

Spatial track: motion modeling

Virginio Cantoni

Computer Vision and Multimedia Lab

Università di Pavia

Via A. Ferrata 1, 27100 Pavia

virginio.cantoni@unipv.it

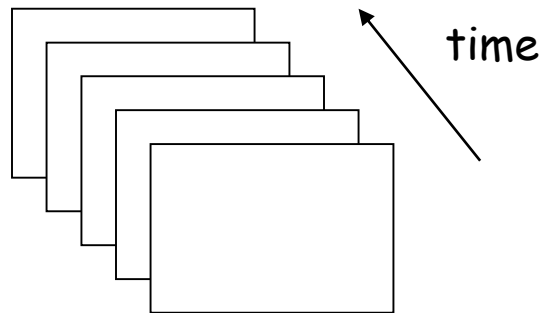
<http://vision.unipv.it/va>

Comparison between Motion and Stereo Analysis

- Stereo: two or more frames



- Motion: N frames



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

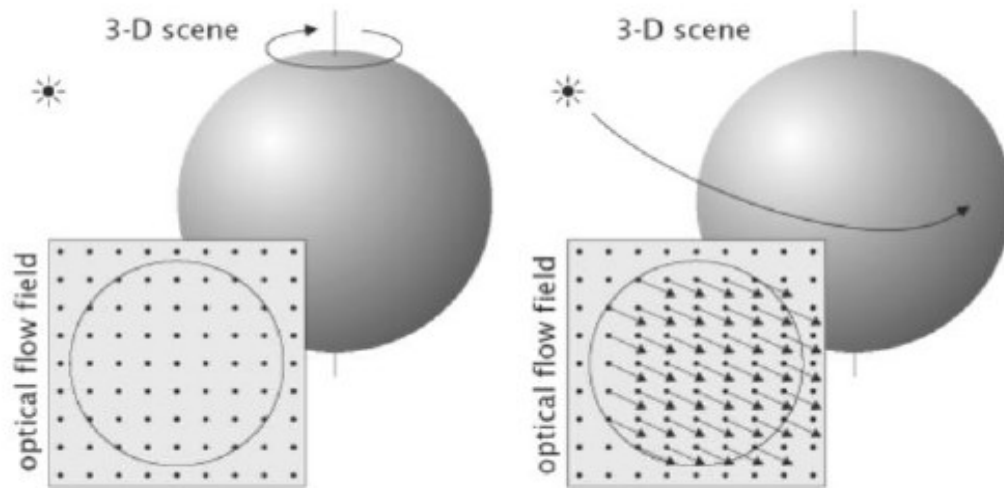
- Motion disparities tend to be smaller

- Stereo images are taken at the same time:

- Motion disparities can be due both to scene and camera motion
- There can be more than one transformation between frames

Ill-posed problem

- As an illustrative example, positively using this limitation to attract attention, consider the barber-shop banner usually displayed outdoors in many countries.
- Typically, a rotation movement of a 3-coloured striped pattern on a cylinder, perceptually suggests that the whole pattern is translated vertically upwards.



- A rotational movement of a homogeneous sphere cannot be perceived, meanwhile a still sphere is perceived rotating if the source of light is rotating around.

Motion versus stereo analysis: correspondences

- **Small displacements**
 - differential algorithms
 - based on gradients in space and time
 - dense correspondence estimates
 - most common with video

- **Large displacements**
 - matching algorithms
 - based on correlation or features
 - sparse correspondence estimates
 - most common with multiple cameras/stereo
 - Correspondence types
 - point correspondences
 - line correspondences
 - curve correspondences
 - region correspondences

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)**

Classes of Techniques and Motion Models

- Motion models exploit two different **invariants**:
 - in the first one, **peculiar object** points are assumed to be recovered from one image to the next
 - in the second all **visible-point intensities are supposed to be maintained** along time.
- Two different approaches of motion are distinguished, respectively named **discrete or sparse** and **continuous or dense**:
 - **motion via correspondence**
 - **motion via local change**
- In the first case corresponding points must be found on different successive images. Note that, in stereo vision, to evaluate the position in space, point correspondences must be similarly found from different images taken simultaneously.
- In the second case, a simple local analysis on the whole image must be performed with the limitation that **only one motion component may be detected, the one orthogonal to the image contour**.

Global Flow

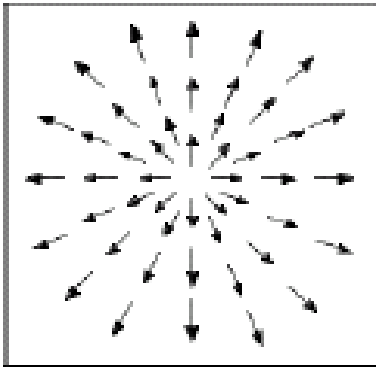
- Dominant motion in the image
 - motion of all points in the scene
 - motion of most of the points in the scene
 - component of motion of all points in the scene
- Global motion is caused by
 - motion of sensor (Egomotion)
 - motion of a rigid scene
- Estimation of global motion can be used to
 - image alignment (Registration)
 - removing camera jitter
 - tracking (by neglecting/eliminating camera motion)
 - video segmentation etc.

Motion Detection and Estimation in Literature

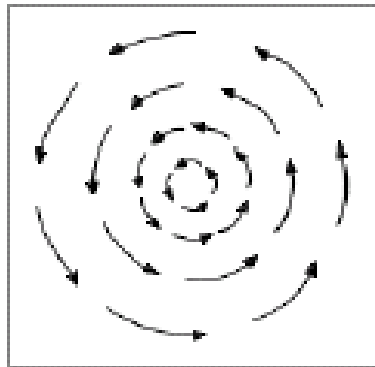
- **Image differencing**
 - based on the thresholded difference of successive images
 - difficult to reconstruct moving areas
- **Background subtraction**
 - foreground objects result by calculating the difference between an image in the sequence and the background image (previously obtained)
 - remaining 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
- Pointwise motion estimation: **OPTICAL FLOW**

Motion Field (MF)

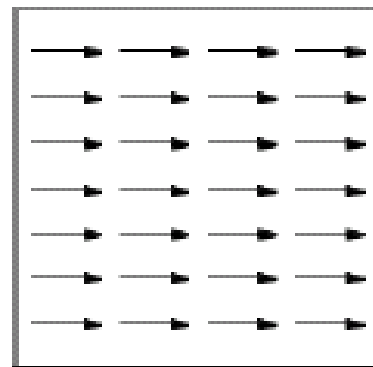
- The **MF** assigns a velocity vector to each pixel in the image.
- These velocities are INDUCED by the RELATIVE MOTION btw the camera and the 3D scene
- The **MF** can be thought as the **projection** of the 3D velocities on the image plane.
- Examples of **MF**:



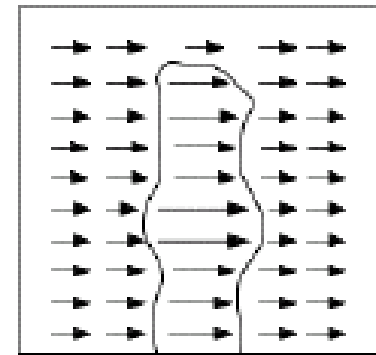
Forward motion



Rotation

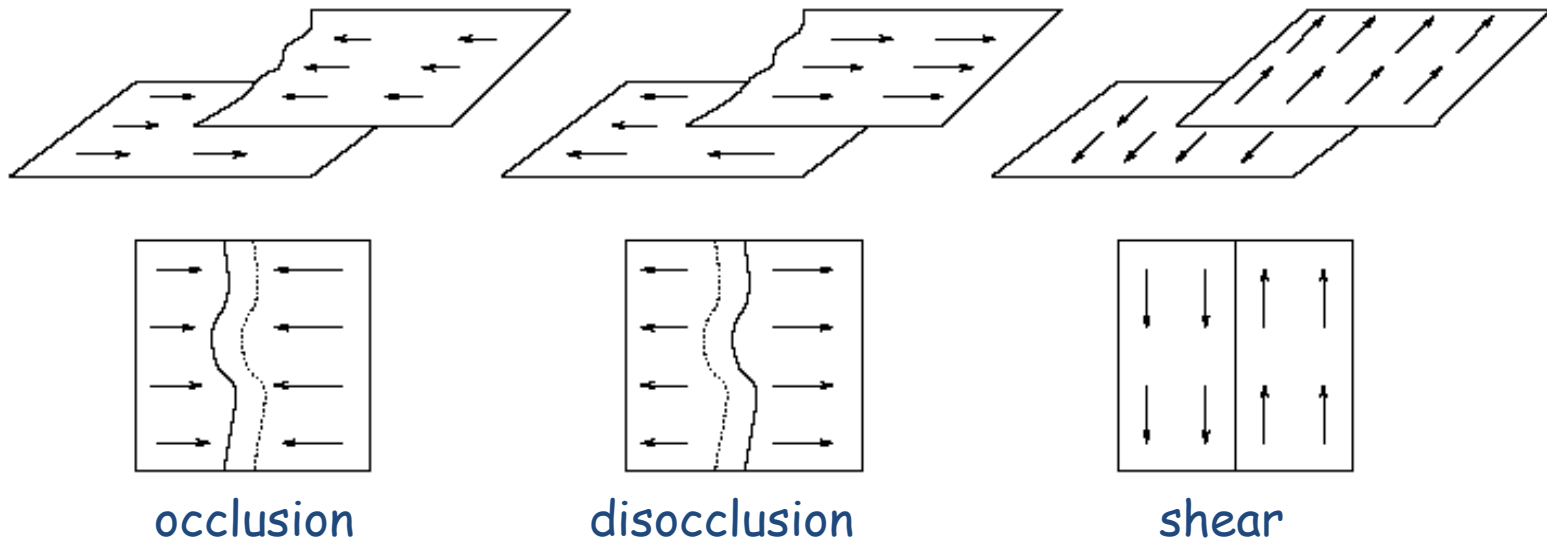


Horizontal translation



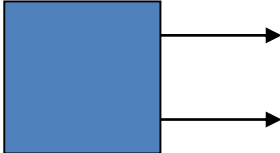
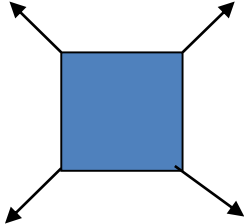
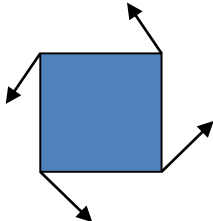
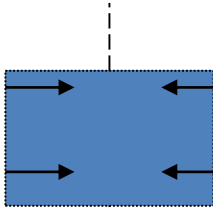
Closer objects
appear to move
faster!!

Occlusion



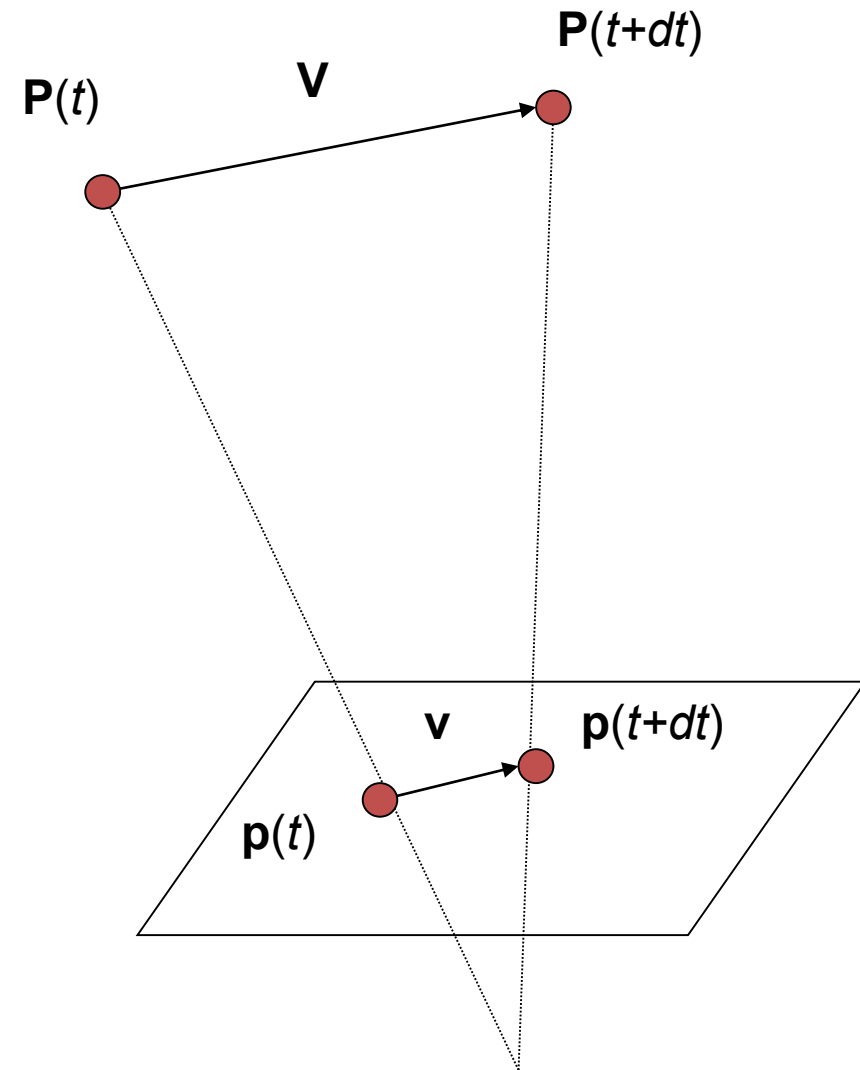
Multiple motions within a finite region.

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.	

Motion field and parallax

- $P(t)$ is a moving 3D point
- Velocity of scene point: $\mathbf{V} = d\mathbf{P}/dt$
- $\mathbf{p}(t) = (x(t), y(t))$ is the projection of P in the image
- Apparent velocity \mathbf{v} in the image: given by components $v_x = dx/dt$ and $v_y = dy/dt$
- These components are known as the **MF** of the image



Motion field

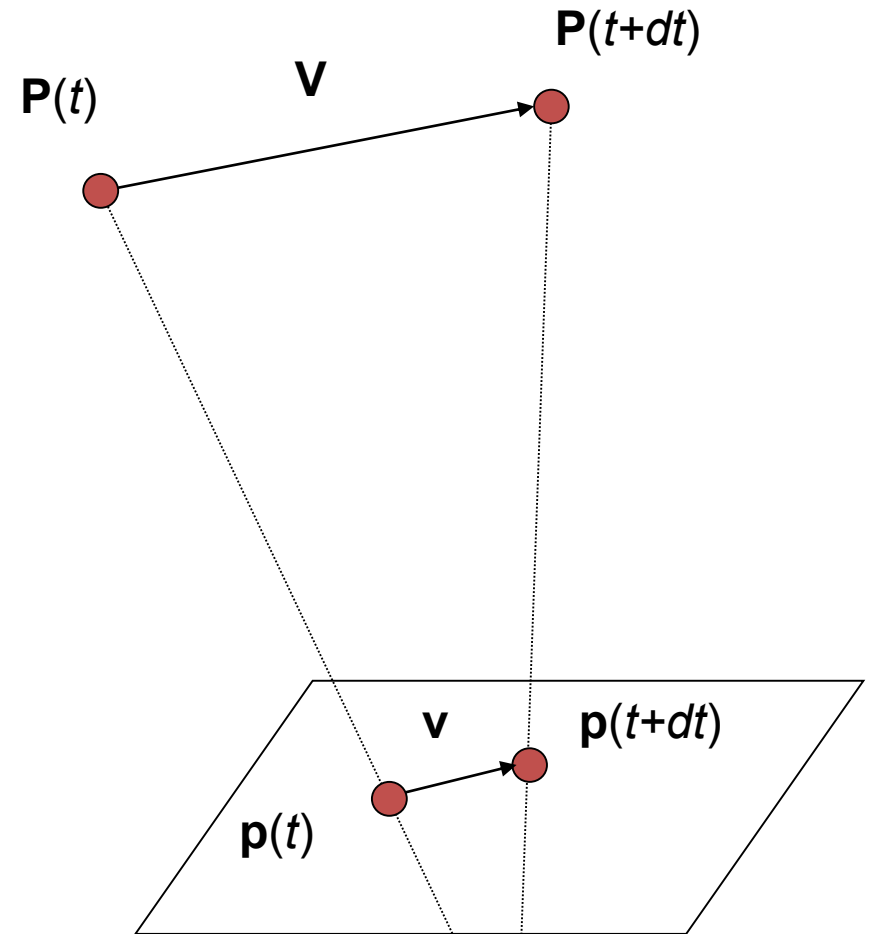
$$\mathbf{p} = f \frac{\mathbf{P}}{Z}$$

To find image velocity \mathbf{v} , differentiate \mathbf{p} with respect to t (using quotient rule):

$$\mathbf{V} = (V_x, V_y, V_z)$$

$$\mathbf{v} = f \frac{Z\mathbf{V} - V_z\mathbf{P}}{Z^2}$$

Image motion is a function of both the 3D motion (\mathbf{V}) and the depth of the 3D point (Z)



Quotient rule:
 $D(h/g) = (g h' - g' h)/g^2$

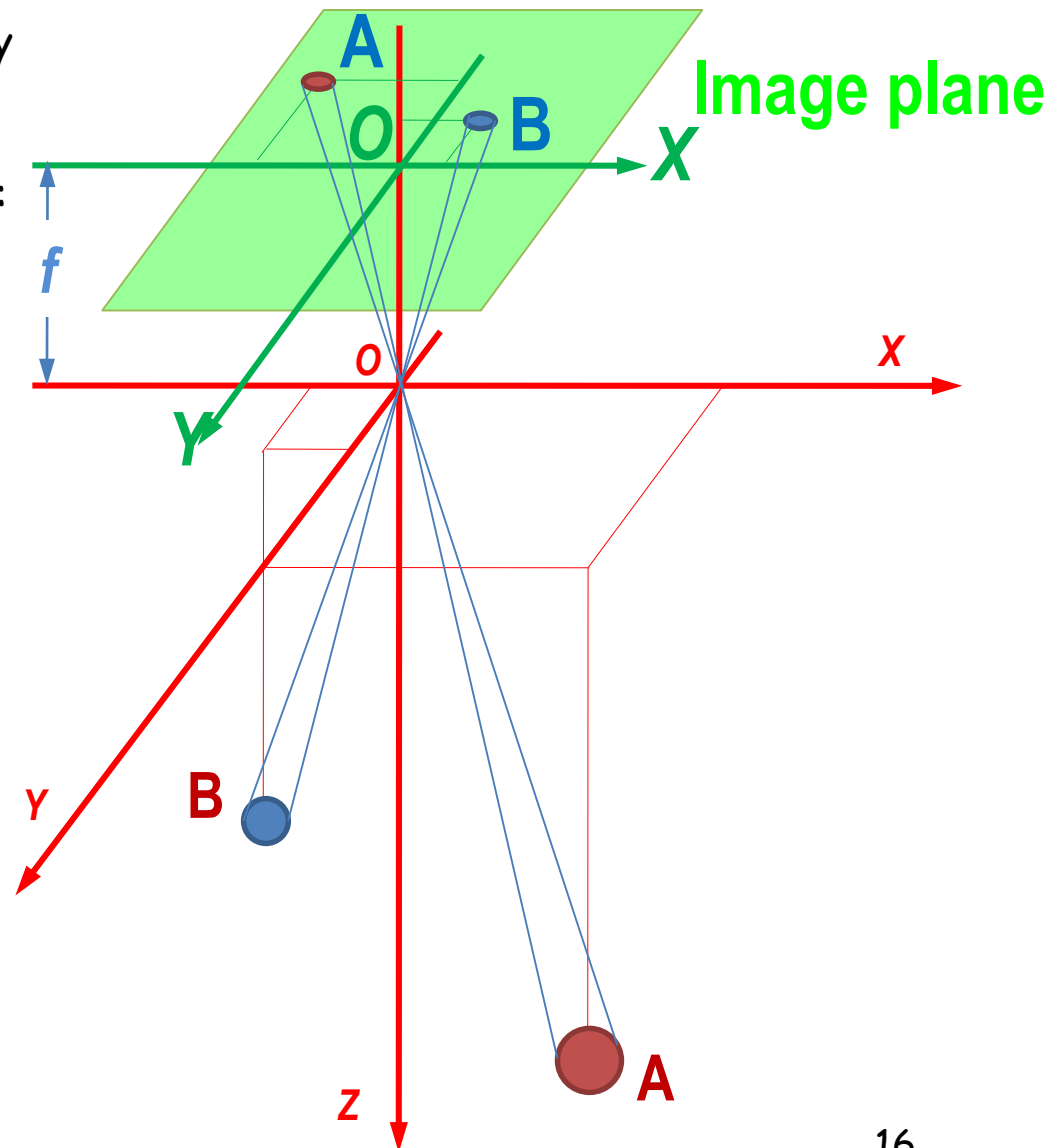
Preliminaries for motion analysis

- 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.



Motion field and parallax

- Pure translation: \mathbf{V} is constant everywhere
- Every motion vector points toward (or away from) \mathbf{v}_0 , the vanishing point of the translation direction
- The length of the motion vectors is inversely proportional to the depth Z

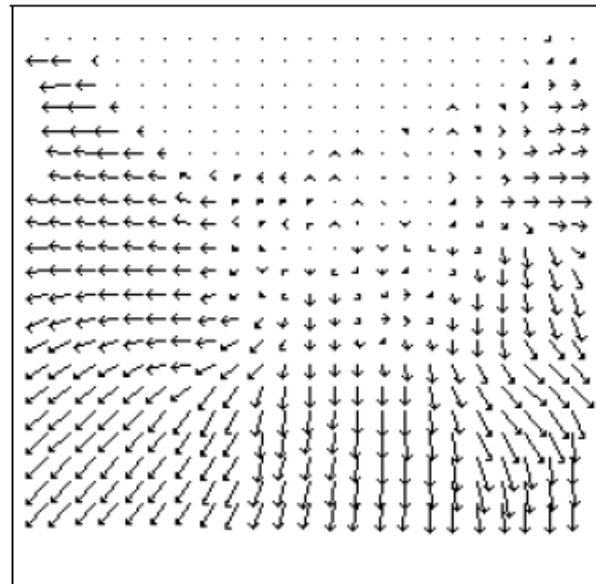


Motion modeling: egomotion

- 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, egomotion is evaluated and compensated next the camera is assumed to remain still.**



Motion field due to camera motion



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

Figure from Michael Black, Ph.D. Thesis

Egomotion

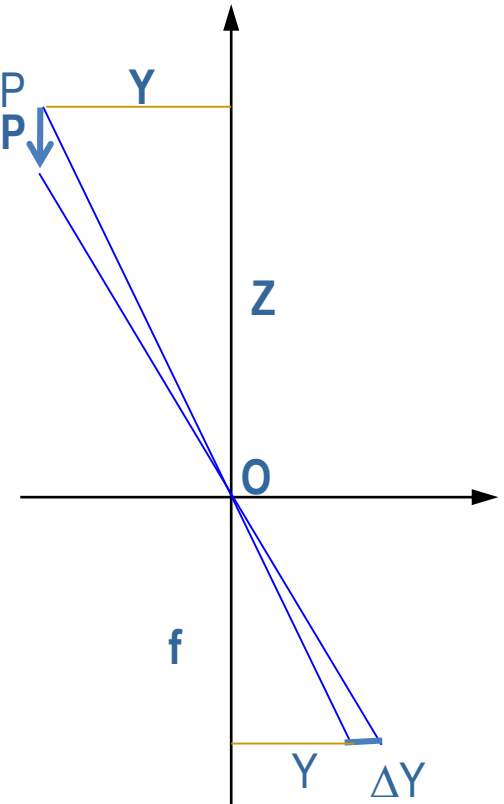
- the **OPTICAL FLOW** is instrumental at evaluating the shape and position of still components from their apparent motion due to the camera movement (**egomotion**). The sketch shows a camera downwards shift along the **Z** axis.
- Thanks to the relativity of perception it is equivalent to assume that 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 ΔY . Considering the triangle similarity:

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

so that the distance **Z** may be derived considering that dZ is the known motion of the camera and (Y, dY) is determined from the image:

$$Z = -\frac{dZ}{dY} Y$$

and consequently the **collision time**.



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 **X_o**, **Y_o**, **Z_o** will be displaced on the image as:

$$X = -f \frac{X_o + ut}{Z_o + wt} ; Y = -f \frac{Y_o + vt}{Z_o + wt}$$

- In order to compute the coordinates of the final/original destination of the moving point we may evaluate these for $t = \pm\infty$ 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**.



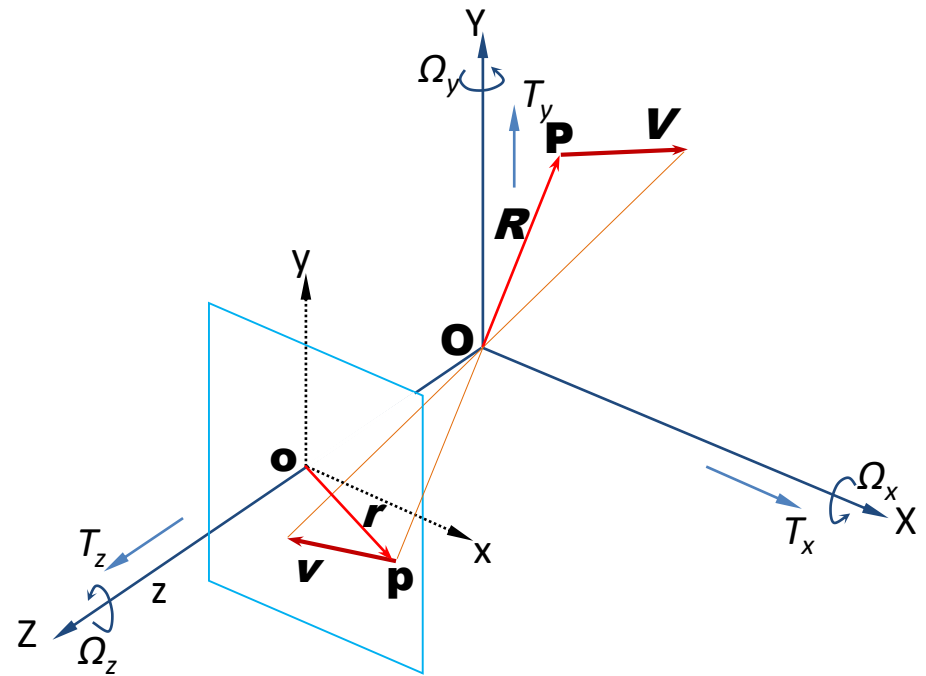
Camera and egomotion

- The egomotion makes all still objects in the scene to verify the same motion model defined by three translations \mathbf{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:

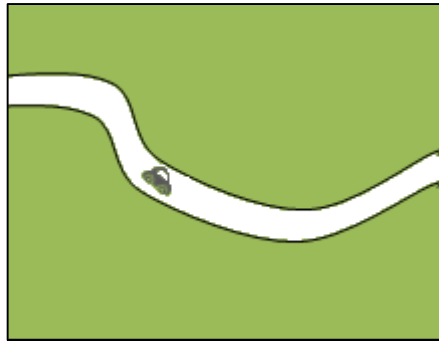
$$u_t = \frac{-fT_X + xT_Z}{Z}, \quad u_r = \frac{-xy}{f}\Omega_X - \left(\frac{-x^2}{f} + 1\right)\Omega_Y + y\Omega_Z$$

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

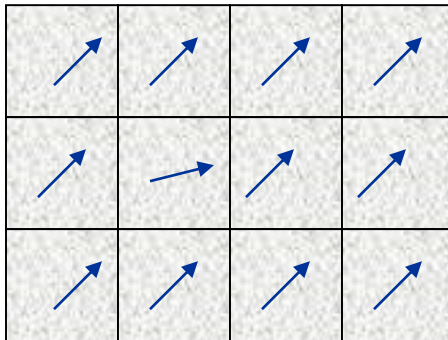
- where $\mathbf{w} = [u, v]^T = [u_t + u_r, v_t + v_r]^T$ stands for the 2-D velocity vector of the pixel under the focal length f .



From egomotion toward object speed

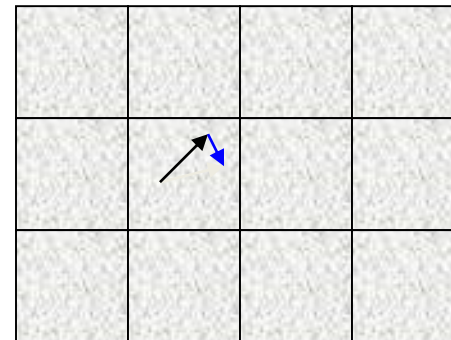


Detected
speed



Average
speed

\nearrow
 $[\bar{u}, \bar{v}]$



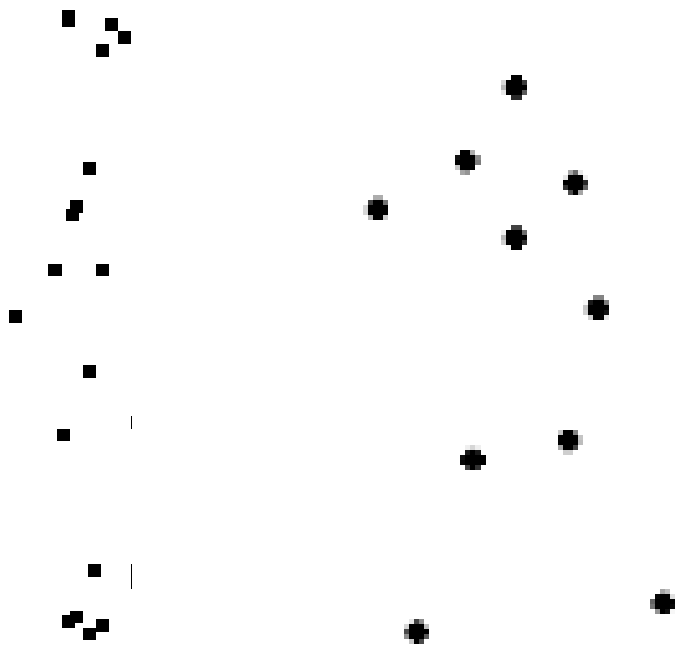
Deviation due to
the weighted
object speed



Motion via correspondences

Even "impoverished" motion data can evoke a strong percept

30 point



Normally, peculiar points on the first image are located so as to search their corresponding points on the second image.

In the triangulation for stereovision, there is no guarantee that such corresponding points exist and the new point of view may not include the 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.

Early to the epipolar segment for stereovision, the corresponding points are searched in a restricted area determined by a heuristic.

Local sketch: locate the position of a pixel in the current image having similarity and **the test Euclidean distance** with respect to a point in the previous frame.

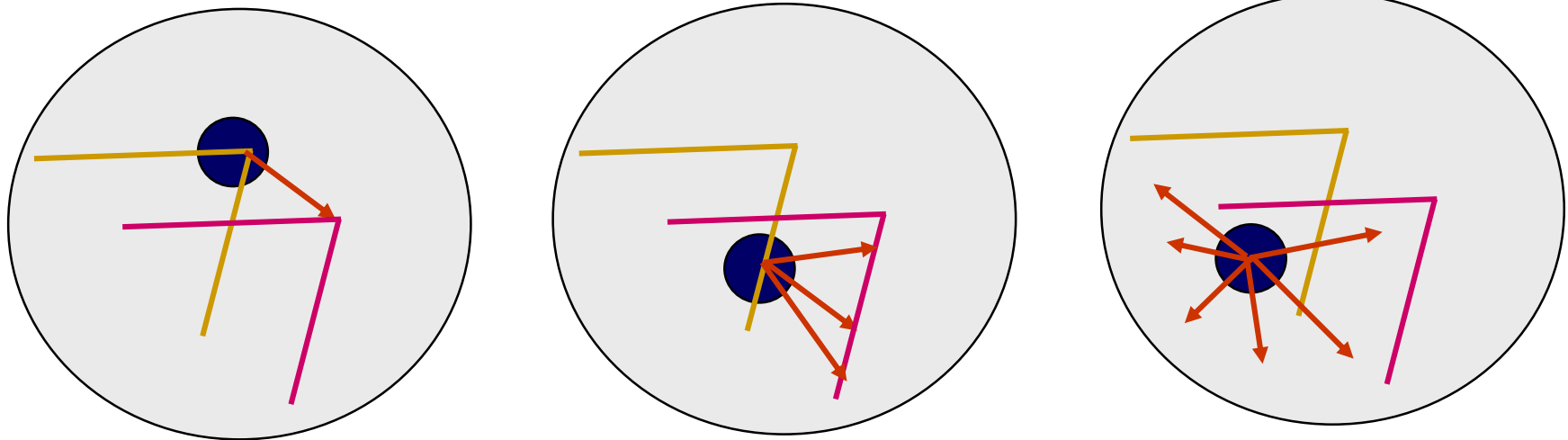
Patch Matching

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



Local Patch Analysis

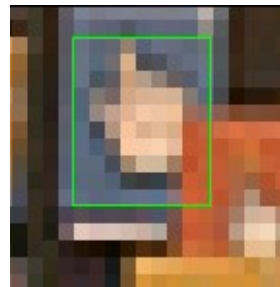
- How *certain* are the motion estimates?



(revisited)

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

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

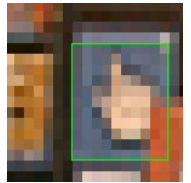
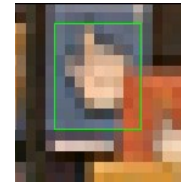


Patch matching: correlation window size

- How do we determine correspondences?

- *block matching* or *SSD* (sum squared differences)

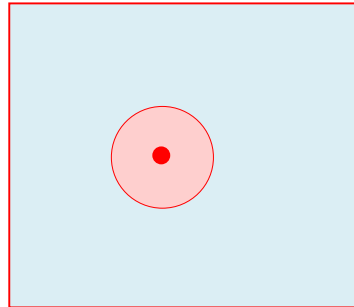
$$E(x, y; d) = \sum_{(x', y') \in N(x, y)} [I_L(x' + d, y') - I_R(x', y')]^2$$



- 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
- Large windows good for global search: less detailed and smoother

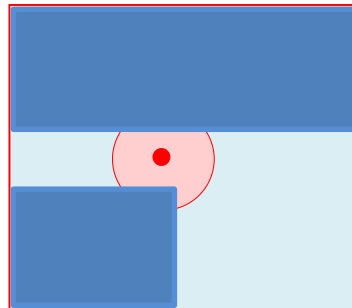
Maximum velocity

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



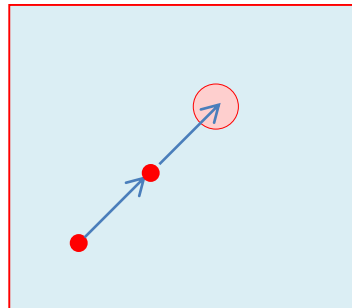
Obstacles

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



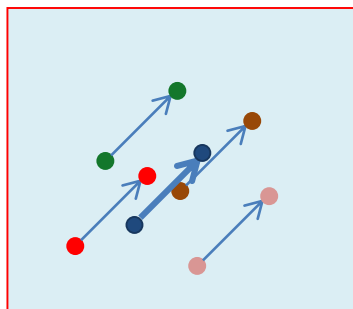
Maximum acceleration

- 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 $\frac{1}{2} A_{\max} Dt^2$ where A_{\max} is the maximum acceleration;



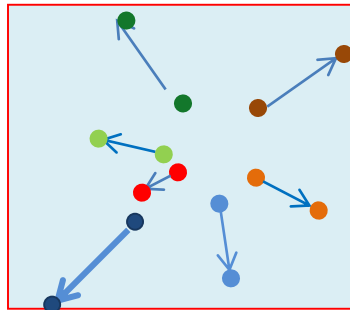
Consistent matching

- 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



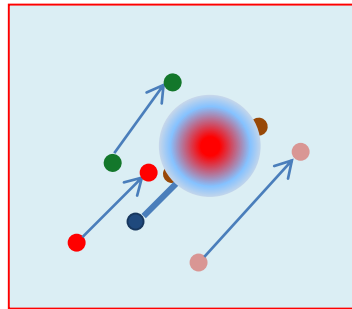
Common motion

- *Common motion (Global Flow)* 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)

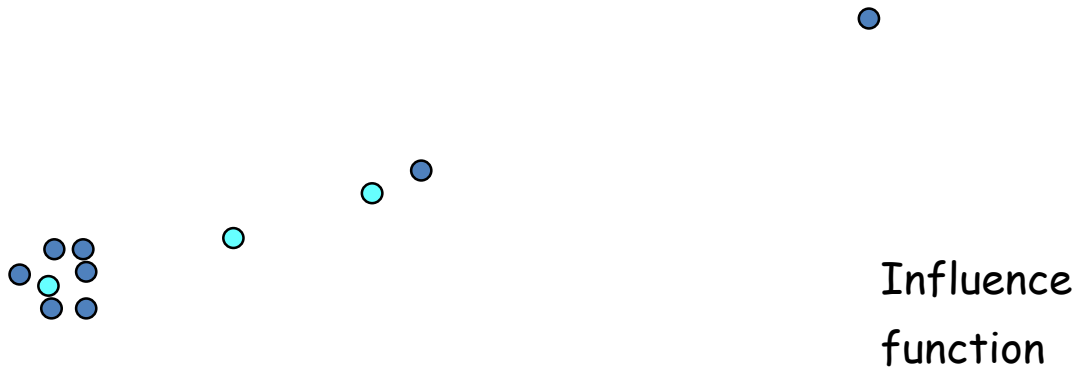


Flexible motion model

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



Robust estimation: outliers



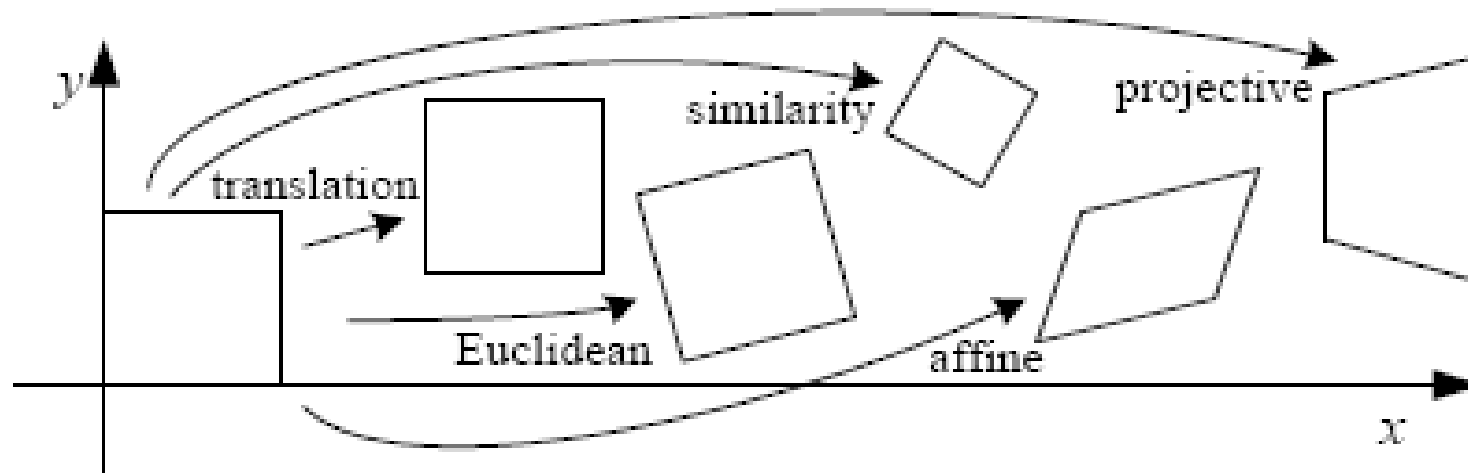
$$E(m) = \sum_i \rho(x_i)$$

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

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

- ✓ Least-squares estimators penalize deviations between data & model with quadratic error (extremely sensitive to outliers)
- ✓ Redescending error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

Motion global models

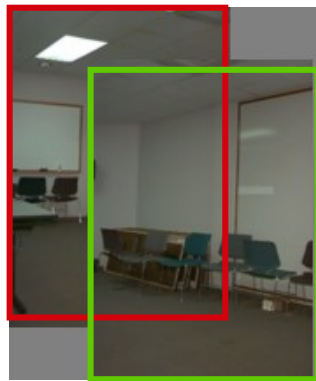


Translation

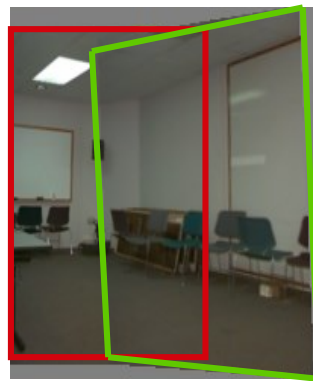
3D rotation

Affine

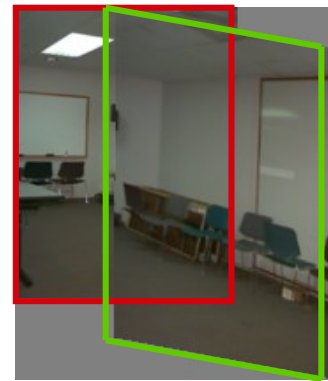
Perspective



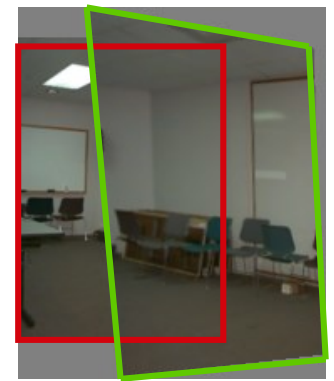
2 unknowns



3 unknowns



6 unknowns

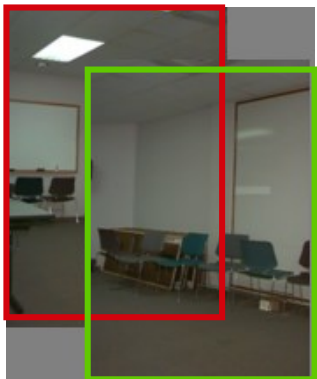


8 unknowns

Szeliski

Translation and Affine

Translation

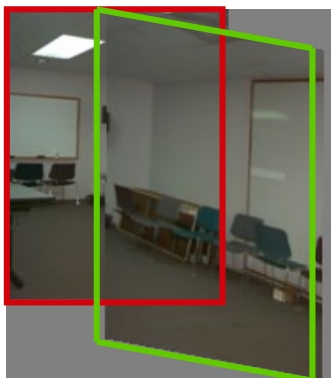


2 unknowns

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

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

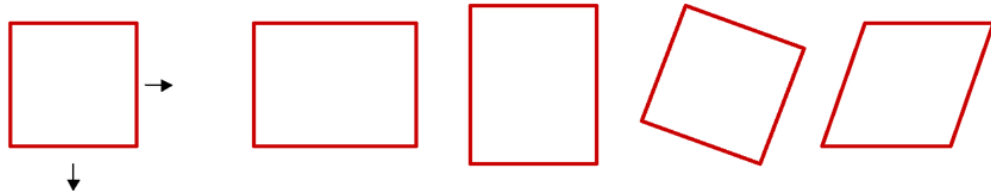
Affine



6 unknowns

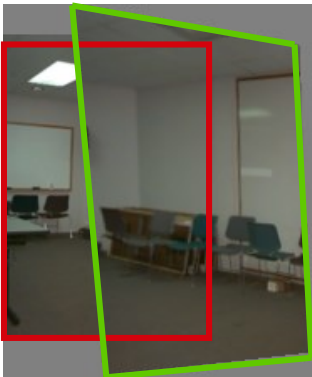
$$E(\mathbf{A}, \mathbf{h}) = \sum_{\mathbf{x} \in \mathbb{R}} [I(\mathbf{Ax} + \mathbf{h}) - I_0(\mathbf{x})]^2$$

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



Planar perspective

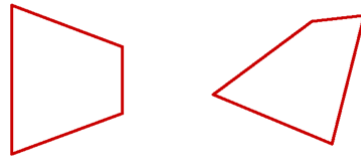
Perspective



8 unknowns

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

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

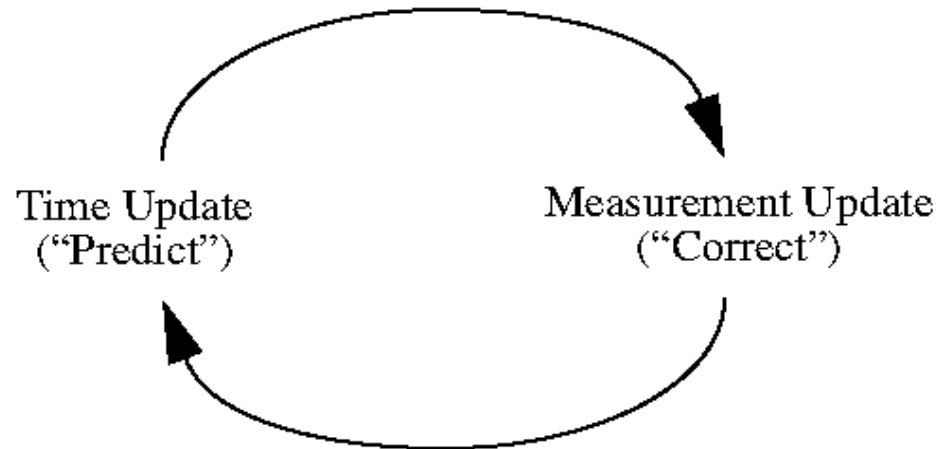


$$E(\mathbf{f}, \mathbf{h}) = \sum_{\mathbf{x} \in R} [I(\mathbf{f}(\mathbf{x}, \mathbf{h})) - I_0(\mathbf{x})]^2$$

Other parametrized transformations

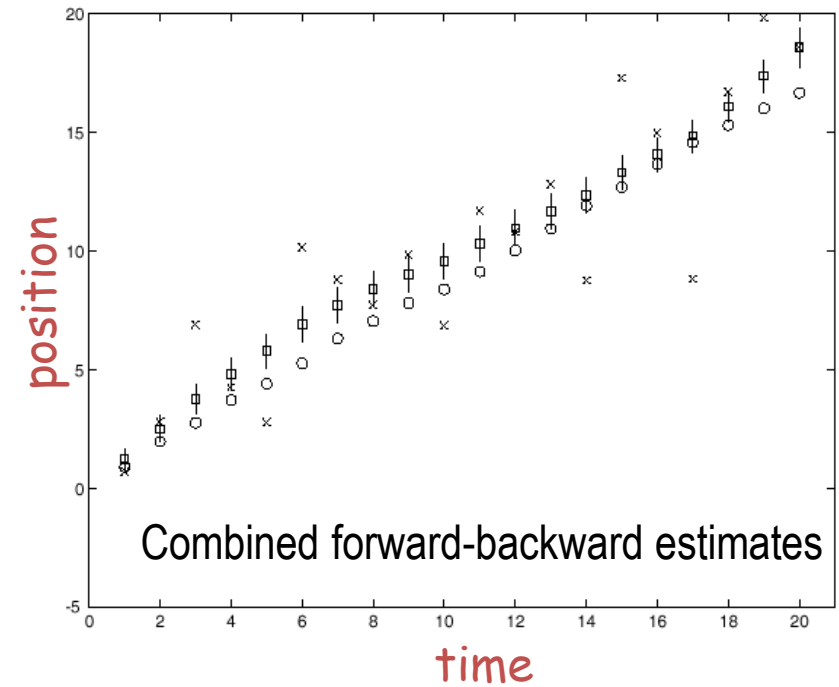
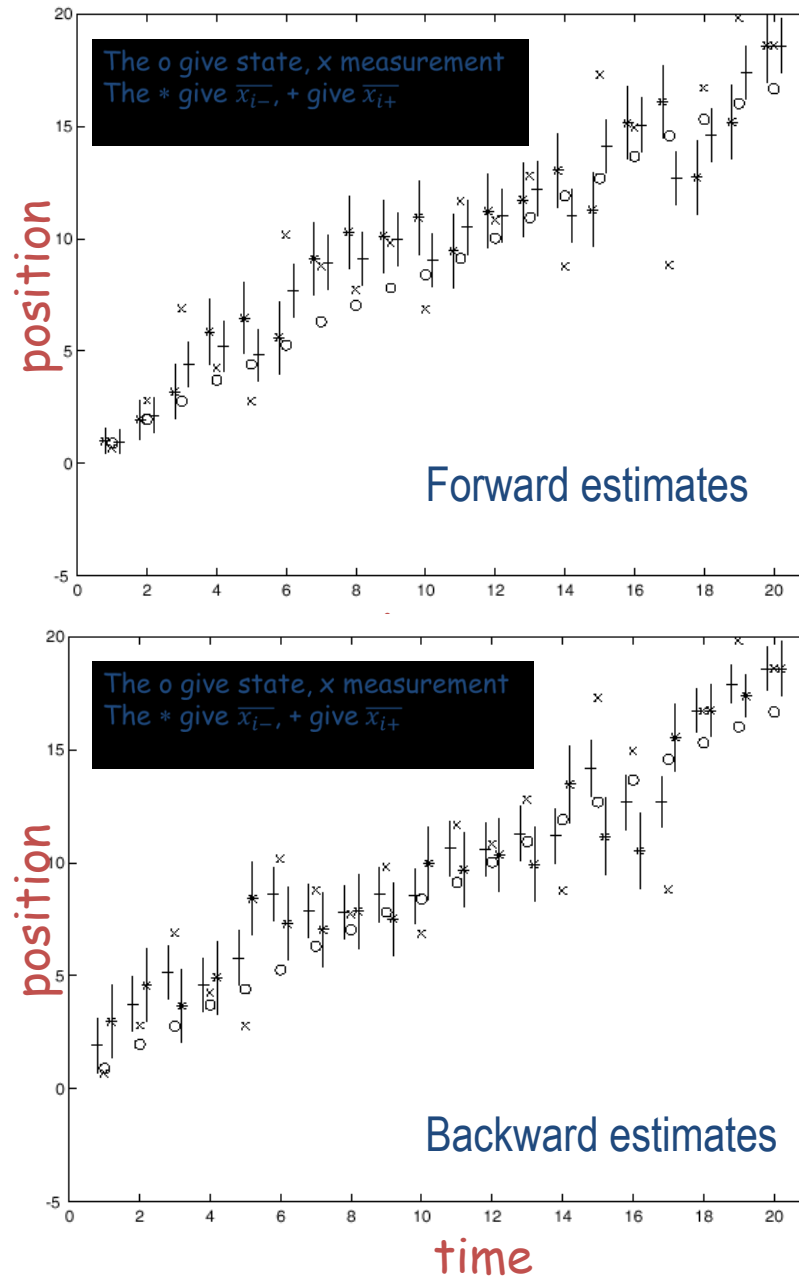
Tracking as induction

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

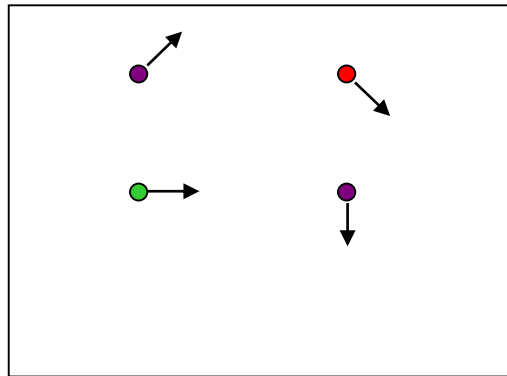


- **Run two filters, one moving forward, the other backward in time.**
- Now combine state estimates
- **It is possible to iterate:** we can obtain a smoothed estimate by viewing the backward filter's prediction as yet another measurement for the forward filter

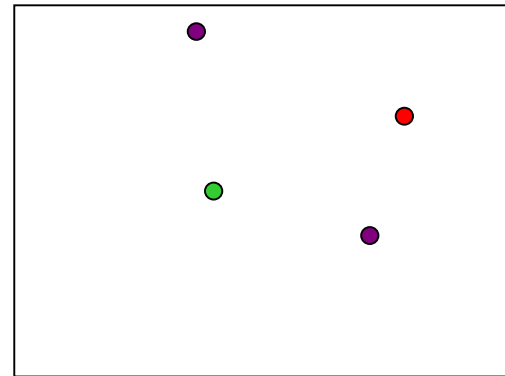
Tracking



Problem definition: optical flow



$H(x, y)$

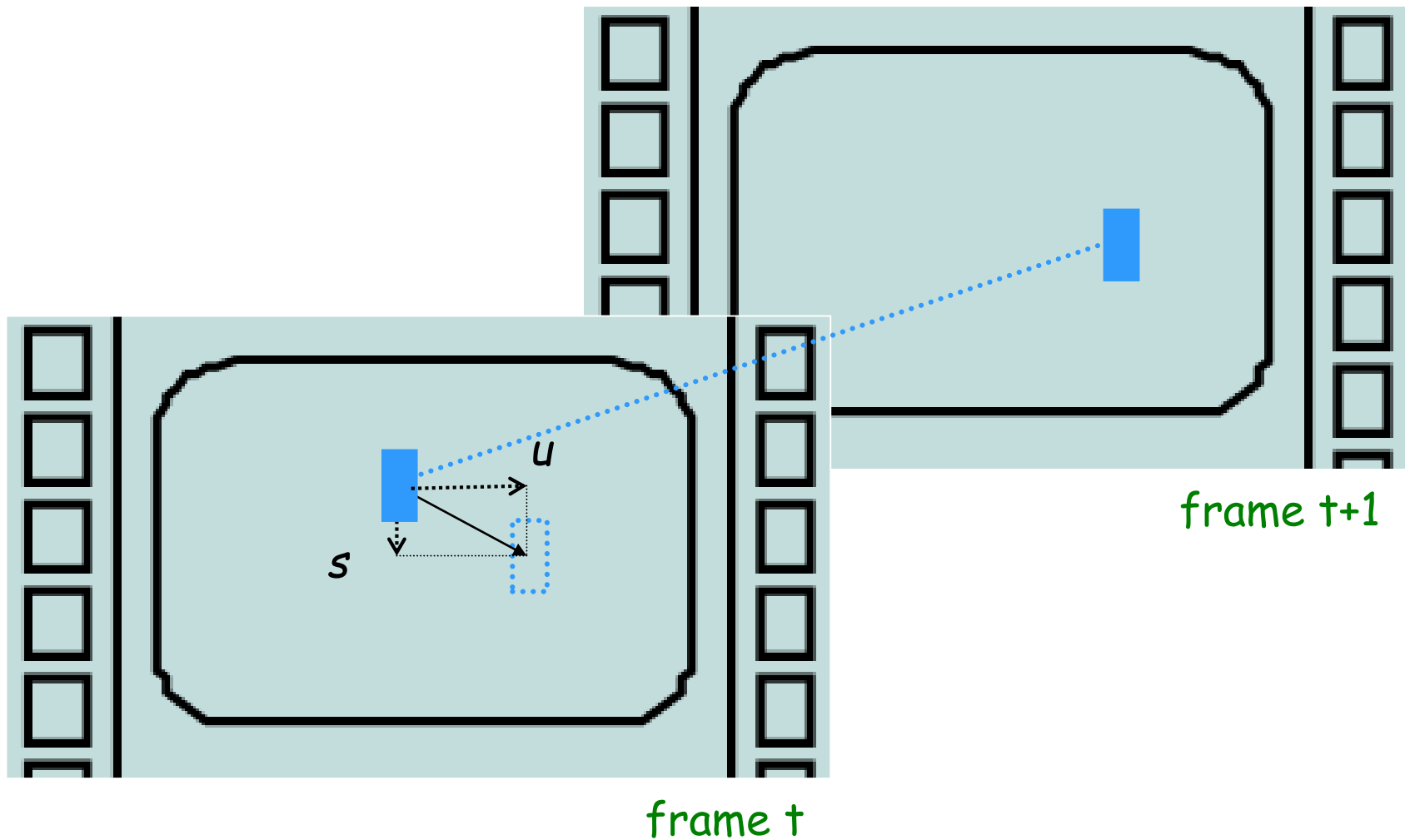


$I(x, y)$

- How to estimate pixel motion from image H to image I ?
 - Solve pixel correspondence problem
 - given a pixel in H , look for **nearby** pixels of the **same color** in I
- Key assumptions
 - **color constancy**: a point in H looks the same in 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

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



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 $\partial x, \partial y$ represents the pixel displacement between images and ∂t is the time interval. By differentiating, one can write:

$$f_x \frac{\partial x}{\partial t} + f_y \frac{\partial y}{\partial t} = -f_t \quad \text{or using } \cdot \text{ for the scalar product and } \nabla \text{ for the gradient} \quad \nabla f \cdot \bar{V} = f_t$$

- 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.

Optical flow: mathematical formulation

Brightness constancy assumption:

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

Optical flow constraint equation :

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

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

$$I_x u + I_y v + I_t = 0$$

$$\nabla I \cdot [u \ v]^T = -I_t$$

1 equation in 2 unknowns

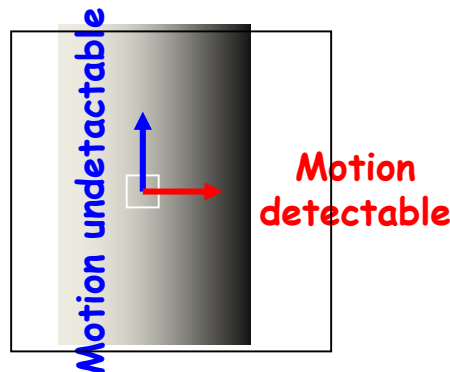
The brightness constancy constraint

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

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

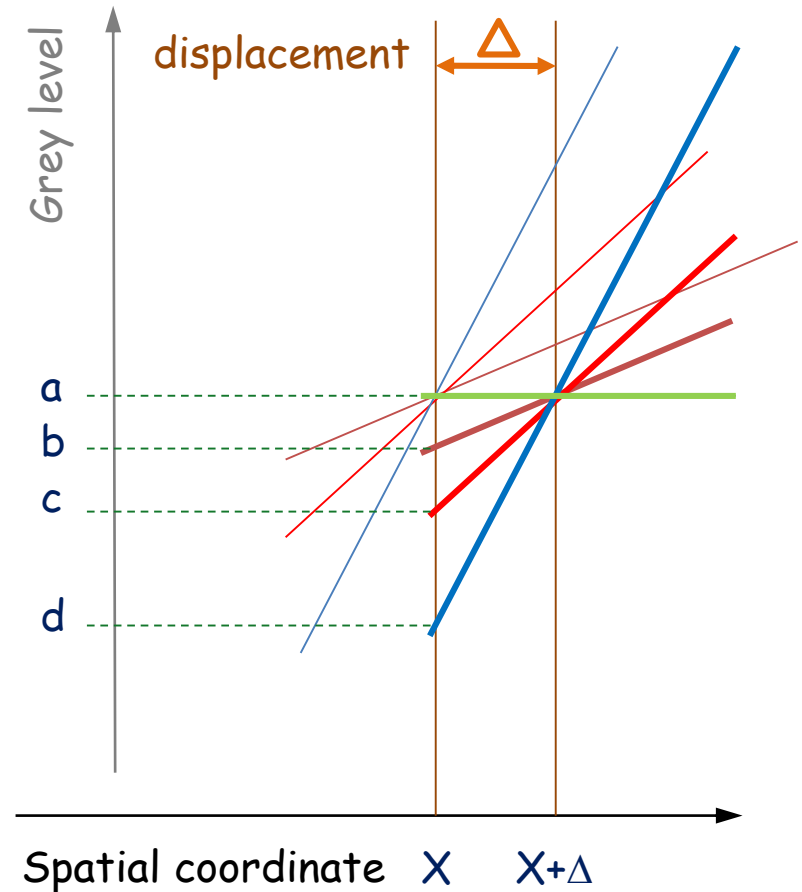
- One equation (this is a scalar equation!), two unknown $\vec{v} = [u, v]$
- The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured
- If $[u,v]$ satisfies the equation, so does $[u+u', v+v']$ if

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

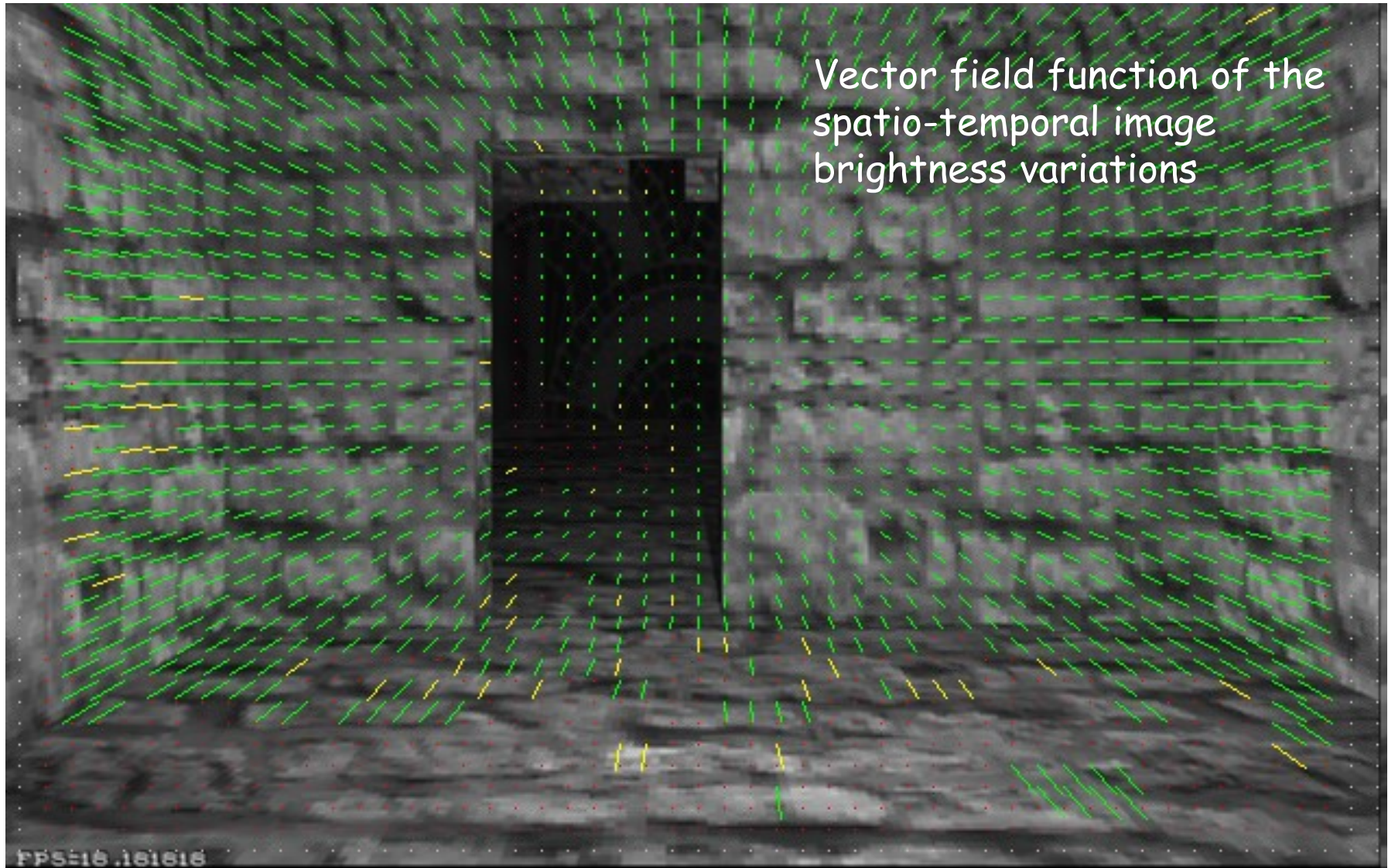


Optical flow

- 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.
- 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.

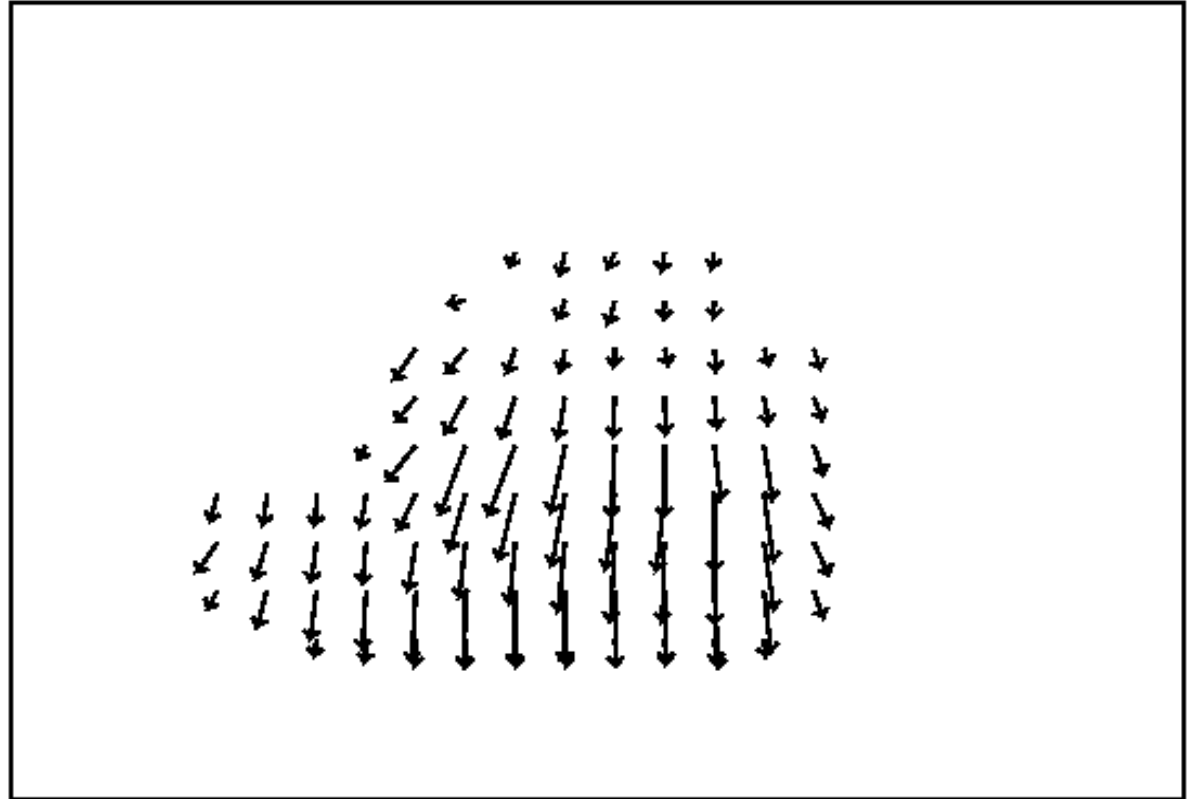


Optical flow



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

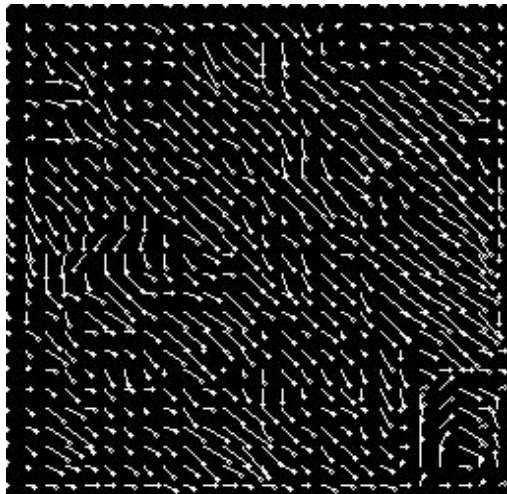
Optical Flow



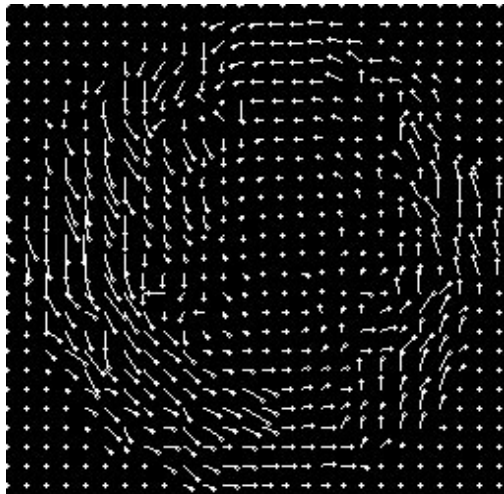
Pierre Kornprobst's Demo

Optical Flow Examples

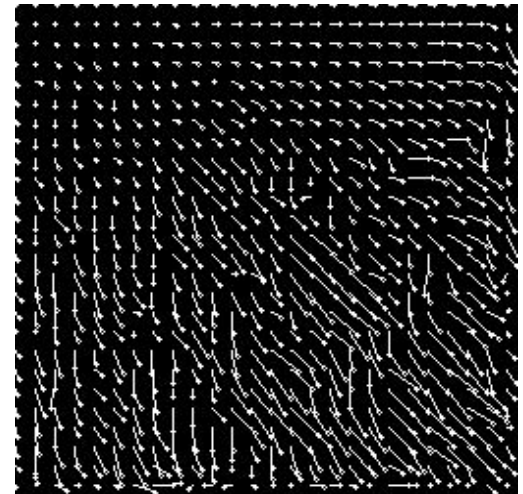
Translation



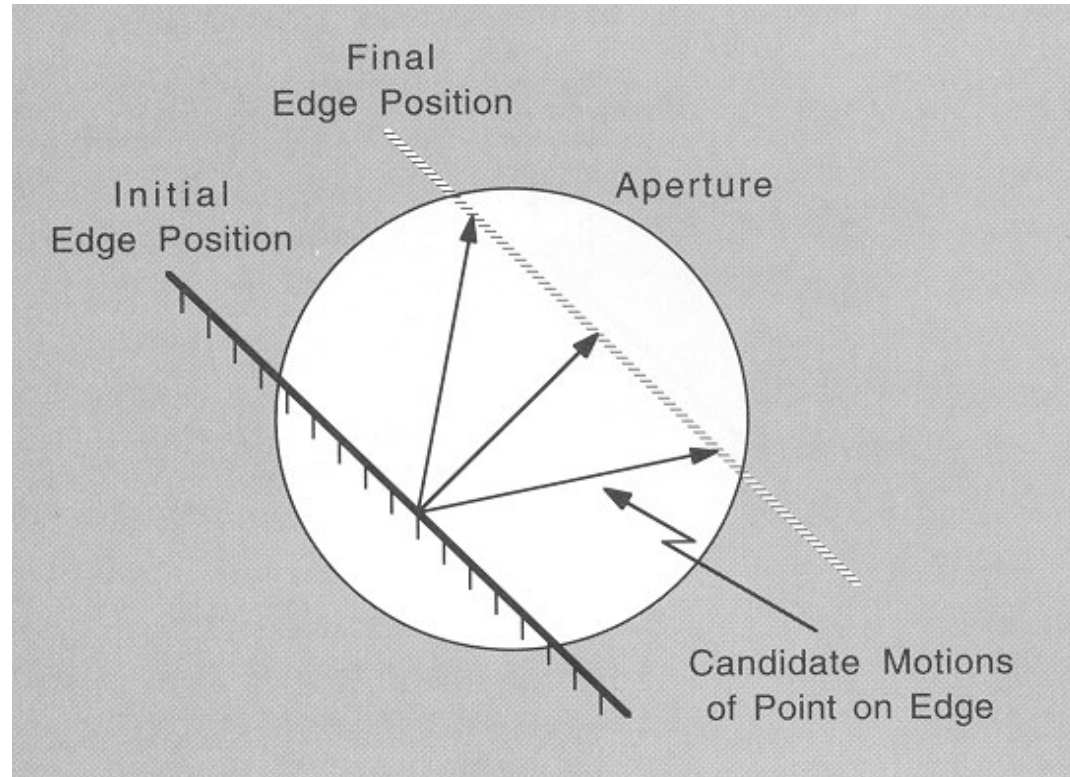
Rotation



Traslation + Scaling

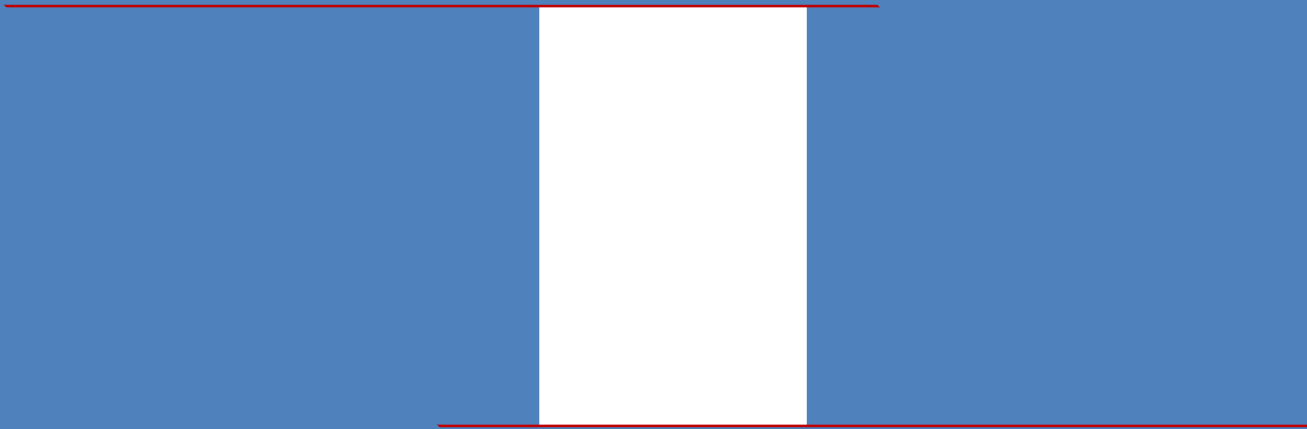


Aperture Problem



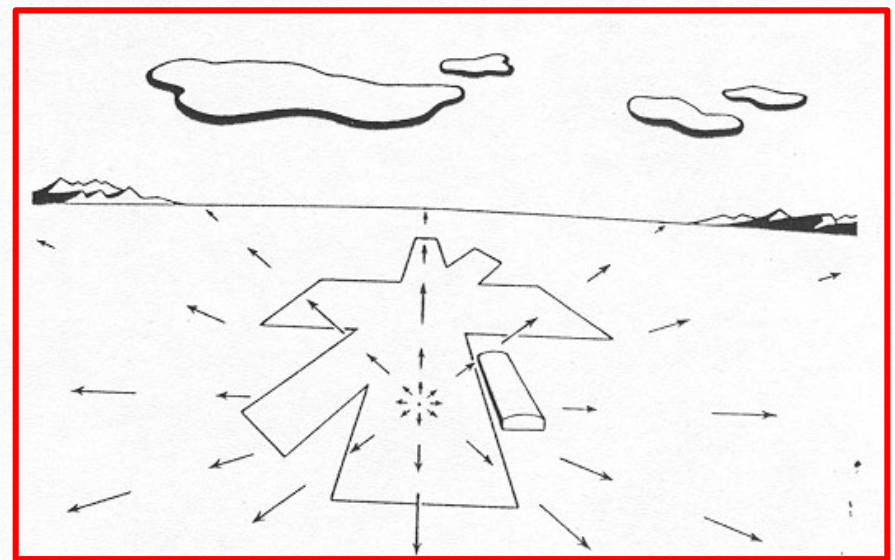
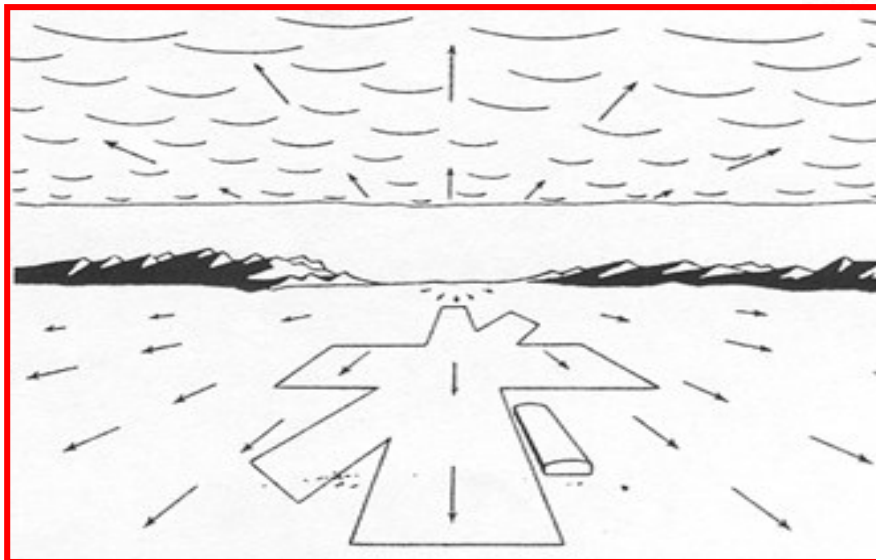
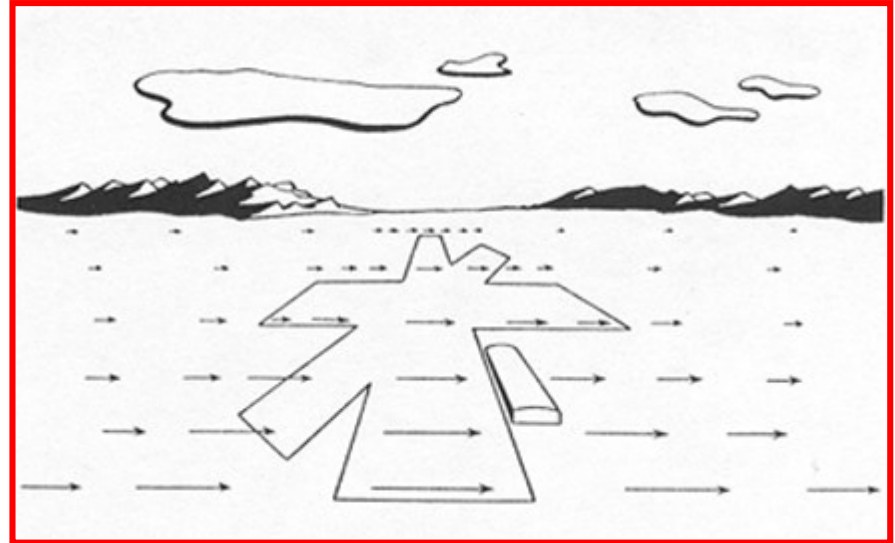
In degenerate local regions, only the normal velocity is measurable.

The aperture problem



Optical Flow as seen from an aircraft

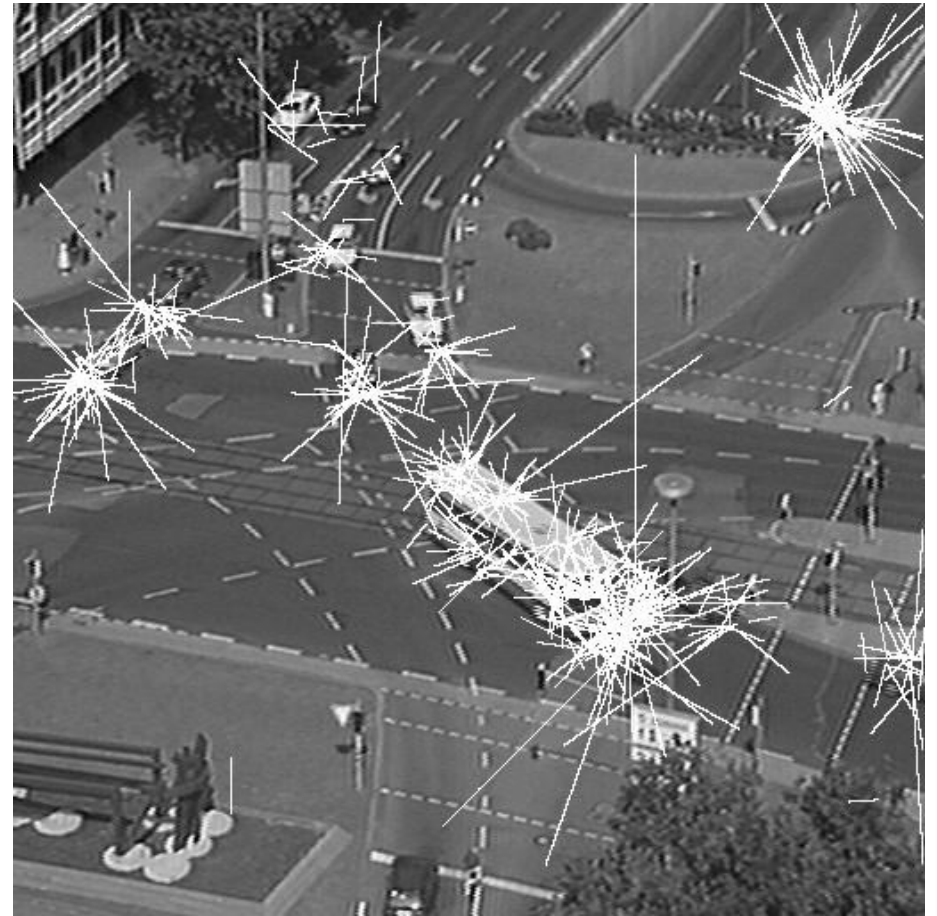
- MF of a pilot looking to the right. FoE is off at infinity to the left or FoC is off to the right
- MF with a plane parallel to the ground FoE at infinity on the horizon.
- MF during landing FoE on the point of impact.



Example



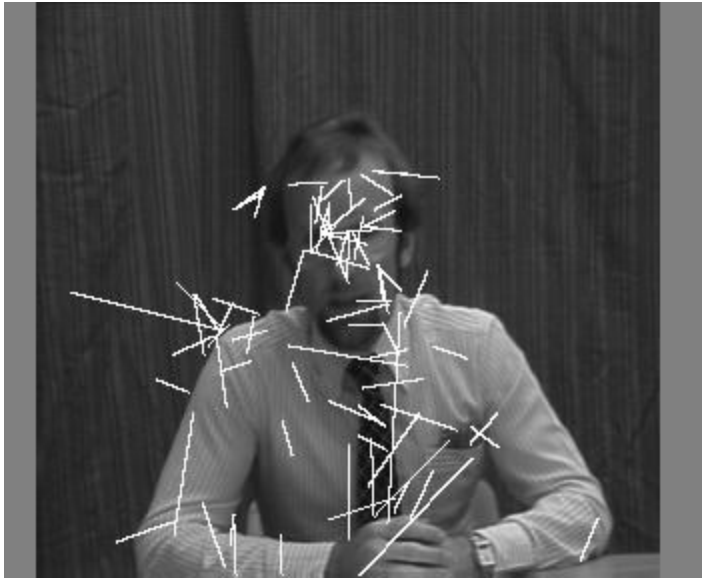
OF results



Example

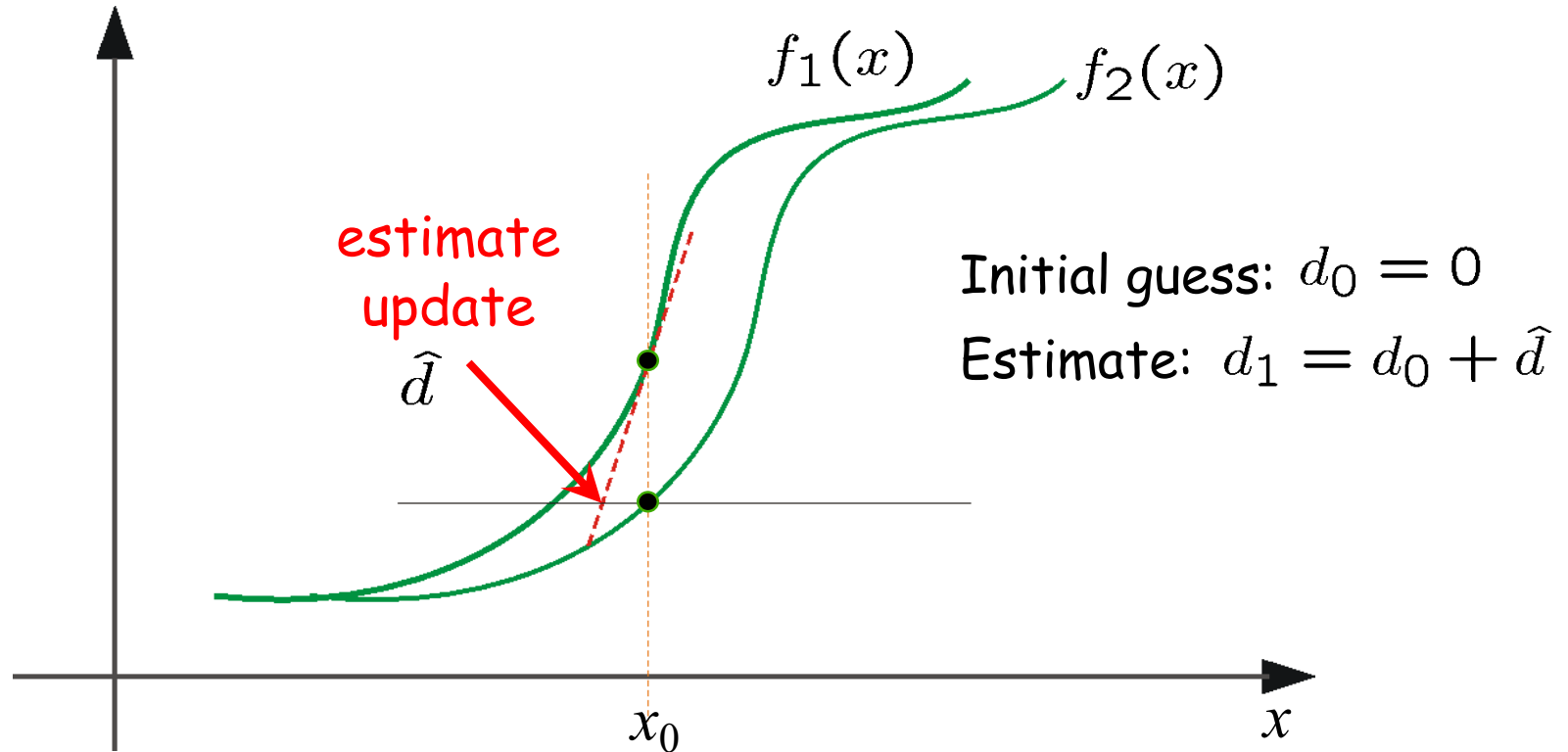


Of results

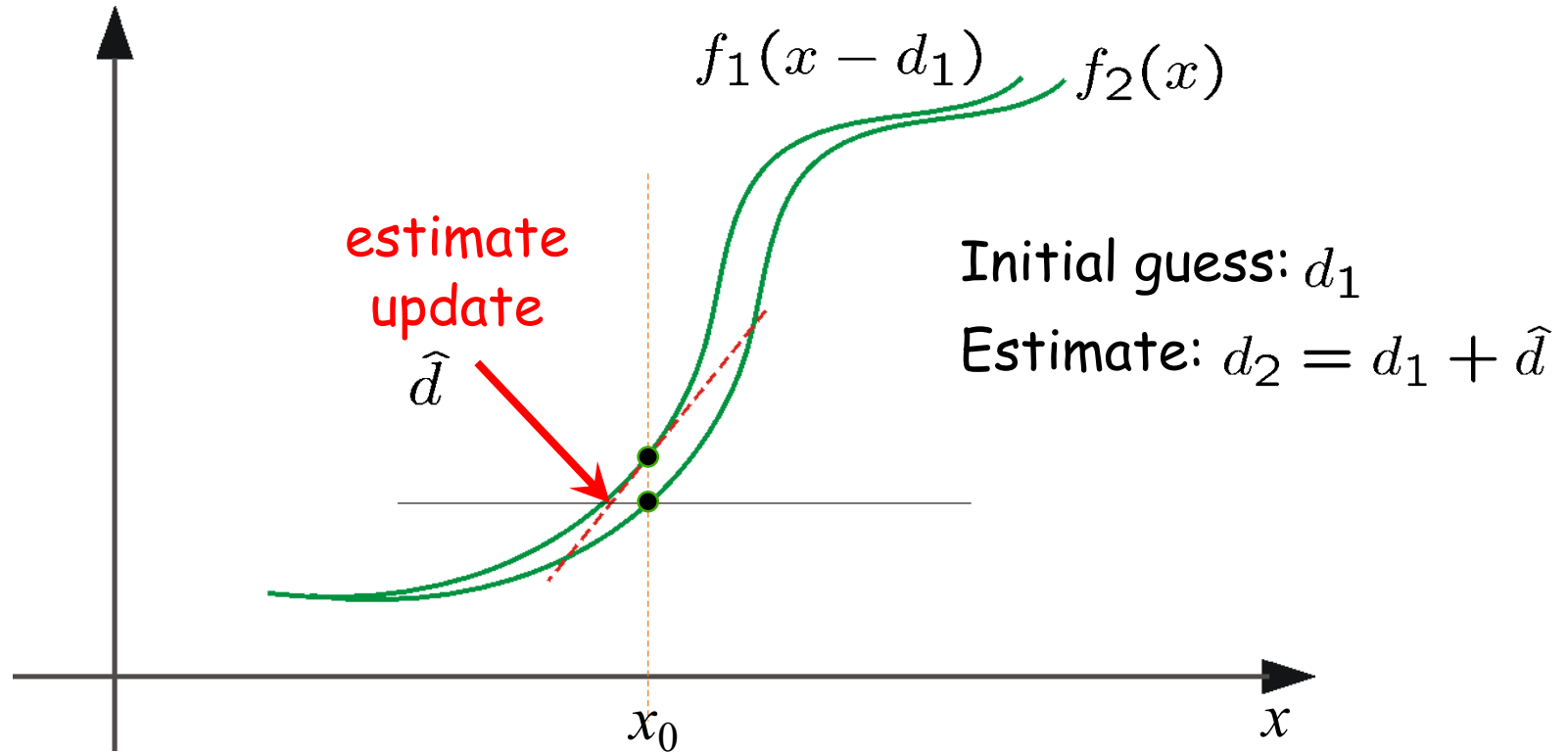


Optical Flow: Iterative Refinement

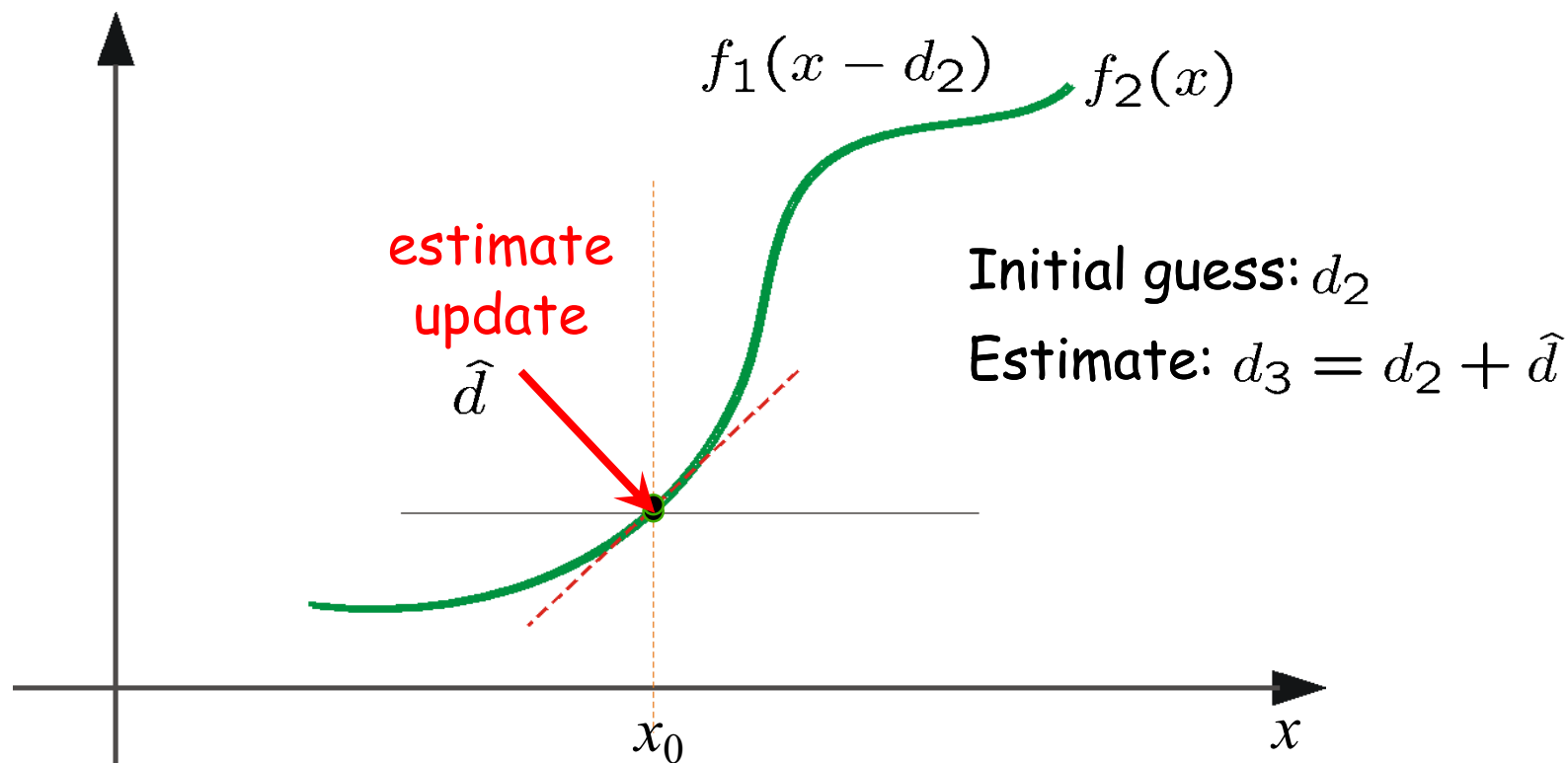
- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation. Refine estimate by repeating the process



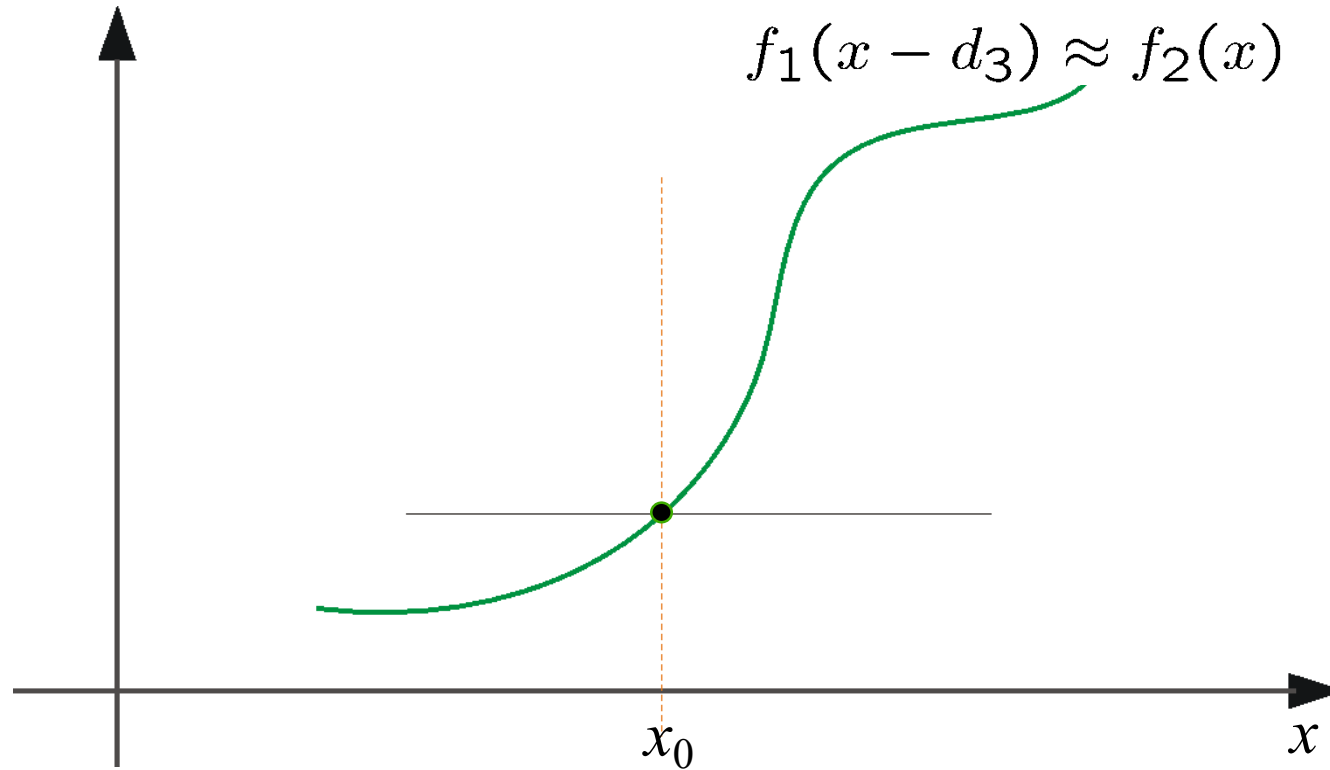
Optical Flow: Iterative Estimation



Optical Flow: Iterative Estimation



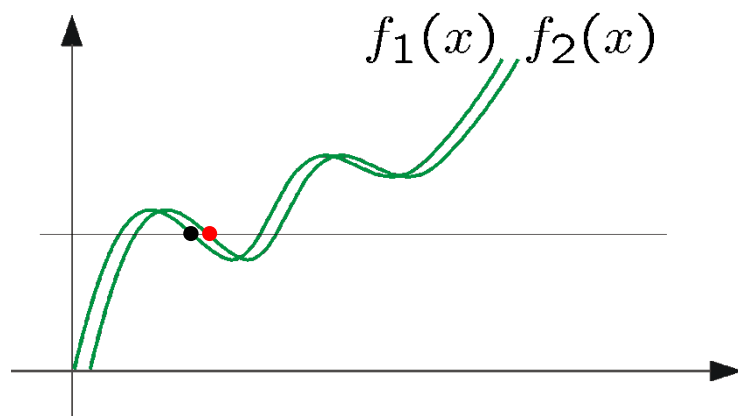
Optical Flow: Iterative Estimation



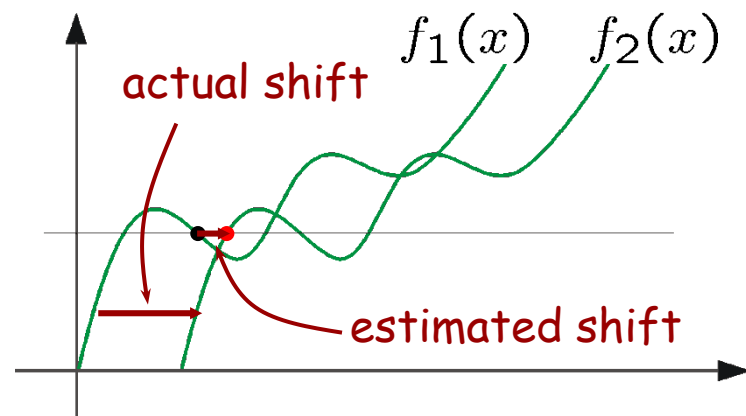
Optical Flow: Aliasing

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.

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

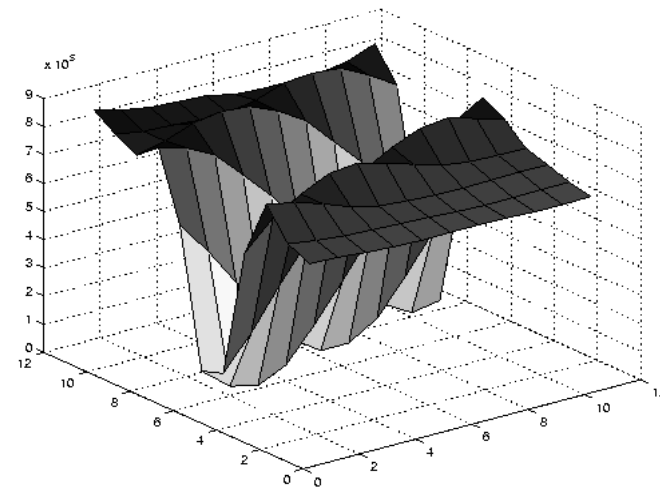
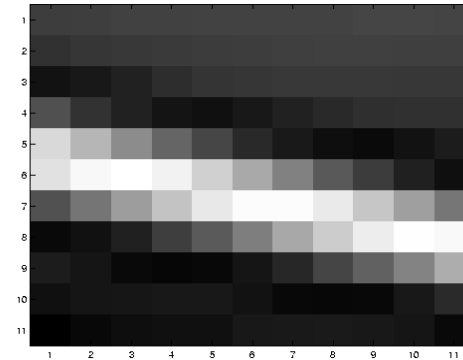


- Is this motion small enough?
 - Probably not—it's much larger than few pixels (2nd order terms dominate)
 - How might we solve this problem?

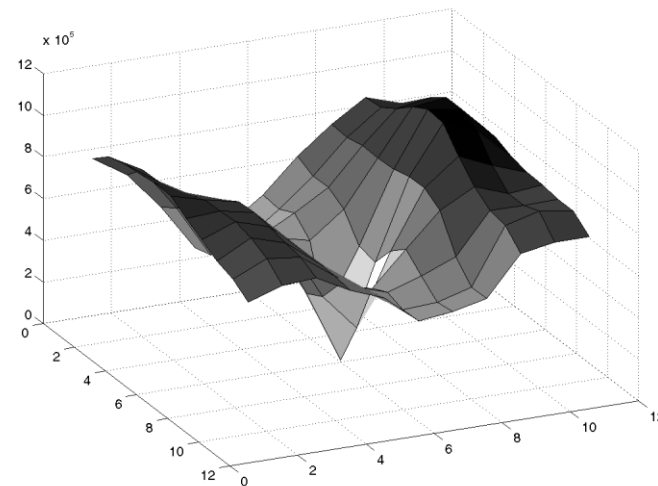
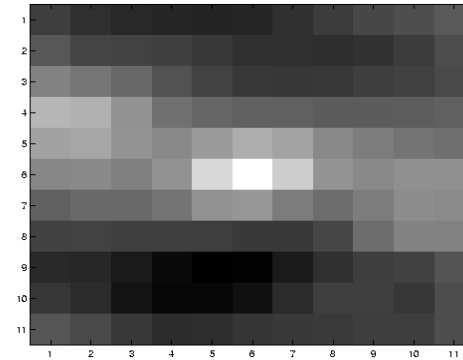
Correlation and SSD

- 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

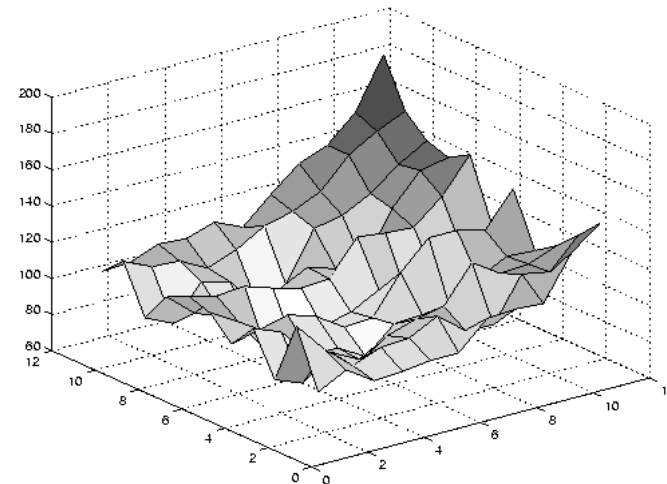
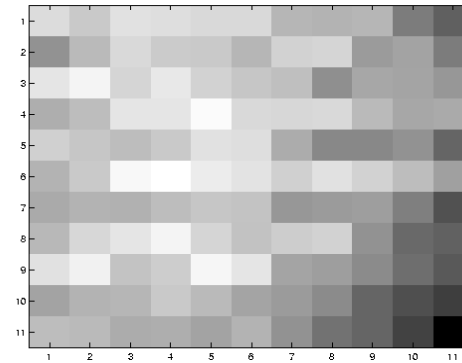
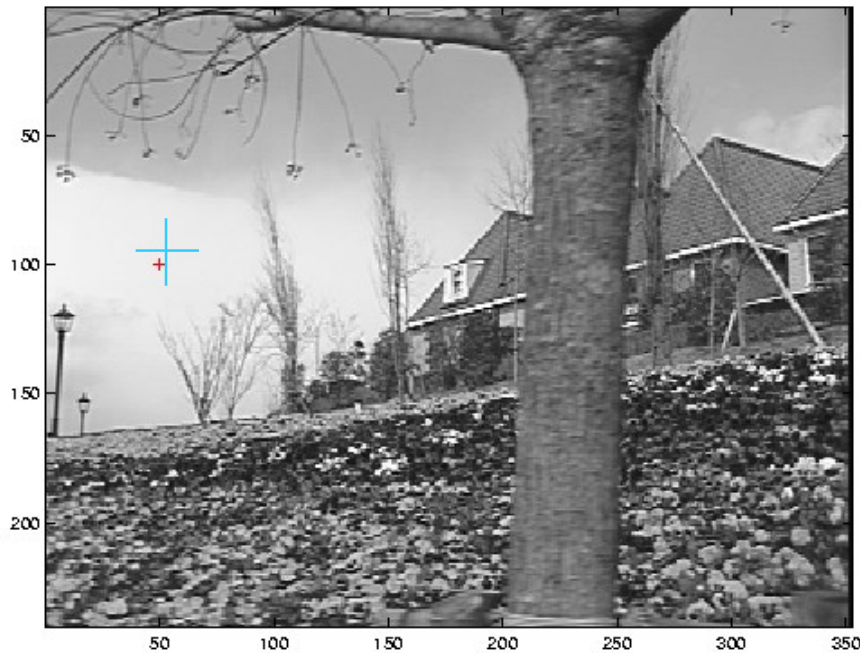
SSD Surface -- Edge



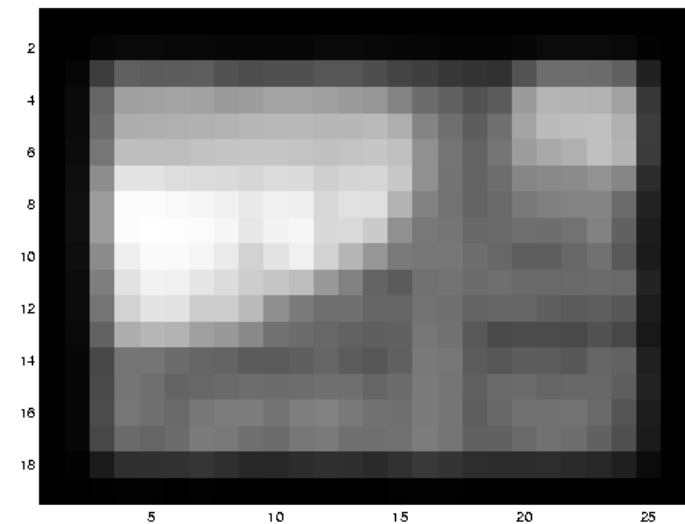
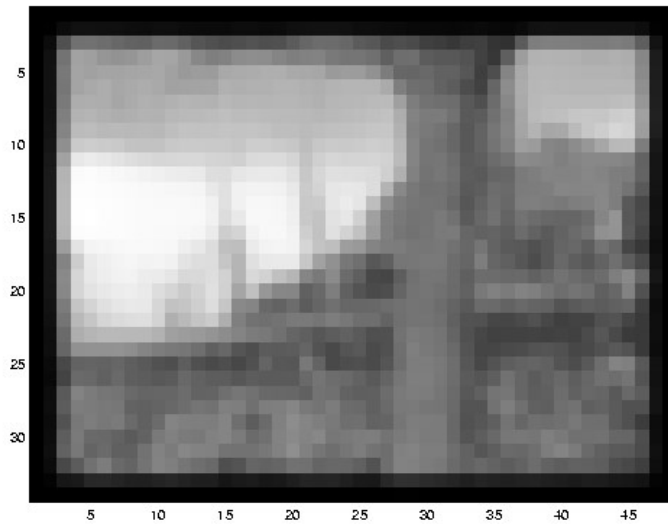
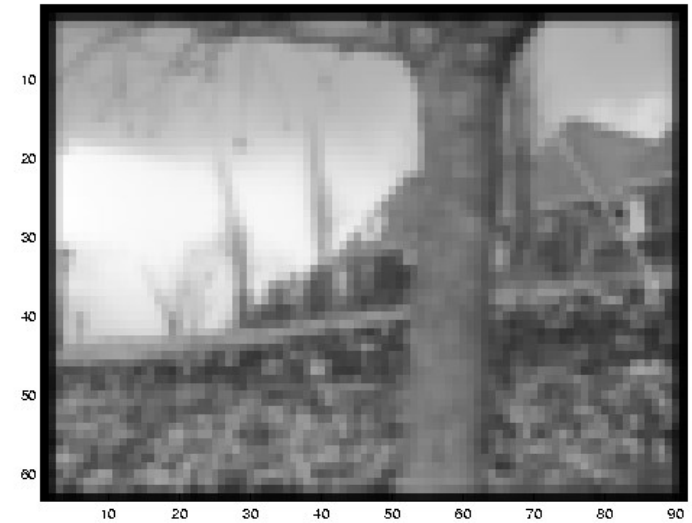
SSD Surface - Textured area



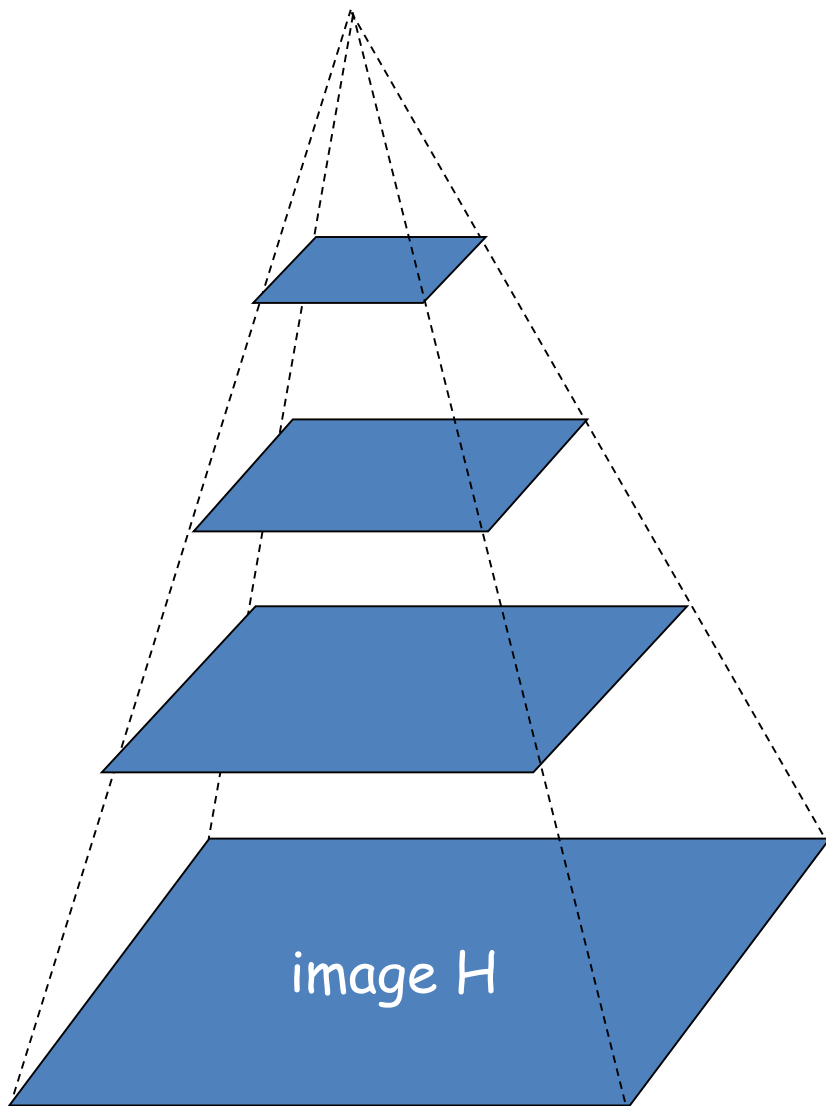
SSD - homogeneous area



Reduce the Resolution!



Coarse-to-fine Optical Flow Estimation



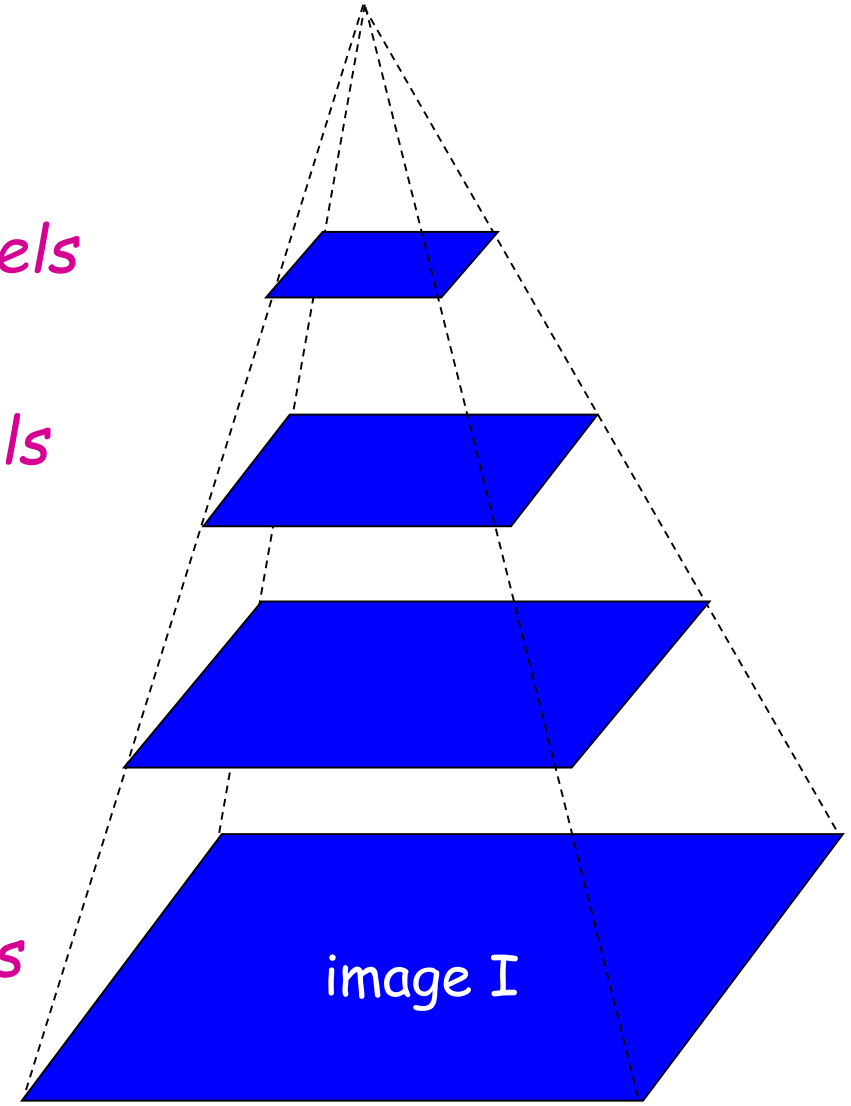
Gaussian pyramid of image H

$u=1.25$ pixels

$u=2.5$ pixels

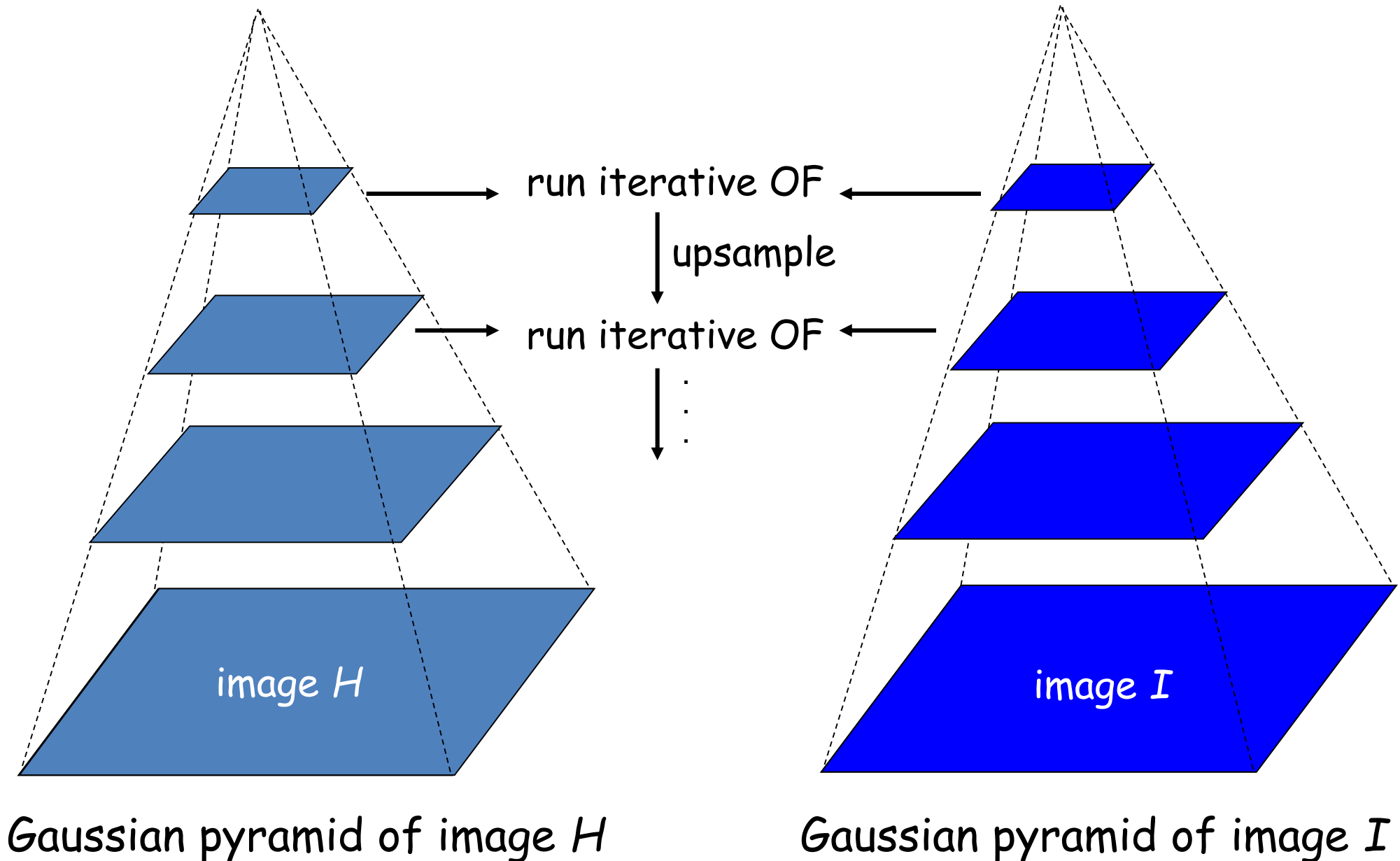
$u=5$ pixels

$u=10$ pixels

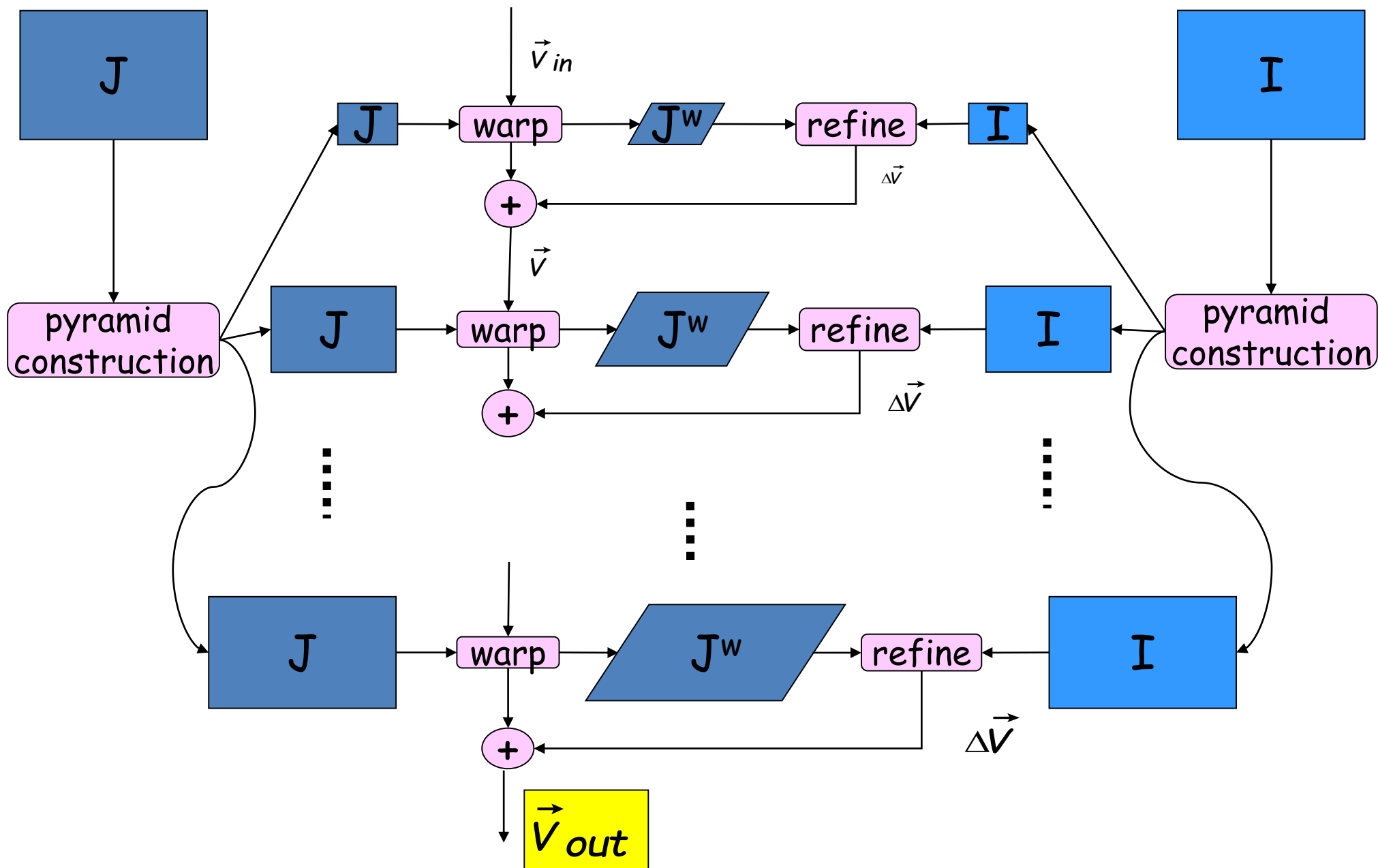


Gaussian pyramid of image I

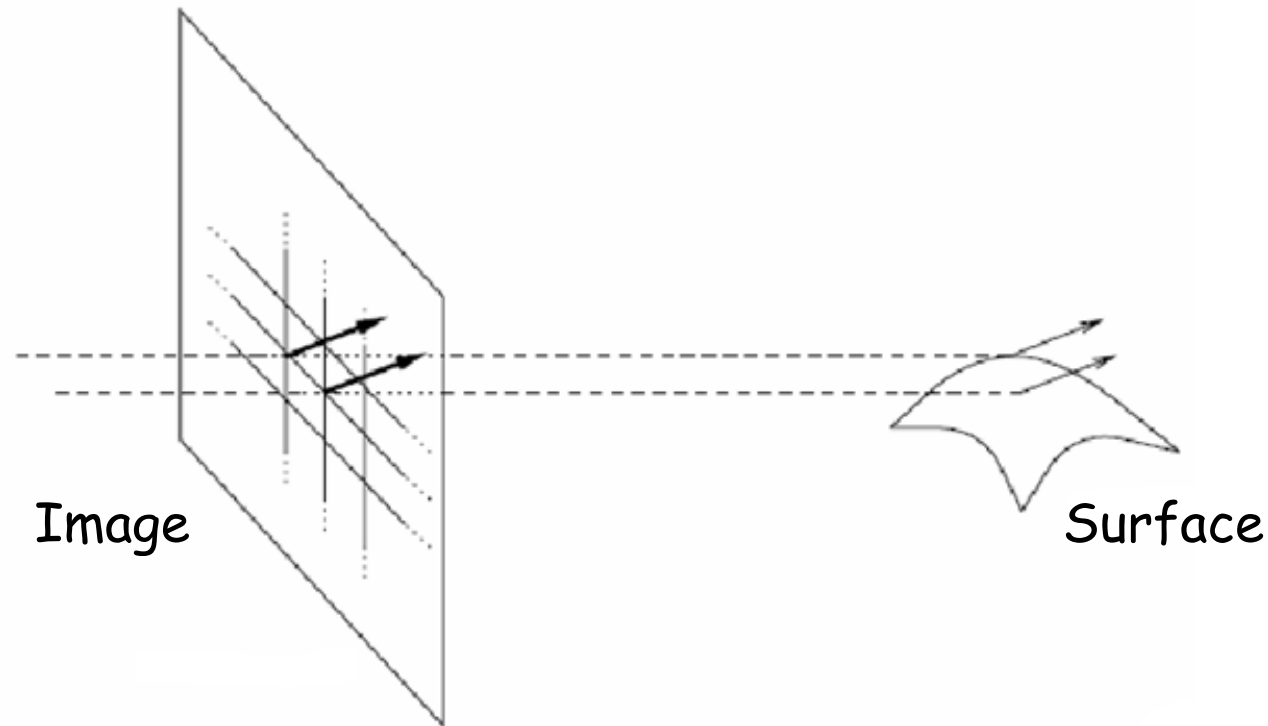
Coarse-to-fine Optical Flow Estimation



Coarse-to-fine estimation



Optical Flow Assumptions: spatial coherence

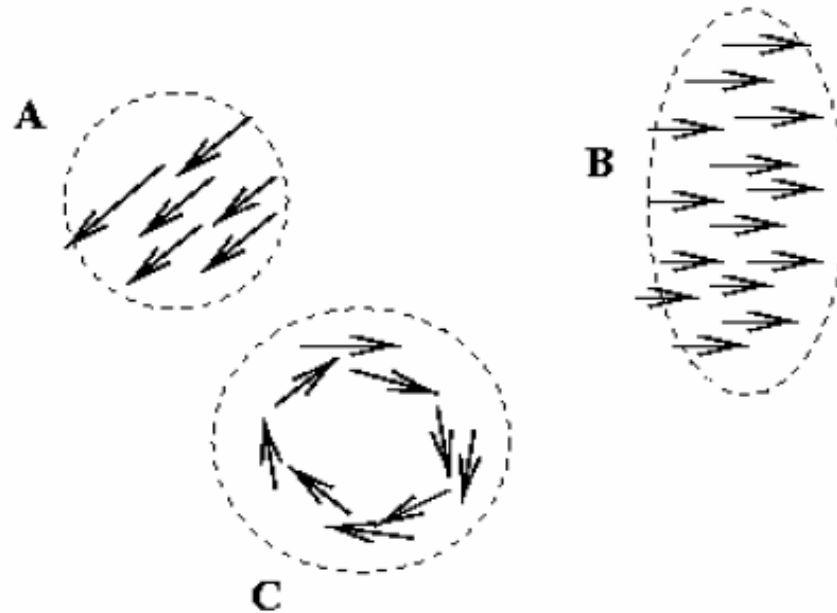


■ Assumption

- 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.

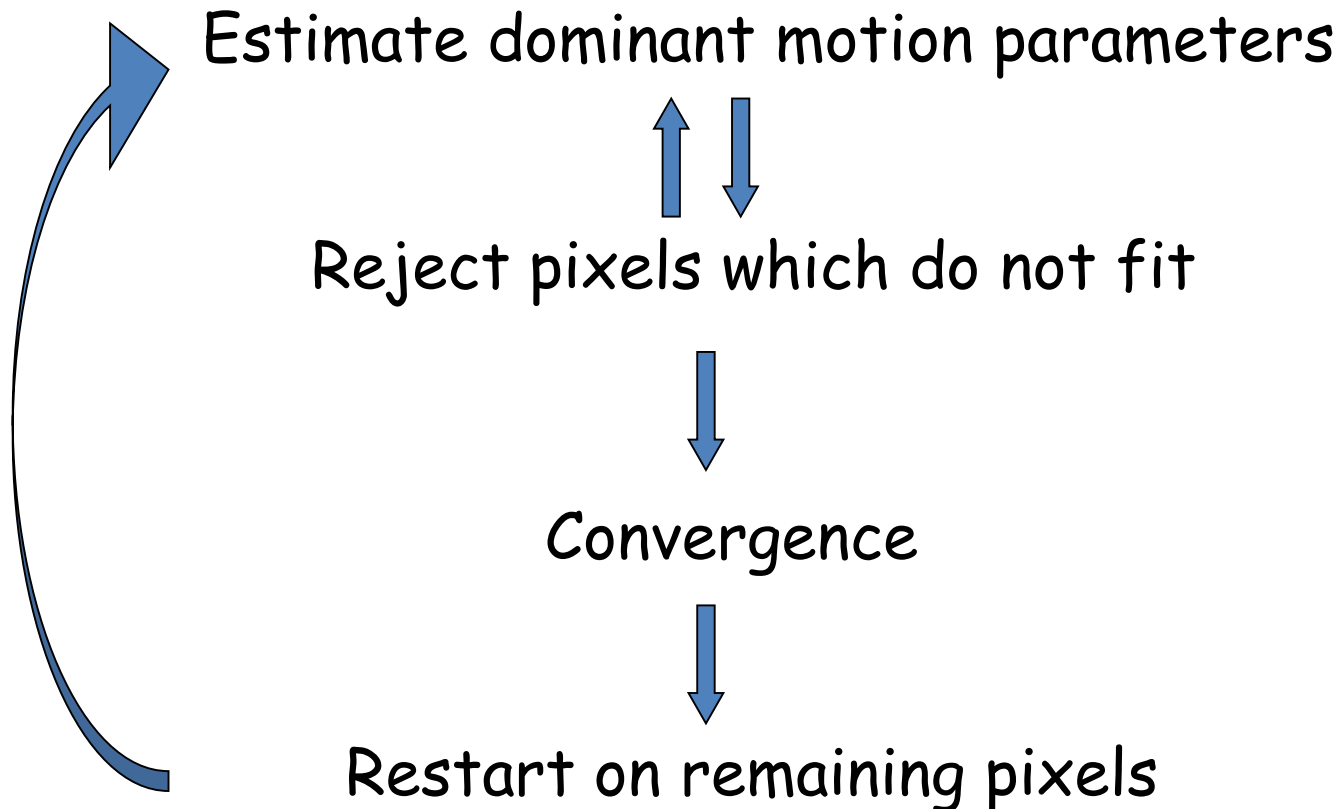
Deployment of Video segmentation

- Segment the video into multiple *coherently* moving objects
 - background subtraction
 - boundary detection
 - motion segmentation



Layered Representation

- For scenes with multiple motions, for each one:



Layered motion

- Break image sequence up into "layers":



- Describe each layer's motion, for each layer:
 - stabilize the sequence with the affine motion
 - compute median value at each pixel
- Determine occlusion relationships

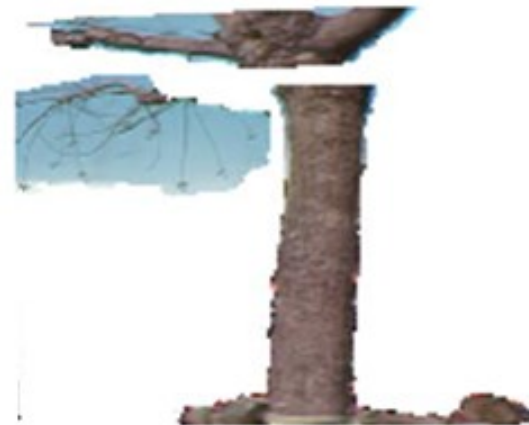
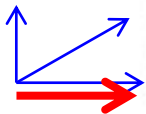
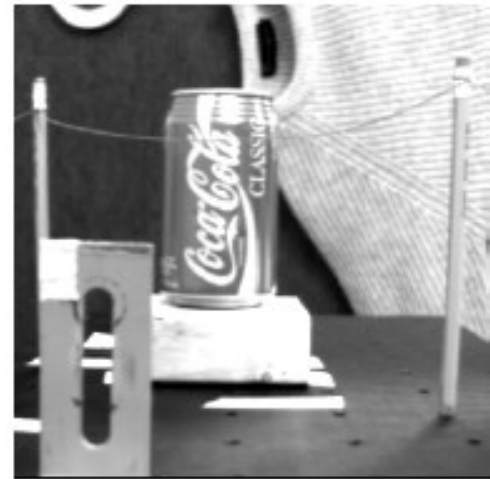


Image Data Sets

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

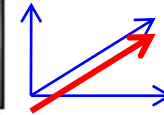


(a) SRI Trees

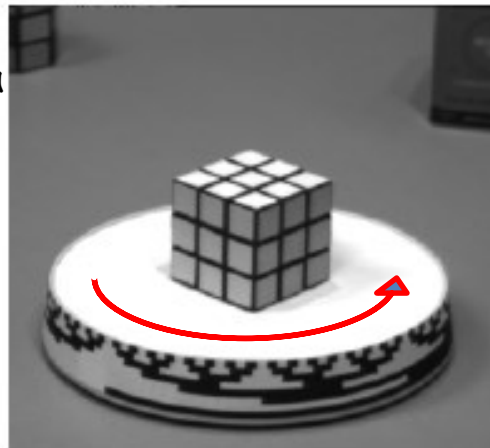


(b) NASA Sequence

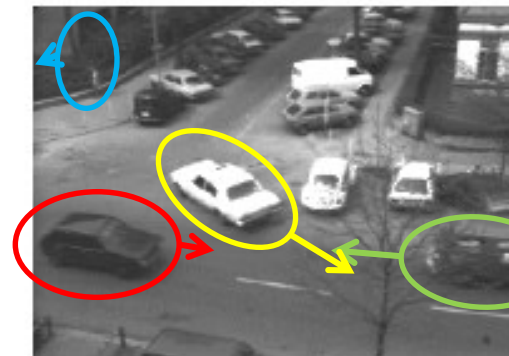
- Primarily dilational
- Velocities < 1 pixel/frame



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



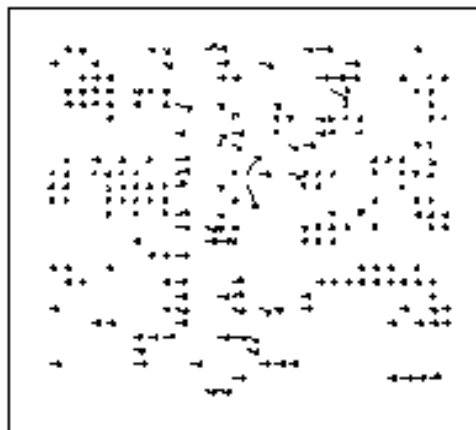
(c) Rubik Cube



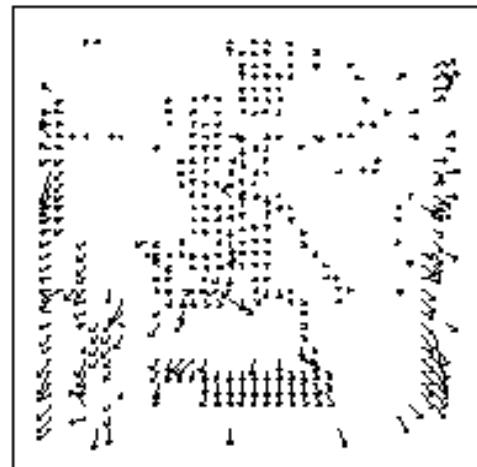
(d) Hamburg Taxi

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

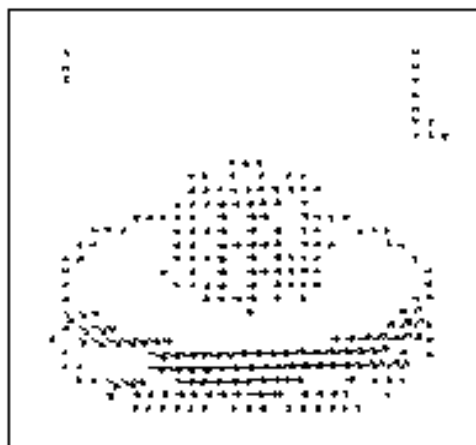
Results: Horn-Schunck



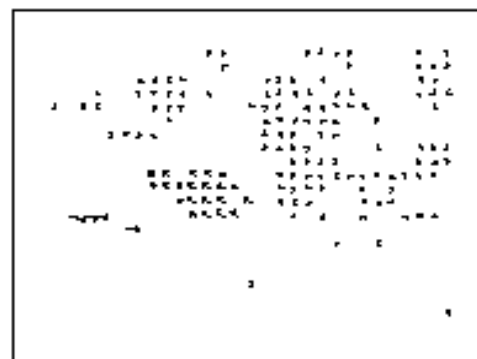
(a) SRI Trees



(b) NASA Sequence



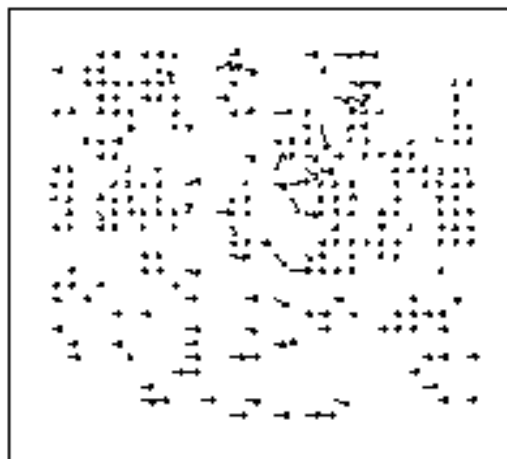
(c) Rubik Cube



(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 $\|\nabla I\| \geq 5.0$.

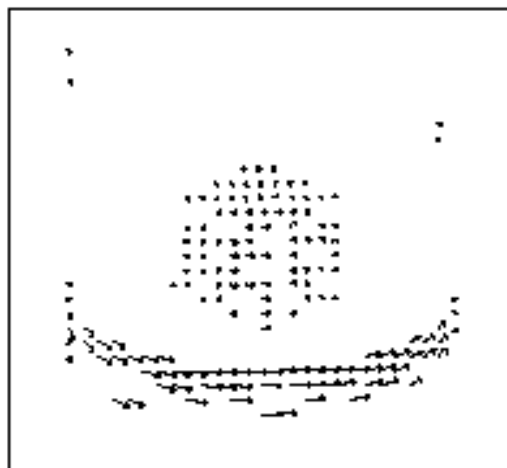
Results: Lucas-Kanade



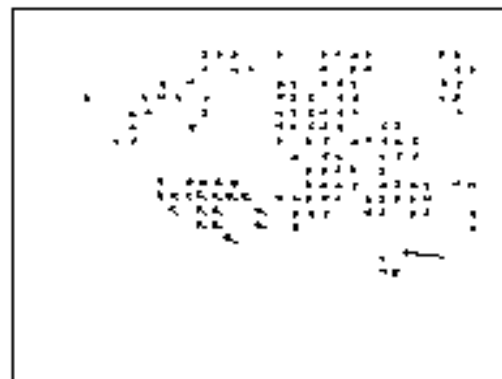
(a) SRI Trees



(b) NASA Sequence



(c) Rubik Cube



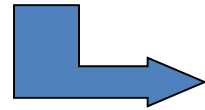
(d) Hamburg Taxi

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 $\lambda_2 \geq 1.0$

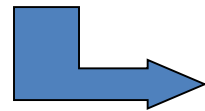
Other break-downs

- Brightness constancy is **not** satisfied



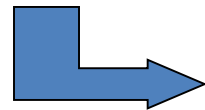
Correlation based methods

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

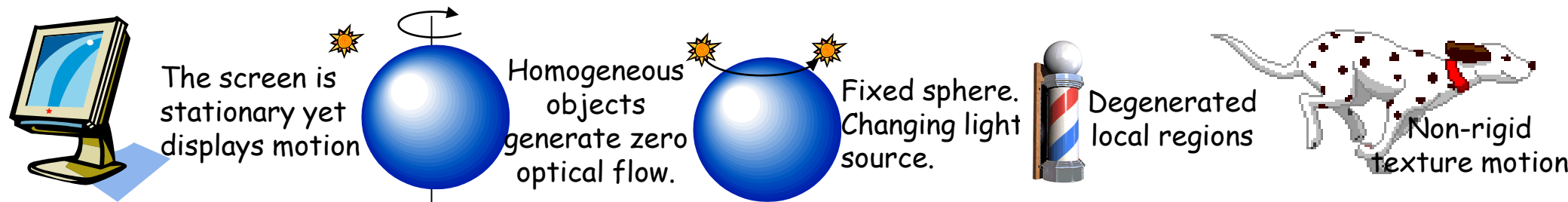


Regularization based methods

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



Use multi-scale estimation



Optical Flow: Where do pixels move to?

