Optical Flow II

Guido Gerig

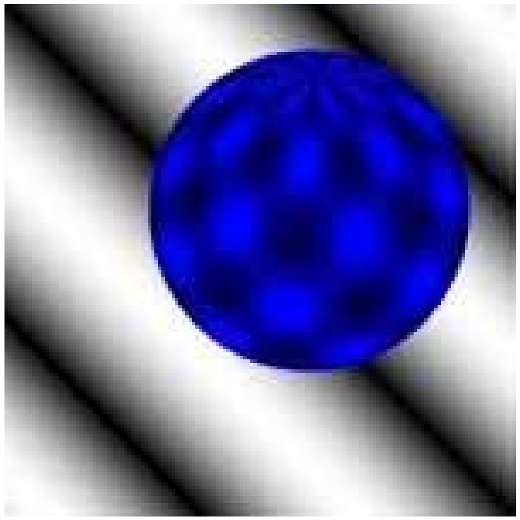
CS 6320, Spring 2015

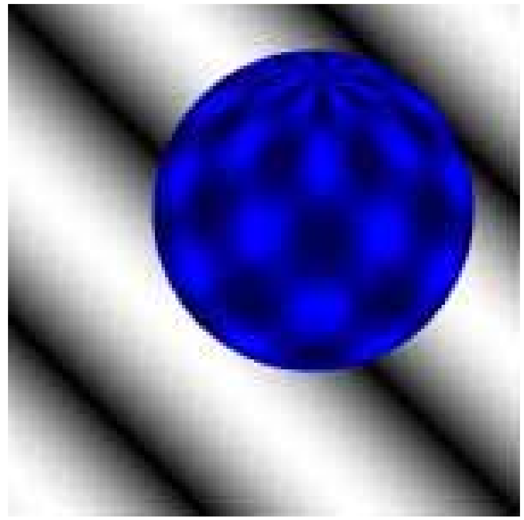
(credits:Pollefeys Comp 256, UNC, Trucco & Verri, Chapter 8, R. Szelisky, CS 223 Fall 2005)

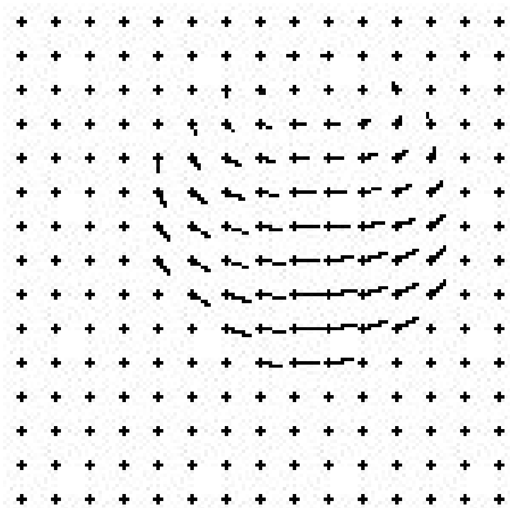
Material

- R. Szelisky Computer Vision: Chapter 7.1-7.2, Chapter 8
- Trucco & Verri Chapter 8 (handout, pdf)
- Hand-written notes G. Gerig (pdf)
- Horn & Schunck Chapter 9
- Pollefeys CV course (ETH/UNC)
- Richard Szeliski, CS223B Fall 2005

Structure from Motion?

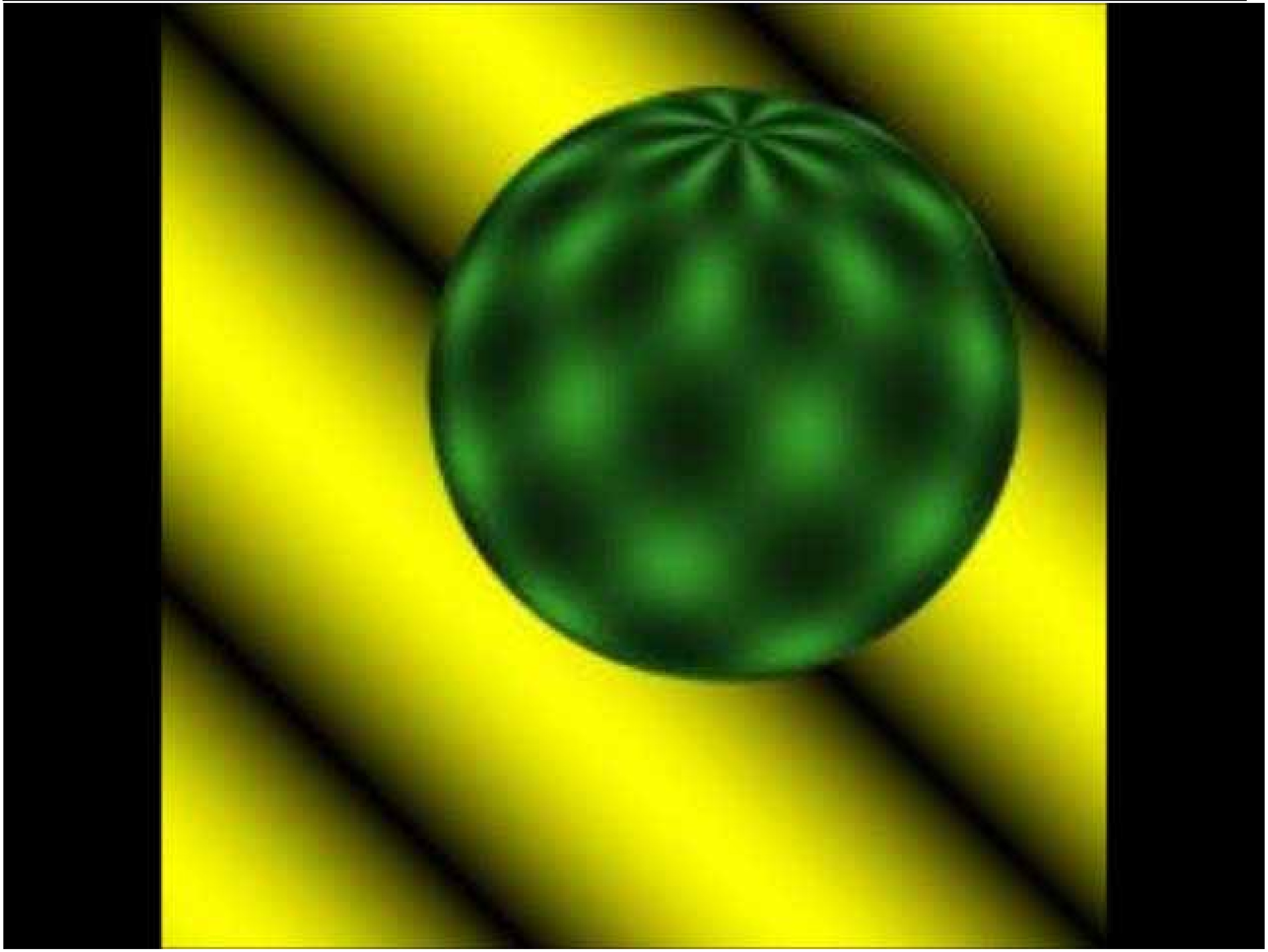






- Known: optical flow (instantaneous velocity)
- Motion of camera / object?

Structure from Motion?

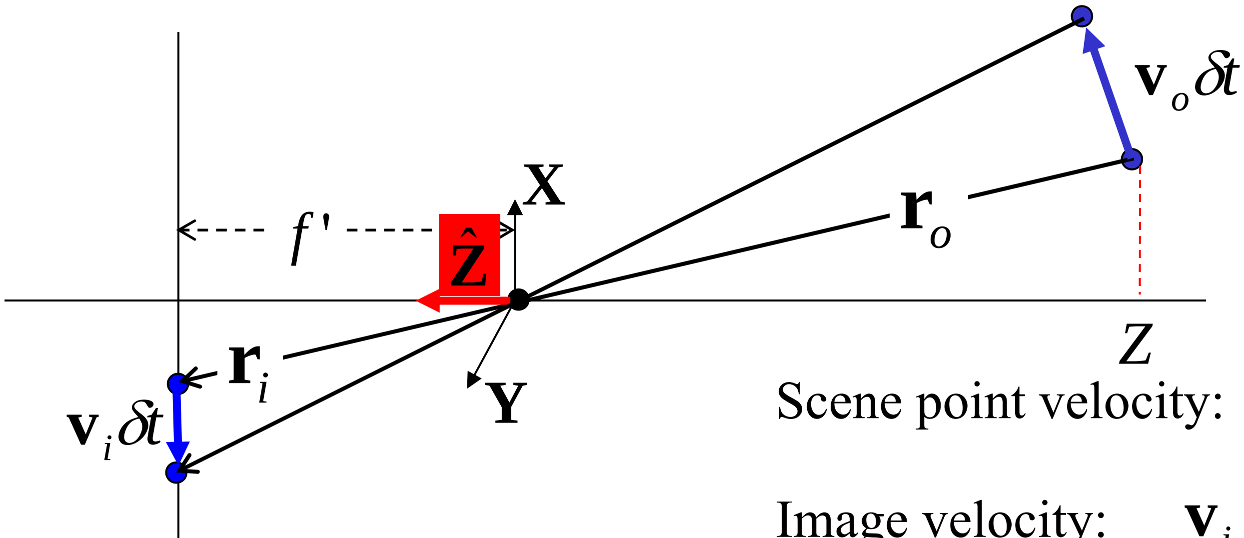


Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

Motion Field: Perspective Projection

Image velocity of a point moving in the scene



Scene point velocity: $\mathbf{v}_o = \dfrac{d\mathbf{r}_o}{dt}$

Image velocity: $\mathbf{v}_i = \dfrac{d\mathbf{r}_i}{dt}$

Perspective projection: $\dfrac{1}{f'}\mathbf{r}_i = \dfrac{\mathbf{r}_o}{\mathbf{r}_o \cdot \hat{\mathbf{Z}}} = \dfrac{\mathbf{r}_o}{Z}$

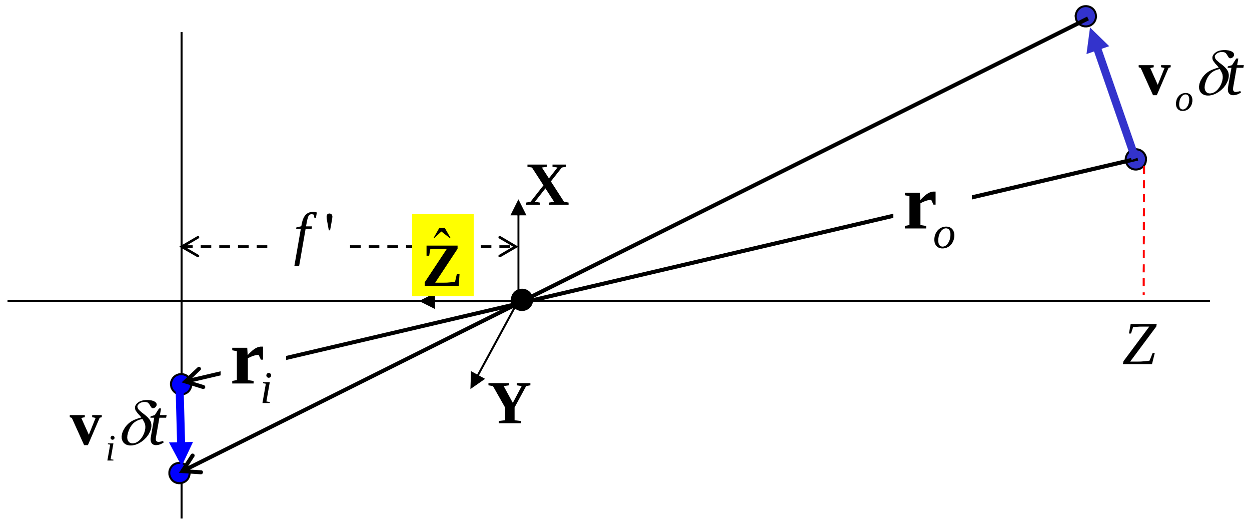
Derivation
(notes GG)

Motion field

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

Motion Field: Perspective Projection

Image velocity of a point moving in the scene



Motion field

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

Discussion: $\mathbf{v}_i$ is orthogonal to $(\mathbf{r}_o\times\mathbf{v}_o)$ and $\hat{Z}$ $\rightarrow$ lies in image plane

Motion Field: Perspective Projection

Motion field

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

Set $\widehat{Z} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix}$ and do the math (see handwritten notes G. Gerig):

$$v_{ix} = \frac{v_{ox} f}{Z} - \frac{x v_{oz}}{Z}$$

$$v_{iy} = \frac{v_{oy} f}{Z} - \frac{y v_{oz}}{Z}$$

Motion Field: Perspective Projection

$$v_{ix} = \frac{v_{ox} f}{Z} - \frac{x v_{oz}}{Z}$$

$$v_{iy} = \frac{v_{oy} f}{Z} - \frac{y v_{oz}}{Z}$$

Discussion:

Component of optical flow in image only due to v_x and v_y , object motion parallel to image plane.

Component of optical flow in image only due to v_z , object motion towards/away from camera.

Motion Field: Perspective Projection

$$v_{ix} = \frac{v_{ox} f}{Z} - \frac{x v_{oz}}{Z}$$

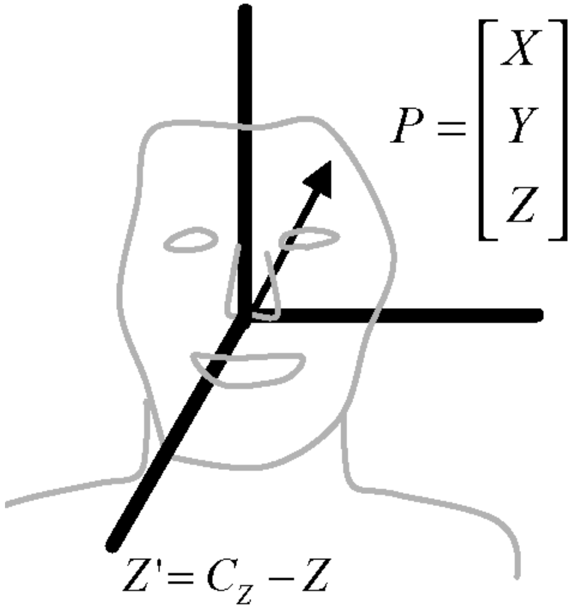
$$v_{iy} = \frac{v_{oy} f}{Z} - \frac{y v_{oz}}{Z}$$

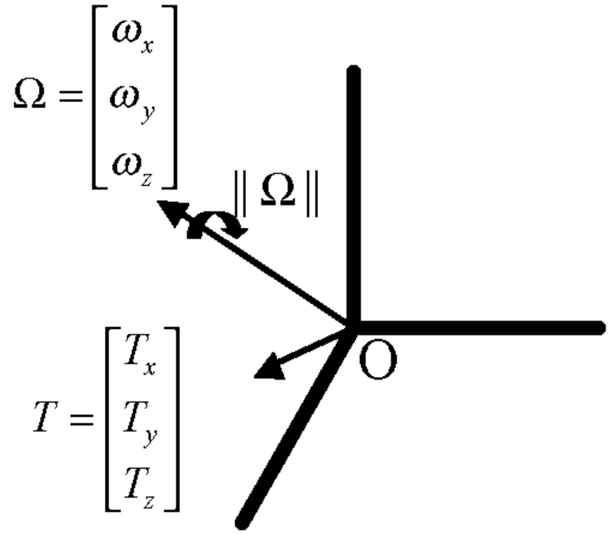
Reformulate: perspective projection of velocity:

$$\begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z} \begin{bmatrix} v_{ox} \\ v_{oy} \\ v_{oz} \end{bmatrix}$$

Rigid pose estimation

- Head pose model: 6 DOF





Please note notation: T stands for translational motion of object, Ω for rotational component.

Optic flow for rigid motion

- 3-D velocity:

$$V = T + \Omega \times P = T - \hat{\mathbf{P}}\Omega = \begin{bmatrix} \mathbf{I} & -\hat{\mathbf{P}} \end{bmatrix} \begin{bmatrix} T \\ \Omega \end{bmatrix}$$

$$\hat{\mathbf{P}} = [\mathbf{P}_x]$$
(skew-sym.)

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix} \begin{bmatrix} T \\ \Omega \end{bmatrix}$$

Optic flow for rigid motion

- Perspective projection

$$\begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z} \begin{bmatrix} v_{ox} \\ v_{oy} \\ v_{oz} \end{bmatrix}$$

Optic flow for rigid motion

- Combine

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix} \begin{bmatrix} T \\ \Omega \end{bmatrix}$$

$$\begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z} \begin{bmatrix} v_{ox} \\ v_{oy} \\ v_{oz} \end{bmatrix}$$

Optic flow for rigid motion

- Rigid Motion (for small v): $\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{H} \begin{bmatrix} T \\ \Omega \end{bmatrix}$

$$\mathbf{H} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix}$$

Perspective projection of 3-D velocity *

3-D velocity at point P (from $\begin{bmatrix} T \\ \Omega \end{bmatrix}$)

Hard to solve with just optic flow vectors! (but see Horn 17.3-17.5).

* Convert from scene to image: $\bar{p} = f\,\frac{\bar{P}}{Z}$

Optic Flow for rigid motion

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{H} \begin{bmatrix} T \\ \Omega \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix}$$

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

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

Flow field of rigid motion

In components, and using (8.5), (8.6) read

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

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

Notice that *the motion field is the sum of two components, one of which depends translation only, the other on rotation only*. In particular, the translational compone of the motion field are

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

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

and the rotational components are

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

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

Trucco & Verri p. 184

Flow field of rigid motion

Notice that *the motion field is the sum of two components, one of which depends translation only, the other on rotation only*. In particular, the translational compone of the motion field are

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

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

and the rotational components are

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

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

Discussion:

- Motion field of translational component depends on T and depth Z. For increasing Z, velocity becomes smaller.
- Motion field that depends on angular velocity does NOT carry information on depth Z!

Trucco & Verri p. 184

Special Case: Pure Translation

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

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

Choose x_0 and y_0 so that v becomes 0

$$x_0 = f T_x / T_z$$

$$y_0 = f T_y / T_z,$$

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

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

Says that motion field of a pure translation is radial, it consists of vectors radiating from a common origin $p_0=(x_0,y_0)$, which is the vanishing point.

Trucco & Verri p. 184/185

See also F&P Chapter 10.1.3 p. 218

Special Case: Pure Translation

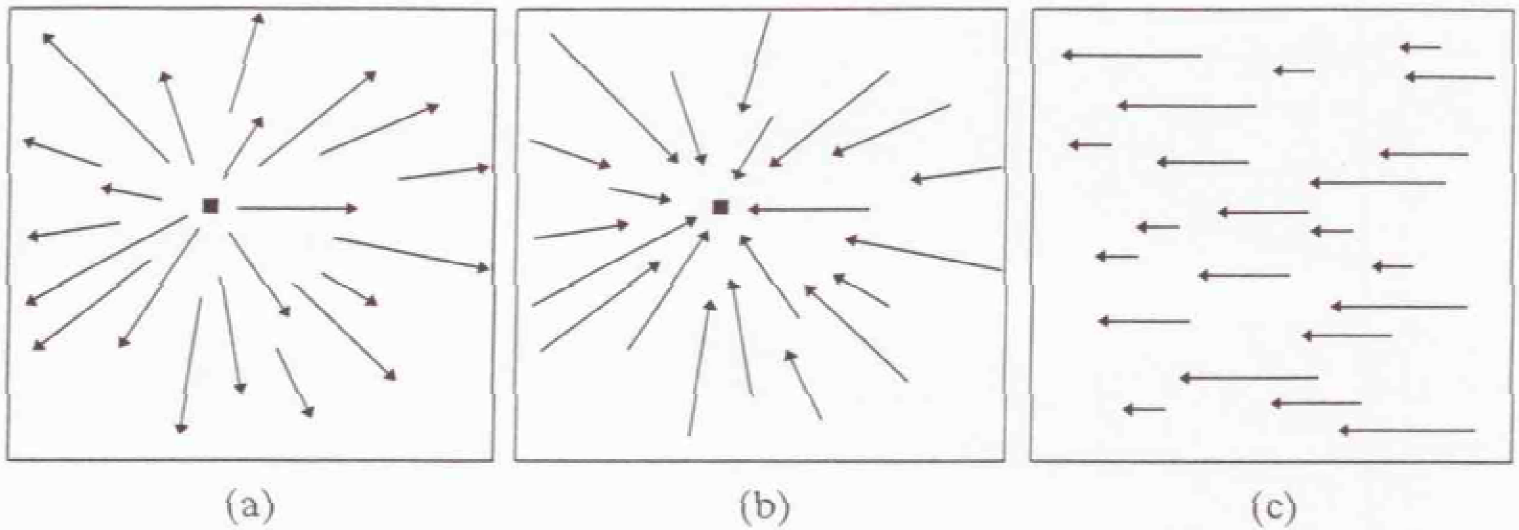


Figure 8.4 The three types of motion field generated by translational motion. The filled square marks the instantaneous epipole.

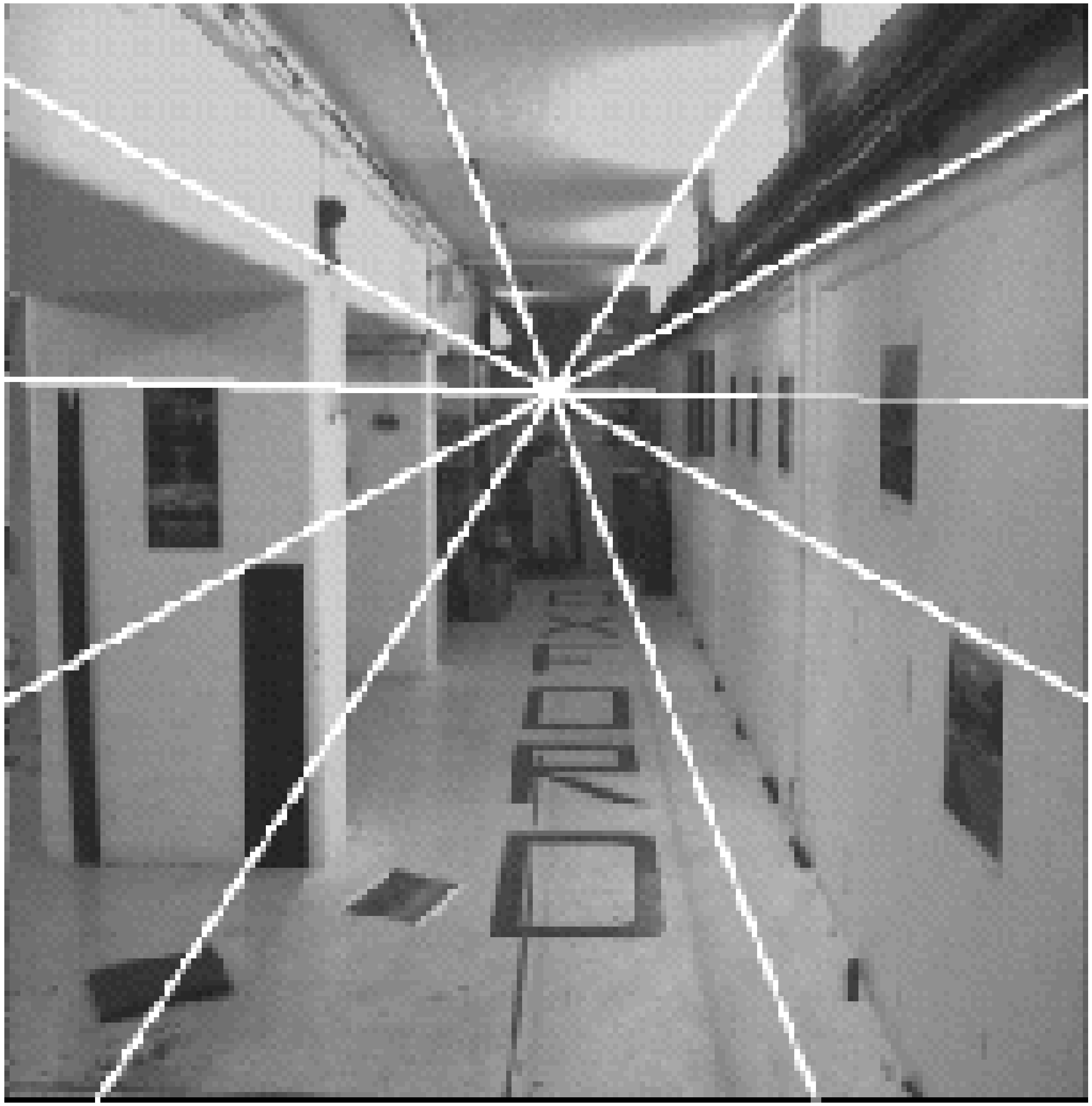
Focus of expansion/contraction:

$$x_0 = fT_x/T_z$$

$$y_0 = fT_y/T_z,$$

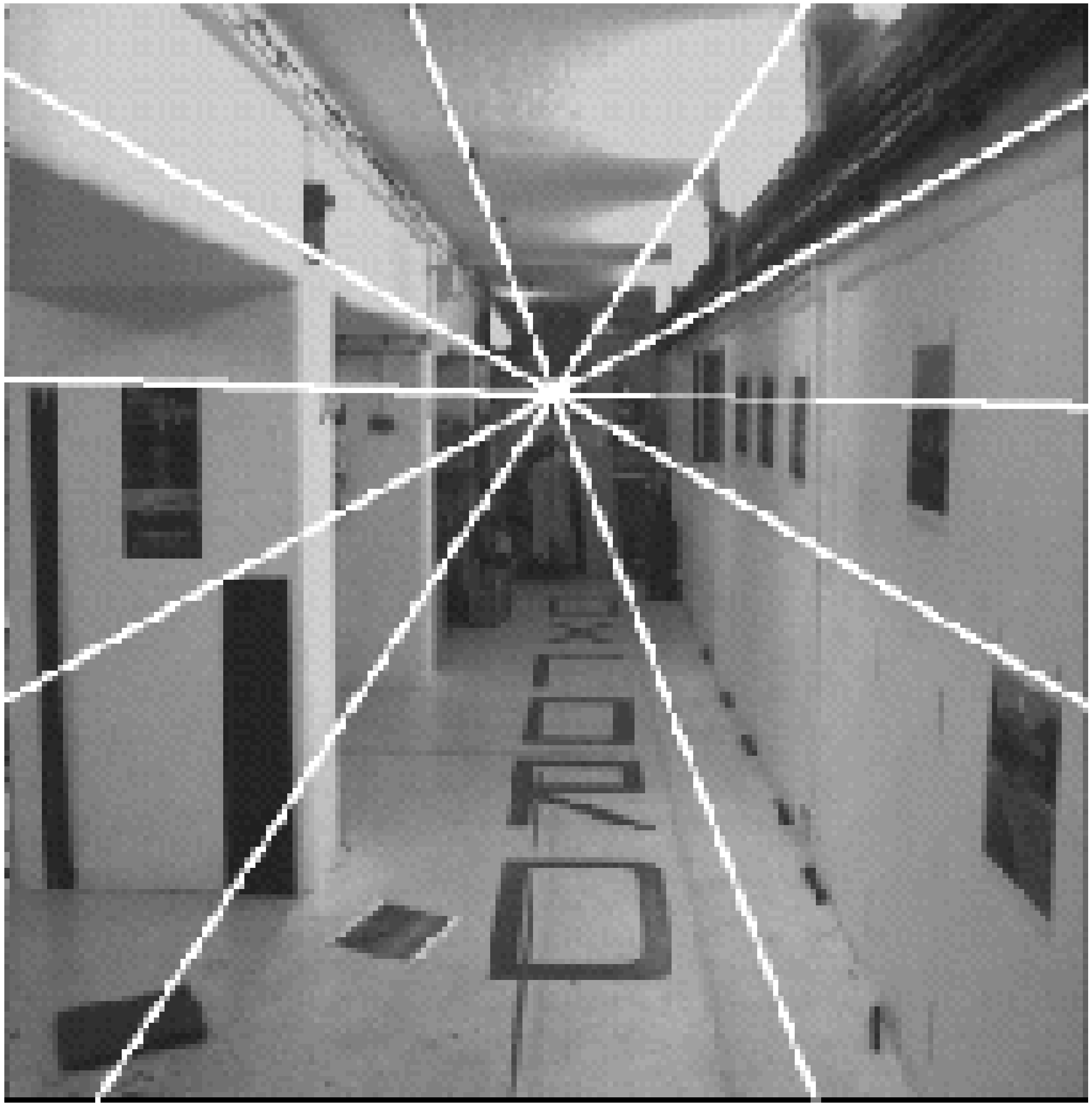
Trucco & Verri p. 184/185
See also F&P Chapter 10.1.3 p. 218

Example: forward motion



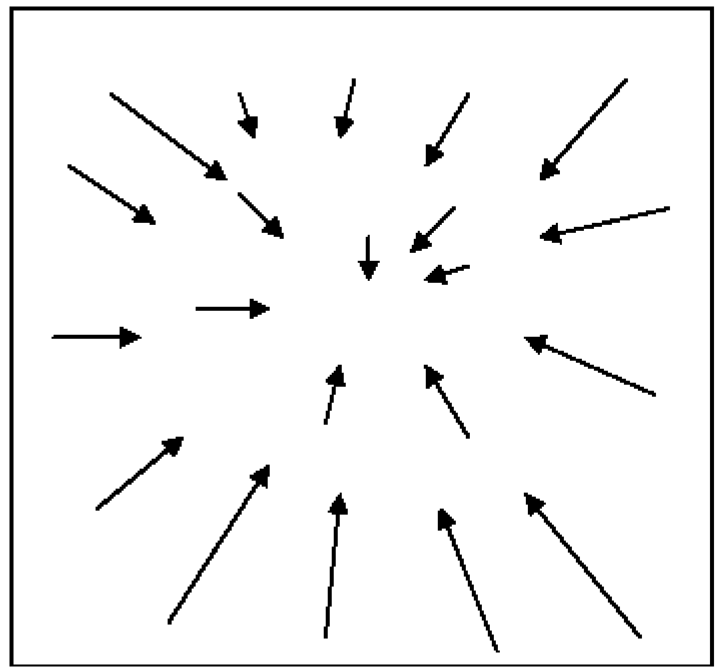
courtesy of Andrew Zisserman

Example: forward motion



courtesy of Andrew Zisserman

FOE for Translating Camera



Moving Plane (Trucco&Verri p.187)

$$v_x = \frac{1}{fd}(a_1 x^2 + a_2 xy + a_3 fx + a_4 fy + a_5 f^2)$$

$$v_y = \frac{1}{fd}(a_1 xy + a_2 y^2 + a_6 fy + a_7 fx + a_8 f^2)$$

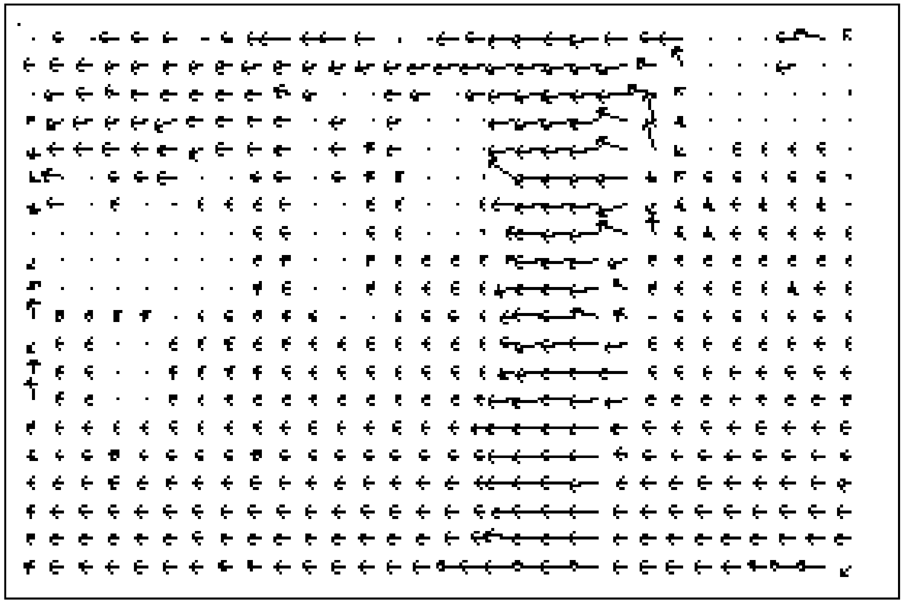
$$a_1 = -d\omega_y + T_z n_x, \quad a_2 = d\omega_x + T_z n_y,$$
$$a_3 = T_z n_z - T_x n_x, \quad a_4 = d\omega_z - T_x n_y,$$
$$a_5 = -d\omega_y - T_x n_z, \quad a_6 = T_z n_z - T_y n_y,$$
$$a_7 = -d\omega_z - T_y n_x, \quad a_8 = d\omega_x - T_y n_z.$$

- Motion field of planar surface is quadratic polynomial in (f,x,y)
- Same motion field produced by two different planes w. two different 3D motions
- Not unique: co-planar set of points (remember 8 point algorithm for calibration)

Application (Szeklisky): Motion representations

- How can we describe this scene?

Optical Flow Field

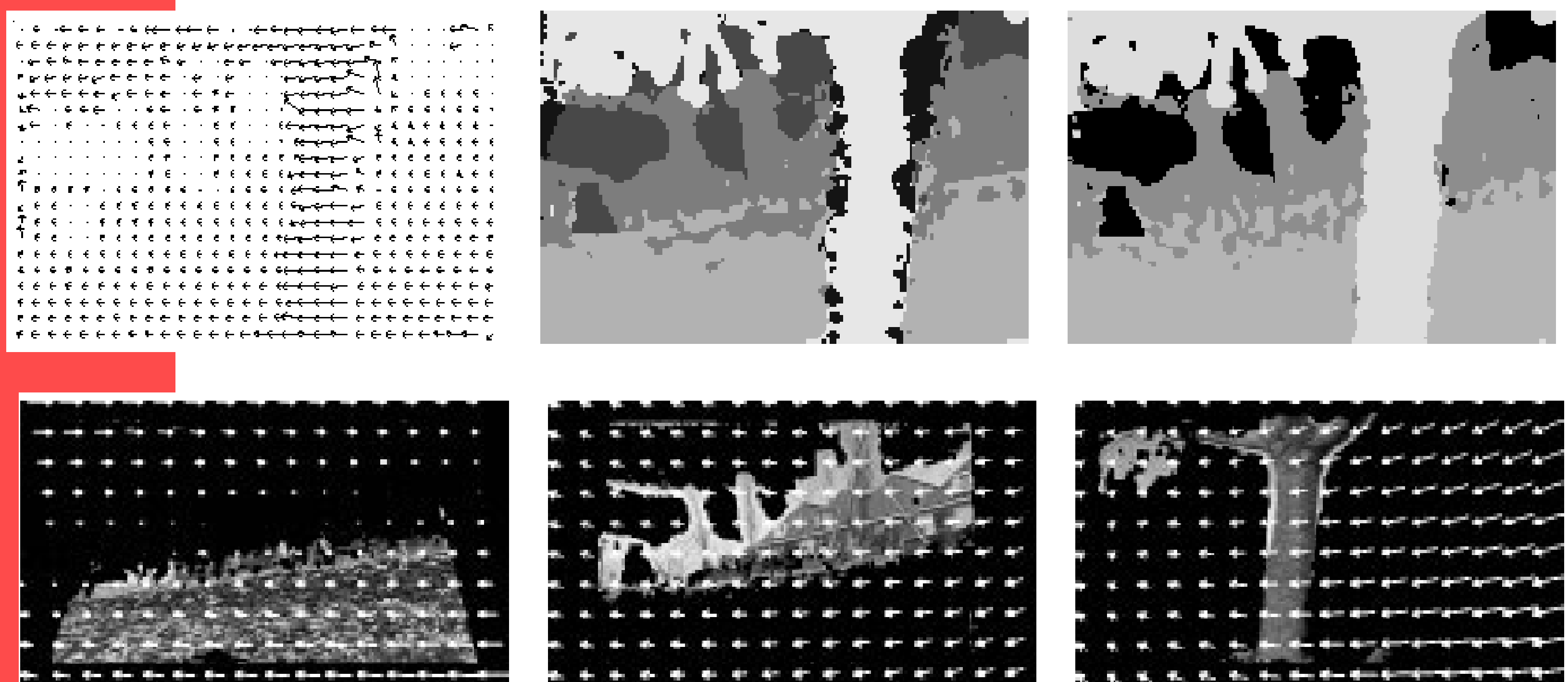


Layered motion

- Break image sequence up into "layers":
-  ÷  =

- Describe each layer's motion

Results



Additional Slides, not discussed in class.

Horn and Schunck

- Least squares formulation:
- Differentiate integrand w.r.t. Z, set to 0, solve for each (x,y)
- Differentiate equation w.r.t. tx, ty, tz, set to 0, solve

- (see Chapter 17 Horn and Schunck)

Optic flow for rigid motion

- Rigid Motion (for small v): $\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{H} \begin{bmatrix} T \\ \Omega \end{bmatrix}$

$$\mathbf{H} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix}$$

Perspective projection of 3-D velocity

3-D velocity at point P

Hard to solve with just optic flow vectors! (but see Horn 17.3-17.5).

Instead, ***Direct method*** combines constraint linearly with BCCE!

(BCCE: Brightness Change Constraint Equation)

Direct Rigid Motion Estimation

- Brightness Change Constraint

$$I(x, y, t) = I(x + v_x, y + v_y, t + 1)$$

$$\frac{dI}{dx} v_x + \frac{dI}{dy} v_y + \frac{dI}{dt} = 0$$

$$\begin{bmatrix} -\frac{dI}{dt} \end{bmatrix} = \begin{bmatrix} \frac{dI}{dx} & \frac{dI}{dy} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

Direct Rigid Motion Estimation

- Brightness Change Constraint

$$\left[-\frac{dI}{dt}\right] = \left[\frac{dI}{dx} \quad \frac{dI}{dy}\right]\begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

- Rigid Motion Model

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{H}\begin{bmatrix} T \\ \Omega \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix}$$

Direct Motion Estimation

- One equation per pixel:

$$\begin{bmatrix} -\frac{dI}{dt} \end{bmatrix} = \begin{bmatrix} \frac{dI}{dx} & \frac{dI}{dy} \end{bmatrix} \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix} \begin{bmatrix} \mathrm{T} \\ \Omega \end{bmatrix}$$

First, convert X,Y from screen coordinates to pixel coordinates….

Direct Motion Estimation

- One equation per pixel:

$$\left[-\frac{dI}{dt}\right]=\left[\frac{dI}{dx}\quad\frac{dI}{dy}\right]\begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix}\frac{1}{Z'}\begin{bmatrix} 1 & 0 & 0 & 0 & Z & -yZ'/f \\ 0 & 1 & 0 & -Z & 0 & xZ'/f \\ 0 & 0 & 1 & yZ'/f & -xZ'/f & 0 \end{bmatrix}\begin{bmatrix} \mathrm{T} \\ \Omega \end{bmatrix}$$

- Still hard!
- Z unknown; assume surface shape…
  - Negahdaripour & Horn - Planar
  - Black and Yacoob - Affine
  - Basu and Pentland; Bregler and Malik - Ellipsoidal
  - Essa et al. - Polygonal approximation
  - ...

Direct Motion Estimation

- One equation per pixel:

$$\left[-\frac{dI}{dt}\right] = \left[\frac{dI}{dx} \quad \frac{dI}{dy}\right] \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -yZ'/f \\ 0 & 1 & 0 & -Z & 0 & xZ'/f \\ 0 & 0 & 1 & yZ'/f & -xZ'/f & 0 \end{bmatrix} \begin{bmatrix} \mathrm{T} \\ \Omega \end{bmatrix}$$

"Direct Depth"

Use real-time stereo!

- Gives Z directly; no approximate model needed
- Express Direct Constraint on Depth Gradient

$$I(x, y, t) = I(x + v_x, y + v_y, t + 1)$$

$$Z(x, y, t) = Z(x + v_x, y + v_y, t + 1) - v_z$$

$$\frac{dZ}{dx} v_x + \frac{dZ}{dy} v_y + \frac{dZ}{dt} - v_z = 0$$

Direct Depth

3-D Depth and Brightness Constraint Equations:

- Orthographic

$$\begin{bmatrix} -dI/dt \\ -dZ/dt \end{bmatrix} = \begin{bmatrix} dI/dx & dI/dy & 0 \\ dZ/dx & dZ/dy & -1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

- Perspective

$$\begin{bmatrix} -dI/dt \\ -dZ/dt \end{bmatrix} = \begin{bmatrix} fdI/dx & fdI/dy & -ydI/dy - xdI/dx \\ fdZ/dx & fdZ/dy & -1 \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Direct Depth

Combined with rigid motion model:

- Orthographic

$$\begin{bmatrix} -dI/dt \\ -dZ/dt \end{bmatrix} = \begin{bmatrix} dI/dx & dI/dy & 0 \\ dZ/dx & dZ/dy & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -y \\ 0 & 1 & 0 & -Z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{bmatrix} \begin{bmatrix} \mathrm{T} \\ \Omega \end{bmatrix}$$

- Perspective

$$\begin{bmatrix} -dI/dt \\ -dZ/dt \end{bmatrix} = \begin{bmatrix} fdI/dx & fdI/dy & -ydI/dy - xdI/dx \\ fdZ/dx & fdZ/dy & -1 \end{bmatrix} \frac{1}{Z'} \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -yZ'/f \\ 0 & 1 & 0 & -Z & 0 & xZ'/f \\ 0 & 0 & 1 & yZ'/f & -xZ'/f & 0 \end{bmatrix} \begin{bmatrix} \mathrm{T} \\ \Omega \end{bmatrix}$$

One system per pixel, same T,Ω. Solve with QR or SVD.

[Harville et. al]

Outline

- Why layers?
- 2-D layers [Wang & Adelson 94; Weiss 97]
- 3-D layers [Baker *et al.* 98]
- Layered Depth Images [Shade *et al.* 98]
- Transparency [Szeliski *et al.* 00]

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?

Layers for video summarization

Frame 0

Frame 50

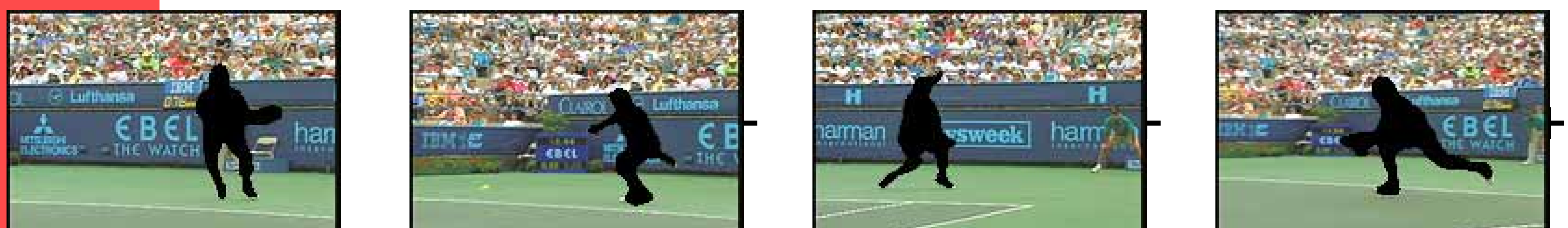
Frame 80

Background scene (players removed)

Complete synopsis of the video

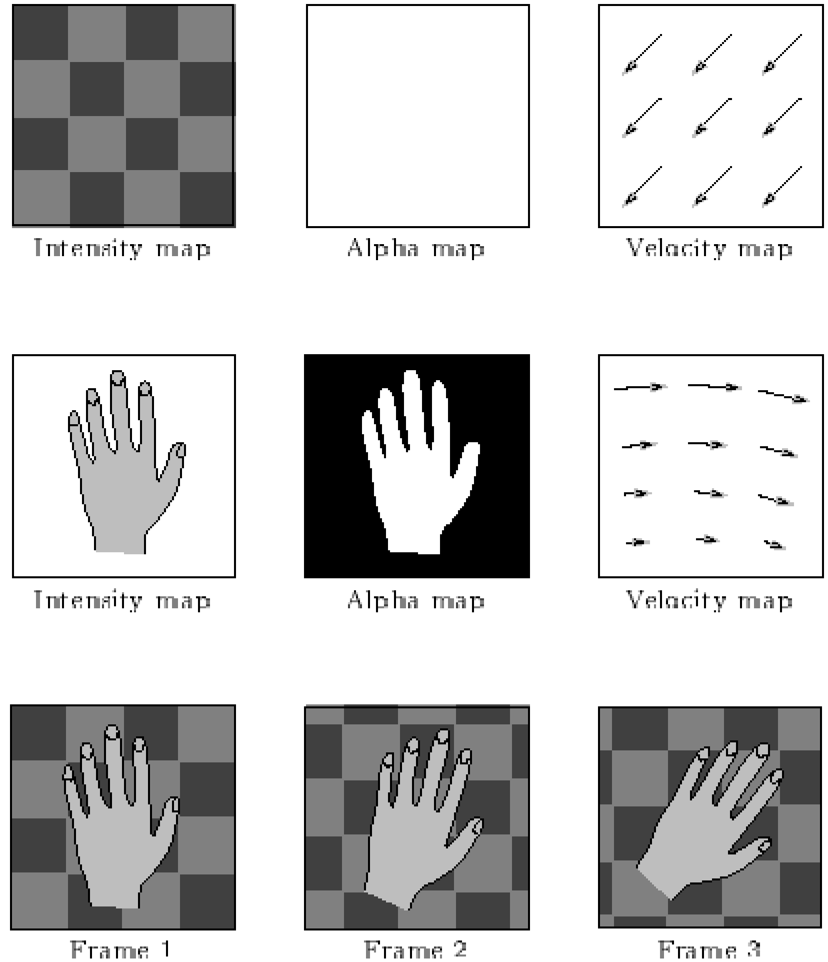
Background modeling (MPEG-4)

- Convert masked images into a background sprite for layered video coding



What are layers?

- [Wang & Adelson, 1994]
- intensities
- alphas
- velocities



How do we composite them?

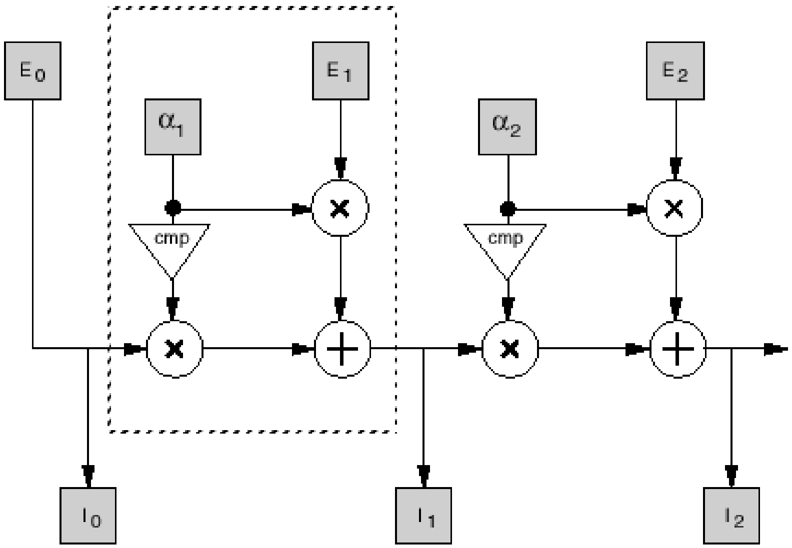


Figure 3: A flow chart for compositing a series of layers. The box labeled "cmp" generates the complement of alpha, $(1 - \alpha)$

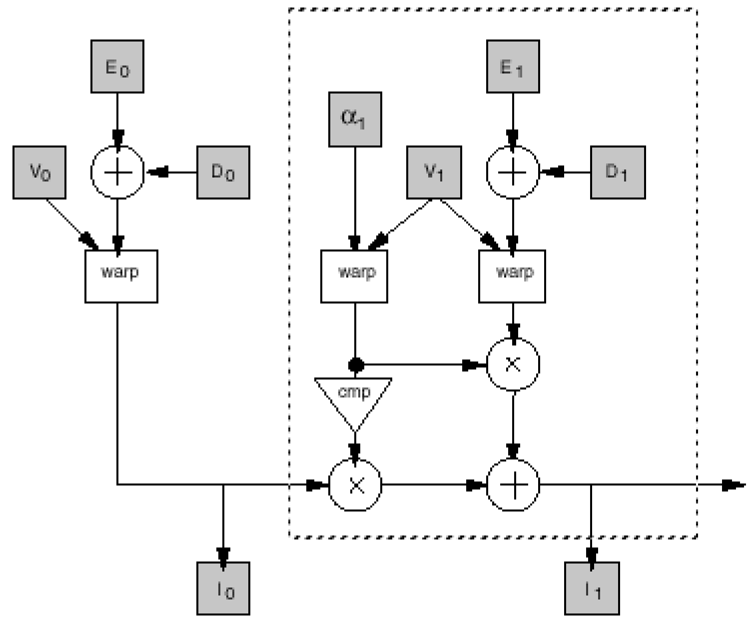
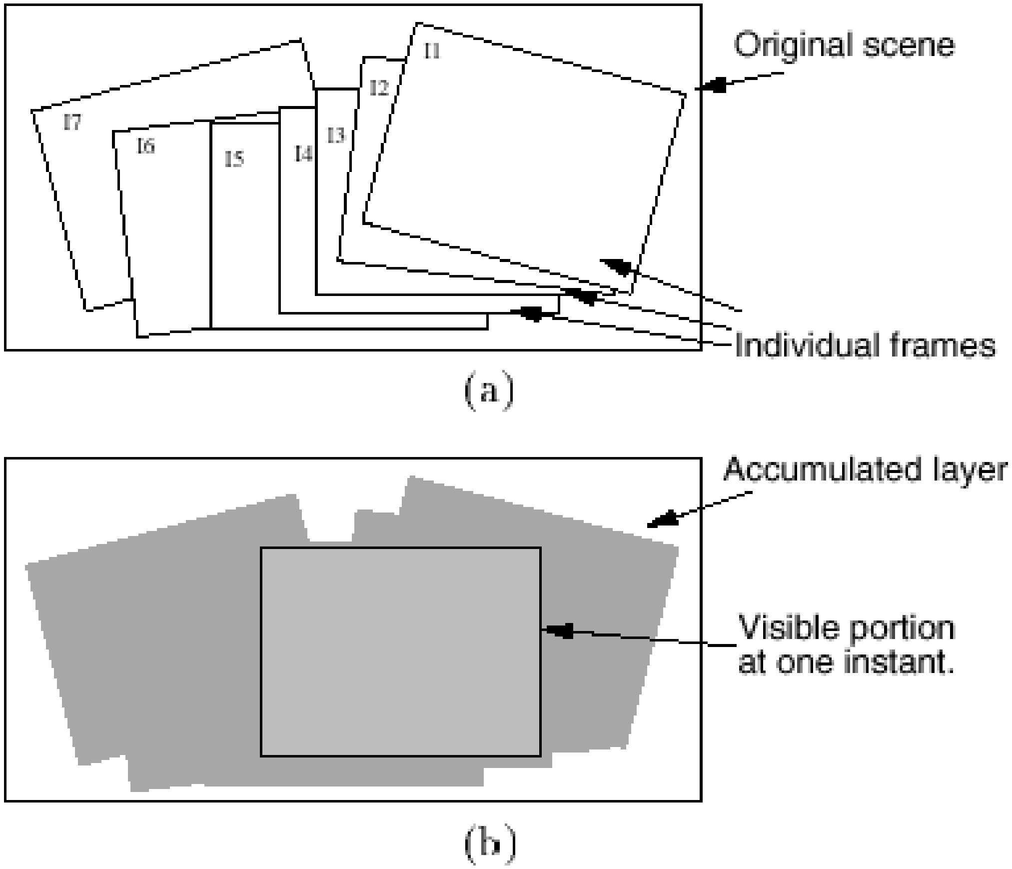


Figure 4: A flow chart for compositing that incorporates velocity maps, V, and delta maps, D.

How do we form them?



How do we form them?

(a) (b) (c)

Figure 7: (a) Frame 1 warped with an affine transformation to align the flowerbed region with that of frame 15. (b) Original frame 15 used as reference. (c) Frame 30 warped with an affine transformation to align the flowerbed region with that of frame 15.

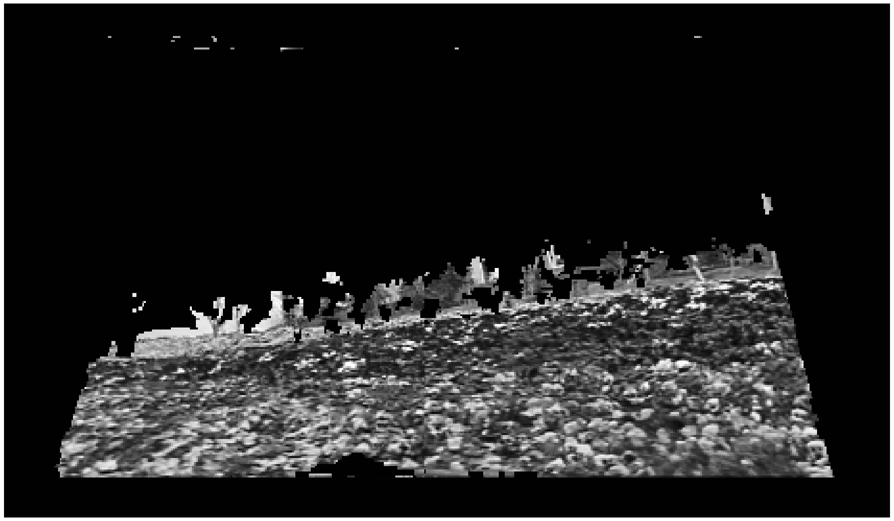
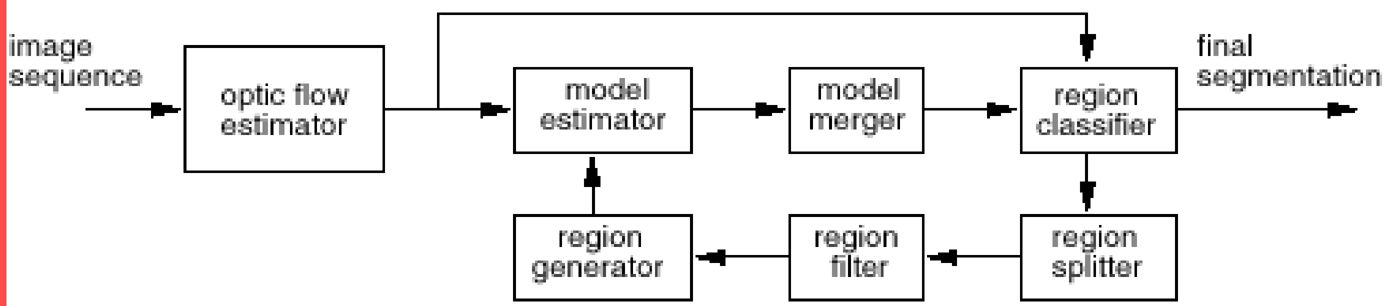


Figure 8: Accumulation of the flowerbed. Image intensities are obtained from a temporal median operation on the motion compensated images. Only the regions belonging to the flowerbed layer is accumulated in this image. Note also occluded regions are correctly recovered by accumulating data over many frames.

How do we estimate the layers?

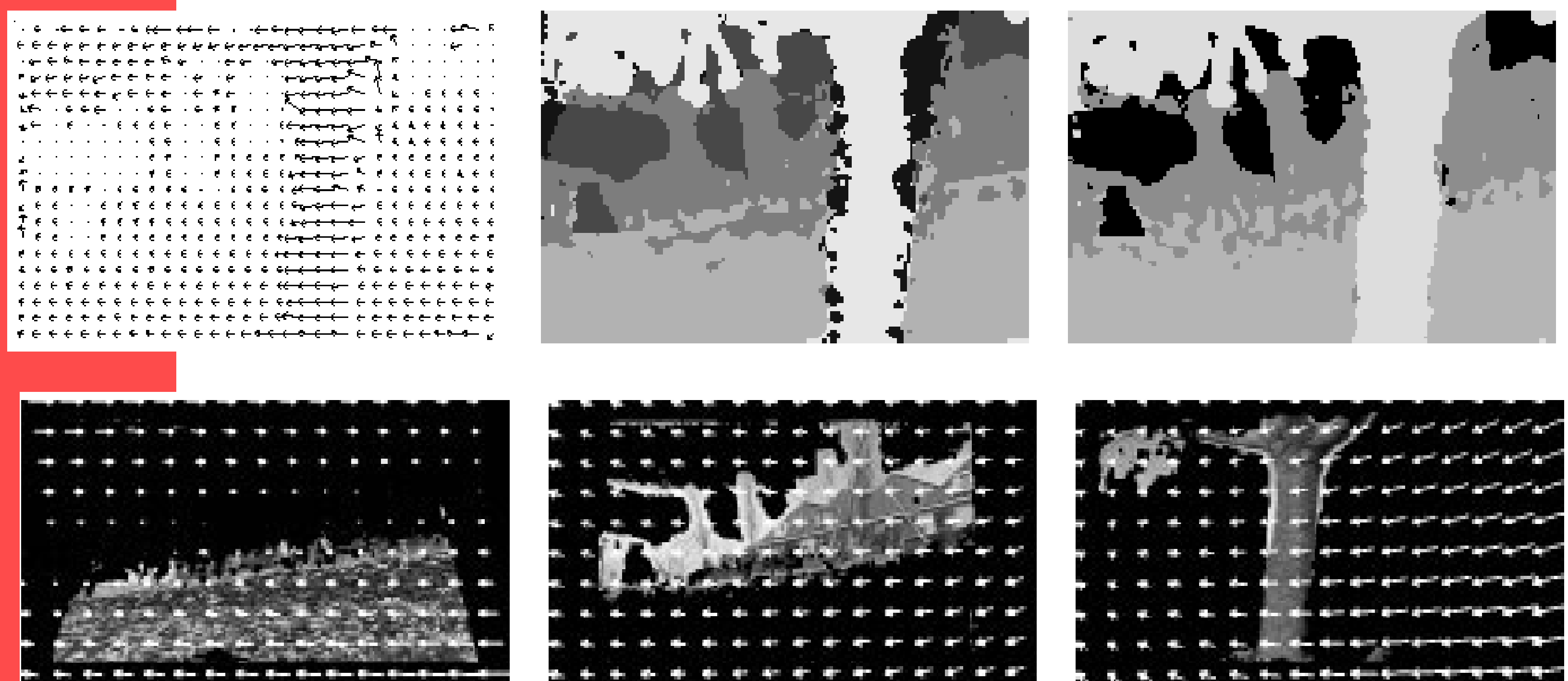
1. compute coarse-to-fine flow
2. estimate affine motion in blocks (regression)
3. cluster with *k-means*
4. assign pixels to best fitting affine region
5. re-estimate affine motions in each



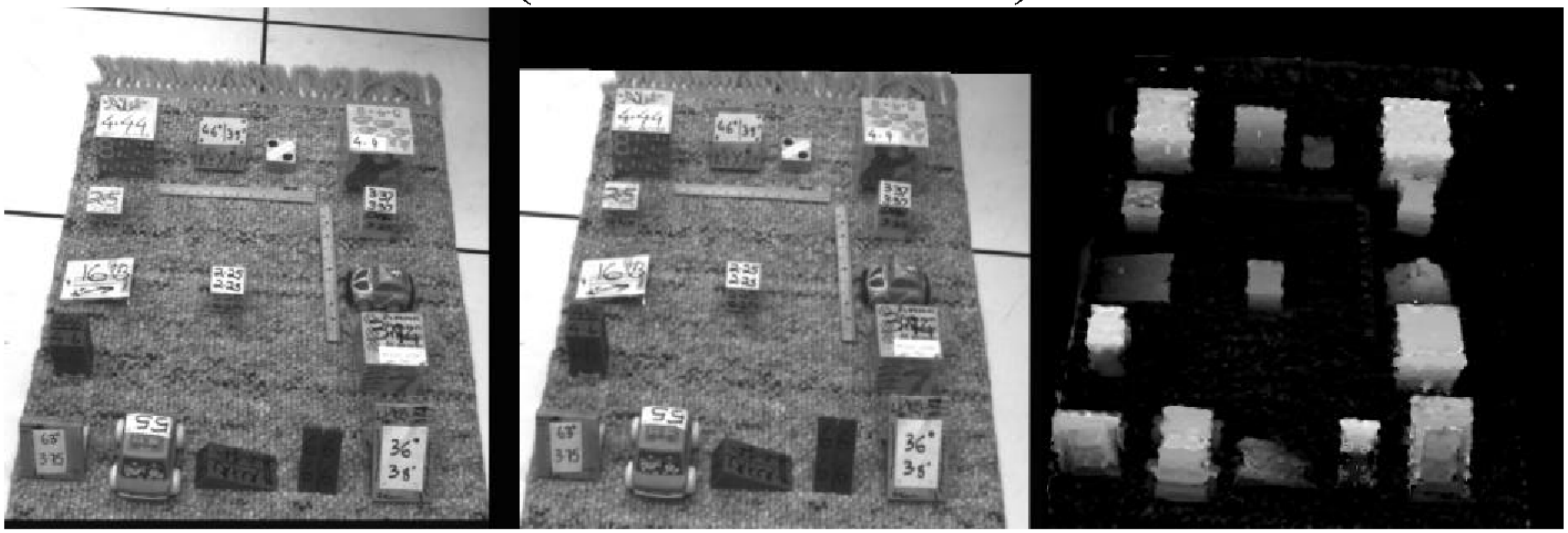
Layer synthesis

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

Results



Residual Planar Parallax Motion (Plane+Parallax)

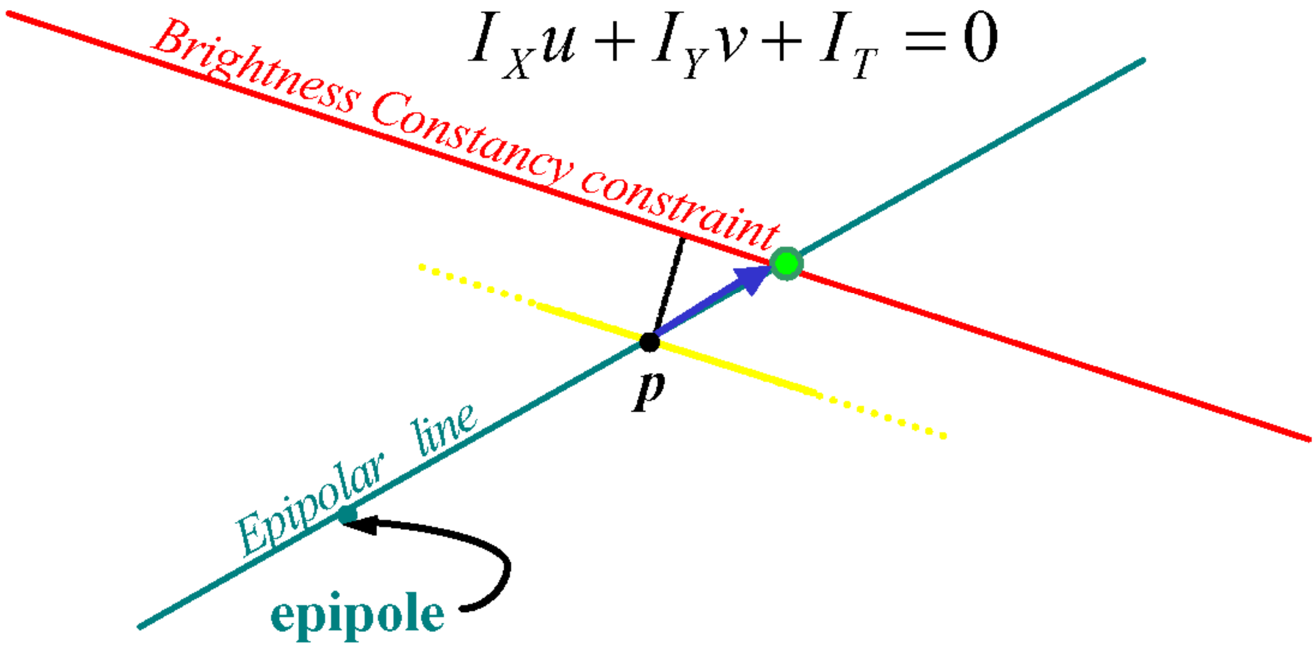


Original sequence **Plane-aligned sequence** **Recovered shape**

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

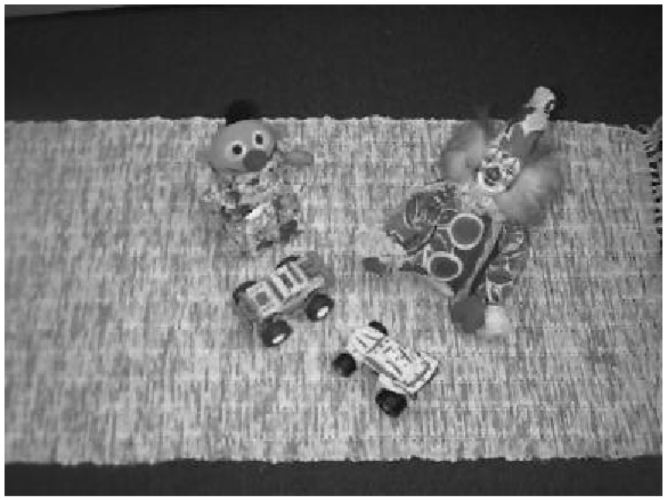
"Given two views where motion of points on a parametric surface has been compensated, the residual parallax is an epipolar field"

Residual Planar Parallax Motion

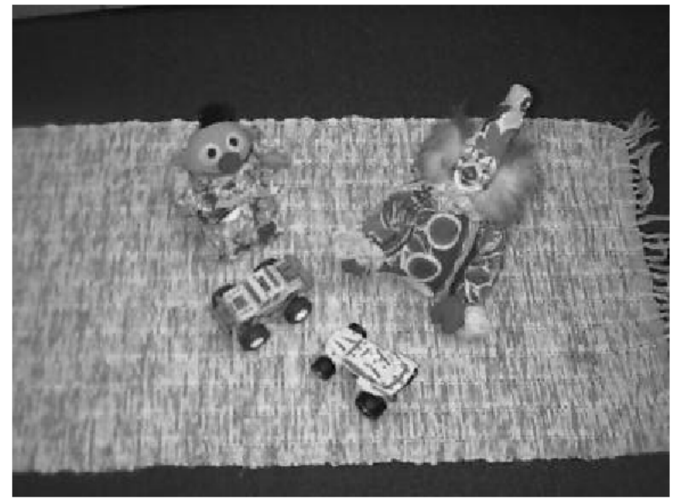


The intersection of the two line constraints <u>uniquely</u> defines the displacement.

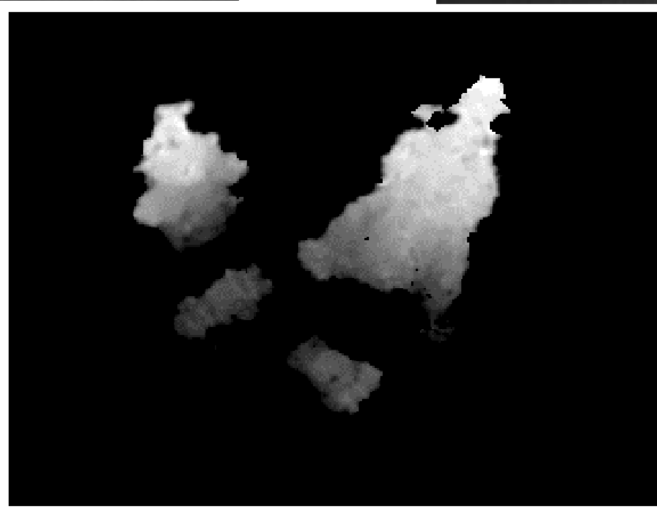
Dense 3D Reconstruction
(Plane+Parallax)



Original sequence



Plane-aligned sequence



Recovered shape