Motion

Prof. Dr. Elli Angelopoulou

Pattern Recognition Lab (Computer Science 5)

University of Erlangen-Nuremberg

Images over Time

- So far we have analyzed either single images, or multiple images acquired simultaneously. We have only captured stationary information about a scene.
- As time passes:
  - objects in the scene may move
  - the camera may move

  either way, there is motion.
- In computer vision when use the term *Motion* to refer to images taken over time.
- In the presence of motion:
  - some objects will move while others will not
  - different objects move in different directions
  - there may be rigid as well as non-rigid motion
  - there may be occlusion.
- What can we tell about images acquired over time? (i.e. movie).

Motion

- There are two main goals within the topic of motion analysis:
  - Detect which objects are moving and in which direction.
  - Extract shape information if possible.
- Motion analysis typically involves:
  - Motion detection.
  - Moving-object detection and location (tracking).
  - Derivation of 3D object properties.
- The information extracted from such an analysis can be used in the following applications:
  - Track object behavior
  - Correct for camera jitter (stabilization)
  - Align images (mosaics)
  - 3D shape reconstruction
  - Special effects

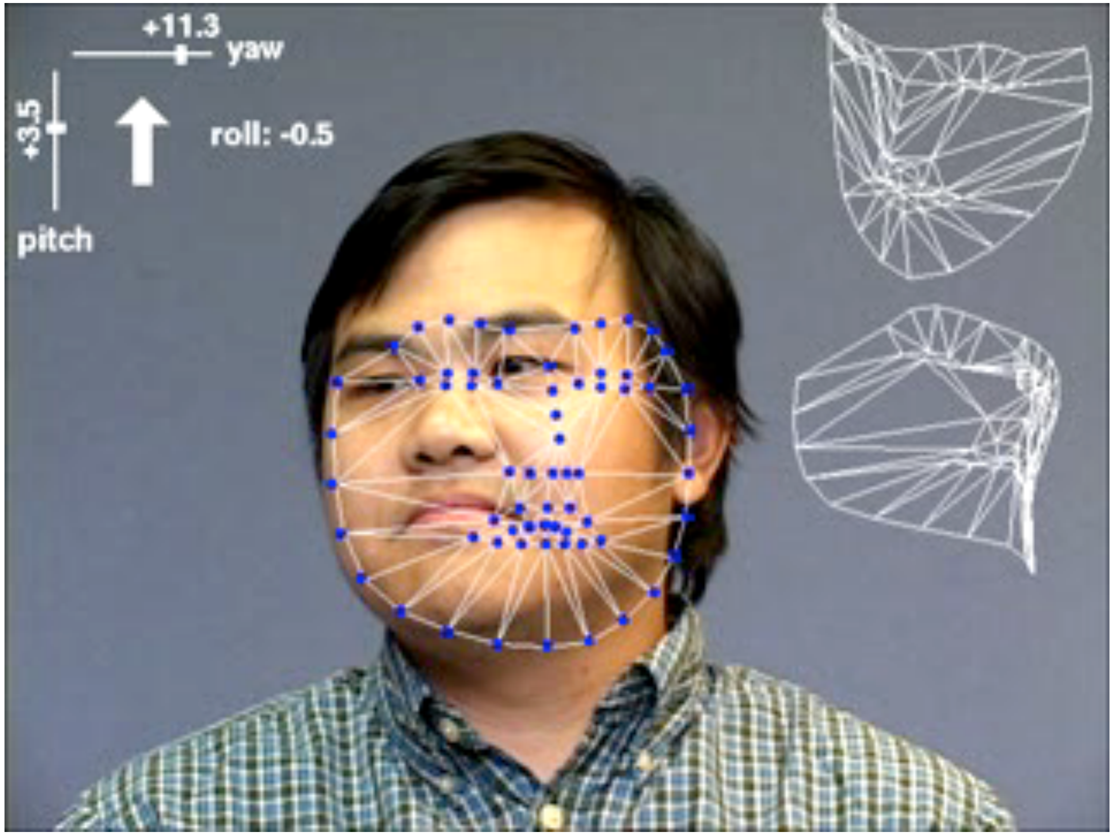
Tracking Rigid Objects

(Simon Baker et al., Carnegie Mellon University)

Tracking Non-Rigid Objects

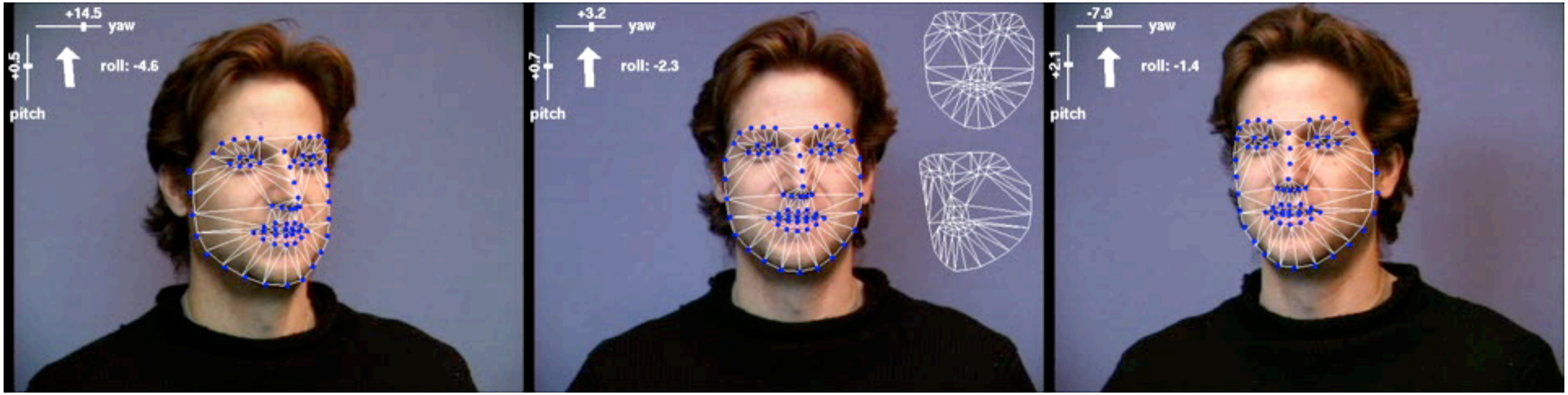
(Dorin Comaniciu et al., Siemens Corporate Research)

Face Tracking - Initialization



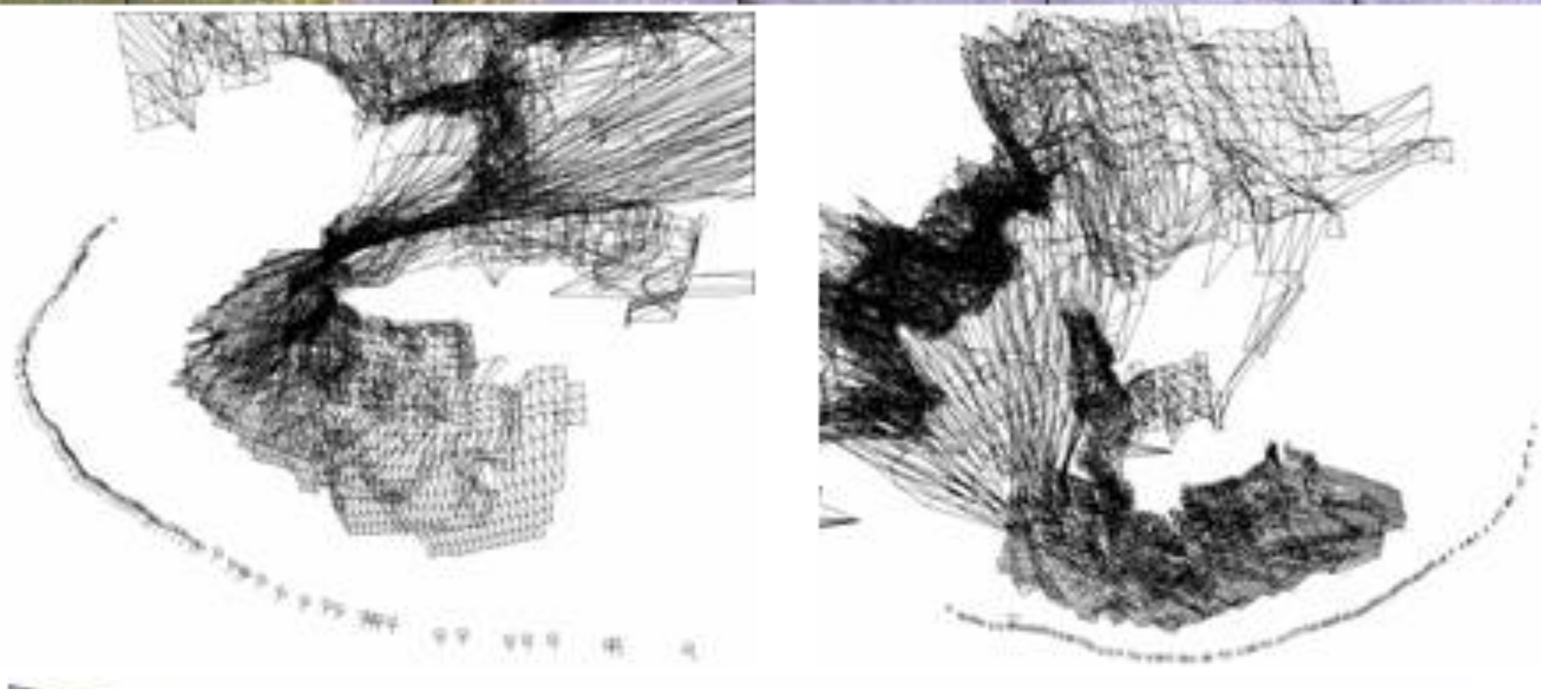
(Simon Baker et al., Carnegie Mellon University)

Face Tracking



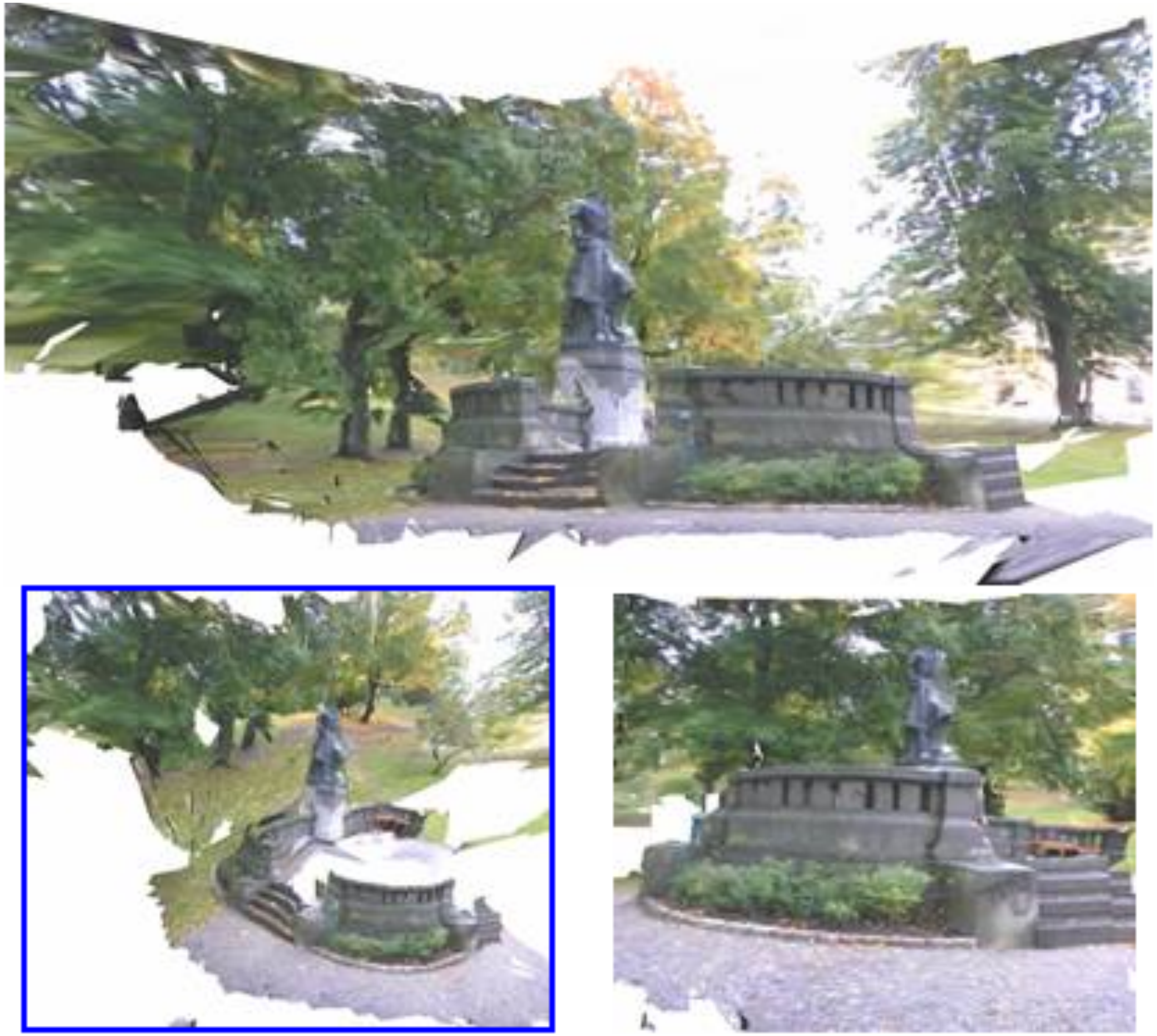
(Simon Baker et al., Carnegie Mellon University)

Structure from Motion



First the unknown camera motion and calibration is recovered. Then through the use of feature-based correspondence over multiple scenes, the 3D geometry of the scene is recovered. (David Nister, University of Kentucky)

Structure from Motion – Final Result



(David Nister, University of Kentucky)

Behavior Analysis

Query

Result

(Michal Irani et al., Weizmann Institute of Science)

Motion Analysis Basics

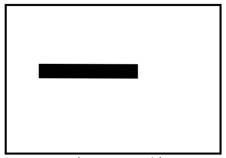
- What visual information can be extracted from the spatial and temporal changes that occur in an image sequence?

- Image sequence: a series of *N* images (frames) acquired at discrete time instants $t_k = t_0 + (k\ \delta t)$, where δt is a fixed time interval and $k = 0,1,\ldots N-1$.

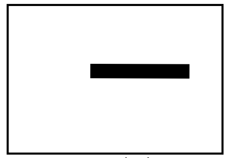
- δt is typically 1/24th sec, 1/30th of a second. This means that the apparent displacement (movement) between frames is at most a few pixels. This observation simplifies the correspondence problem (at the expense of accuracy).

Image Differencing

- Assuming the illumination conditions do not vary, image changes are caused by a *relative motion* between the camera and the scene.
- Simple motion example:



$I(t-1)$

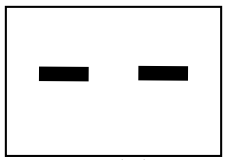


$I(t)$

- Idea: Subtract images. If there is a difference, then there is motion. Accordingly, no change means stationary part.

$$M(t) = I(t-1) - I(t)$$

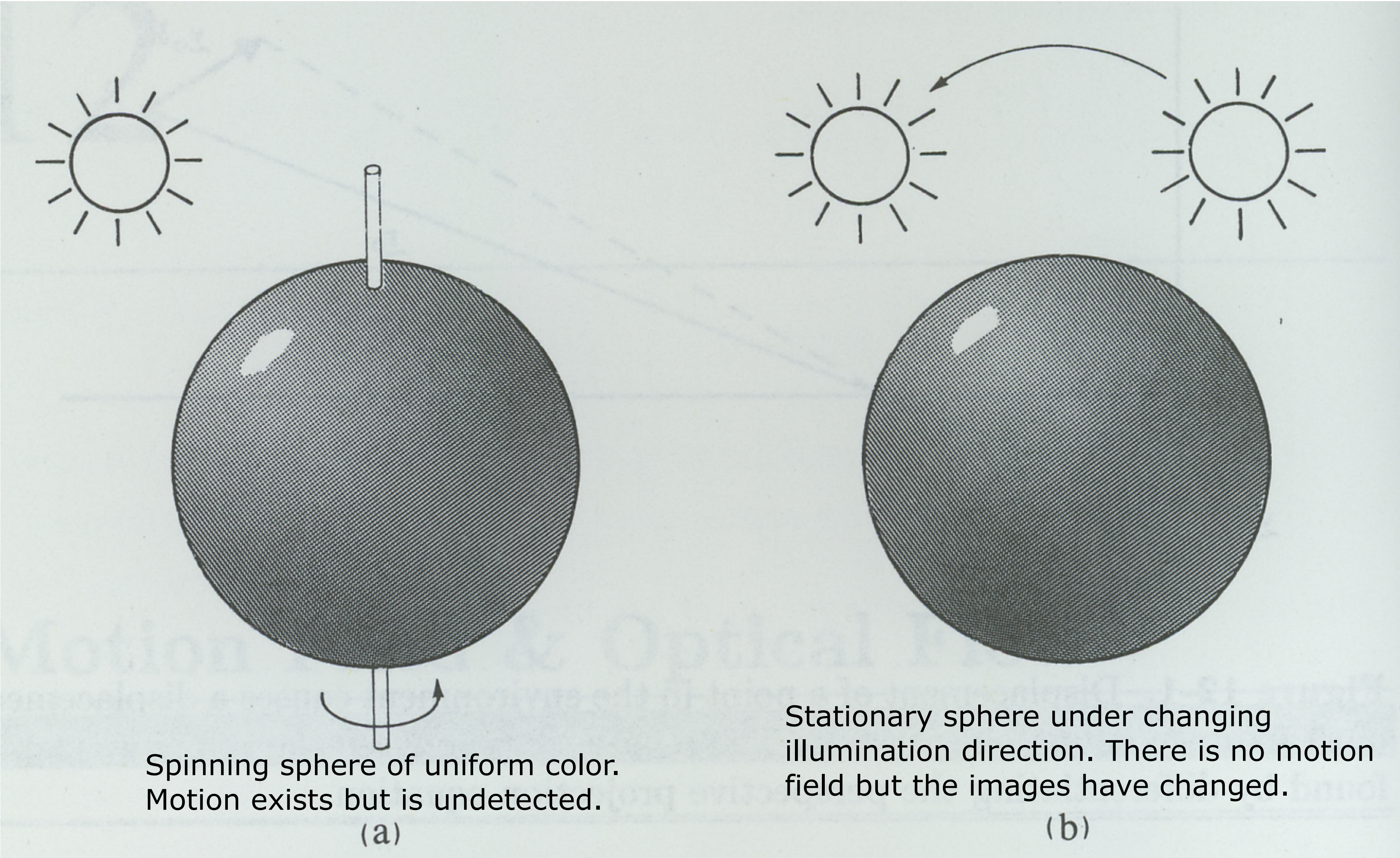
- In the previous example:



$M(t)$

- Either the line moved to the right, or the camera moved to the left. We are interested in relative motion.

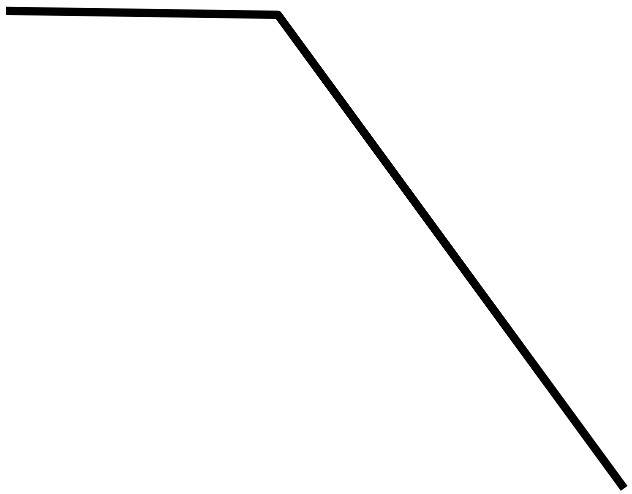
Does Differencing Suffice?



Spinning sphere of uniform color. Motion exists but is undetected.

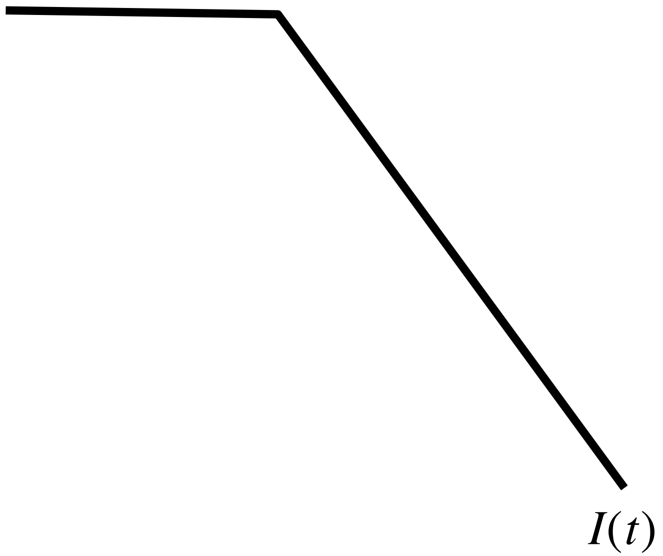
Stationary sphere under changing illumination direction. There is no motion field but the images have changed.

Aperture Problem

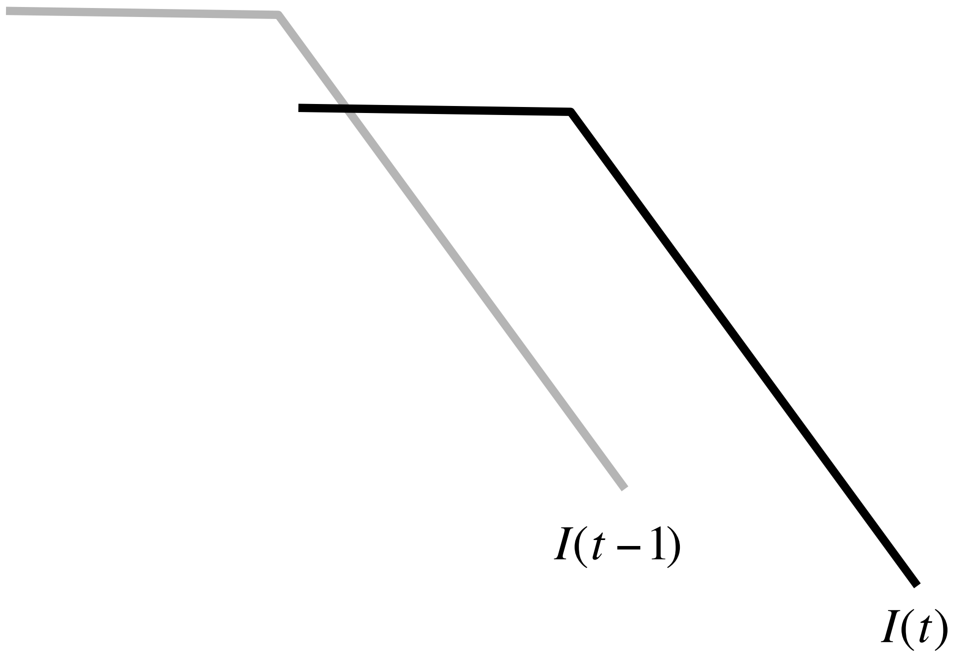


$I(t-1)$

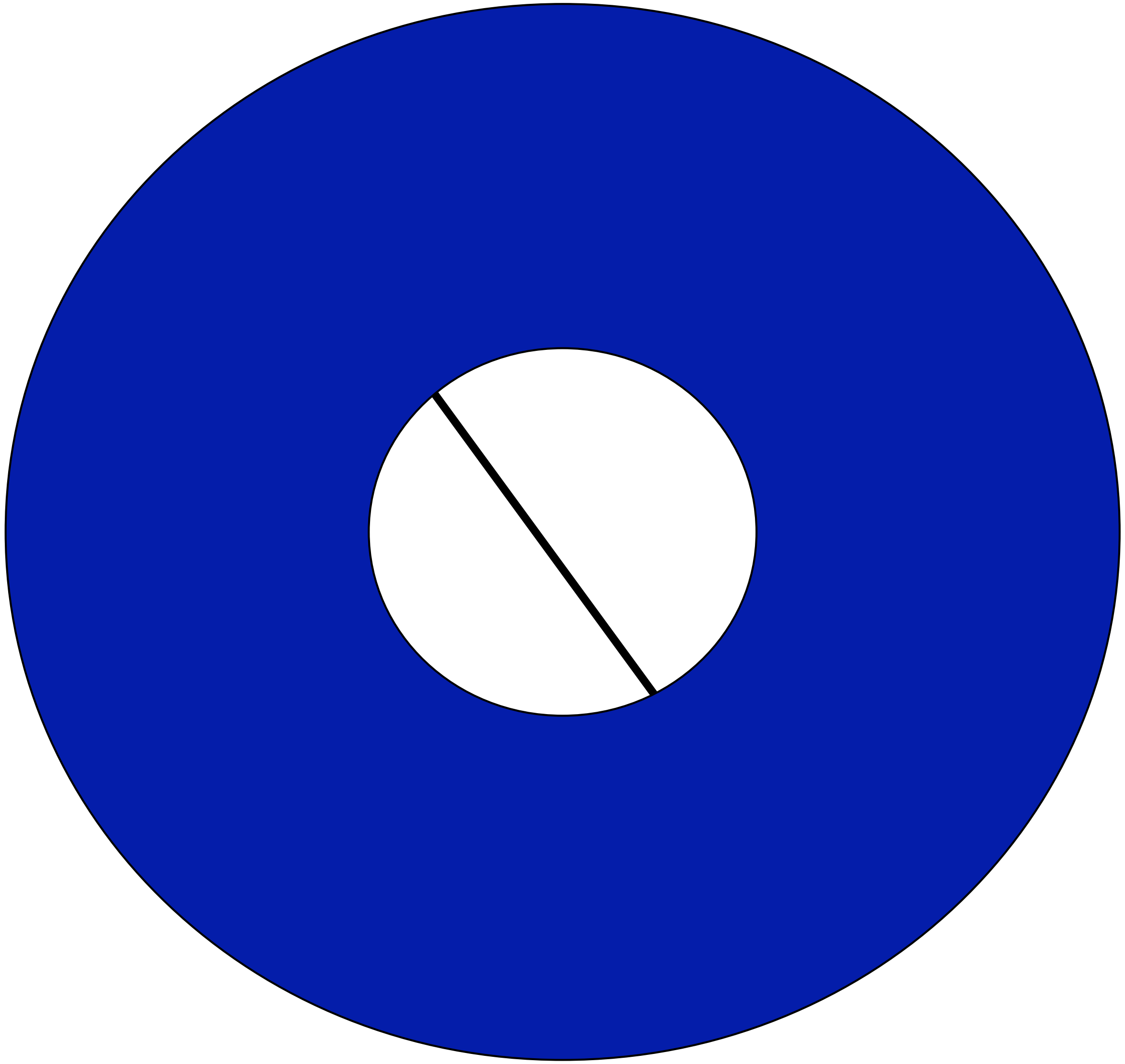
Aperture Problem - continued



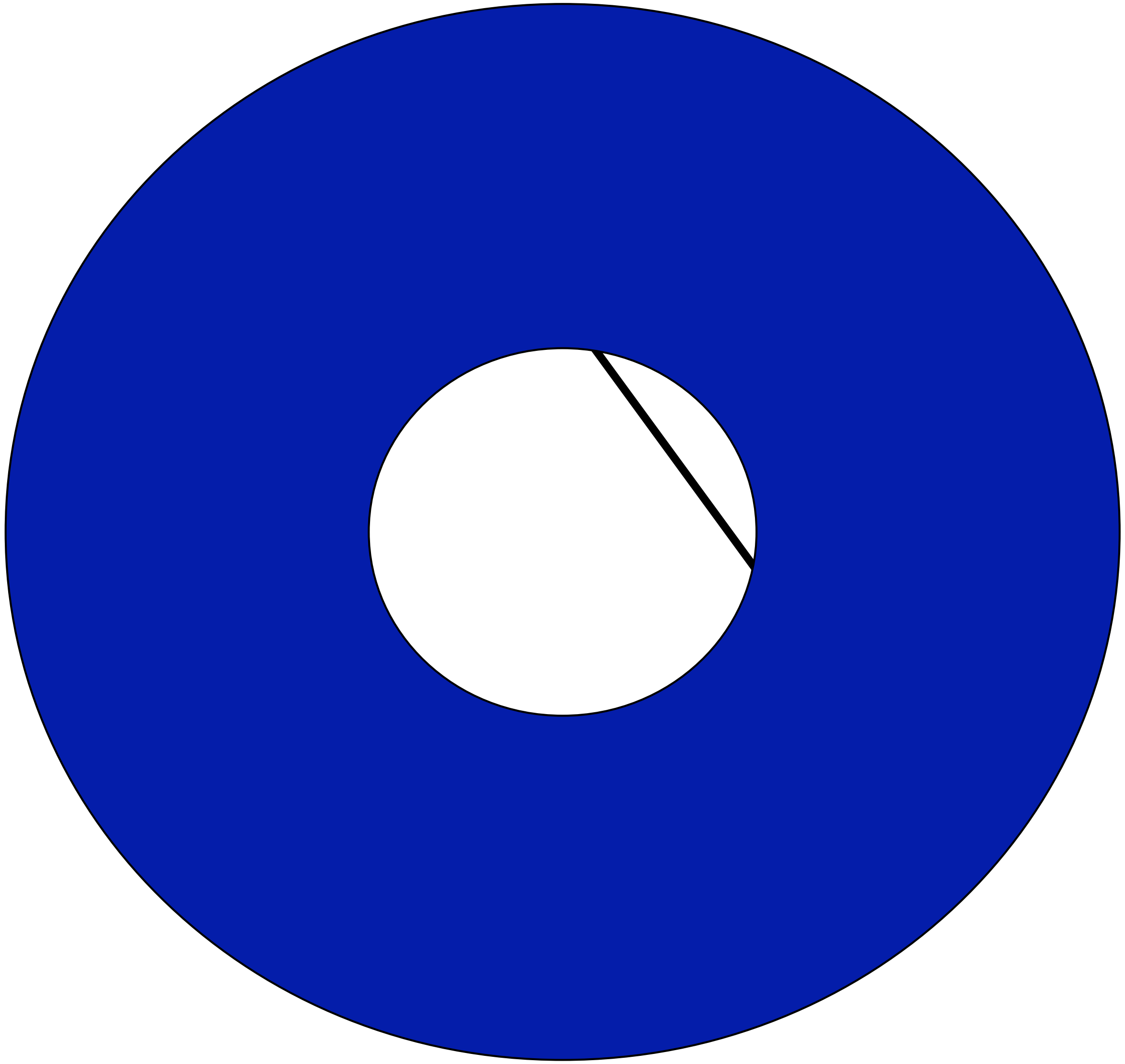
Aperture Problem - continued



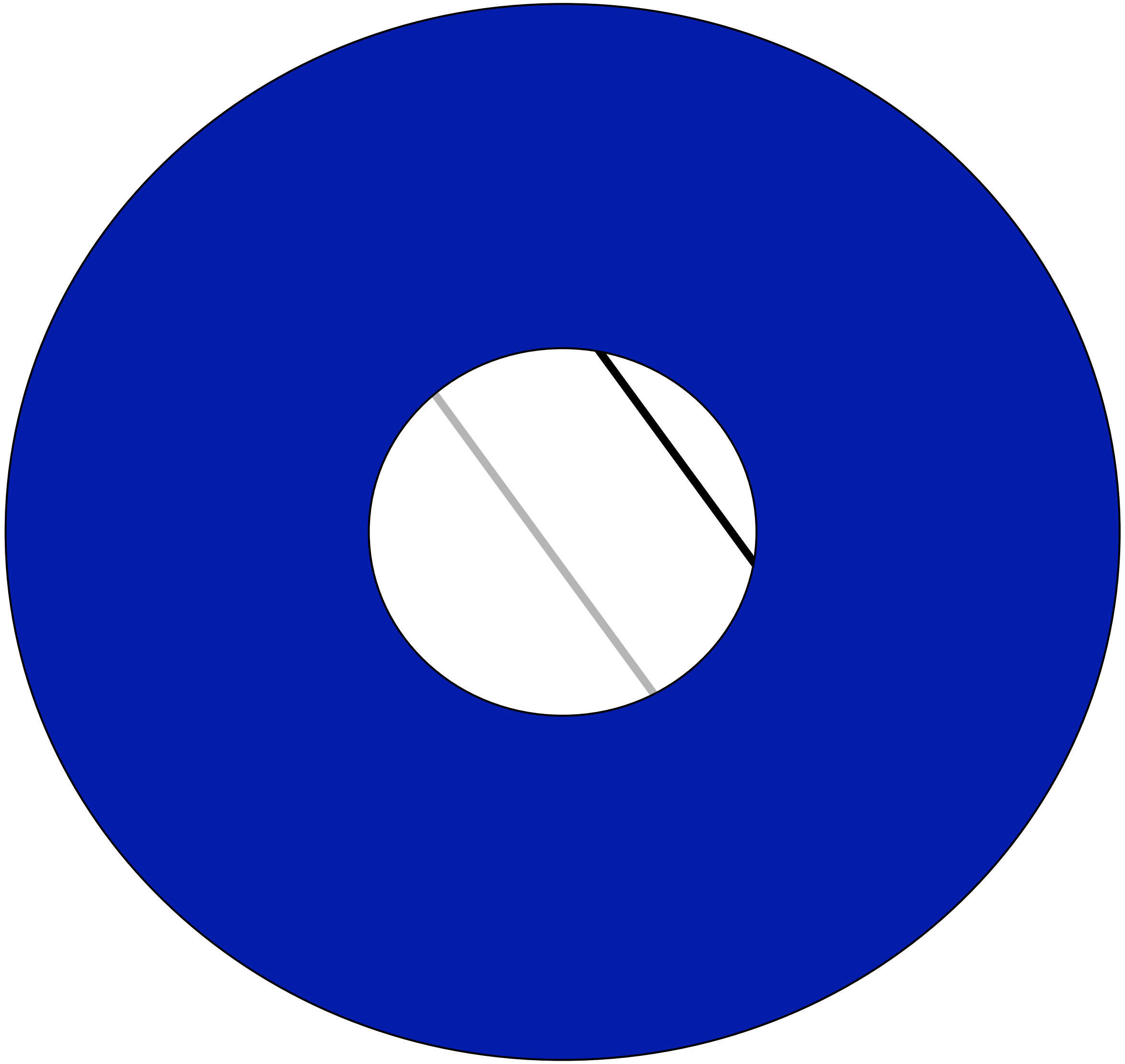
Aperture Problem - continued



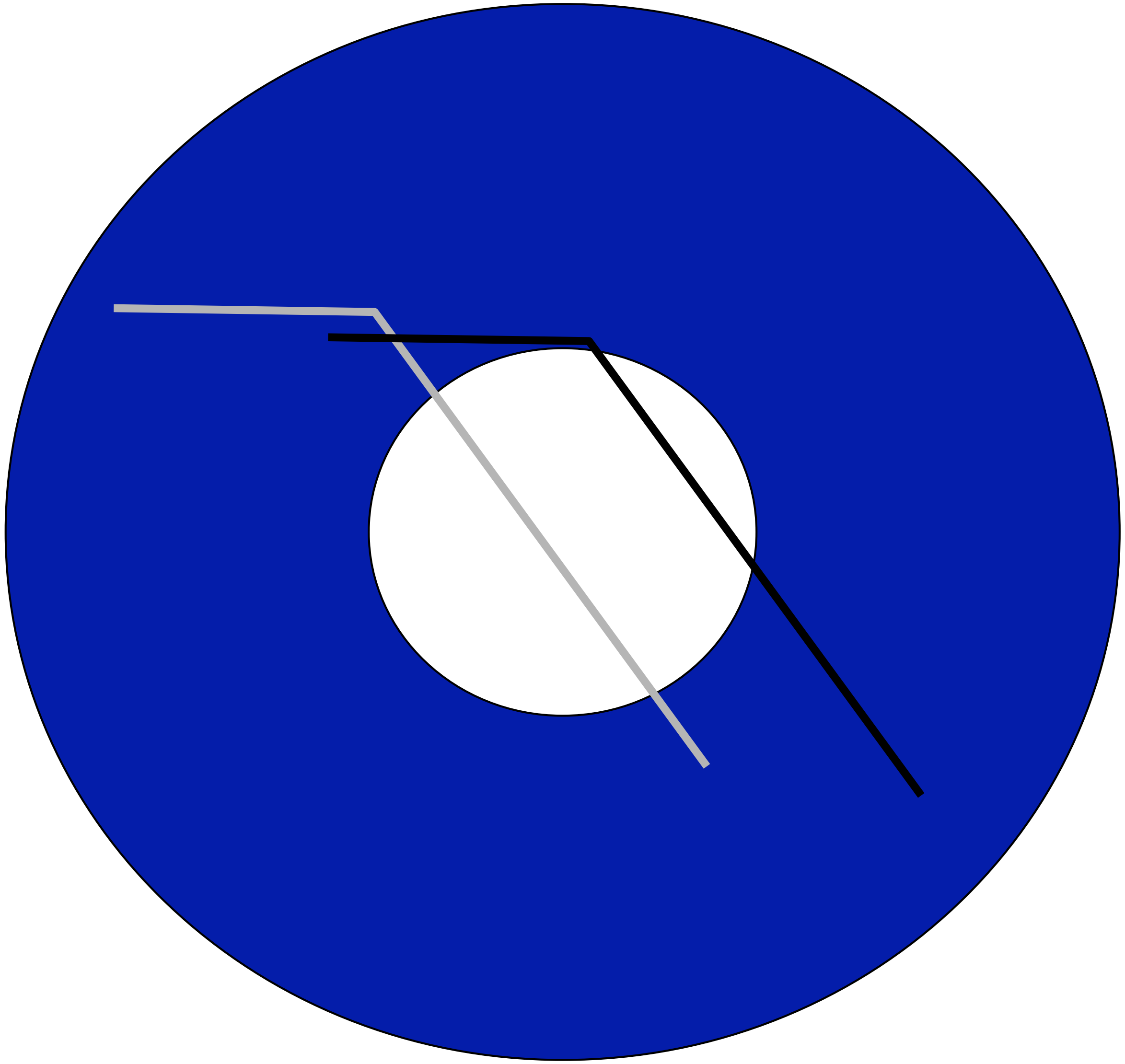
Aperture Problem - continued



Aperture Problem - continued



Aperture Problem - continued



Motion Recovery

- When dealing with image sequences over time, given the constraints in image capture, motion analysis can be summarized as follows:

1. Between $I(t_k)$ and $I(t_{k+1})$ we observe a change in intensity in a pixel *p*.
2. We associate this change with motion.
3. We try to infer which motion in 3D caused this motion in 2D.

Background Subtraction

- First we must estimate where motion occurs.
- If we have a relatively stationary (or slowly changing background) we can remove it from the image.
- Subtract the last two images:

$$d(i,j) = \begin{cases} 1 & \text{if} \quad \left| I_{t+1}(i,j) - I_t(i,j) \right| \le \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

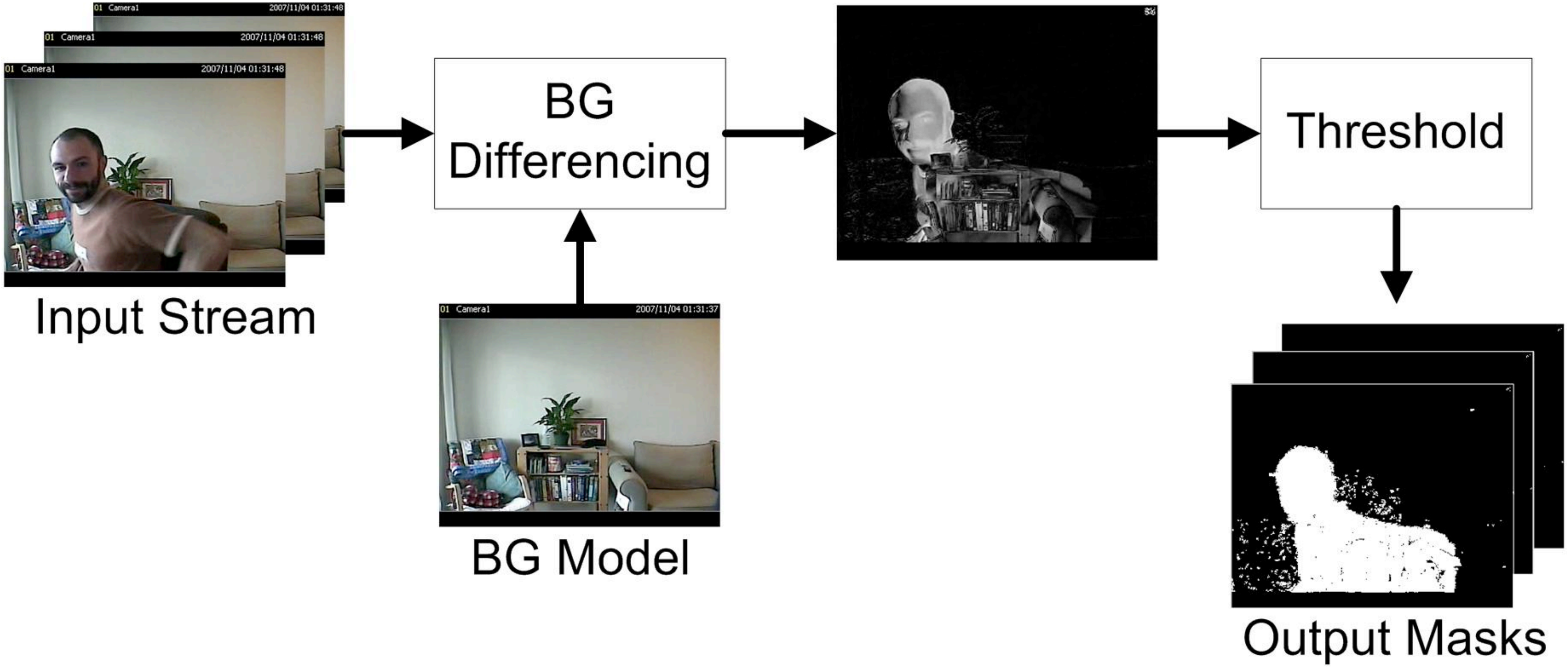
- Or compute a cumulative background image:

$$B_{t+1} = \left(w_a I_t + \sum_{i=1}^{t-1} w_i B_{t-i} \right) \Big/ w_c$$

- and then subtract:

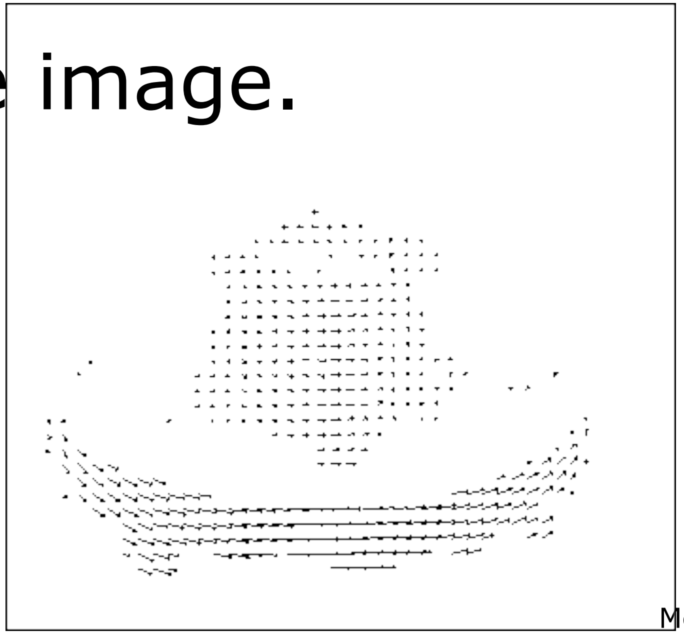
$$d(i,j) = \begin{cases} 1 & \text{if} \quad \left| I_{t+1}(i,j) - B_{t+1}(i,j) \right| \le \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

Background Subtraction Example



Optical Flow

- Optical Flow: The apparent (observed) motion of the image brightness pattern.
- It is a collection of 2D velocity vectors, each of them describing the velocity by which the brightness pattern moved.
- It is a 2D vector field on the image.

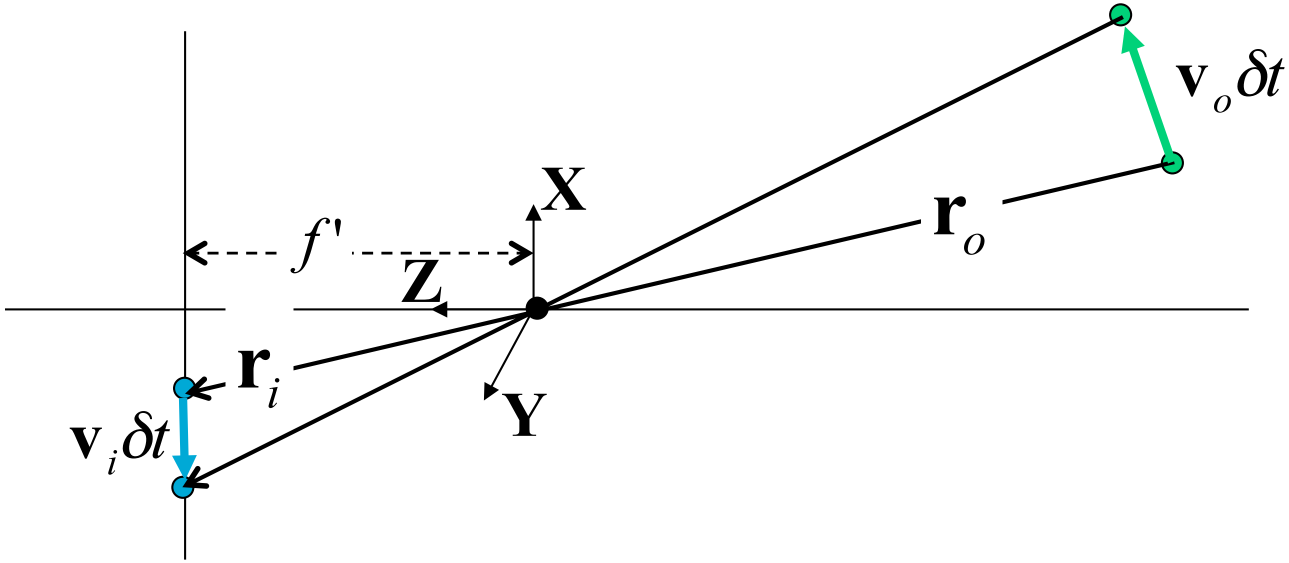


Motion Field

- The projection of the motion of the points in the scene.
- It is a collection of 2D vectors, each vector being the projection of the 3D velocity of a scene point on the image plane.
- It is a 2D array of 2D vectors representing the motion in 3D.
- It is induced by the relative motion between the viewing camera and the observed scene.

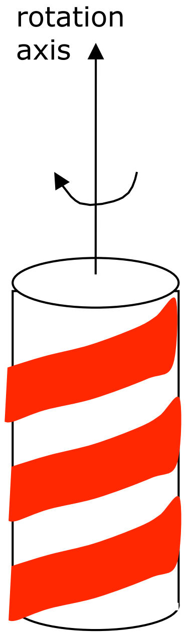
Motion Field

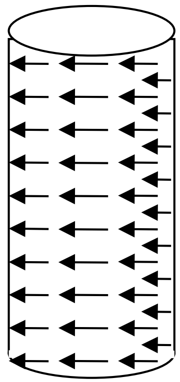
- Image velocity of a point moving in the scene and its projection on the image plane



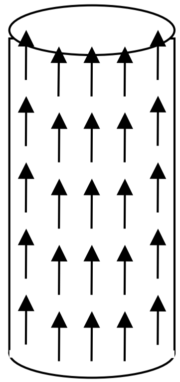
Optical Flow ≠ Motion Field

Barber's Pole Illusion





Motion Field



Optical Flow

Barber's pole

Velocity Basics

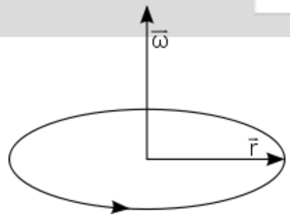
- For motion on a straight line, the velocity is simply distance traveled per unit time:

$$\mathbf{v} = {d\mathbf{s}}/{dt} = \left({dx}/{dt}, {dy}/{dt} \right)$$

- If a point is moving on a circle (consider for example a nail stuck on a wheel), then the best way to describe its speed, is by how many degrees in travels per unit time, i.e. its angular velocity:

$$\omega = {d\vartheta}/{dt}$$

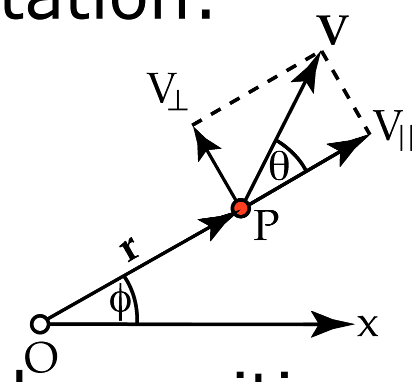
Angular Velocity



- In 3D angular velocity is a pseudo-vector.
- It now has not only a magnitude, but also a direction.
- The magnitude is the angular speed, $|\vec{\omega}| = |\vec{r}||\vec{v}|\sin\theta$

and the direction describes the axis of rotation:

$$\vec{\omega} = \frac{(\vec{r} \times \vec{v})}{|\vec{r}|^2} = \frac{|\vec{v}|\sin\theta}{|\vec{r}|}\vec{n}$$



where $\vec{r}$ is the linear vector connecting the position of the particle with the origin of the rotation, $\vec{v}$ is the linear momentum vector and $\vec{n}$ is a vector parallel to the axis of rotation.

Motion Field Basics

- Let *P=(X,Y,Z)* point in scene and *p=(x,y,f)* its projection.

$$p = P(f/Z) \qquad (1)$$

- Assume that *P* moved relative to the camera in such a way that both pure translation as well as rotation may be involved.
- The relative motion between the point *P* and the camera can be described as:

$$\vec{V} = -\vec{T} - \vec{\omega} \times \vec{P} \qquad (2)$$

where $\vec{T}$ is the pure translation part of the motion of *P* and $\vec{\omega}$ is the angular velocity.

- Then:

$$\begin{aligned} V_x &= -T_x - \omega_y Z + \omega_z Y \\ V_y &= -T_y - \omega_z X + \omega_x Z \\ V_z &= -T_z - \omega_x Y + \omega_y X \end{aligned} \qquad (3)$$

Motion Field Basics 2

- The motion field is the projection of the *3D* motion of *P* on the image plane. The same projective relationship $p = P(f/Z)$ applies for the velocities too. So, by taking the time derivative of eq. (1)

$$\vec{v} = f\left(\frac{Z\vec{V} - V_z\vec{P}}{Z^2}\right) \qquad (4)$$

- By combining equations (3) and (4):

$$v_x = \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}$$
$$v_y = \frac{T_z y - T_y f}{Z} + \omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}$$

Motion Field Basics 3

- The translational components of the motion field are:

$$^{T}v_x = \frac{T_z x - T_x f}{Z}$$

$$^{T}v_y = \frac{T_z y - T_y f}{Z}$$

- The rotational components of the motion field are:

$$^{\omega}v_x = -\omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}$$

$$^{\omega}v_y = +\omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}$$

- Note that the rotational component of the motion field does not convey any information about depth.

Pure Translation

- In the case of pure translation we have:

$$v_x = \frac{T_z x - T_x f}{Z}$$
$$v_y = \frac{T_z y - T_y f}{Z} \qquad (5)$$

- Consider first the case where there is a change in depth also, i.e. $T_z \neq 0$. Let us define a point $p_0=(x_0, y_0)$ such that:

$$x_0 = f\frac{T_x}{T_z} \Rightarrow T_x f = x_0 T_z$$
$$y_0 = f\frac{T_y}{T_z} \Rightarrow T_y f = y_0 T_z \qquad (6)$$

Pure Translation 2

- By combining eqs. (5) and (6):

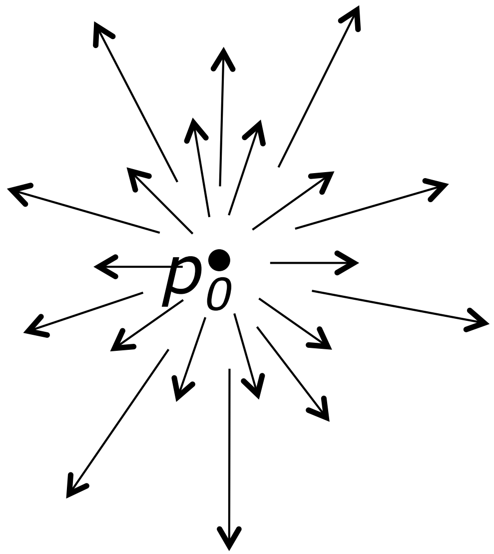
$$v_x = (x - x_0)\frac{T_z}{Z}$$

$$v_y = (y - y_0)\frac{T_z}{Z}$$

- This shows that the length of *v(p)* is proportional to the distance between *p* and p_0 and inversely proportional to the depth of the 3D point *P*.
- The motion field of a pure translation when there is a change in depth is radial, i.e. all vectors emanate/radiate from a common origin, the point p_0, which is known as the *vanishing point* of the translation direction. It is the intersection of the ray parallel to the translation vector with the image plane.

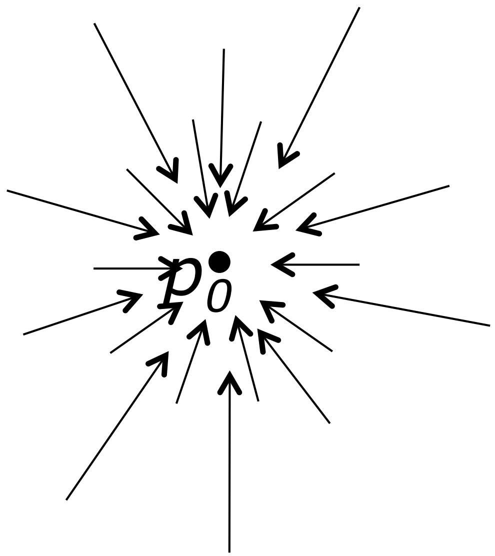
Focus of Expansion

- If $T_z < 0$ (i.e. *Z* is decreasing, object moves towards the camera) the vectors point away from p_0 and p_0 is the *focus of expansion*.



Focus of Contraction

- If $T_z > 0$ (i.e. *Z* is increasing, object moves away from the camera) the vectors point away towards p_0 and p_0 is the *focus of contraction*.



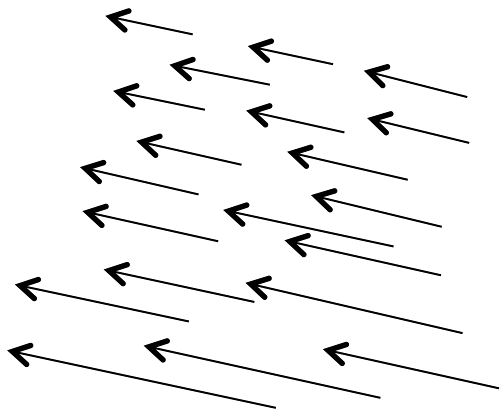
Parallel Motion Field

- In the special case that $T_z = 0$ eq. (5) becomes

$$v_x = -T_x \left(\frac{f}{Z} \right)$$

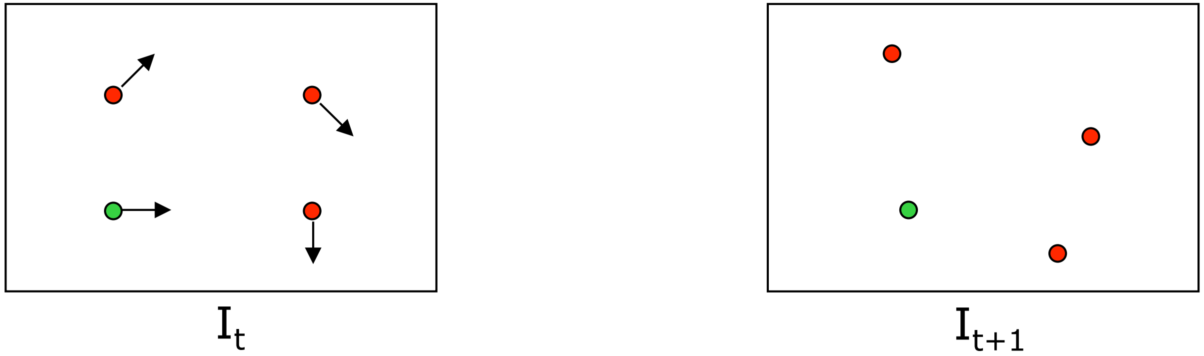
$$v_y = -T_y \left(\frac{f}{Z} \right)$$

- All the motion field vectors are parallel to each other.
- The length of v(p) is inversely proportional to the depth of the 3D point P.



Optical Flow Estimation

- We compute the optical flow and we assume that it is almost equivalent to the motion field



- How to estimate pixel motion from image I_t to image I_{t+1}?
  - Find pixel correspondences: Given a pixel in I_t, look for nearby pixels of the same appearance (e.g. color) in I_{t+1}.
- There are 2 main strategies for computing the Optical Flow:
  - Differential Methods: motion is computed at every pixel; these techniques are based on time derivatives and thus require small δt.
  - Matching/Prediction Methods: motion is estimated only on selected features; these methods make predictions about possible positions in the next frame.

Assumptions

1. Assumption 1: The image brightness is continuous and differentiable. (This is a key assumption in differential methods).
2. Assumption 2: The image brightness value (more properly the image irradiance *E*) of objects doesn't change over δt, in other words,

$$\frac{dE}{dt} = 0$$

   This last assumption is known as the **image brightness constancy assumption**.
3. Assumption 3: Points do not move very far. It is also known as the **small motion assumption**.

Differential Method

- For each image point *(x,y)* at time *t* we have a value *E(x(t),y(t),t)*, so (by the chain rule):

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

v_x v_y

Gradient-based edge detector

- Thus, this last equation can be written more compactly as:

$$\frac{dE}{dt} = G_x v_x + G_y v_y + E_t = 0$$

Differential Method 2

- In vector form we have:

$$\vec{G}^T \vec{v} + E_t = 0$$

Image Brightness Constancy Equation

- We can compute ***G*** and E_t . Can we then directly estimate the motion field ***v***?

$$\vec{G}^T \vec{v} + E_t = 0$$

$$\vec{G}^T \vec{v} = -E_t$$

$$\frac{\vec{G}^T \vec{v}}{\|G\|} = -\frac{E_t}{\|G\|}$$

Differential Method 3

- We can compute $$\frac{\vec{G}^T\vec{v}}{\|G\|} = -\frac{E_t}{\|G\|}$$
- But this is not the motion field. Rather, what we compute is: $$\hat{v}_n = \frac{\vec{G}^T\vec{v}}{\|G\|}$$

  which is the component of the motion field ***v*** in the direction of the spatial image gradient.
- So with the *Image Brightness Constancy Equation*, there is only sufficient information to determine the velocity in the direction parallel to the image gradient.

Error Analysis

- Besides this limitation, how accurate is the estimate that we get?
- Let Δv be the difference between the true v_n and the one estimated through the image's optical flow.

$$|\Delta v| = |v_n - \hat{v}_n|$$

- Let's use information from the image formation process.
- Additional Assumption: Lambertian Surface

$$E = \rho \vec{L}^T \vec{n}$$

where ρ is the albedo, ***L*** the direction and intenisty of illumination and ***n*** the surface normal.

Error Analysis - continued

- Under the Lambertian assumption

$$dE/dt = \rho \vec{L}^T \left(d\vec{n}/dt \right)$$

- If we assume distant light sources and a distant camera position, then only a rotation will cause a change in image irradiance, *E*.

$$dE/dt = \rho \vec{L}^T \left(\vec{\omega} \times \vec{n} \right)$$

- By incorporating the previous equations:

$$\vec{G}^T \vec{v} + E_t = \rho \vec{L}^T \left(\vec{\omega} \times \vec{n} \right)$$

$$\frac{\vec{G}^T \vec{v} + E_t}{\left\| \vec{G} \right\|} = \frac{\rho \vec{L}^T \left(\vec{\omega} \times \vec{n} \right)}{\left\| \vec{G} \right\|}$$

Error Analysis - continued

- We estimate: $\hat{v}_n = -\dfrac{E_t}{\|G\|}$

- So the difference between what we measure and the true v_n is:

$$|\Delta v| = \rho \left| \frac{\vec{L}^T (\vec{\omega} \times \vec{n})}{\|\vec{G}\|} \right|$$

- This means that $|\Delta v| = 0$ only:
  - under pure translation or
  - under rigid motion where the illuminant direction is parallel to ω.

- Δv decreases as the magnitude of ***G*** increases.

Implementation of the Differential Method

- There exist a large number of differential techniques:
  - Iteratively solve for the image brightness constancy equation.
  - Solve a system of partial differential equations (sometimes iteratively).
  - Use 2nd or higher order derivatives of image brightness, *E*.
  - Use a least squares method.
- We will focus on the Least Squares Method. It tends to be more stable (Iterative methods may converge to the wrong solution and are sensitive to discontinuities; Higher order derivatives are noisy due to the approximations used in computing them).

Least Squares Method

- Assume that over a small NxN patch Q, i.e. 5x5 region, all the pixels move with the same velocity.

1. Compute the spatial and temporal derivatives, i.e. $\boldsymbol{G}$ and E_t for each of the N^2 pixels.

   E_t is a derivative over time, so one can use the same approximations as in edge detection, but over the time domain. For example, once can use Sobel

$$H_t = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

   but this time the horizontal axis it t.

Least Squares Method - continued

2. We want to find a value v that keeps $\vec{G}^T \vec{v} + E_t$ close to 0 for all the N^2 pixels.

   Minimize the functional: $f[\vec{v}] = \sum_{p \in Q} \left(\vec{G}^T \vec{v} + E_t \right)^2$

   One way to do this is by solving an over-constrained linear system:

$$A^T A v = A^T b \Rightarrow v = (A^T A)^{-1} A^T b$$

$$A = \begin{bmatrix} \vec{G}(p_1) \\ \vec{G}(p_2) \\ \vdots \\ \vec{G}(p_{N^2}) \end{bmatrix}$$

A is an N^{2}x2 matrix

$$b = -\begin{bmatrix} E_t(p_1) \\ E_t(p_2) \\ \vdots \\ E_t(p_{N^2}) \end{bmatrix}$$

b is an N^2 vector

v is the optical flow at the center of the NxN patch Q.

Least Squares Algorithm

1. Smooth spatially with a Gaussian of $\sigma = 1.5$
2. Smooth temporally with a Gaussian of $\sigma = 1.5$
3. Perform edge detection in the spatial domain. In other words, compute the spatial gradient $\boldsymbol{G}$.
4. Perform edge detection in the temporal domain. In other words, compute the time derivative E_t.
5. For each patch Q
   - Construct A and b
   - Compute $\boldsymbol{v}$

Weighted Least Squares

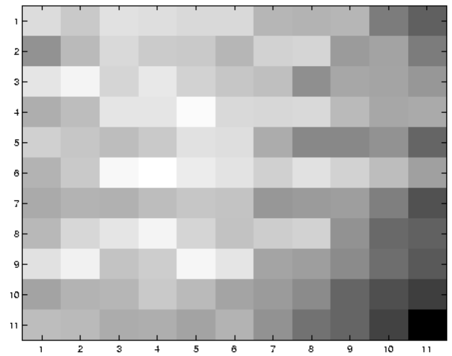
- There is an expected error in ***v*** as we incorporate spatial and temporal derivatives from pixels farther away from the center of the patch Q.
- Solution: use a weighted least squares method.

$$v = (A^T W A)^{-1} A^T W b$$

- W is a weight matrix where the weight decreases with distance from the center of the patch Q.
- It is an $N^2 x N^2$ diagonal matrix, where $W_{ii} = \frac{1}{d(p_i, c)}$

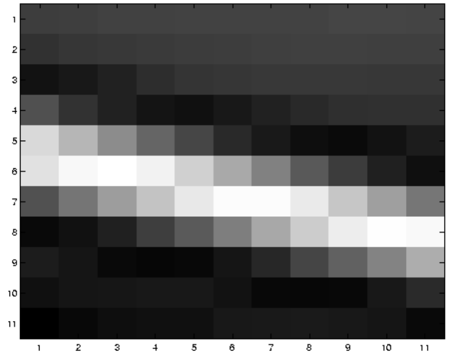
  where *c* is the location of the center of the patch Q and p_i is the location of a pixel in the patch Q.

Low Texture Region - Bad



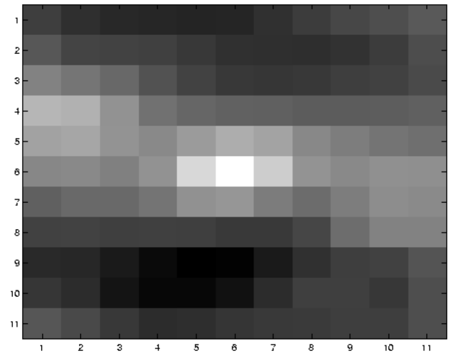
– gradients have small magnitude

Edges Can Be Problematic – Aperture Problem



– large gradients, but all the same
– could cause "limited-aperture" inaccuracies

High Textured Region - Good



– gradients are different, large magnitudes

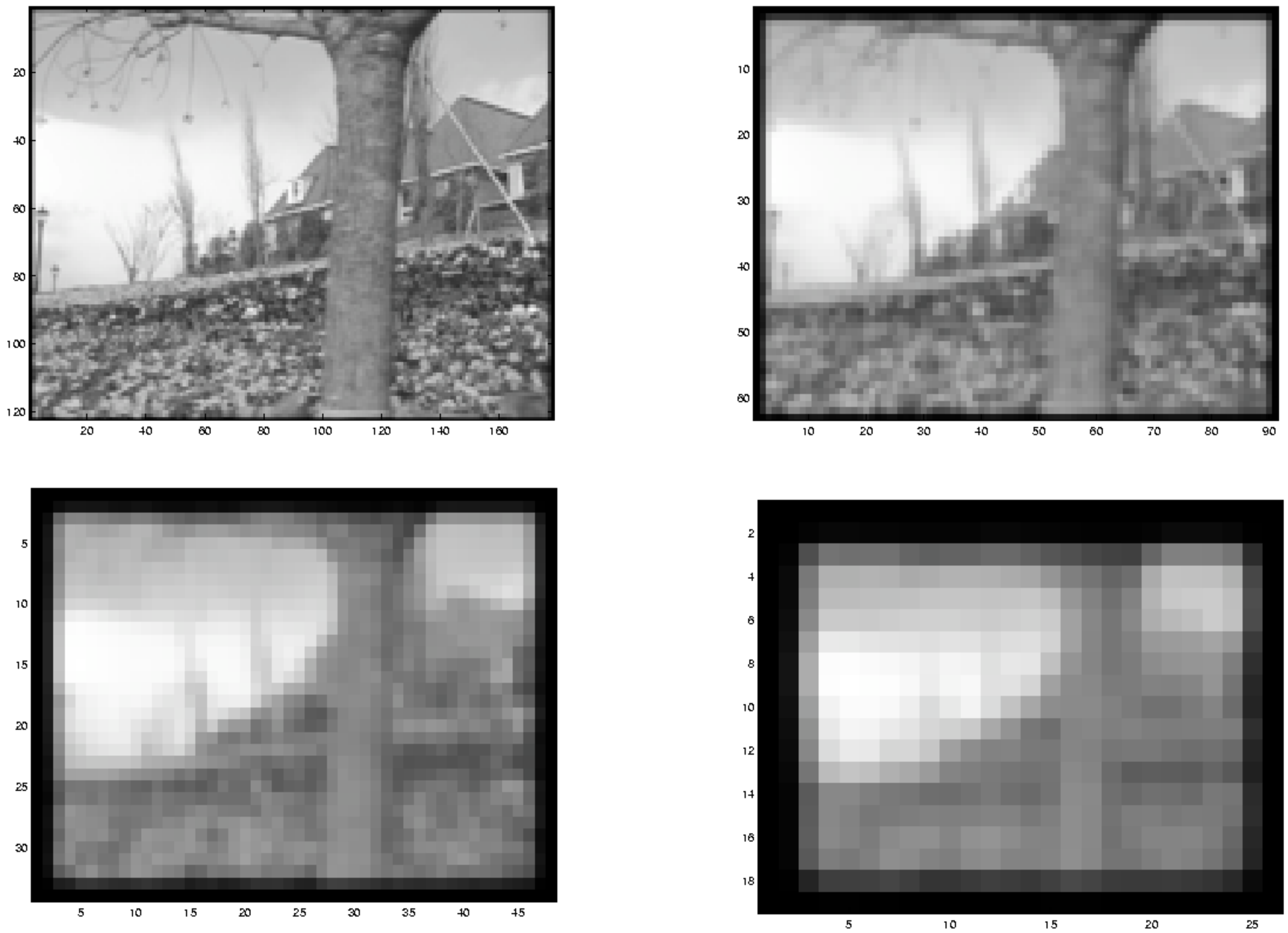
Small Motion Assumption

- Is such a motion small enough?

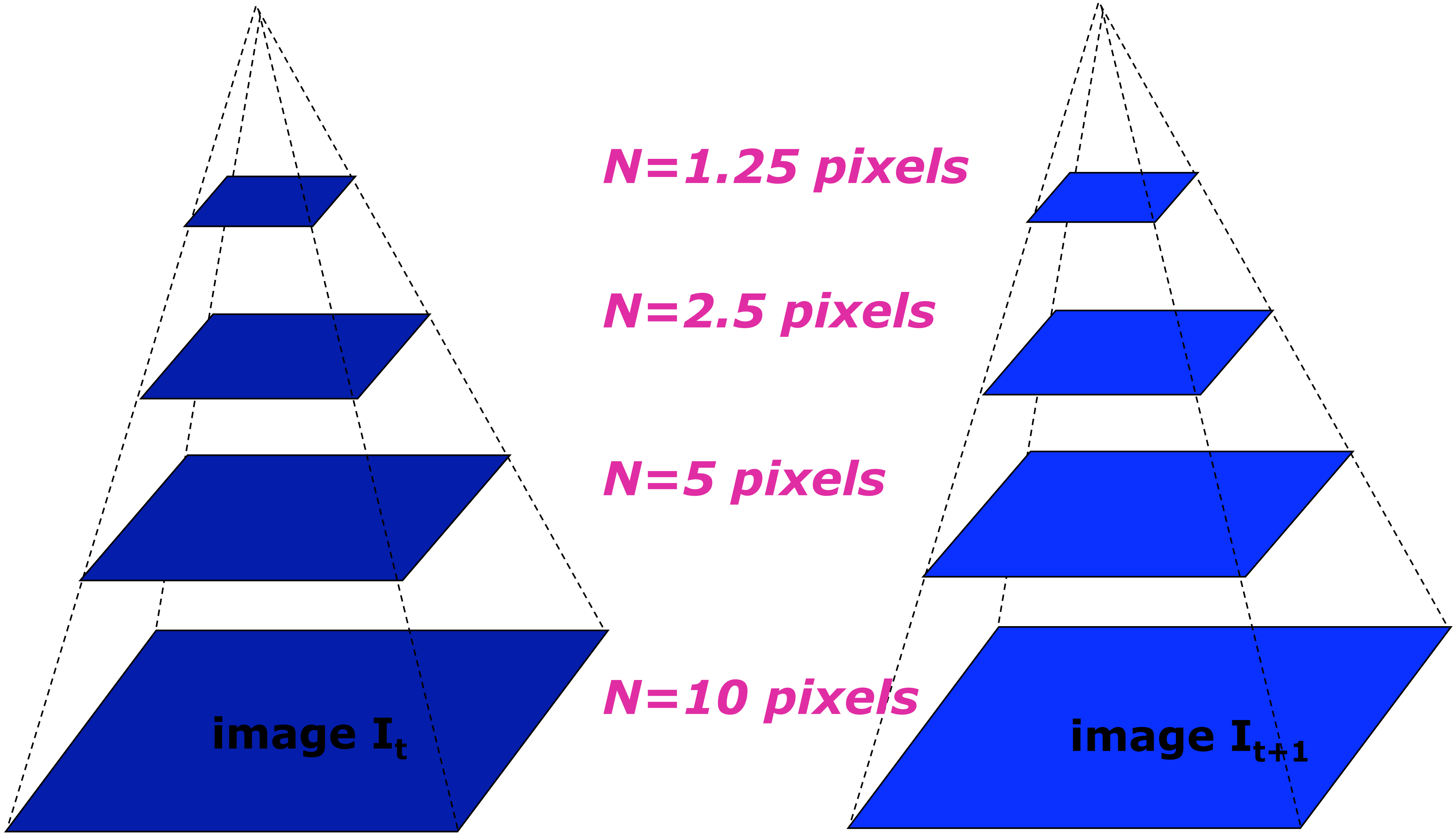
Small Motion Assumption

- Is such a motion small enough?
  - Probably not—it's much larger than one pixel
  - How might we solve this problem?

Reduce the Resolution

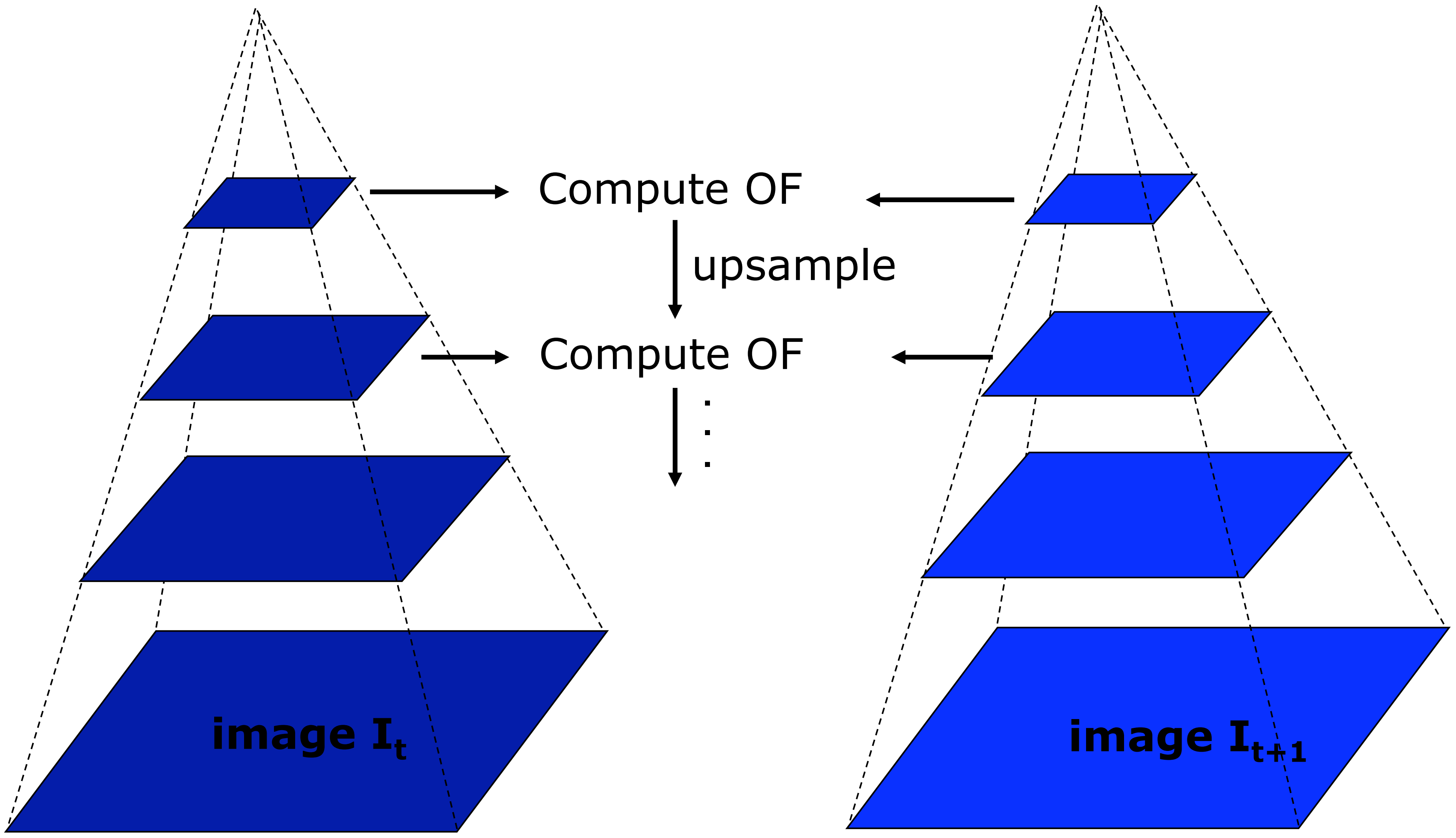


Coarse to Fine Estimation



Gaussian pyramid of image I_t Gaussian pyramid of image I_{t+1}

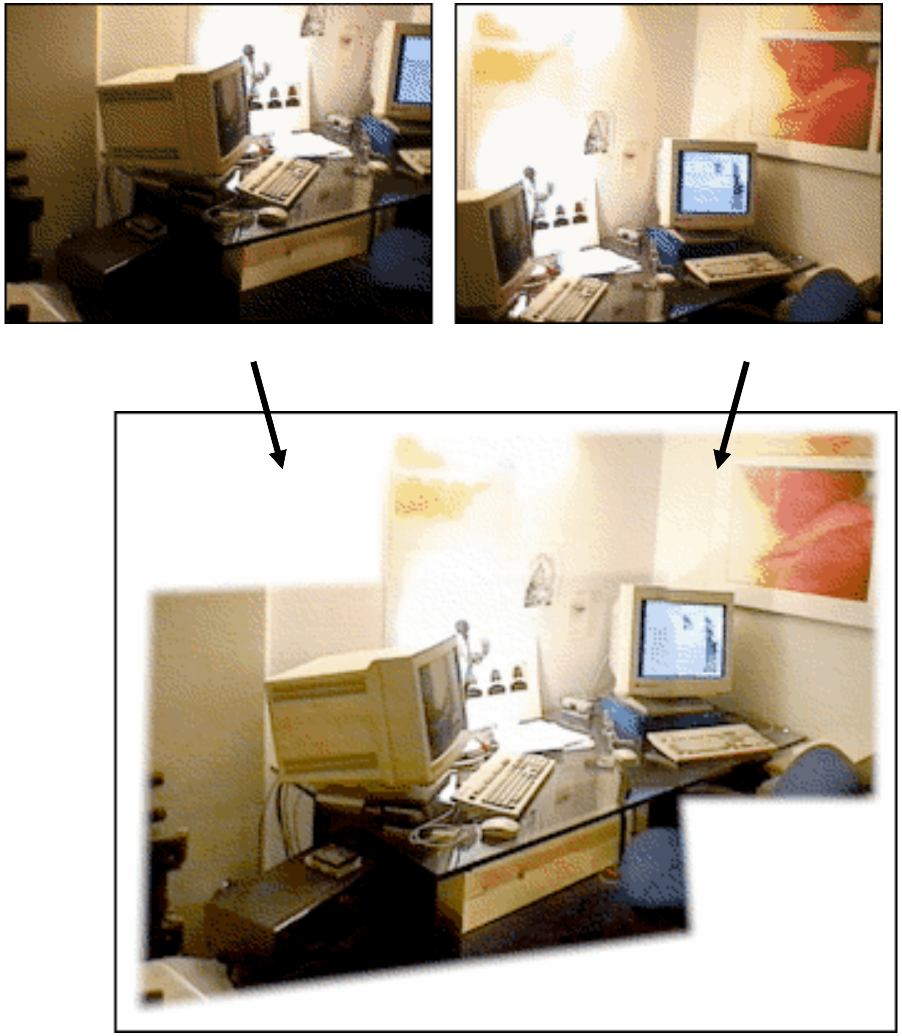
Coarse to Fine Computation



Gaussian pyramid of image I_t

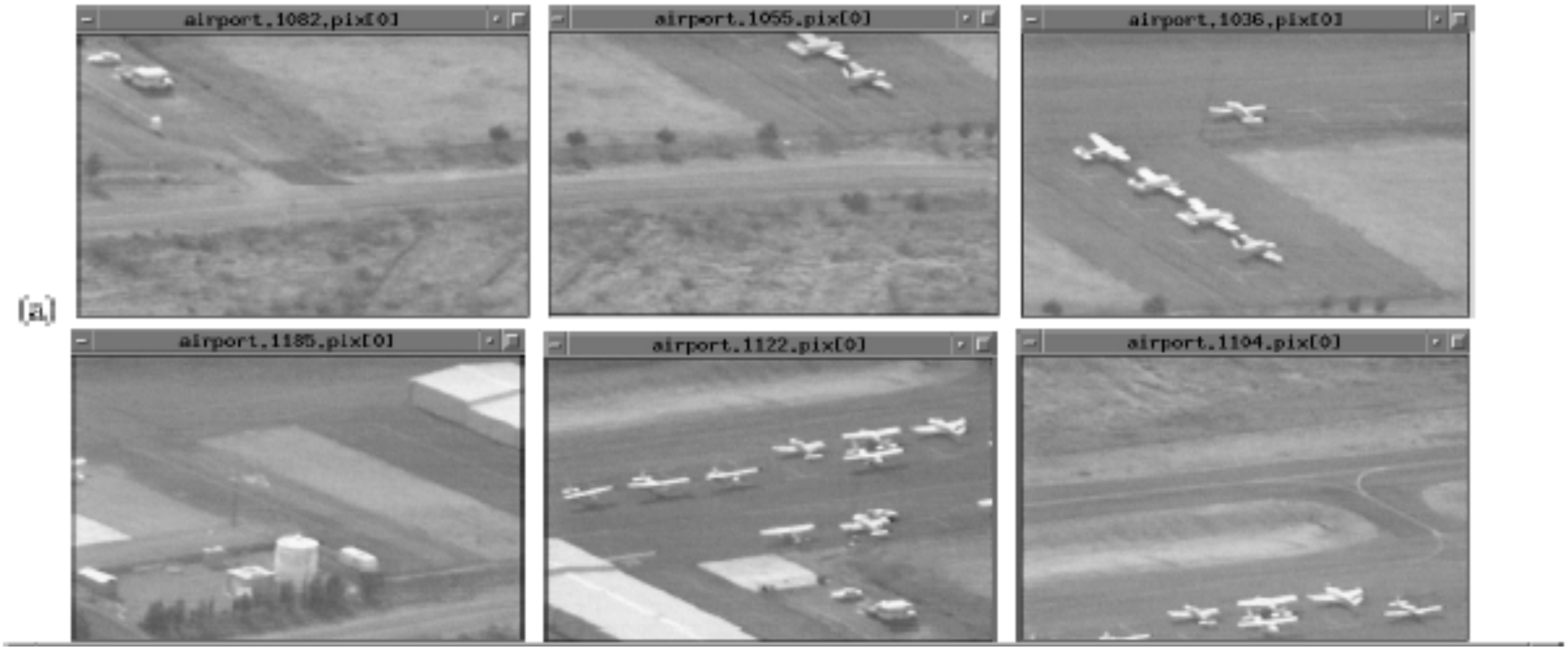
Gaussian pyramid of image I_{t+1}

Image Alignment



- Goal: Estimate a *single* **v** translation (transformation) for the entire image.
- The entire image has the same translation value so the optical flow values for every pixel is the same.
- This is typically an easier problem than general motion estimation.
- We can compute it very well with pyramid-based methods like the Lucas-Kanade one.

Mosaicing – input images



Mosaicing – Final Result

b)

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

Image Sources

1. The car tracking example is courtesy of S. Baker, http://www.ri.cmu.edu/research_project_detail.html?project_id=513&menu_id=261
2. The American football tracking sequence is courtesy of D. Comaniciu, http://comaniciu.net/
3. The face tracking example is courtesy of S. Baker, http://www.ri.cmu.edu/research_project_detail.html?project_id=448&menu_id=261
4. The Structure-from-Motion example is courtesy of D. Nister, http://www.vis.uky.edu/~dnister/Research/research.html
5. The behavior analysis example is courtesy of M. Irani http://www.wisdom.weizmann.ac.il/~vision/BehaviorCorrelation.html
6. The background subtraction figure is courtesy of D. Parks, http://dparks.wikidot.com/background-subtraction
7. The spinning barber's pole is from Wikipedia http://en.wikipedia.org/wiki/Barber's_pole
8. The figures on angular velocity are from Wikipedia http://en.wikipedia.org/wiki/Angular_velocity
9. The mosaicing example is courtesy of M. Irani http://www.wisdom.weizmann.ac.il/~vision/
10. A number of slides in this presentation have been adapted by the presentation of S. Narasimhan, http://ww.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/lec-16.ppt