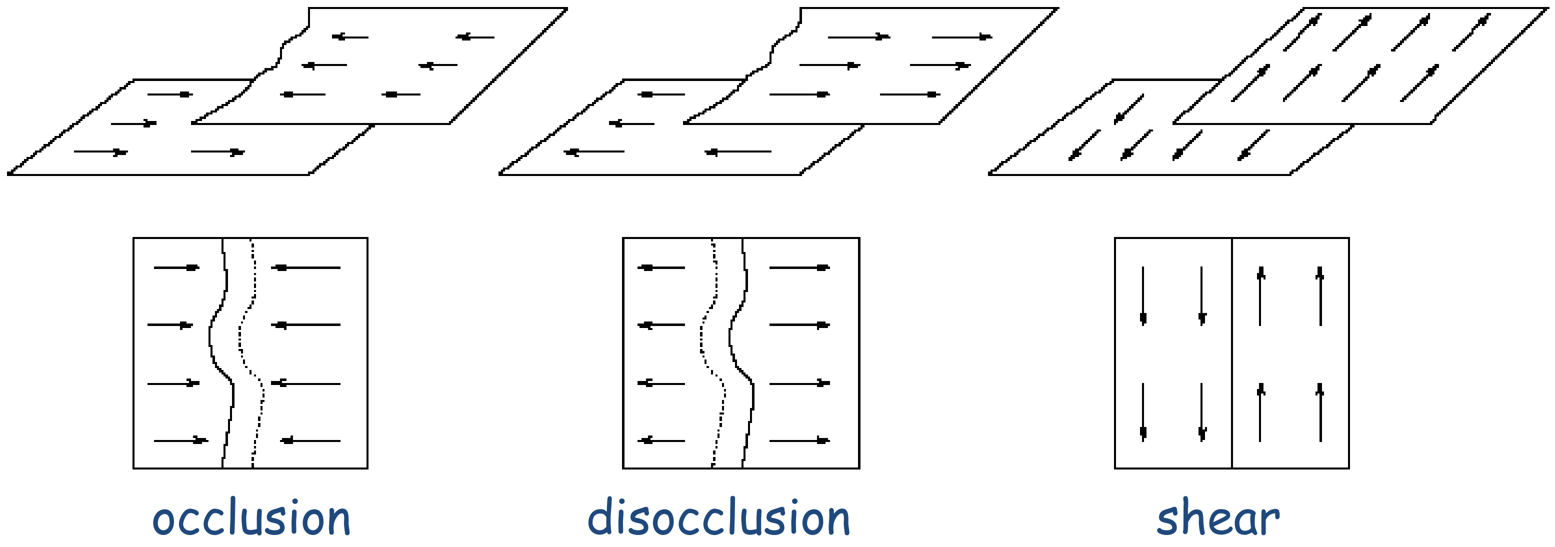




Forms of motion

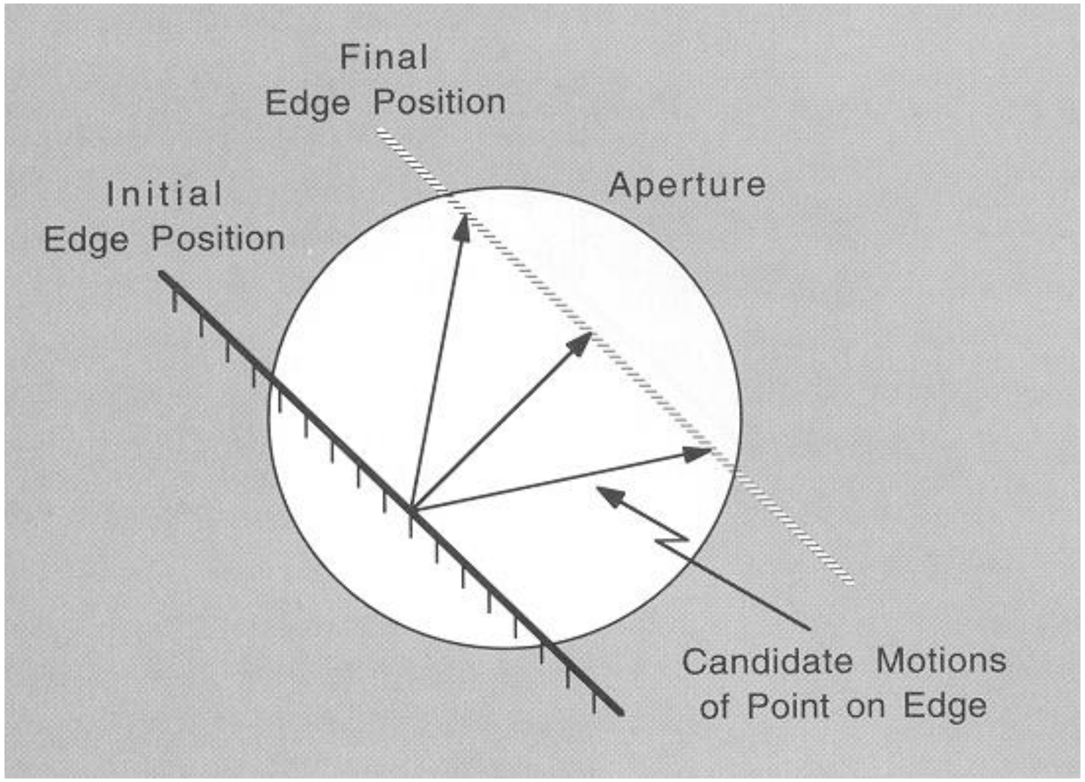
Translation at constant distance from the observer.	Set of parallel motion vectors.	
Translation in depth relative to the observer.	Set of vectors having common focus of expansion.	
Rotation at constant distance from view axis.	Set of concentric motion vectors.	
Rotation of planar object perpendicular to the view axis.	One or more sets of vectors starting from straight line segments.	

Occlusion



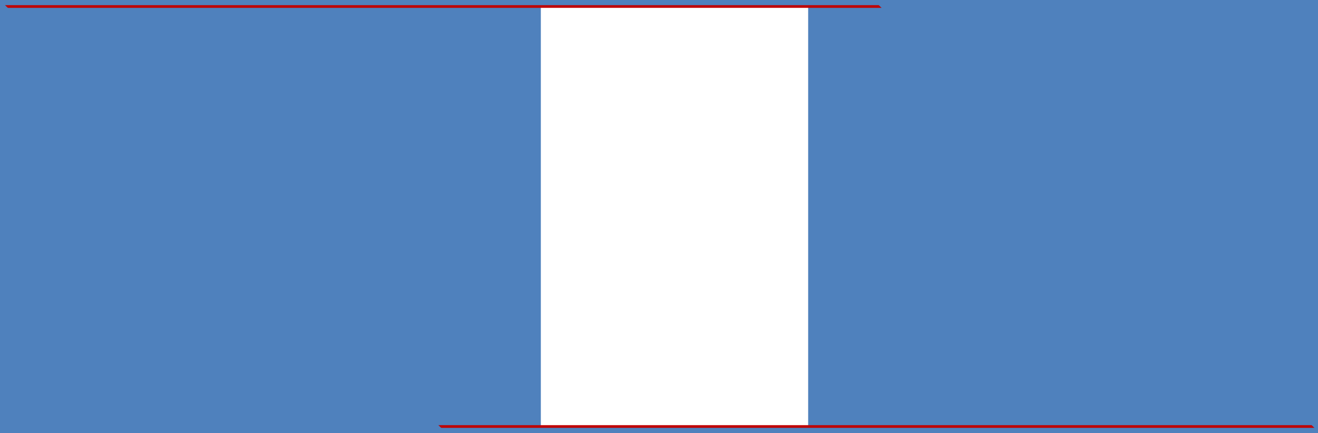
Multiple motions within a finite region.

Aperture Problem

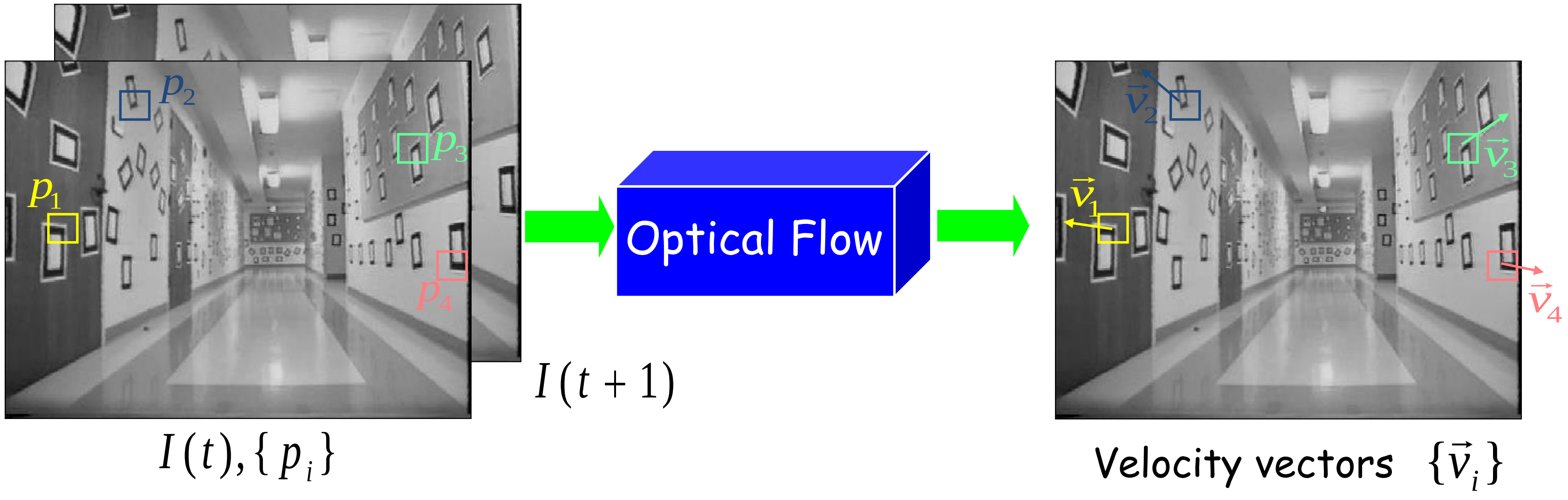


In degenerate local regions, only the normal velocity is measurable.

The aperture problem



What is Optical Flow?



Velocity vectors $\{\vec{v}_i\}$

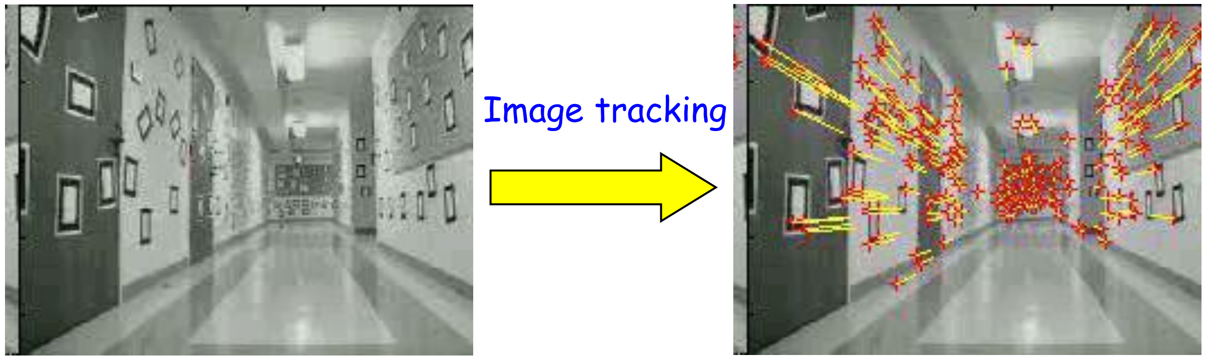


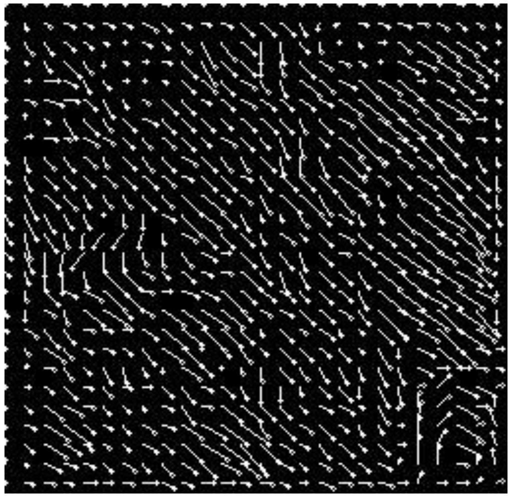
Image sequence (single camera)

Tracked sequence

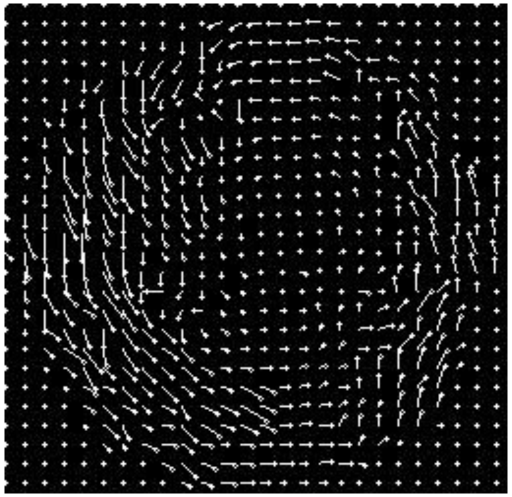
Note: more elaborate tracking models can be adopted if more frames are process all at once

Optical Flow Examples

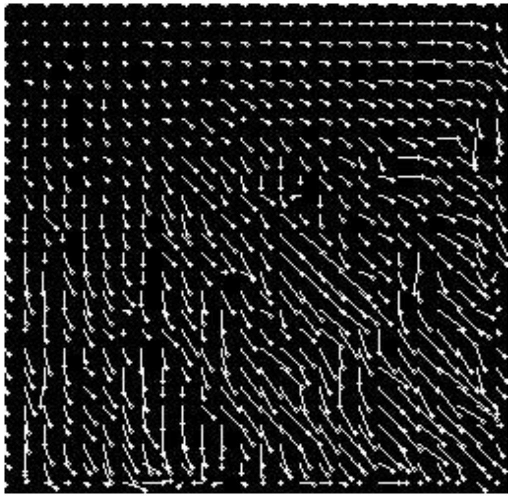
Translation



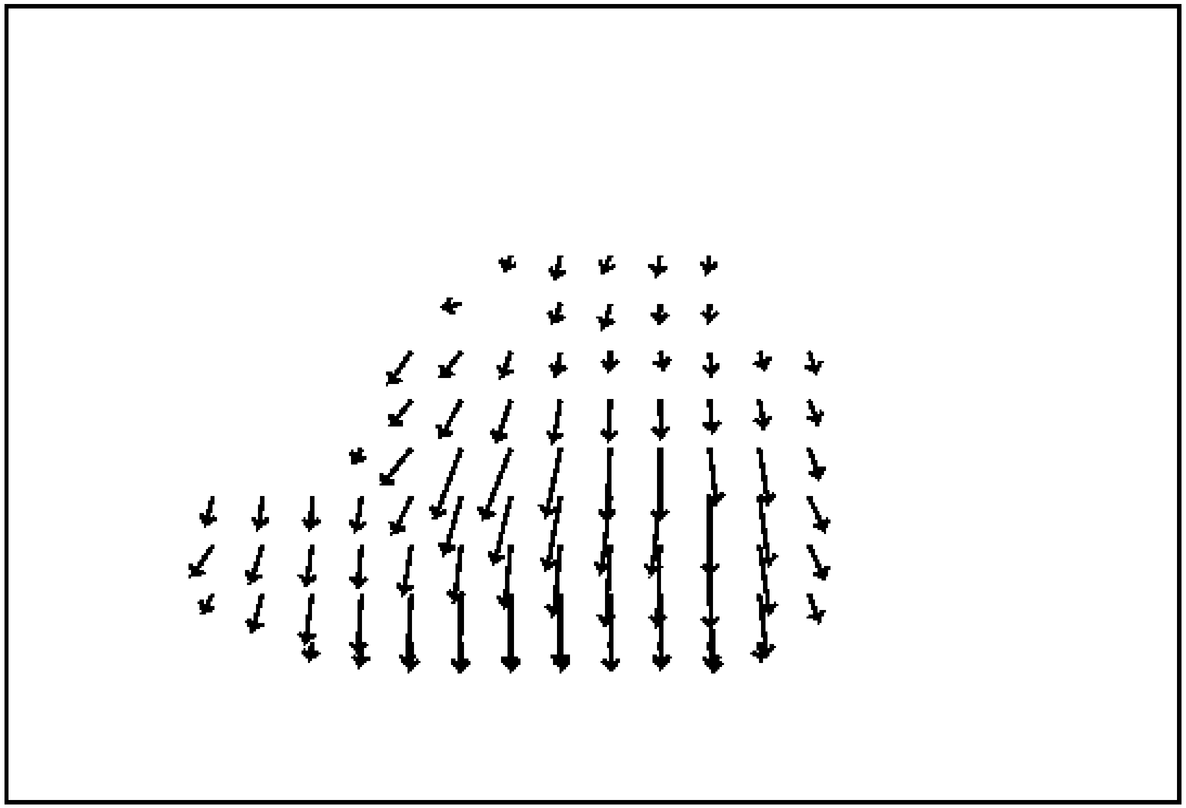
Rotation



Scaling



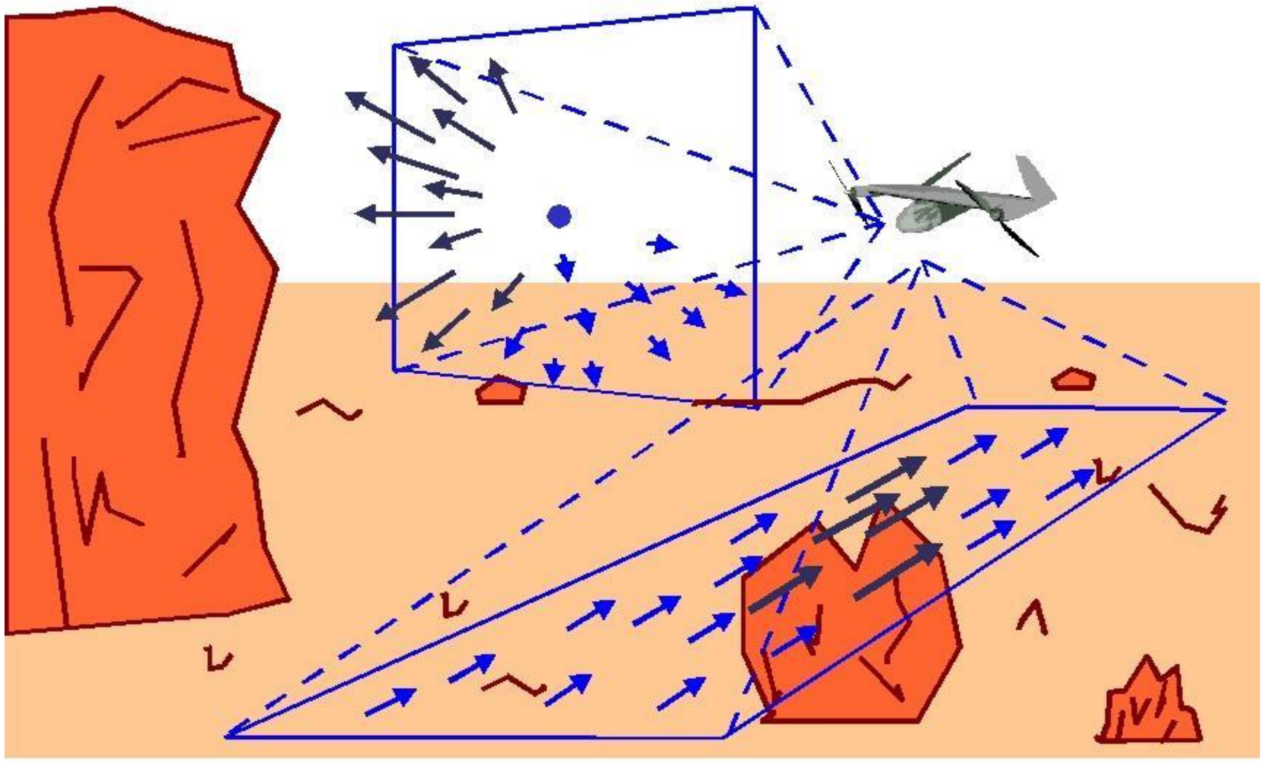
Optical Flow



Pierre Kornprobst's Demo

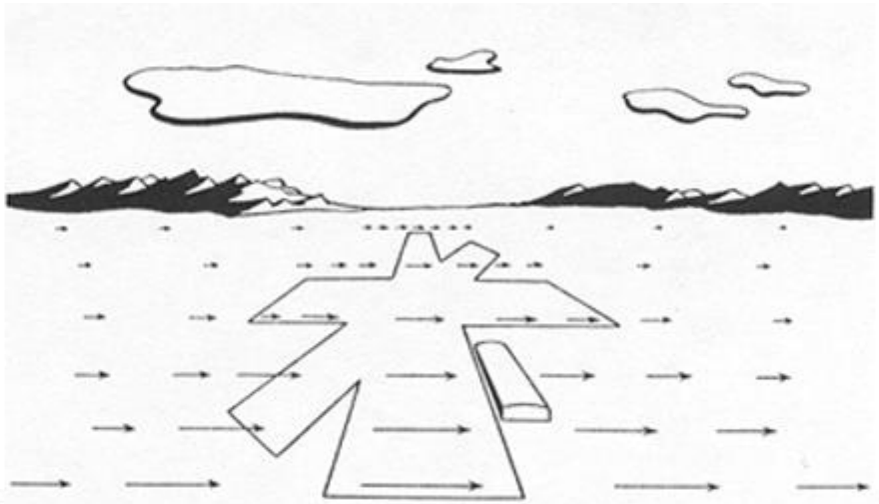
Optical Flow as Seen from an Aircraft

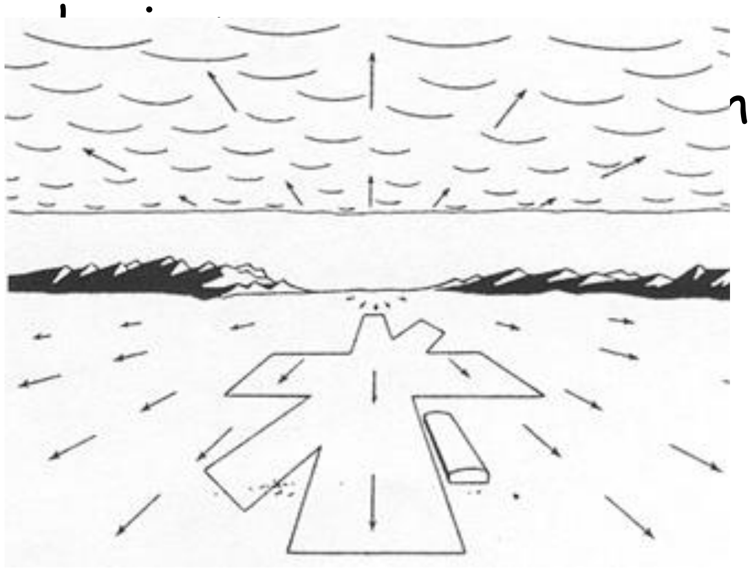
- Blue arrows show the optic flow
- Blue circle directly at the center shows the "focus of expansion", which tells the aircraft the specific direction it is flying.

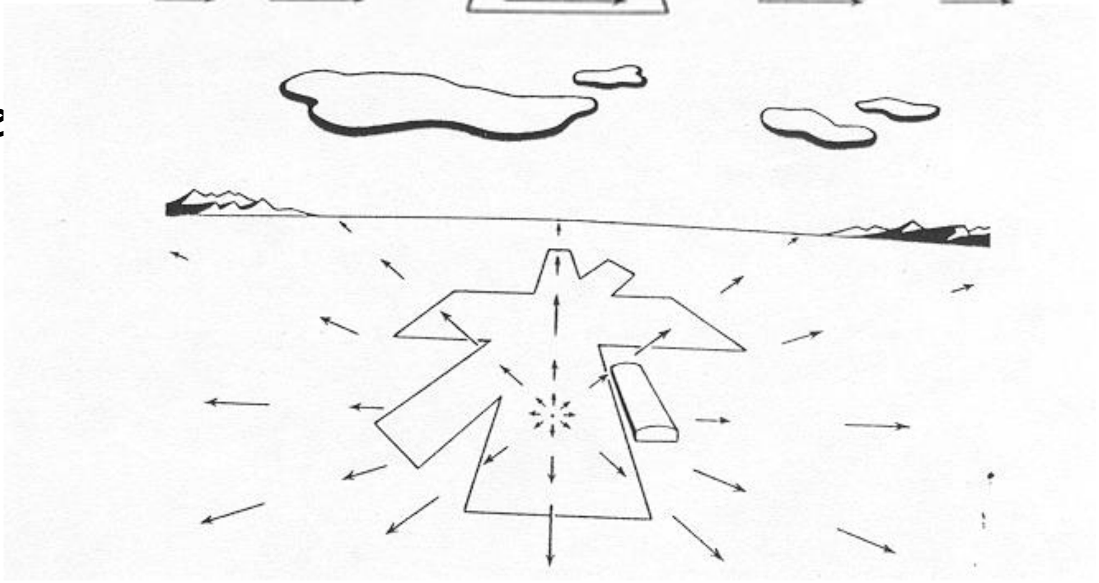


Optical Flow as Seen from an Aircraft

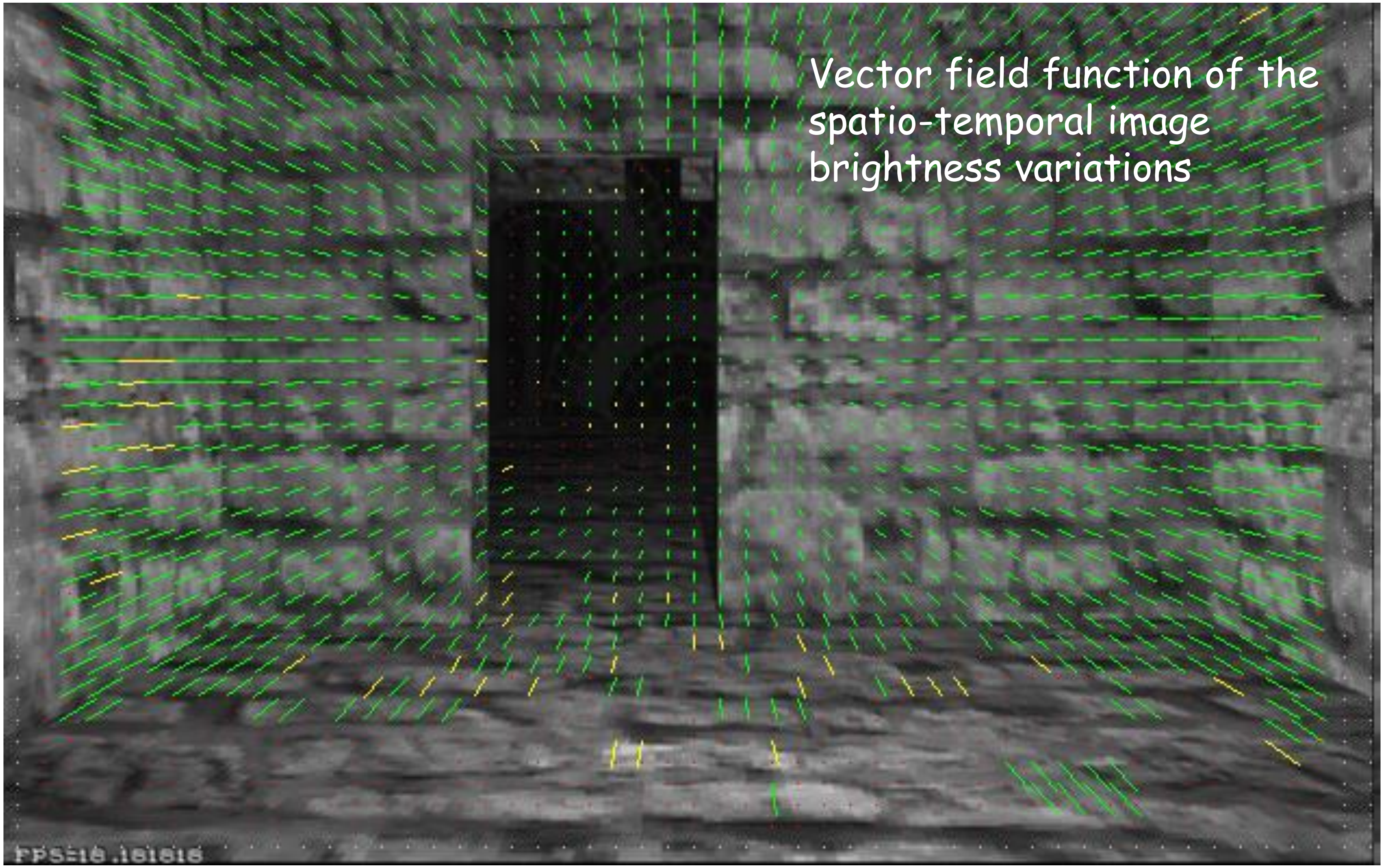
- MF of a pilot looking to the right in level flight. FoE is off at infinity to the left or equivalently FoC is off to the right
- MF with a plane parallel to the ground FoE at infinity on the







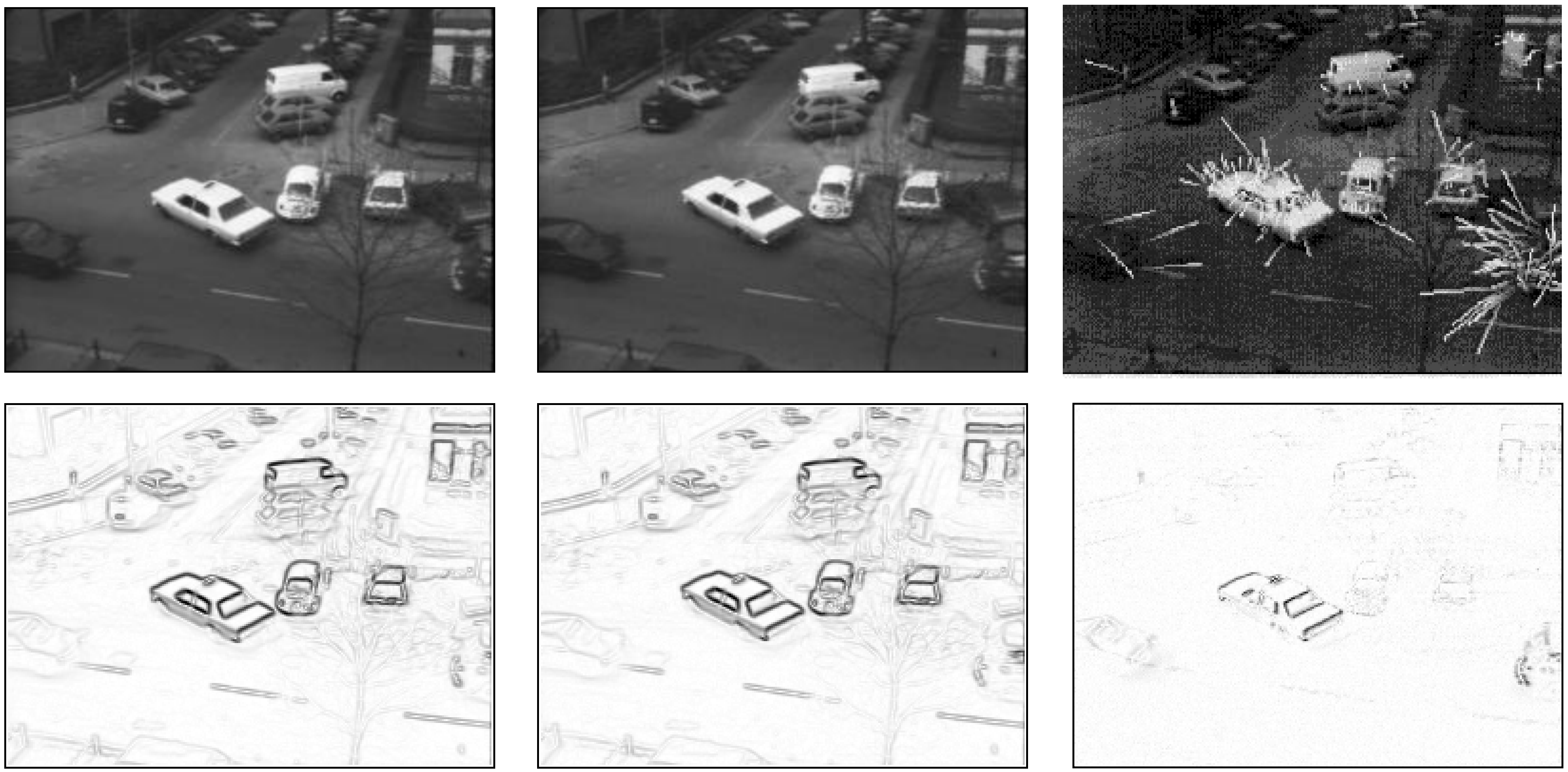
Optical flow



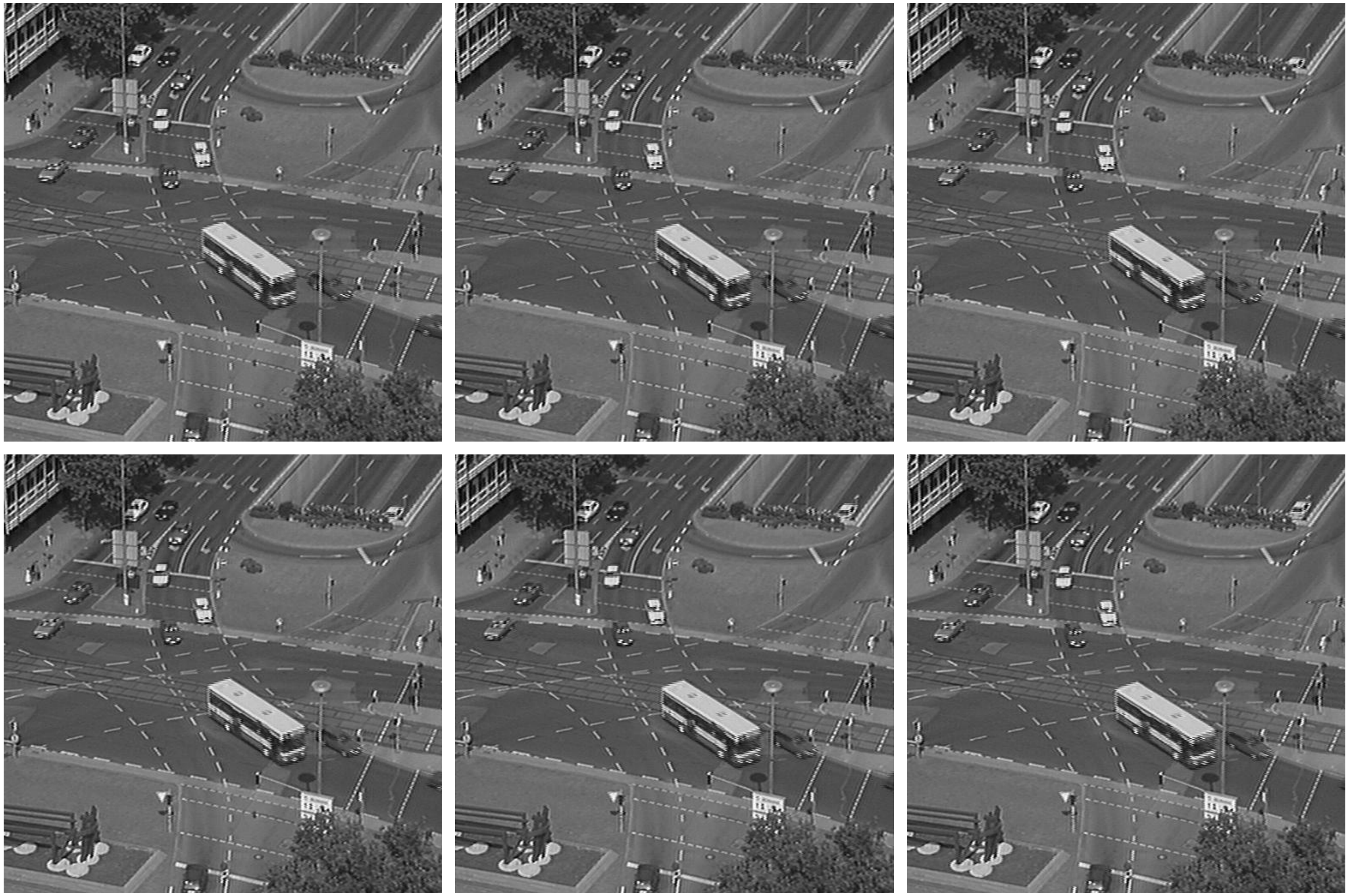
Vector field function of the spatio-temporal image brightness variations

Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

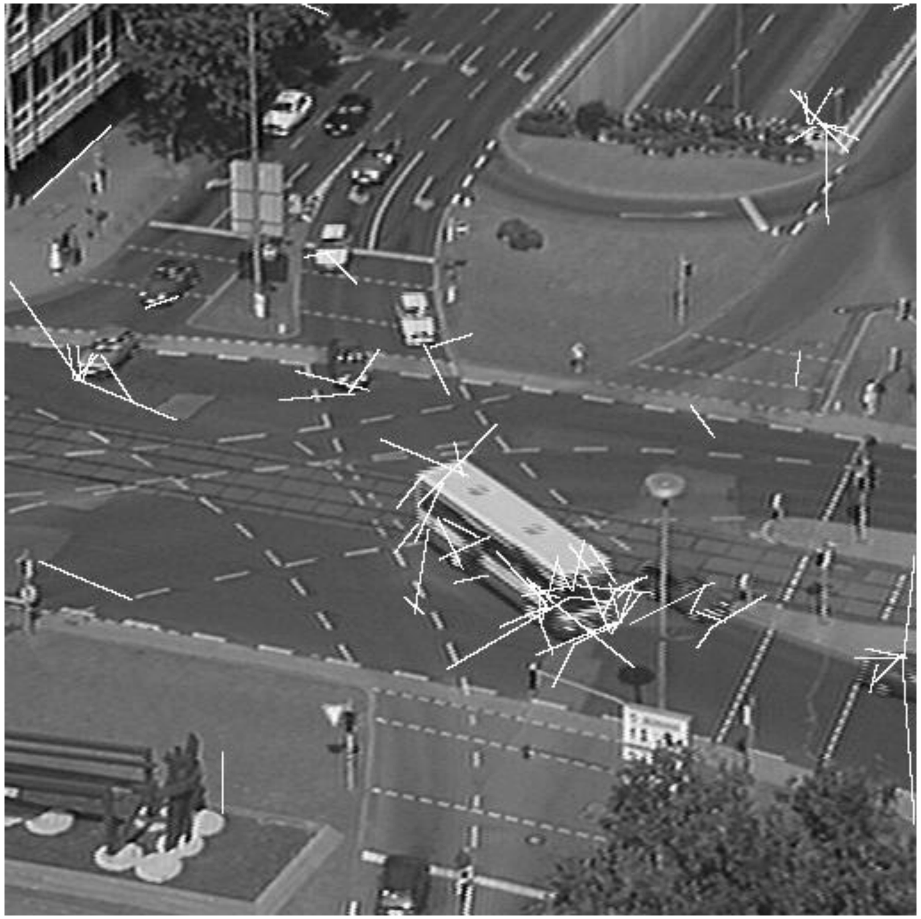
Example

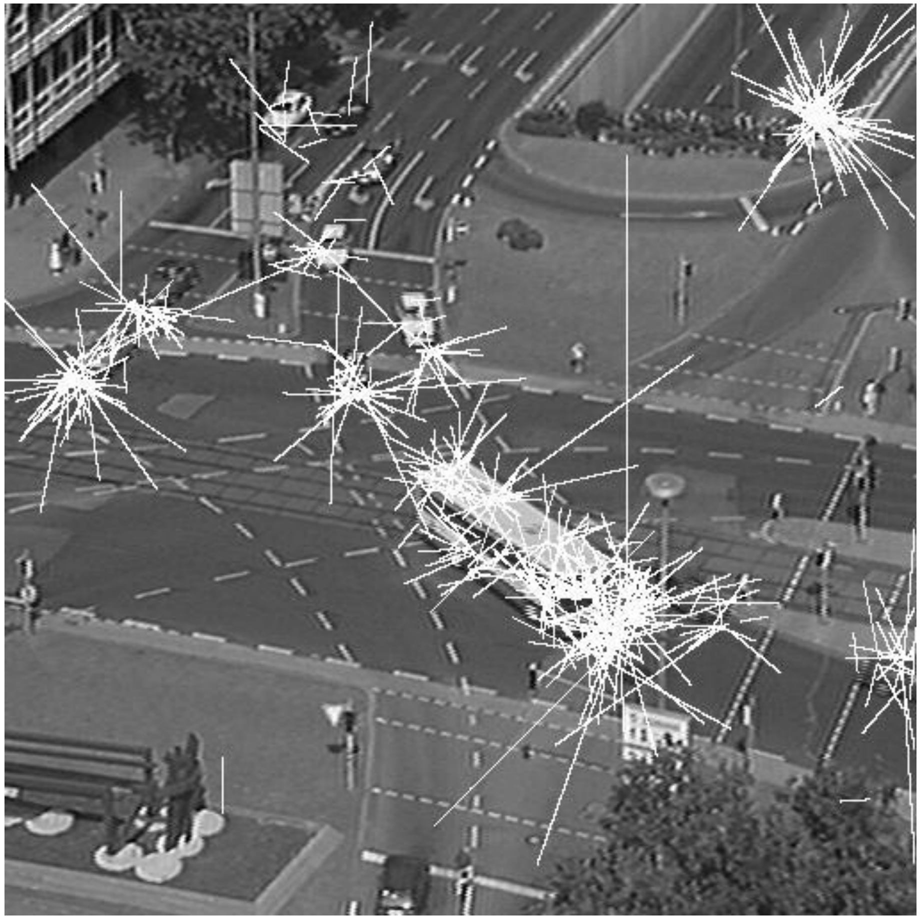


Example



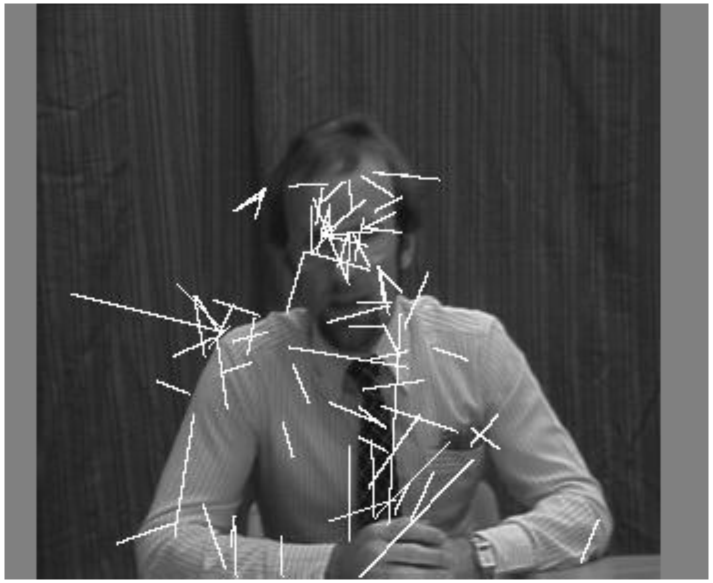
OF results

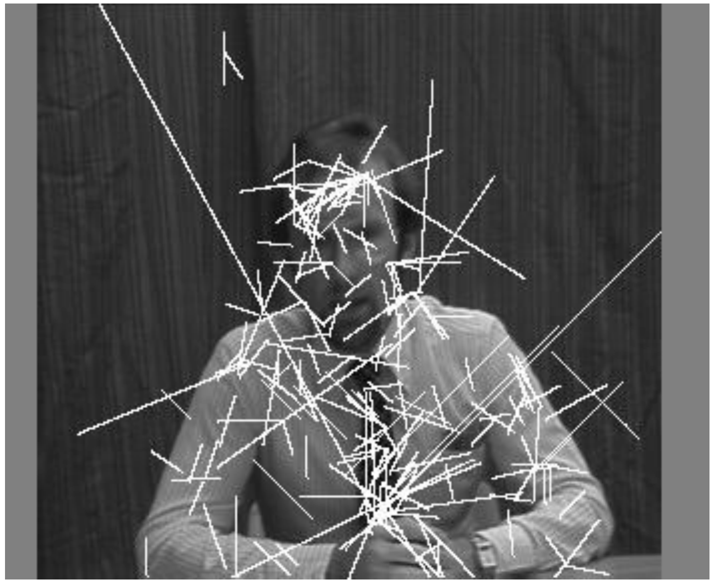




Example

Of results

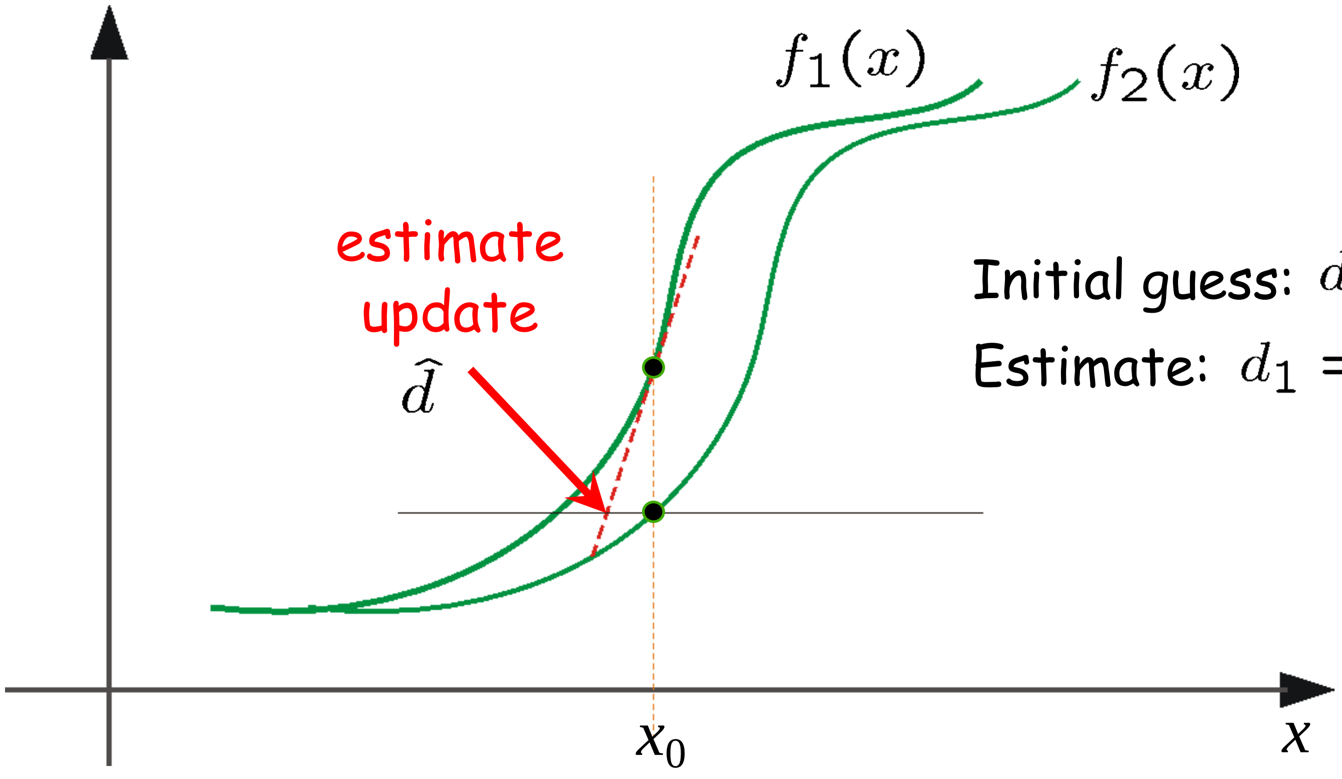




Iterative Refinement

- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation
- Warp one image toward the other using the estimated flow field
  (easier said than done)
- Refine estimate by repeating the process

Optical Flow: Iterative Estimation

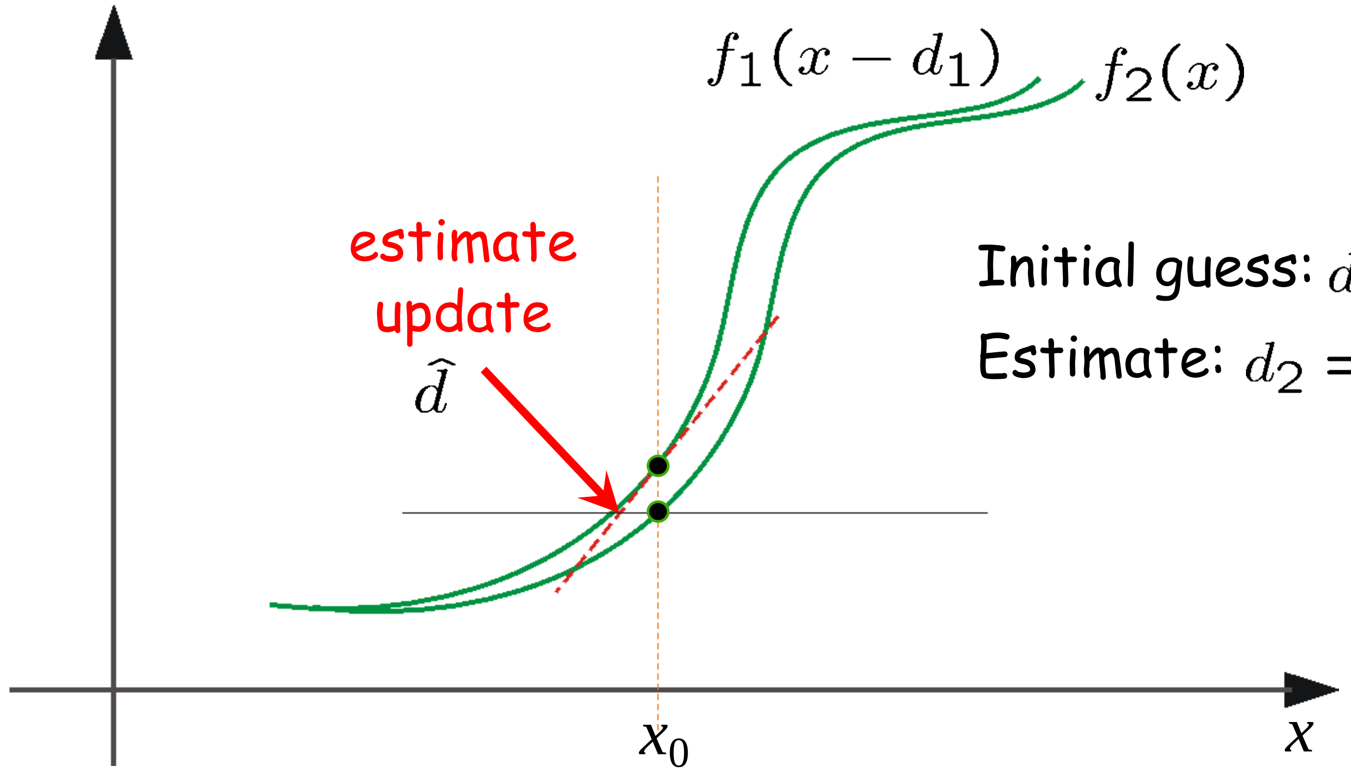


Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

(using *d* for *displacement* here instead of *u*)

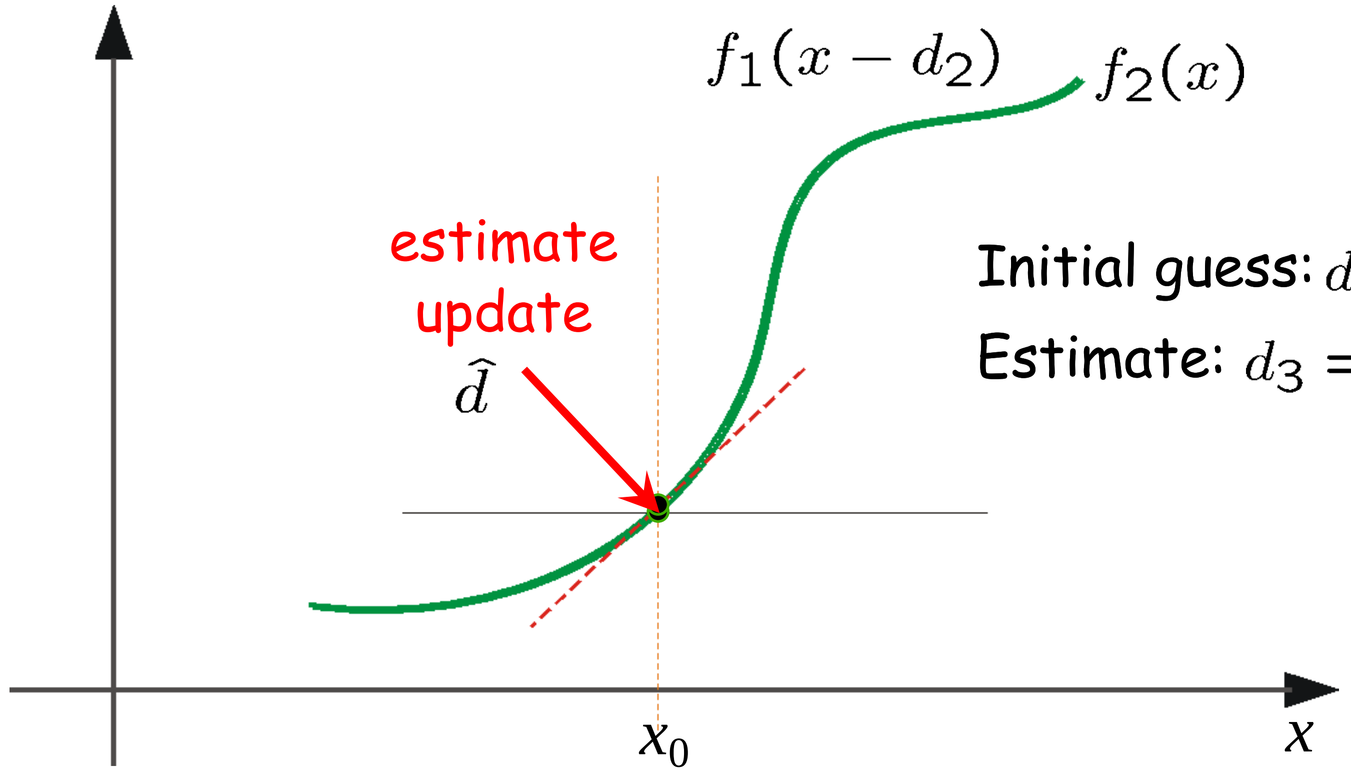
Optical Flow: Iterative Estimation



Initial guess: d_1

Estimate: $d_2 = d_1 + \hat{d}$

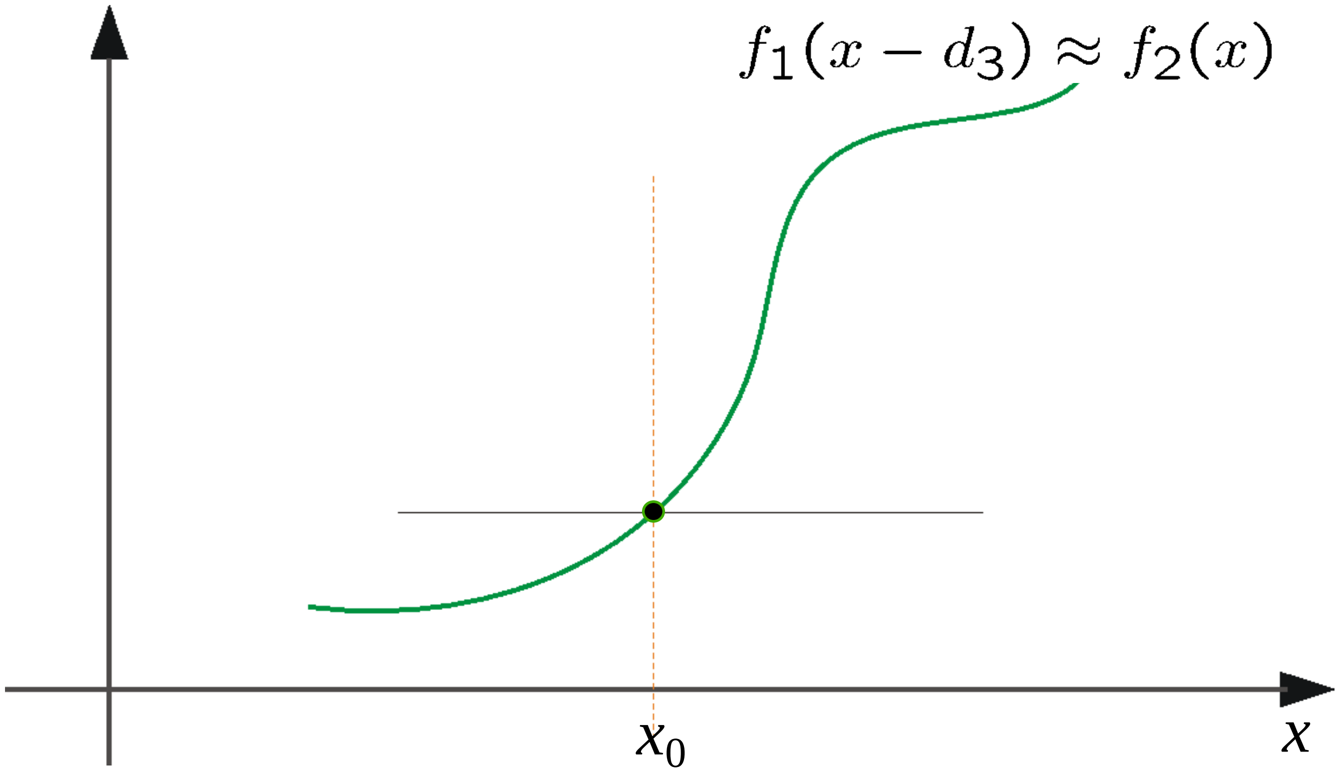
Optical Flow: Iterative Estimation



Initial guess: d_2

Estimate: $d_3 = d_2 + \hat{d}$

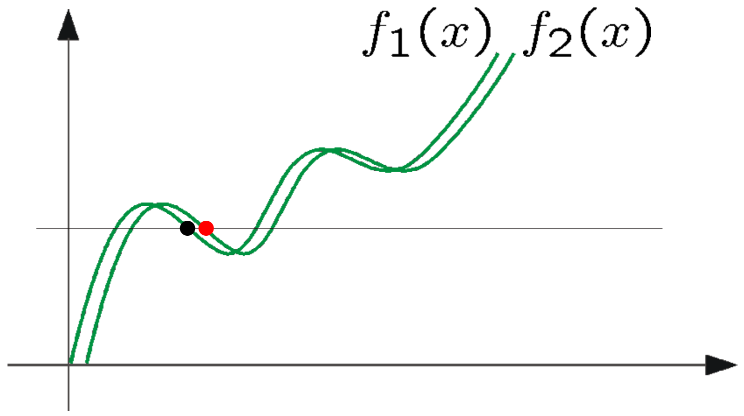
Optical Flow: Iterative Estimation



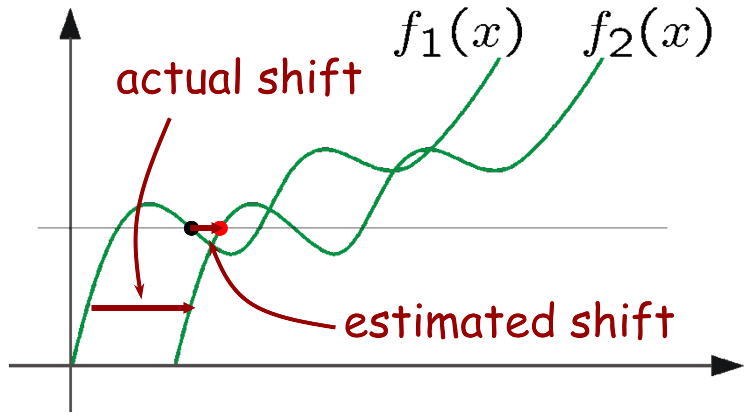
Optical Flow: Aliasing

Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.

I.e., how do we know which 'correspondence' is correct?



nearest match is correct (no aliasing)

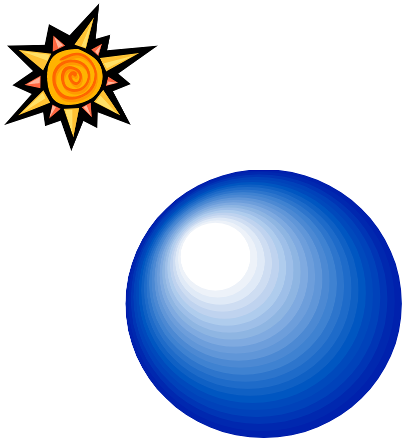


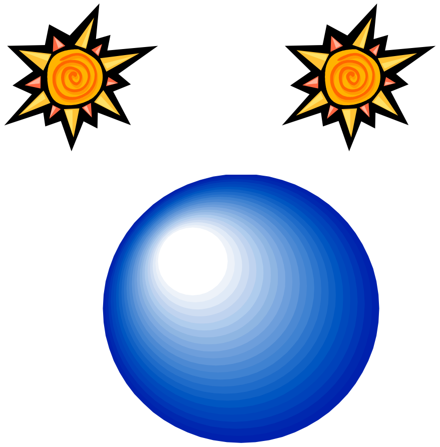
nearest match is incorrect (aliasing)

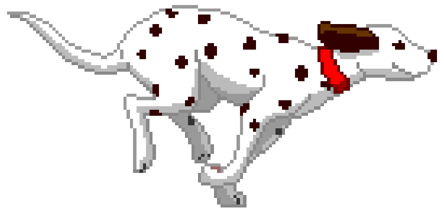
To overcome aliasing: coarse-to-fine estimation.

When does it break?

The screen is stationary yet displays motion


Homogeneous objects generate zero optical flow.


Fixed sphere. Changing light source.


Non-rigid texture motion

Computing Optical Flow: Improvements

- Larger motion: how to maintain “differential” approximation?
  - Solution: iterate
- Even better: adjust window / smoothing
  - Early iterations: use larger Gaussians to allow more motion
  - Late iterations: use less blur to find exact solution, lock on to high-frequency detail

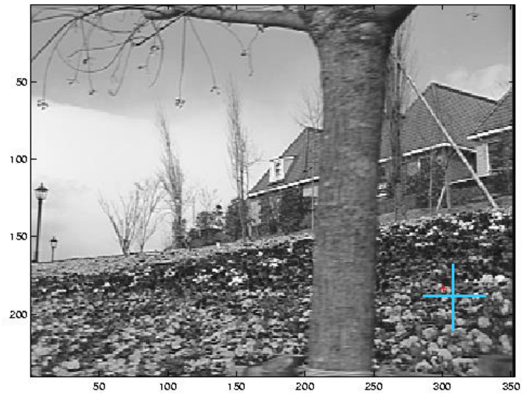
Revisiting the Small Motion Assumption

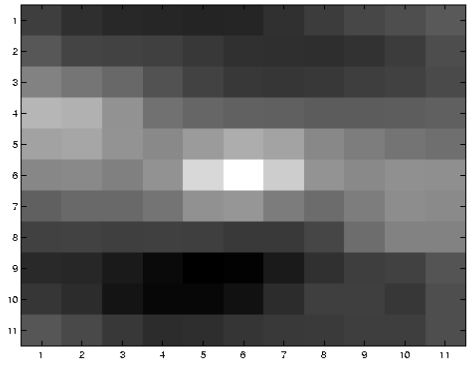
- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)
  - How might we solve this problem?

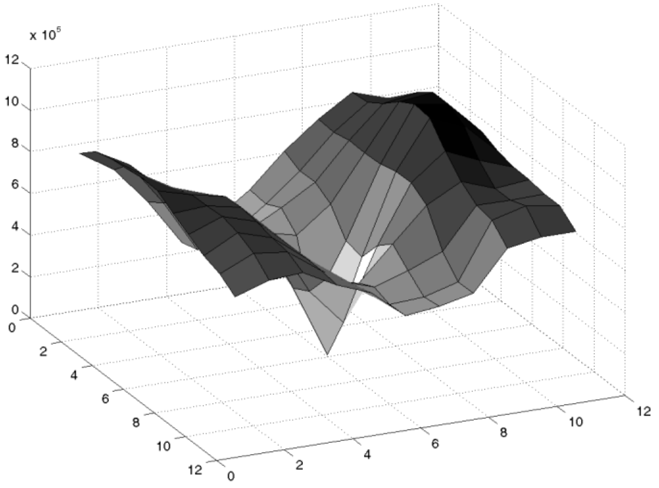
Correlation and SSD

- For large displacements, do template matching as was used in stereo disparity search.
  - Define a small area around a pixel as the template
  - Match the template against each pixel within a search area in next image.
  - Use a match measure such as correlation, normalized correlation, or sum-of-squares difference
  - Choose the maximum (or minimum) as the match
  - Sub-pixel interpolation also possible

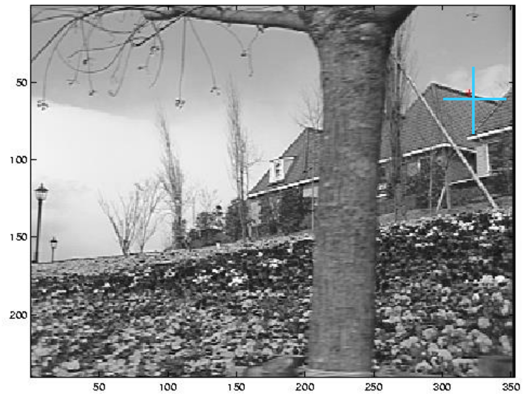
SSD Surface – Textured area

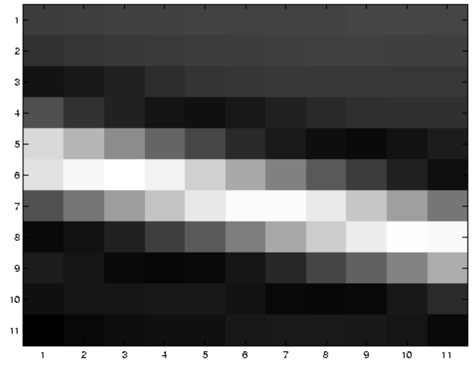


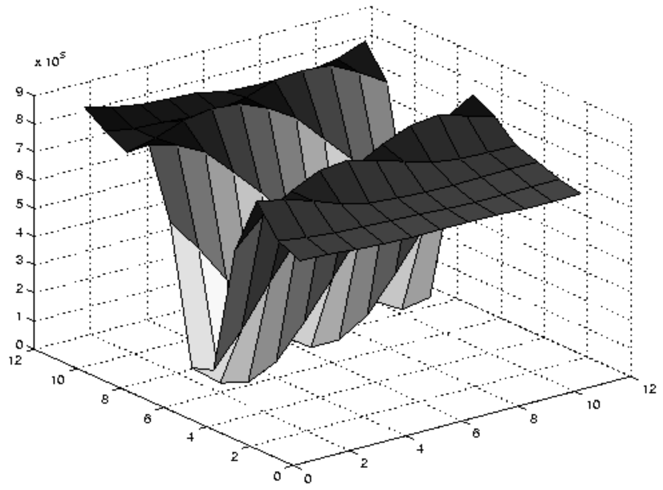




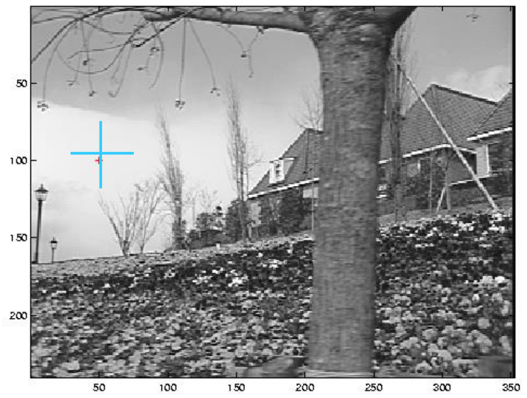
SSD Surface -- Edge

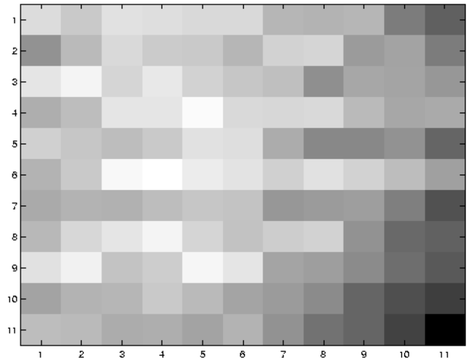


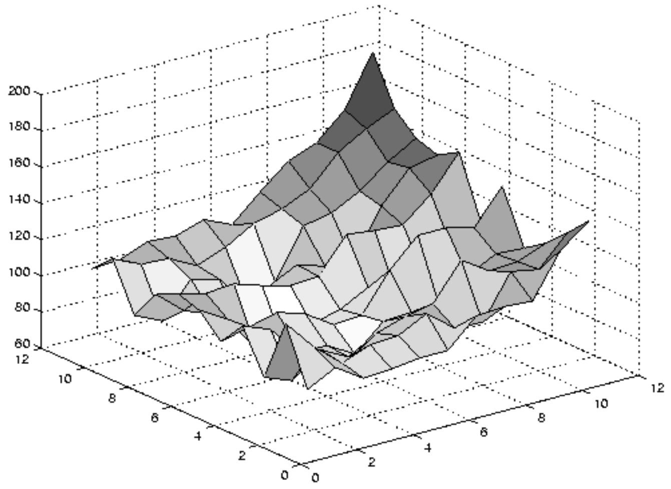




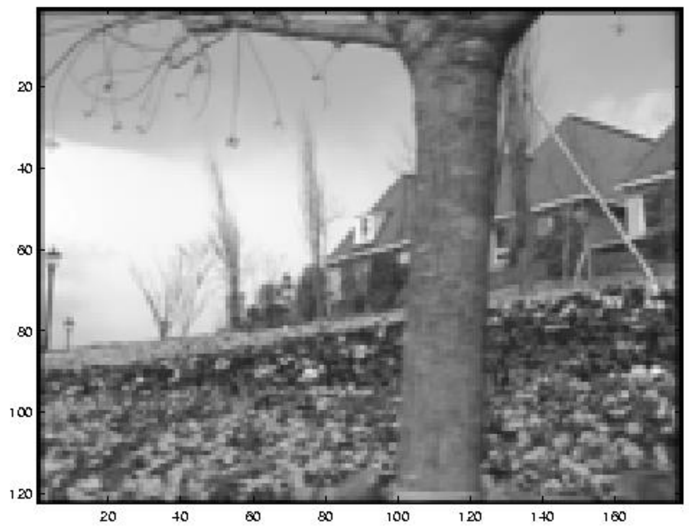
SSD - homogeneous area

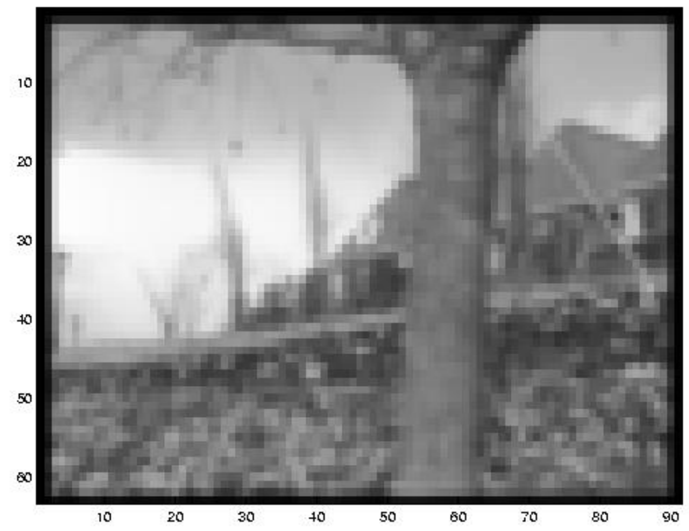


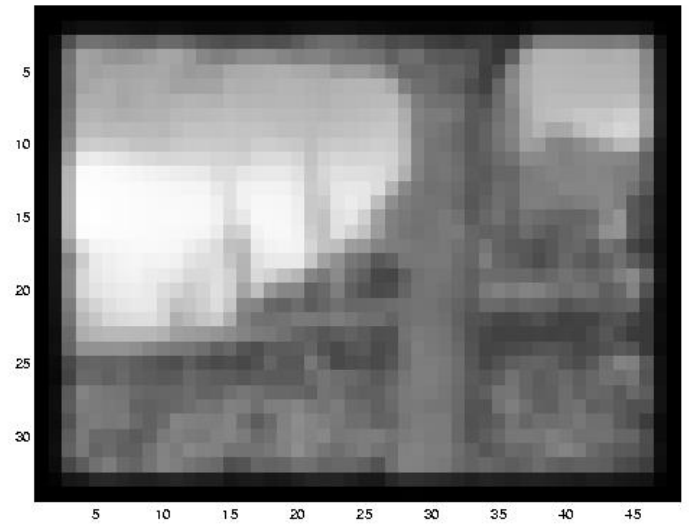


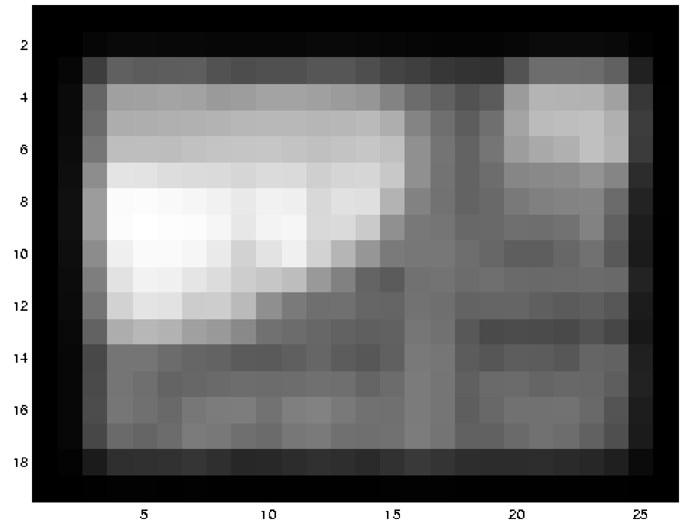


Reduce the Resolution!

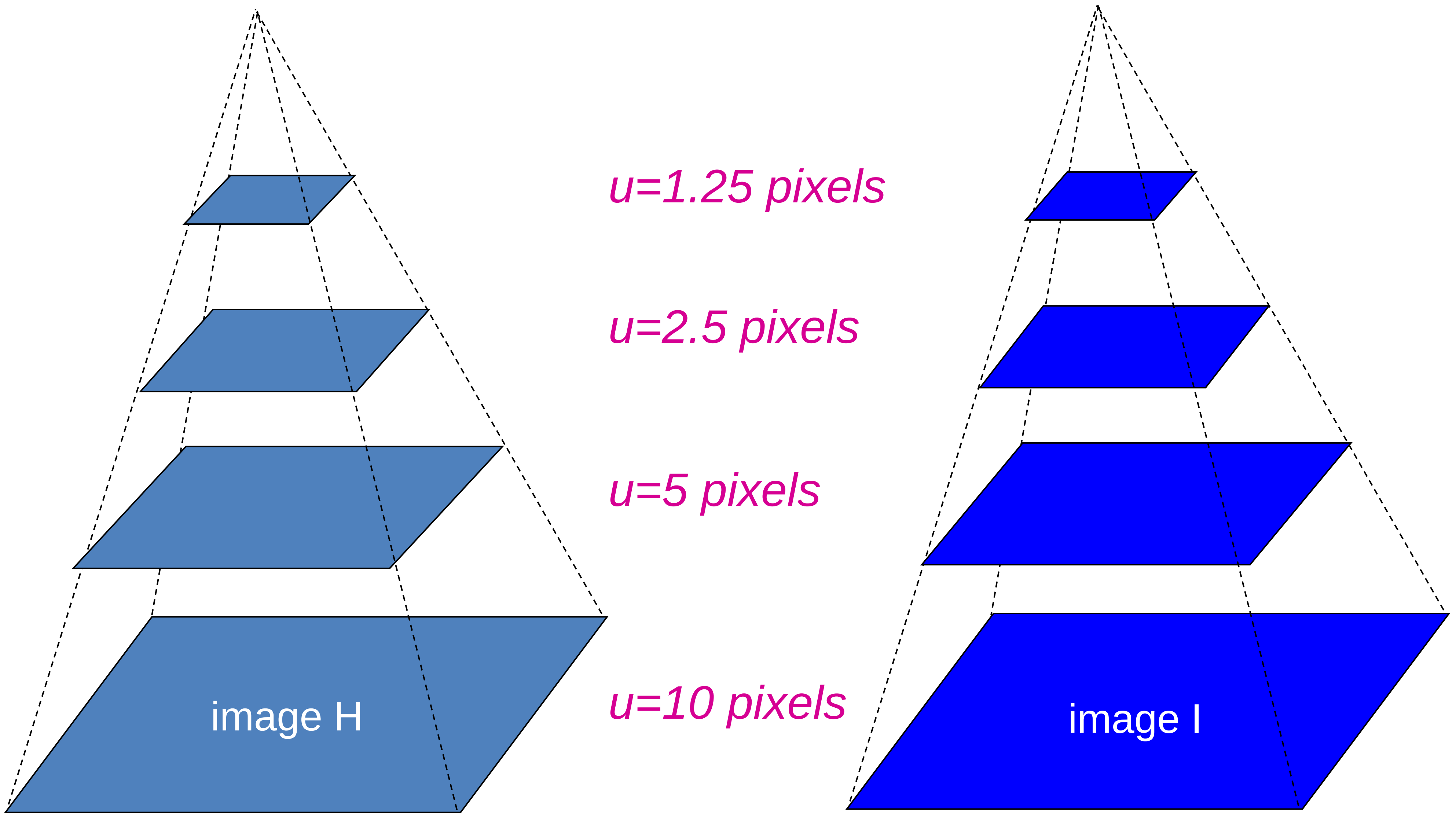








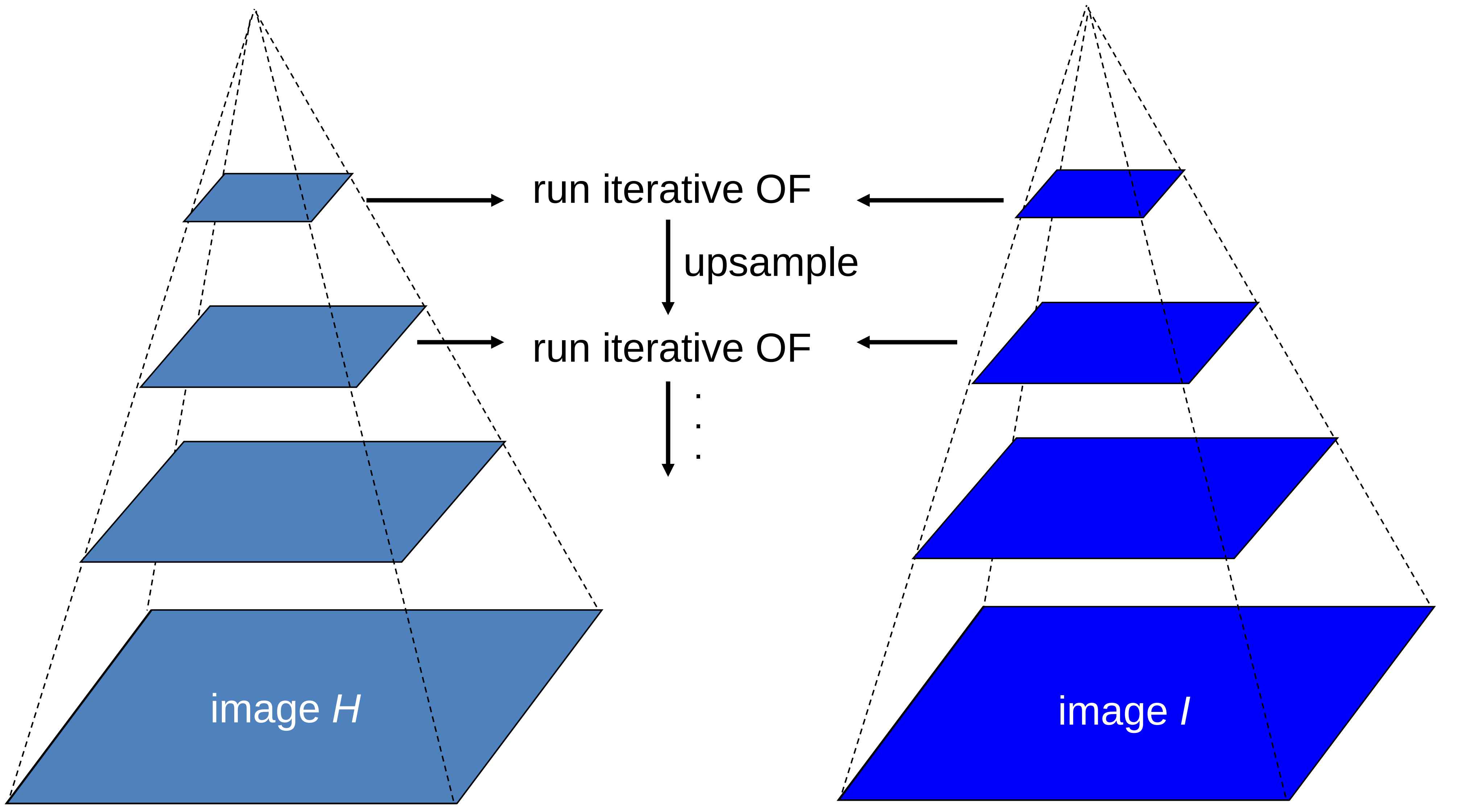
Coarse-to-fine Optical Flow Estimation



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Coarse-to-fine Optical Flow Estimation



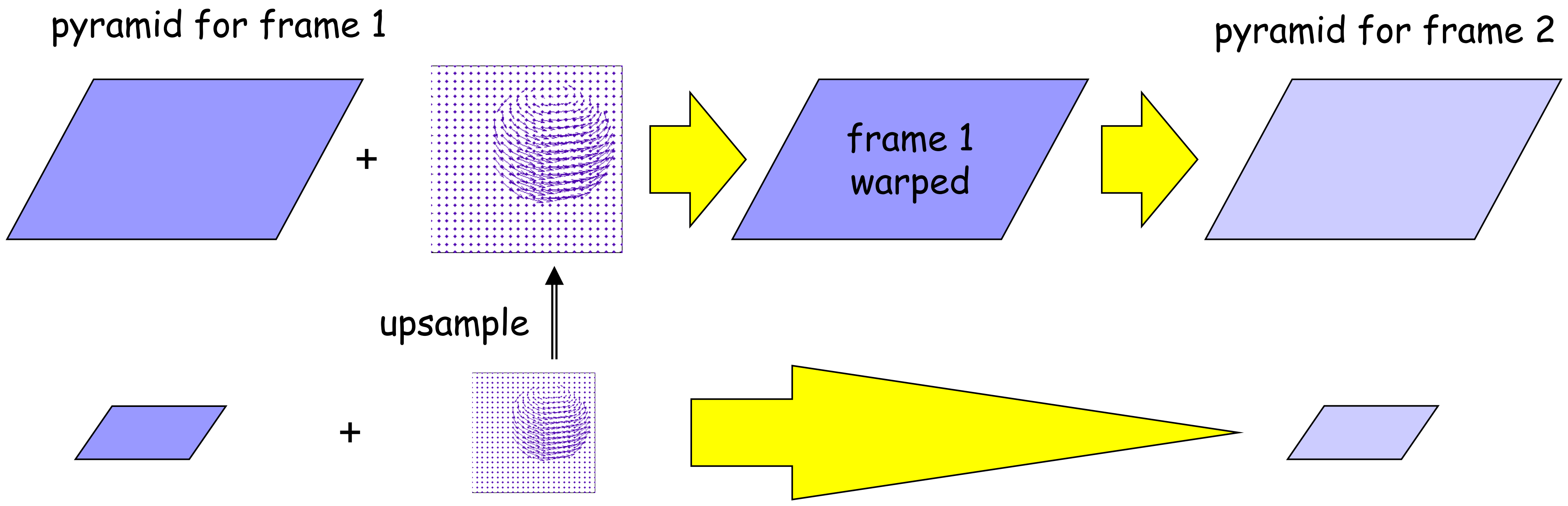
Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Multiscale Optical Flow

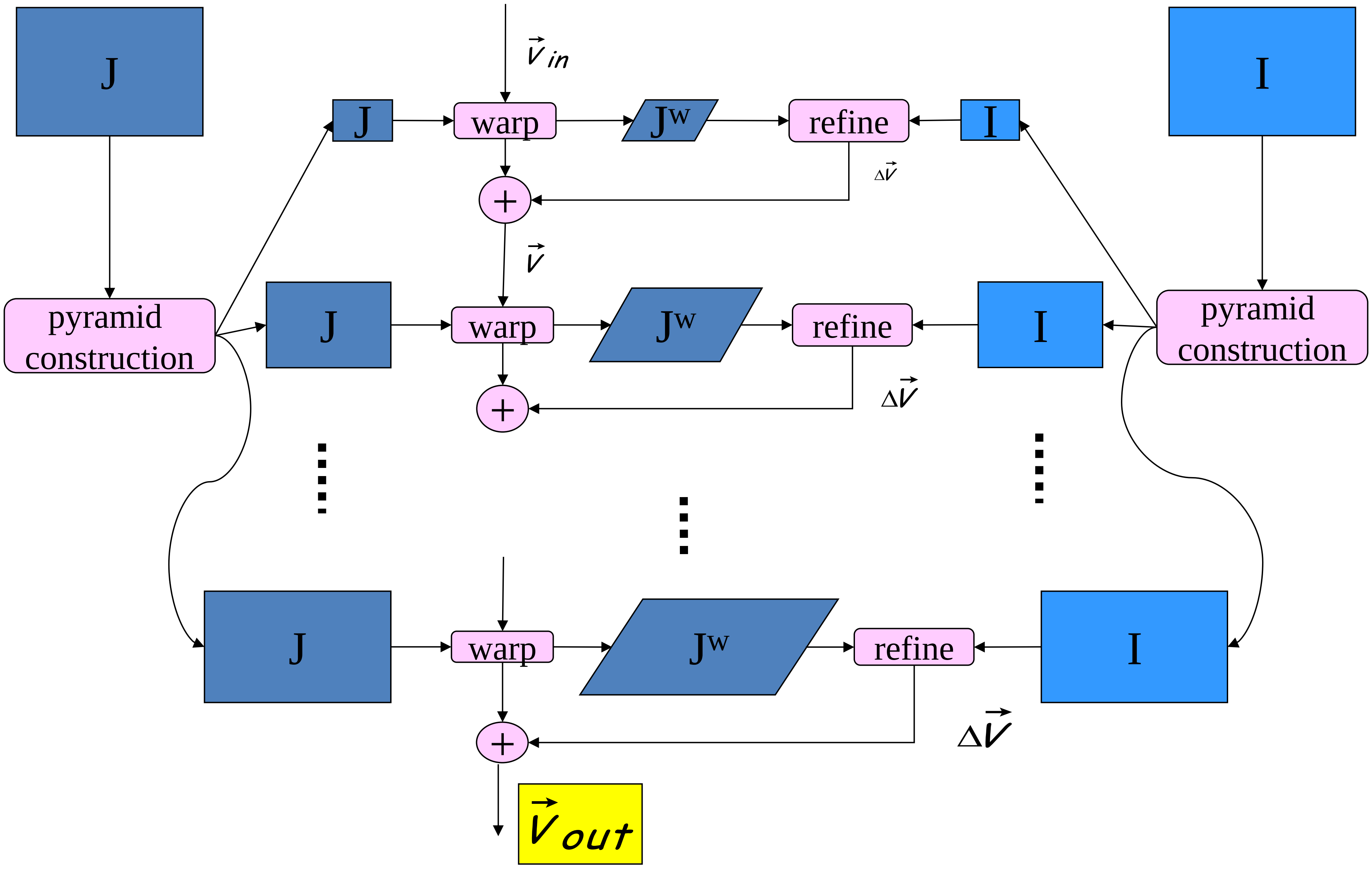
Linearization: valid only for small flow

$$I(x+u, y+v, t+1) - I(x, y, t) = I_x u + I_y v + I_t$$



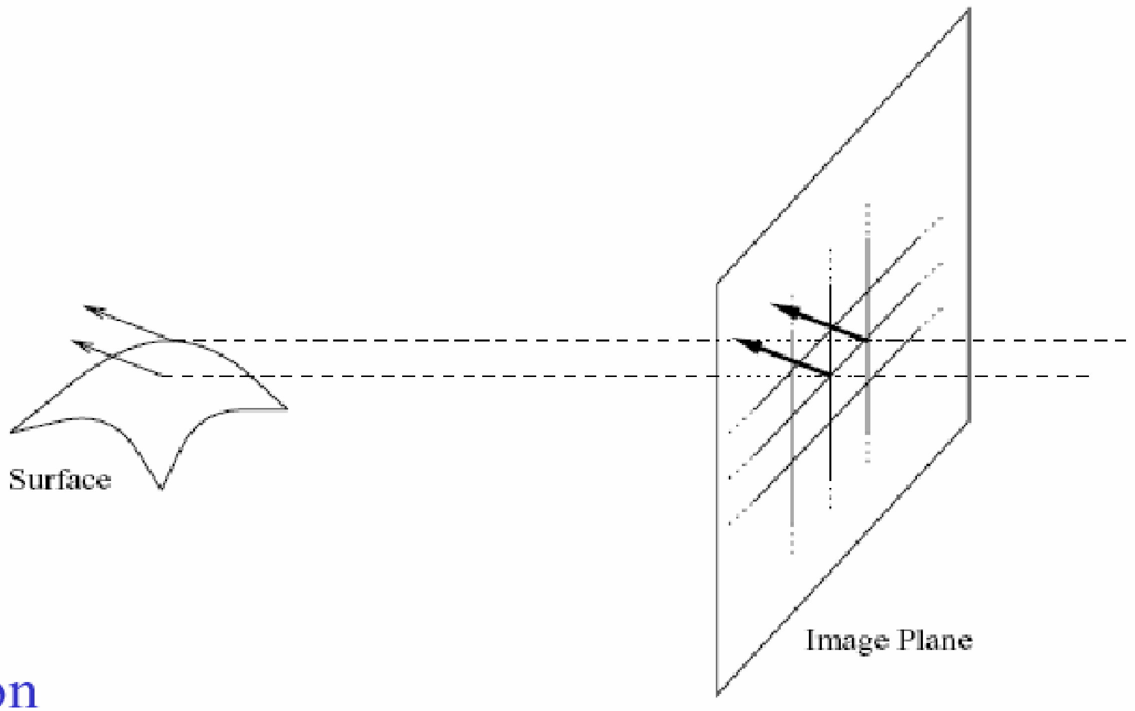
(other names: “warping”, “coarse-to-fine”, “multiresolution”)

Coarse-to-fine estimation



Optical Flow Assumptions:

Spatial Coherence



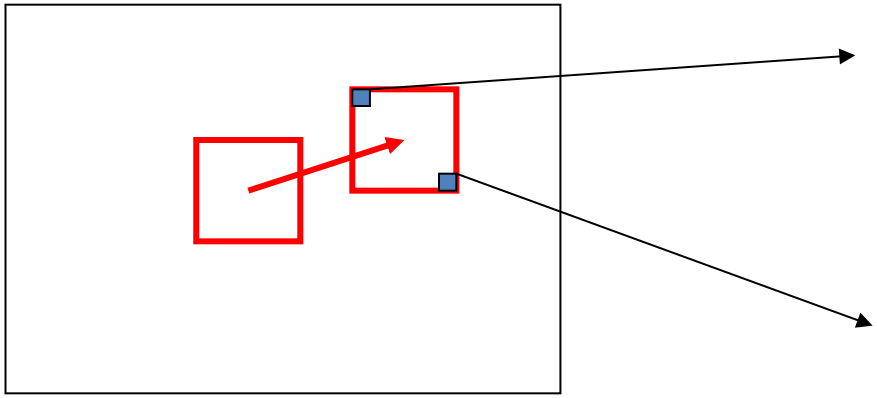
Assumption

* Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
* Since they also project to nearby points in the image, we expect spatial coherence in image flow.

Local smoothness Lucas Kanade (1984)

$$I_x u + I_y v = -I_t \quad \Longrightarrow \quad \begin{bmatrix} I_x & I_y \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

Assume constant (u,v) in small neighborhood



$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \end{bmatrix}$$

$$\boxed{A\vec{u} = b}$$

Goal: Minimize $\left\| A\vec{u} - b \right\|^2$ Method: Least-Squares

Regularization Horn and Schunk (1981)

Add global smoothness term

Smoothness error:

$$E_s = \iint_D \left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right) dx\, dy$$

Error in brightness constancy equation

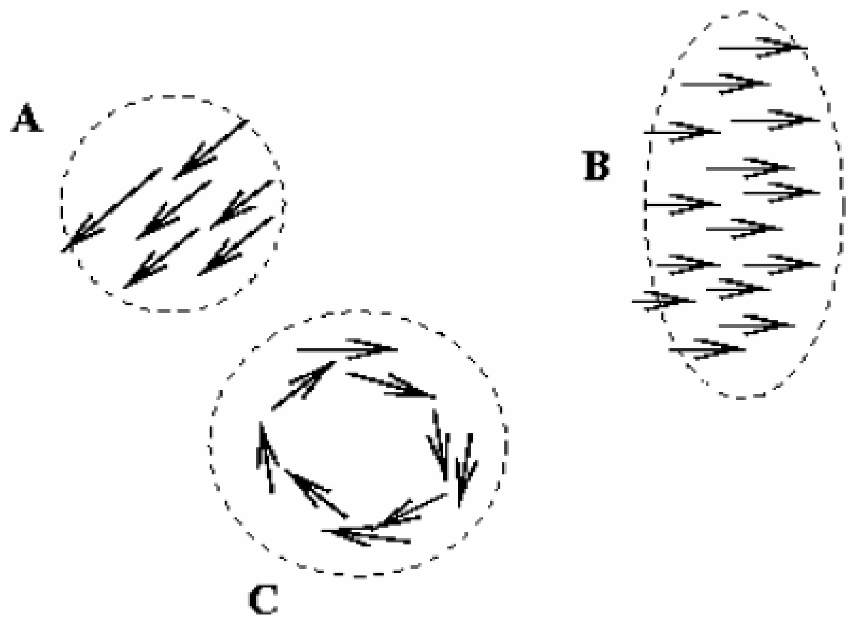
$$E_c = \iint_D \left(I_x u + I_y v + I_t\right)^2 dx\, dy$$

Minimize: $E_c + \lambda E_s$

Solve by calculus of variations

Deployment of Video segmentation

- Segment the video into multiple *coherently* moving objects
  - Background subtraction
  - Shot boundary detection
  - Motion segmentation



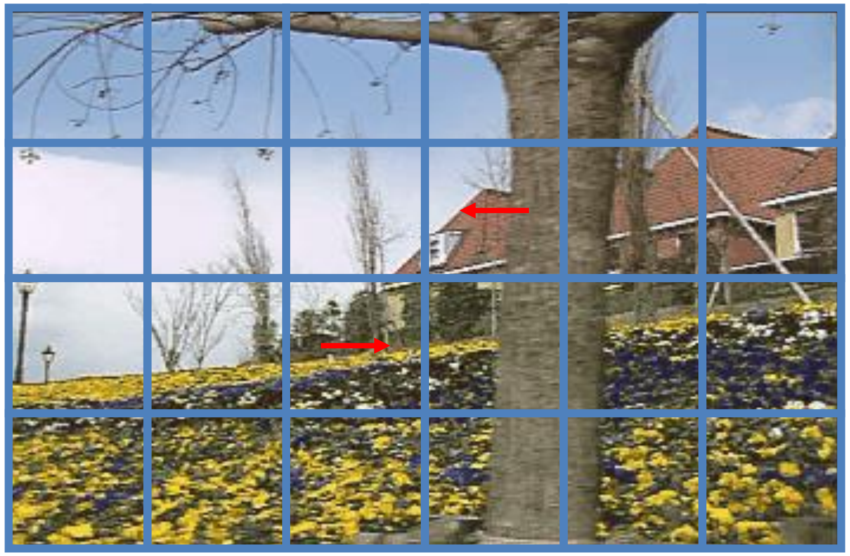
Layered Representation

For scenes with multiple affine motions

$$\begin{tikzcd}
\text{Estimate dominant motion parameters} \arrow[d, shift left] \\
\text{Reject pixels which do not fit} \arrow[u, shift left] \arrow[d] \\
\text{Convergence} \arrow[d] \\
\text{Restart on remaining pixels} \arrow[uuu, bend left]
\end{tikzcd}$$

Block-based motion prediction

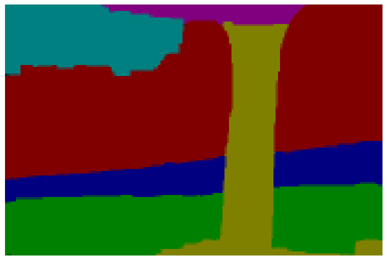
- Break image up into square blocks
- Estimate translation for each block
- Use this to predict next frame, code difference



Layered motion

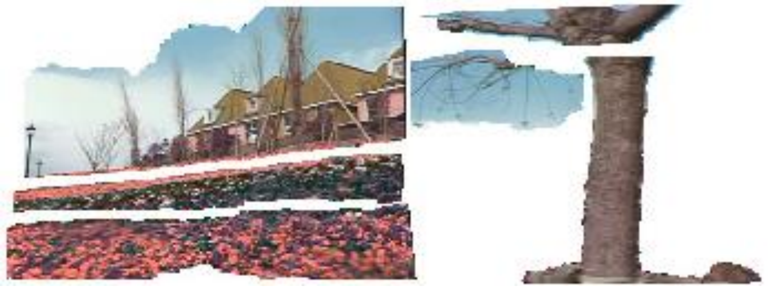
- Break image sequence up into "layers":

÷



=

- Describe each layer's motion

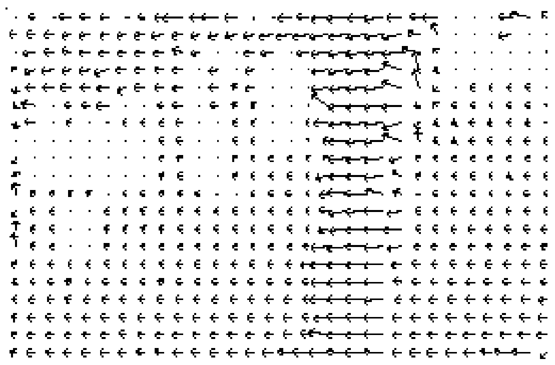
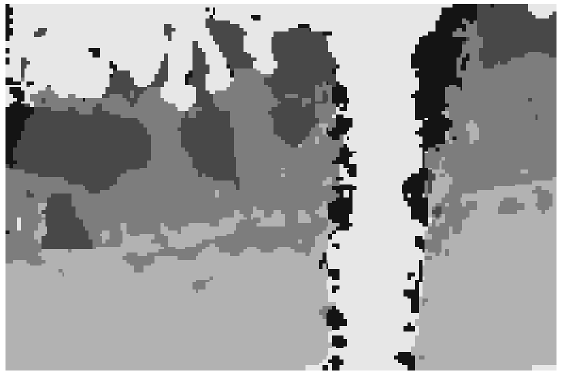
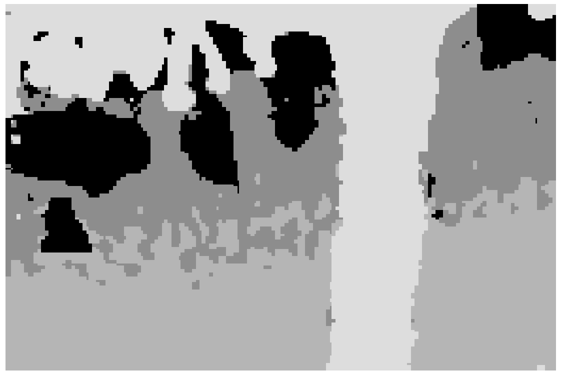
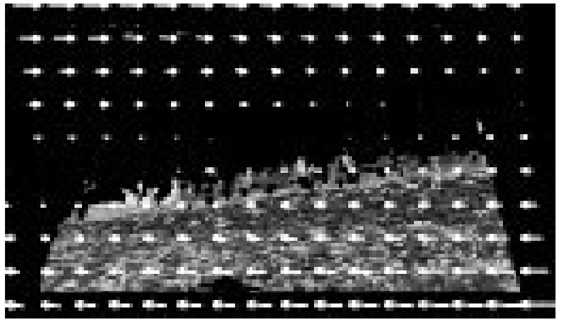
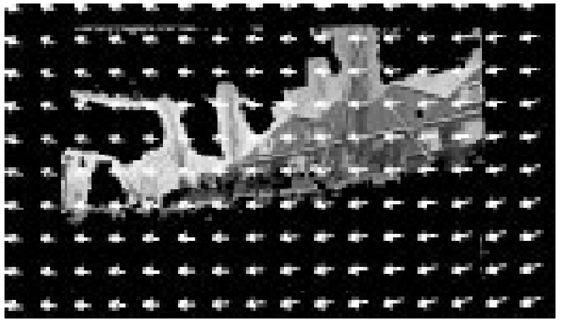
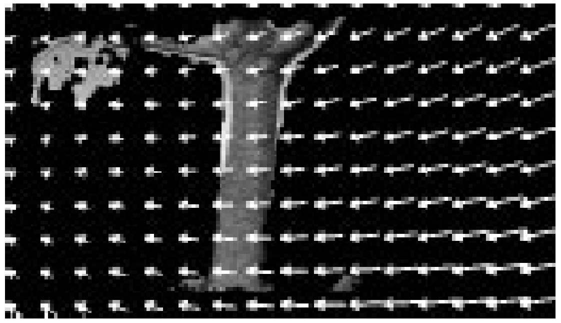


J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. IEEE Transactions on Image Processing, 3(5):625--638, September 1994.

Layer synthesis

- For each layer:
  - stabilize the sequence with the affine motion
  - compute median value at each pixel
- Determine occlusion relationships

Results

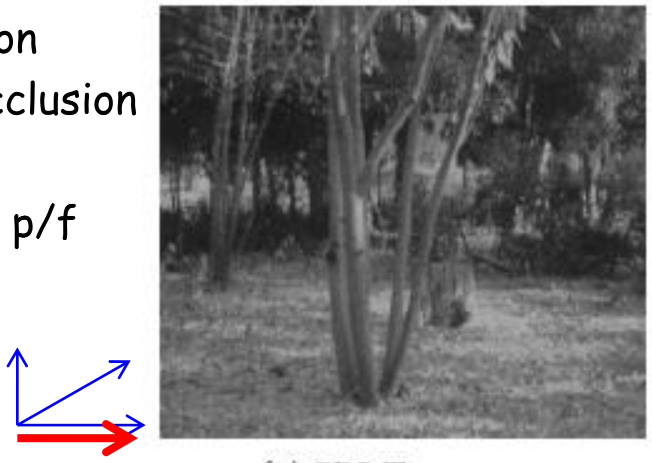







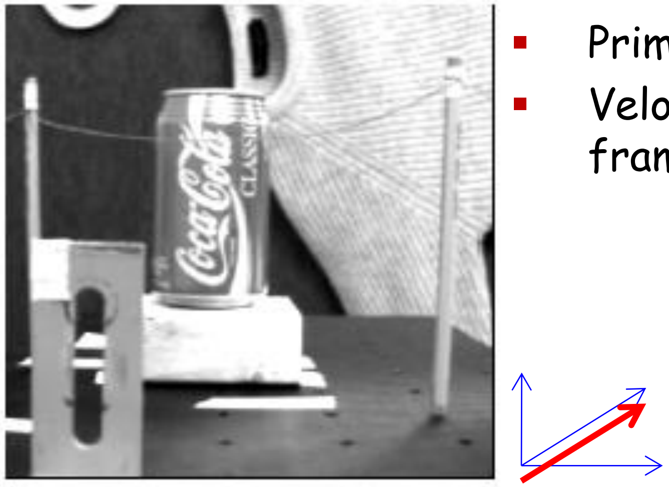
Layered motion

- Advantages:
  - can represent occlusions / disocclusions
  - each layer's motion can be smooth
  - video segmentation for semantic processing
- Difficulties:
  - how do we determine the correct number?
  - how do we assign pixels?
  - how do we model the motion?

Image Data Sets

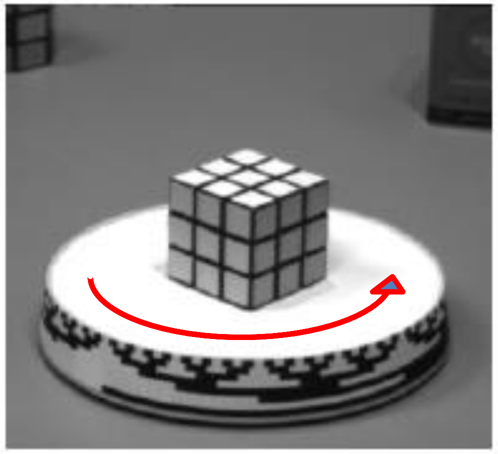
- Poor resolution
- Amount of occlusion
- Low contrast
- Velocities ~2 p/f


(a) SRI Trees


(b) NASA Sequence

- Primarily dilational
- Velocities <1 pixel/ frame

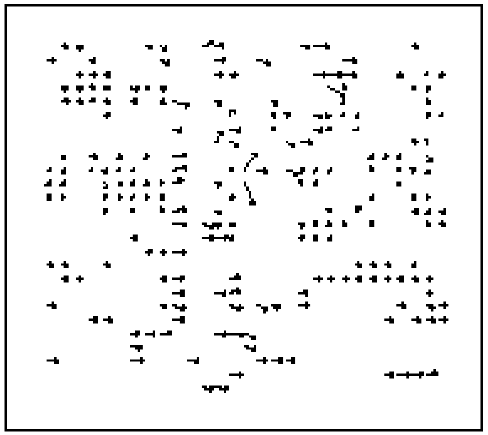
- The cube is rotating counterclockwise on a turntable
- Velocities on the table 1.2~1.4 p/f
- Velocities on the cube 0.2~0.5 p/f

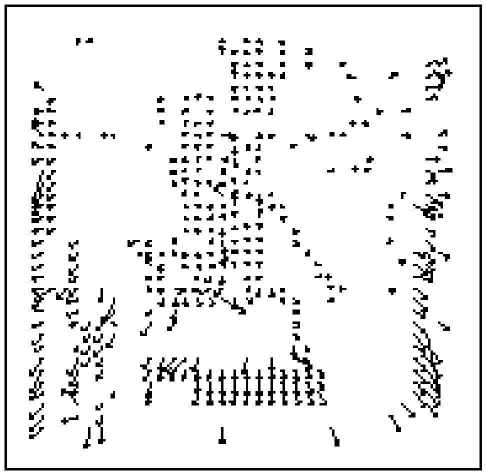

(c) Rubik Cube

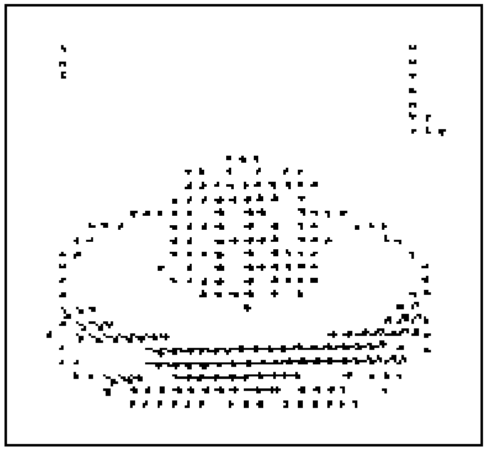
(d) Hamburg Taxi

- Four moving objects
- Speeds
  - Taxi 1.0 p/f
  - Car 3.0 p/f
  - Van 3.0 p/f
  - Pedestrian 0.3 p/f

Results: Horn-Schunck


(a) SRI Trees


(b) NASA Sequence


(c) Rubik Cube

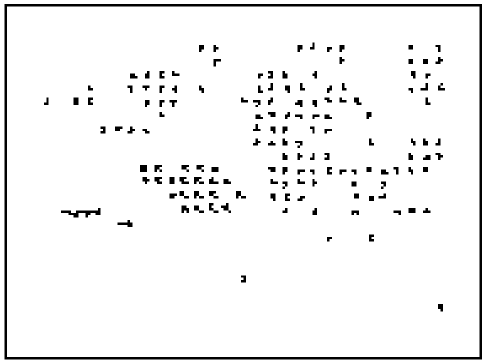
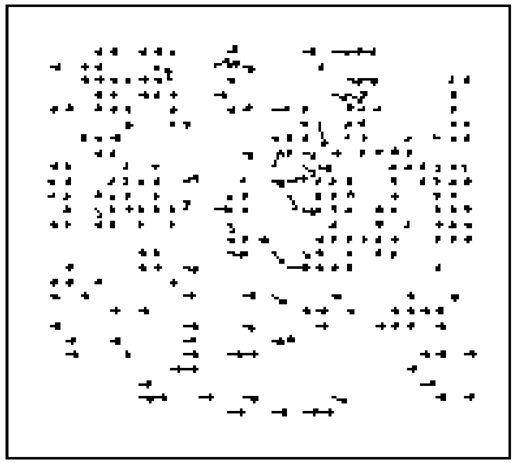
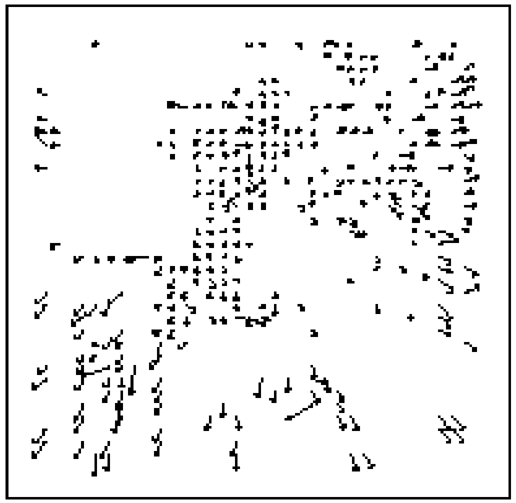
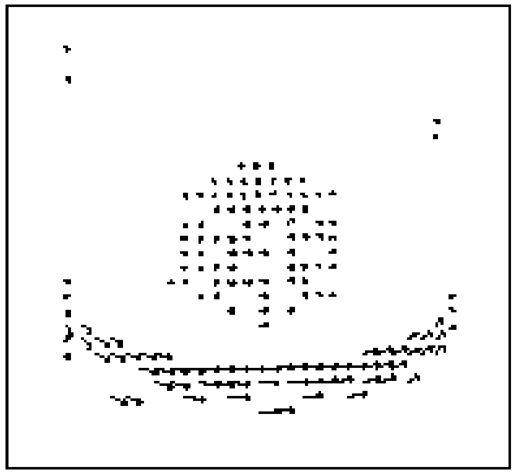

(d) Hamburg Taxi

Figure 5.1: *Flow fields for the modified* **Horn and Schunck** *technique (spatiotemporal Gaussian presmoothing and 4-point central differences) applied to real image data.* *The velocity estimates were thresholded using* $\| \nabla I \| \geq 5.0$.

Results: Lucas-Kanade


(a) SRI Trees


(b) NASA Sequence


(c) Rubik Cube

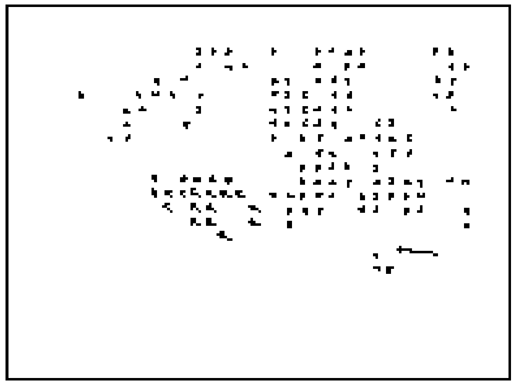
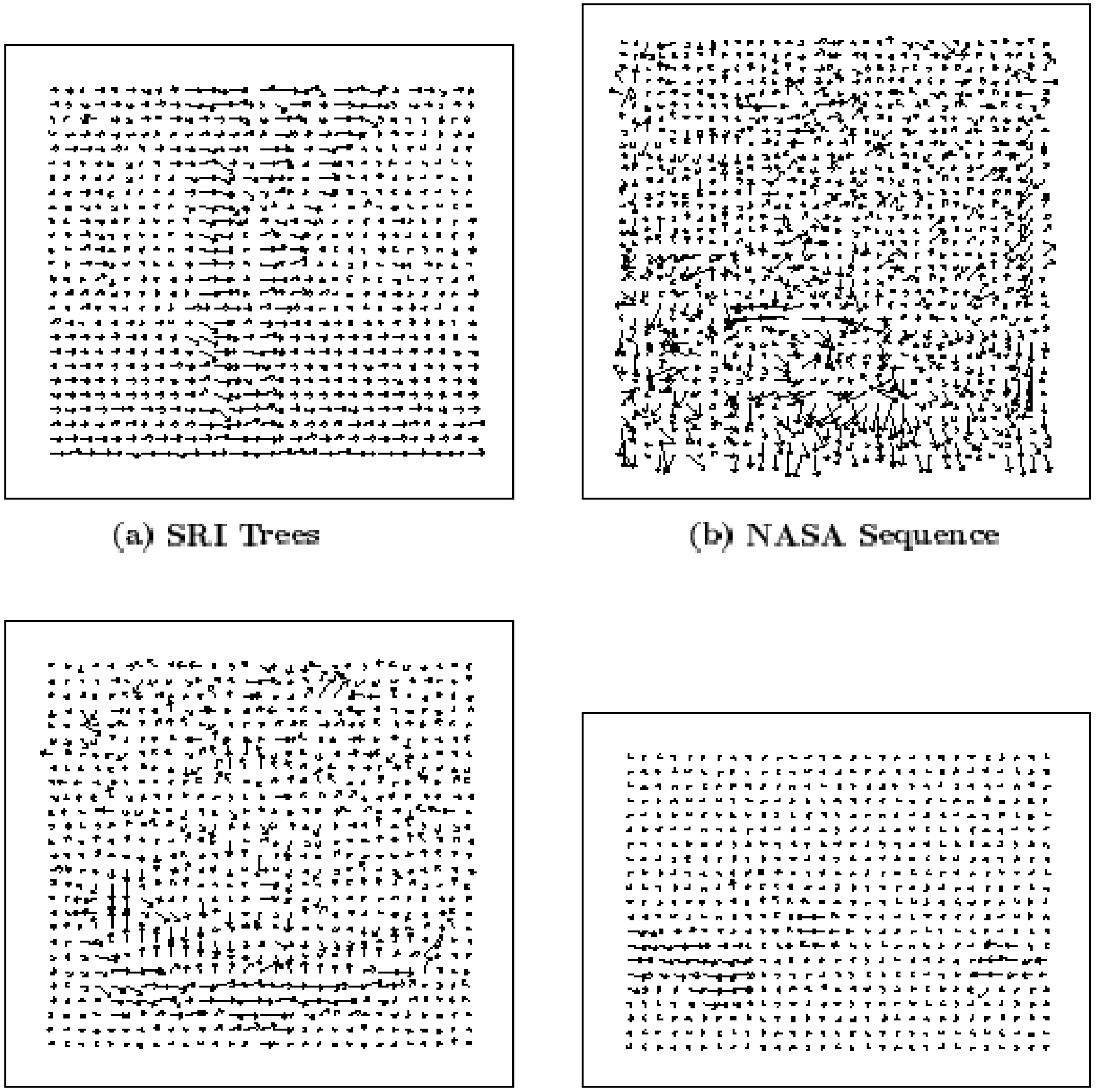

(d) Hamburg Taxi

Figure 5.2: *Flow fields for the* **Lucas and Kanade** *technique applied to real image data. All flow fields were produced with a threshold of* $\lambda_2 \geq 1.0$

Results: Anandan



(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 5.5: *Flow fields for the technique of* **Anandan** *applied to real image data. The results are unthresholded.*

References

Anandan, “A computational framework and an algorithm for the measurement of visual motion,” *IJCV* vol. 2, pp. 283-310, 1989.

Barron, Fleet, and Beauchemin, “Performance of Optical Flow Techniques,” *IJCV* 12:1, pp. 43-77, 1994.

Beauchemin and Barron, “The Computation of Optical Flow,” *ACM Computing Surveys*, 27:3, pp. 433-467, 1995.

Fleet and Jepson, “Computation of component image velocity from local phase information,” *IJCV*, vol. 5, pp. 77-104, 1990.

Heeger, “Optical flow using spatiotemporal filters,” *IJCV*, vol. 1, pp. 279-302, 1988.

Horn and Schunck, “Determining Optical Flow,” *Artificial Intelligence*, vol. 17, pp. 185-204, 1981.

Lucas and Kanade, “An iterative image registration technique with an application to stereo vision,” *Proc. DARPA Image Understanding Workshop*, pp. 121-130, 1981.

Singh, “An estimation-theoretic framework for image-flow computation,” *Proc. IEEE ICCV*, pp. 168-177, 1990.

Other break-downs

- Brightness constancy is **not** satisfied

  ↳ Correlation based methods

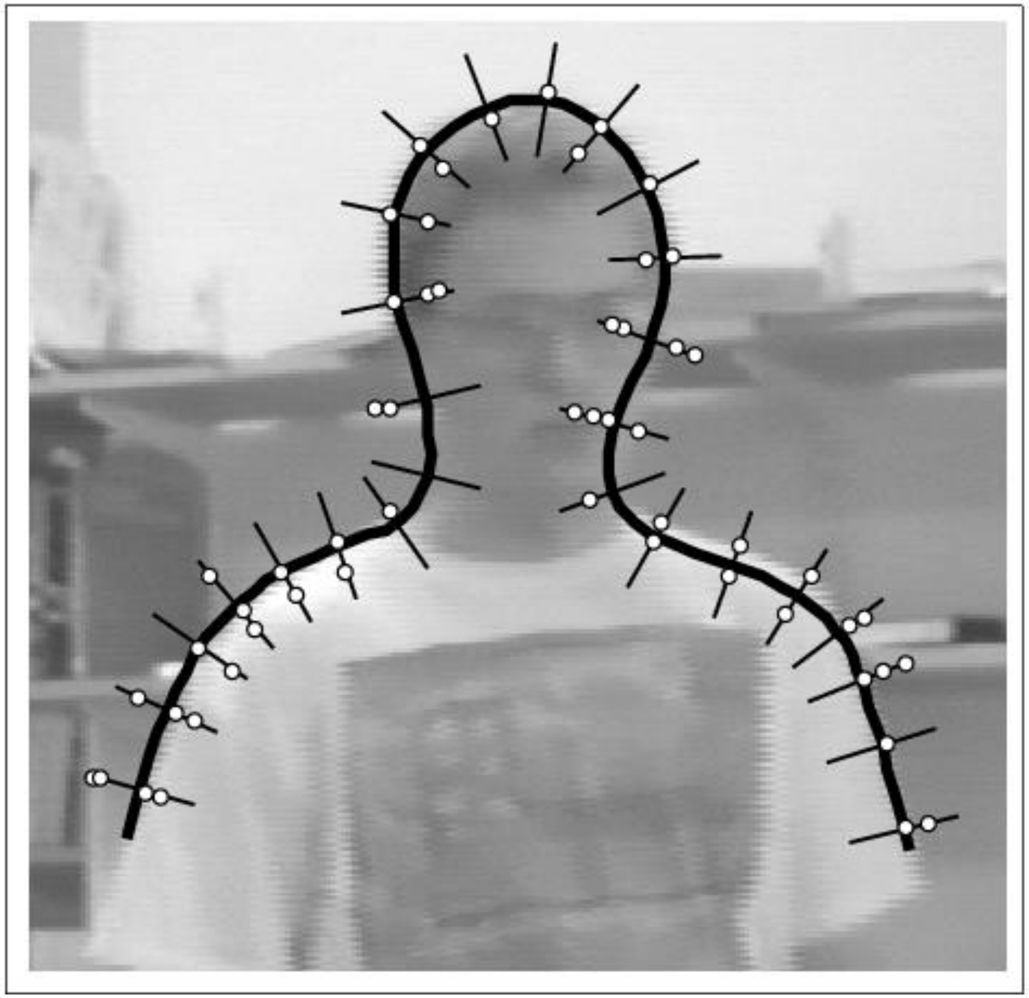
- A point does **not** move like its neighbors
  - what is the ideal window size?

  ↳ Regularization based methods

- The motion is **not** small (Taylor expansion doesn't hold)

  ↳ Use multi-scale estimation

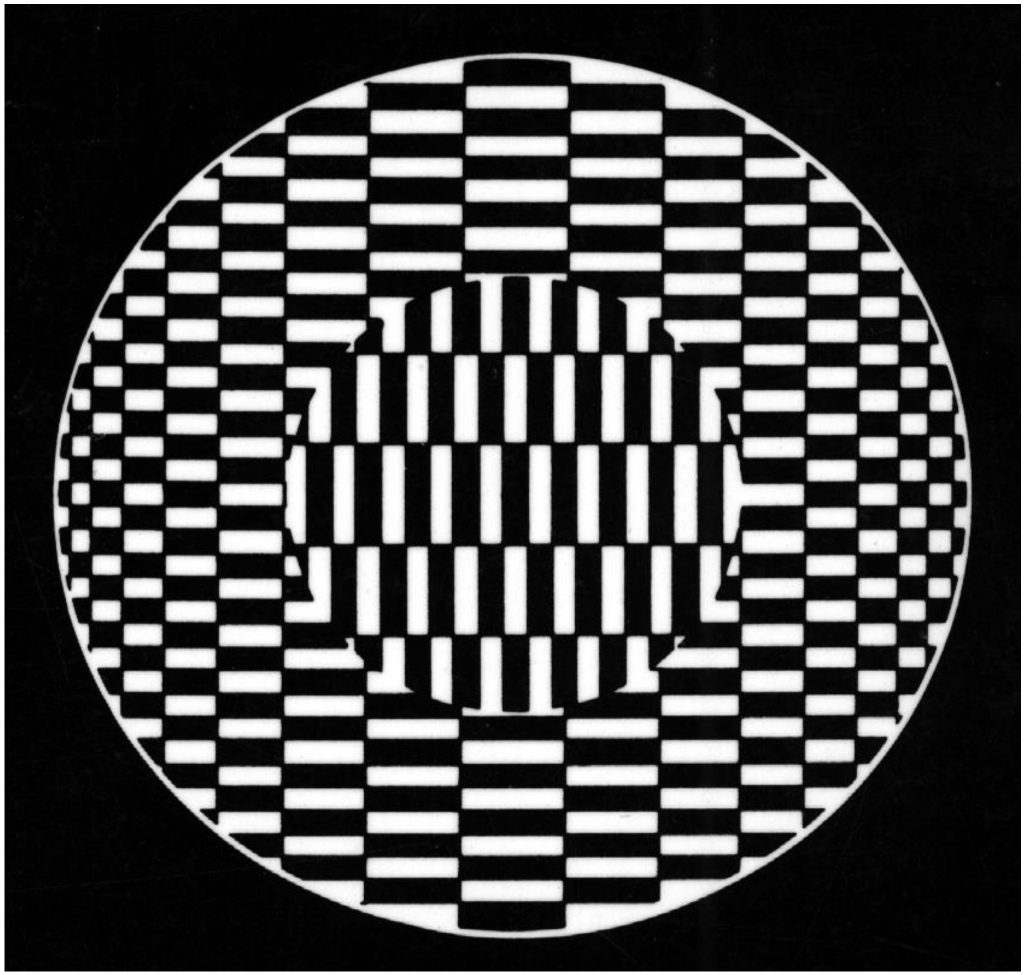
Contour tracking



Textured Motion

- http://civs.stat.ucla.edu/Yizhou_Research/Texturedmotion.htm
- Natural scenes contain rich stochastic motion patterns which are characterized by the movement of a large amount of particle and wave elements, such as falling snow, water waves, dancing grass, etc. We call these motion patterns "*textured motion*".

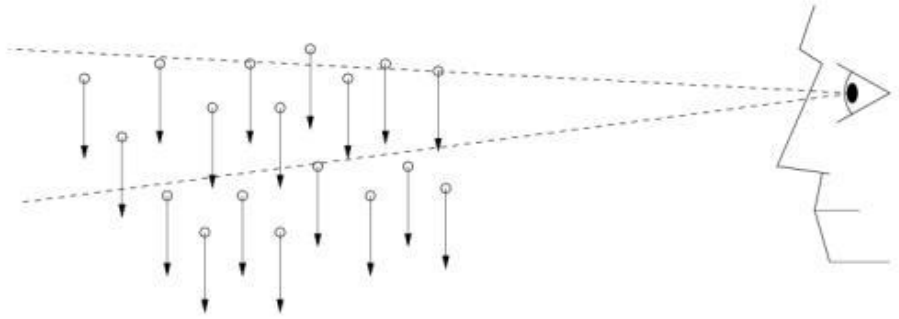
Optical Flow Break Down



Optical Flow:
Where do pixels move to?

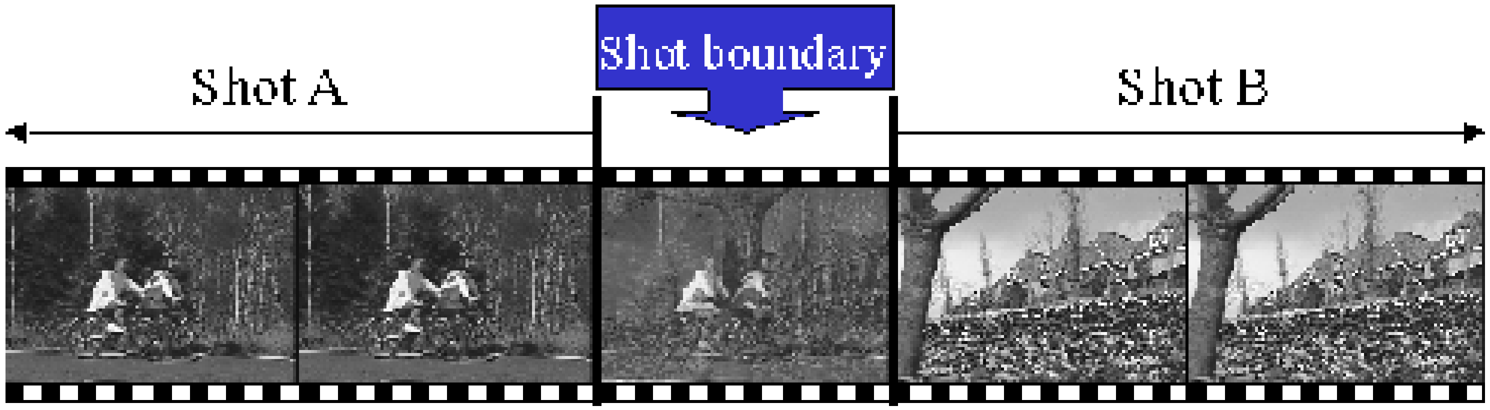
Optical Snow

- http://www.cim.mcgill.ca/~langer/research-optical-snow.html
- Optical snow is the type of motion an observer sees when watching a snow fall. Flakes that are closer to the observer appear to move faster than flakes which are farther away.



Applications of segmentation to video

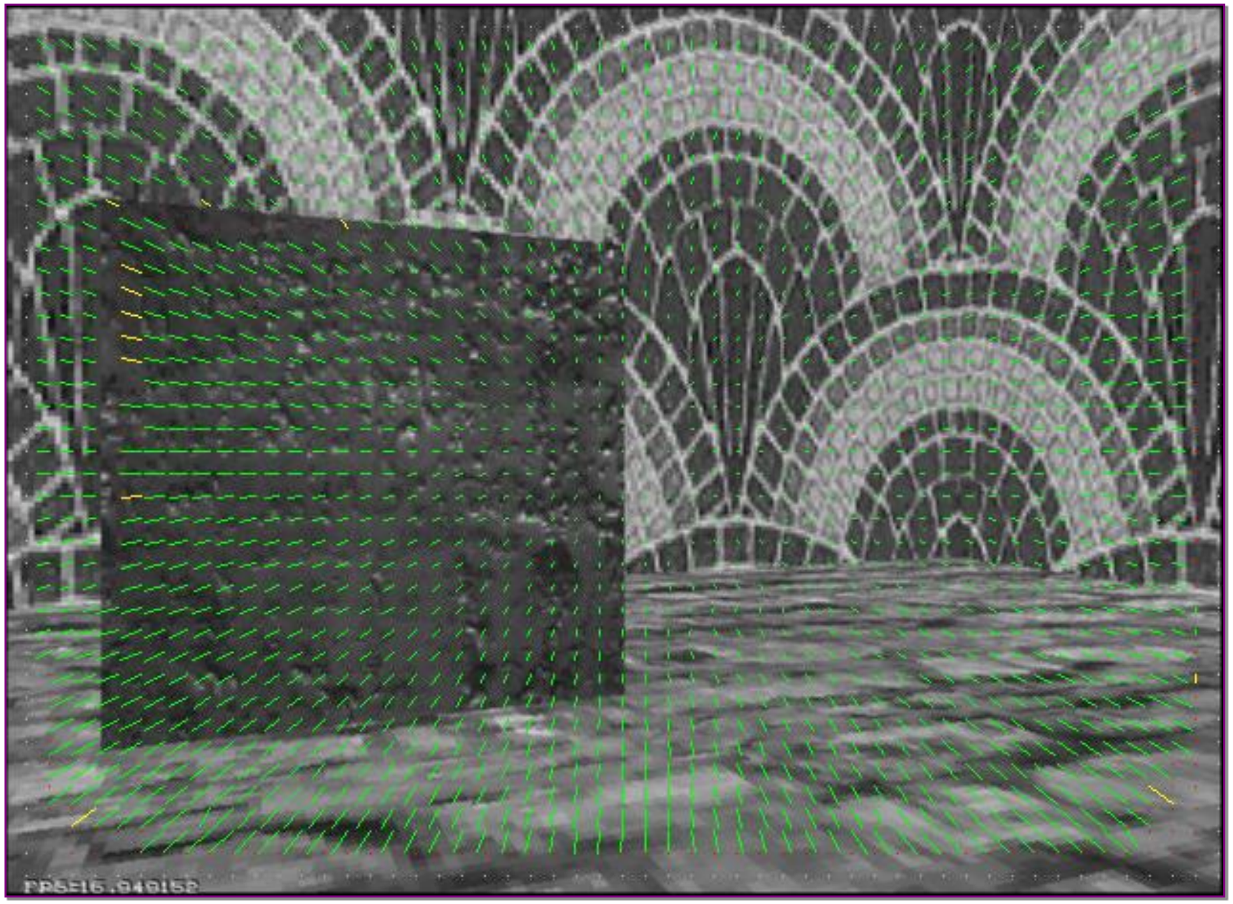
- Background subtraction
- Shot boundary detection
  - Commercial video is usually composed of *shots* or sequences showing the same objects or scene
  - Goal: segment video into shots for summarization and browsing (each shot can be represented by a single keyframe in a user interface)
  - Difference from background subtraction: the camera is not necessarily stationary



Applications of segmentation to video

- Background subtraction
- Shot boundary detection
  - For each frame
    - Compute the distance between the current frame and the previous one
      - Pixel-by-pixel differences
      - Differences of color histograms
      - Block comparison
    - If the distance is greater than some threshold, classify the frame as a shot boundary

Optical Flow Applications



Obstacle Detection: Unbalanced Optical Flow

Temizer