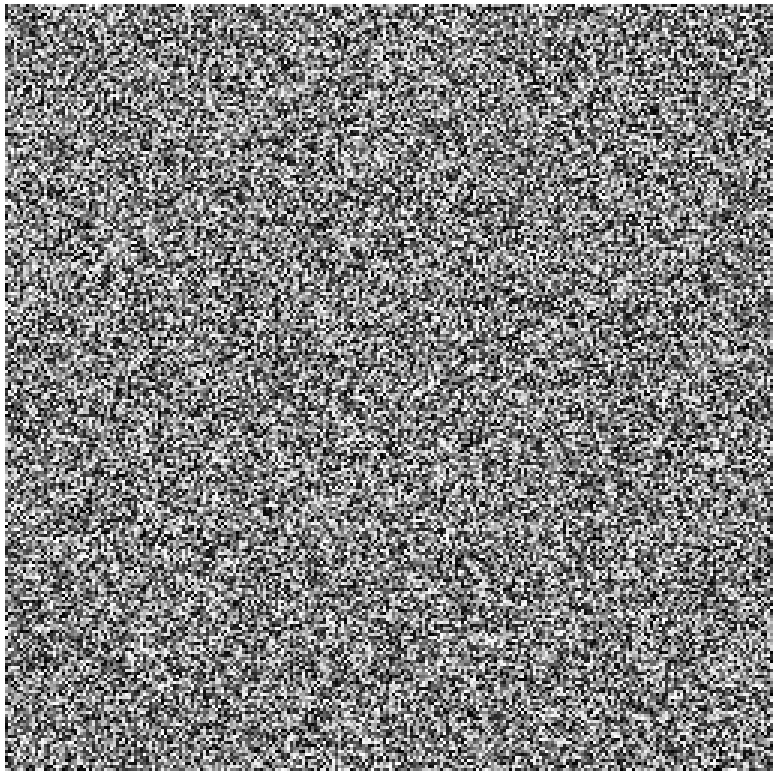
Gebze Institute of Technology
Department of Computer Engineering

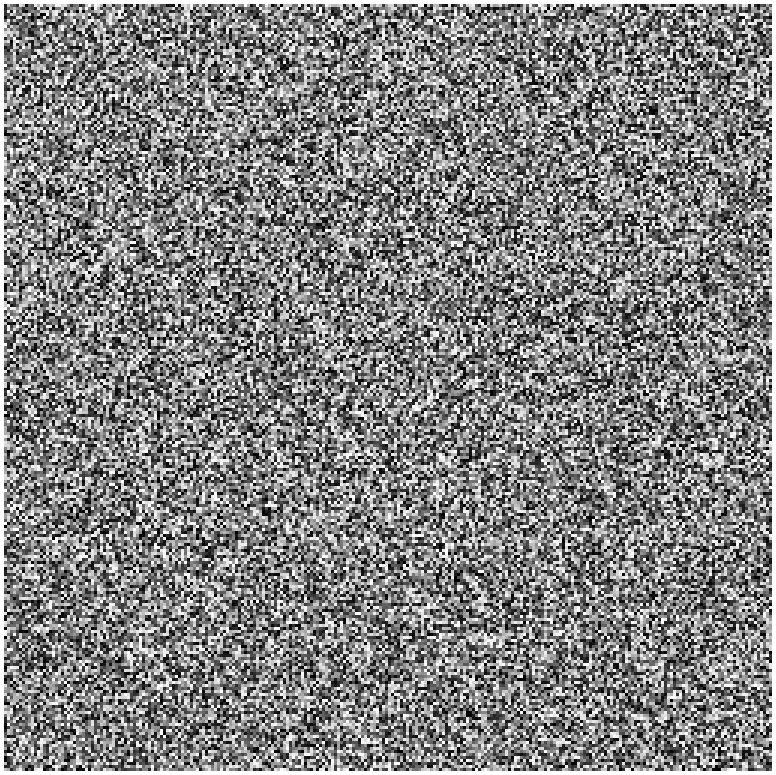
Computer Vision

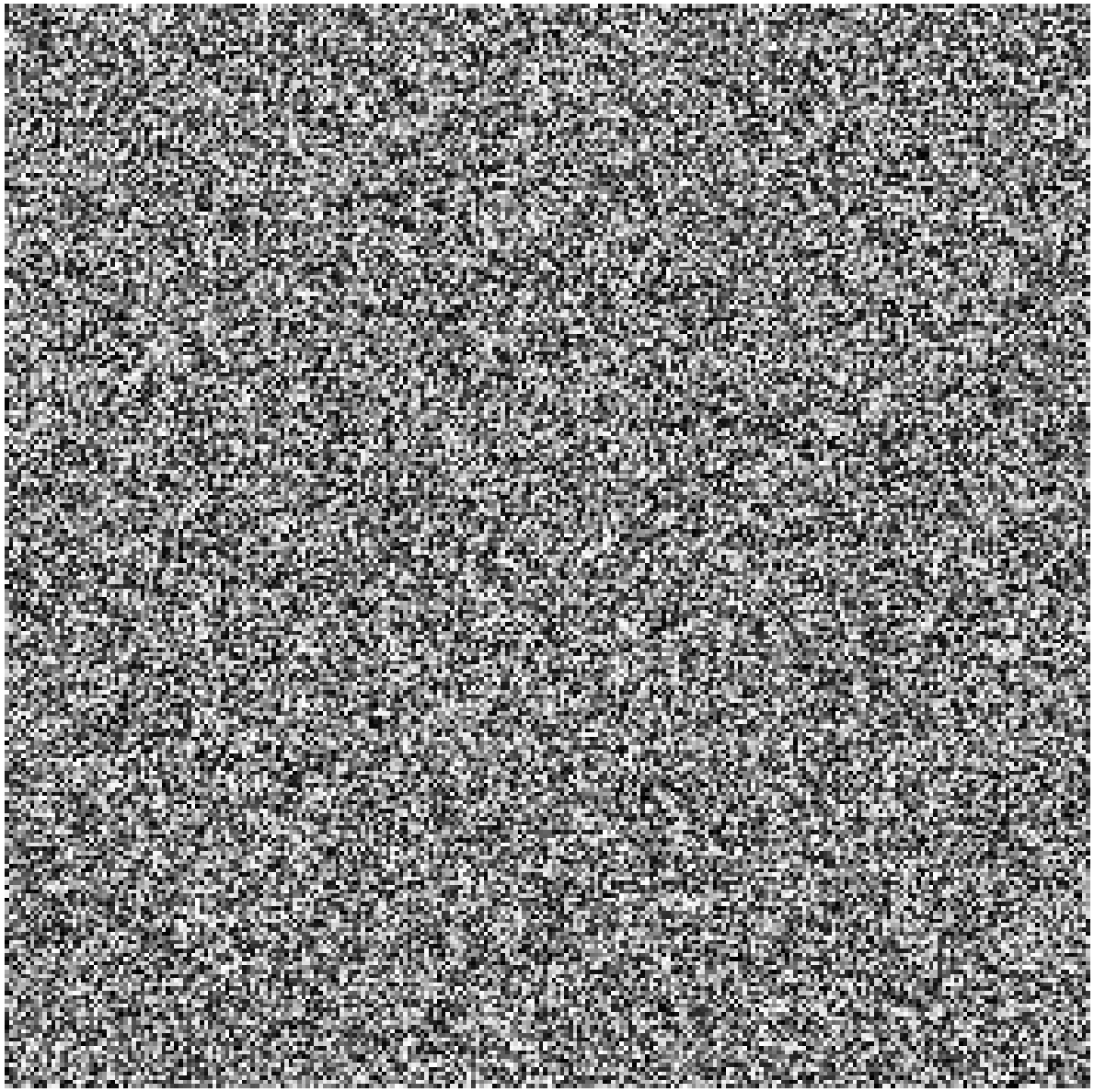
Visual Motion

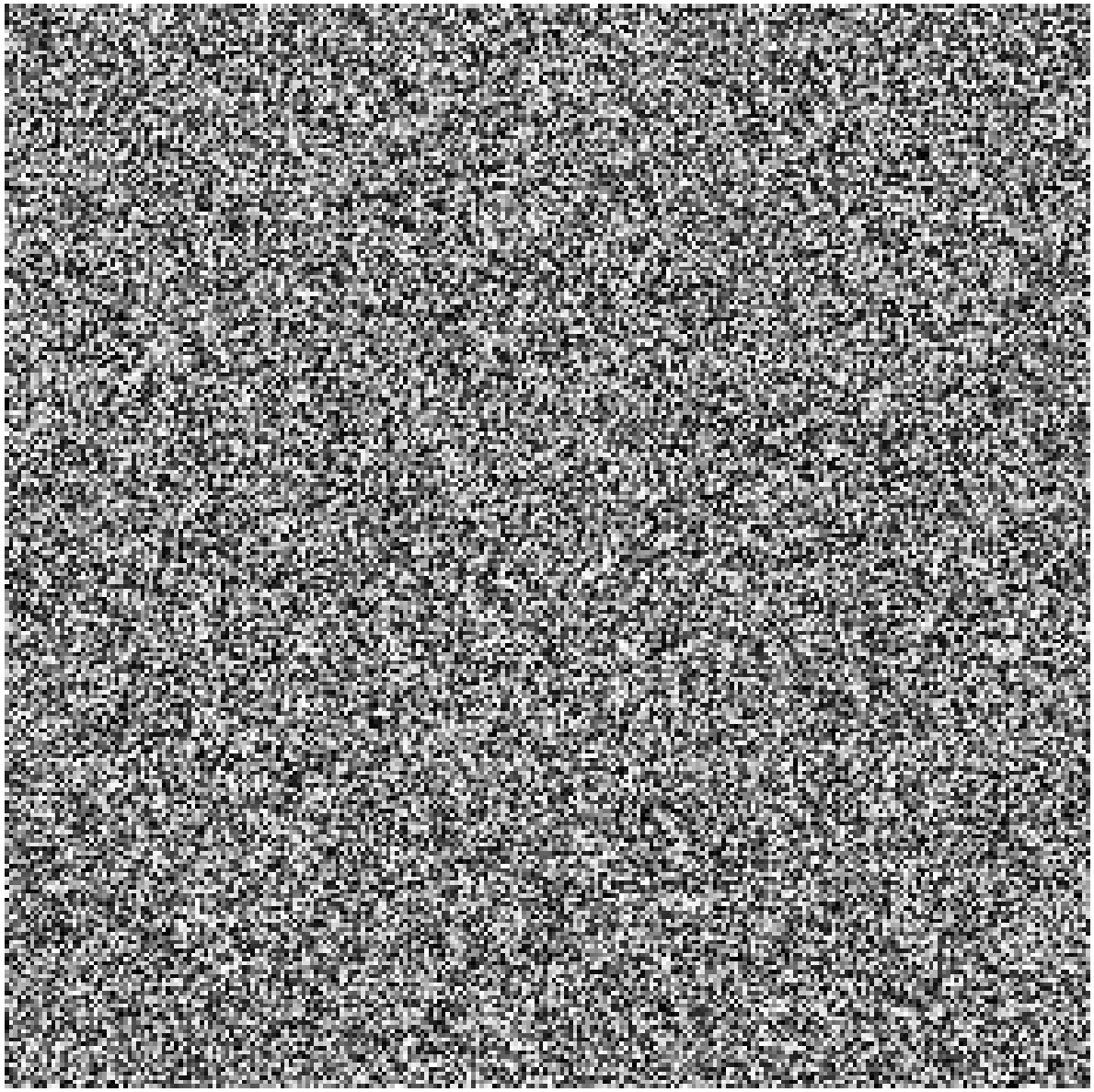
The Importance of Visual Motion

- Structure from Motion
  - Apparent motion is a strong visual clue for 3D reconstruction
    - More than a multi-camera stereo system
- Recognition by motion (only)
  - Biological visual systems use visual motion to infer properties of 3D world with little a priori knowledge of it
    - Random dot example
- Other uses of Visual Motion or Video
  - Video Coding and Compression: MPEG 1, 2, 4, 7…
  - Video Mosaicing and Layered Representation
  - Surveillance (Human Tracking and Traffic Monitoring)
  - HCI using Human Gesture (video camera)
  - Automated Production of Video Instruction Program (VIP)
  - Video Texture for Image-based Renderin









Process of images over time.

Image sequence:

- An image sequence is a series of N images, or frames, acquired at discrete time instants $t_k=t_0+k\Delta t$, where Δt is a fixed time interval and k=0,1,…,N-1

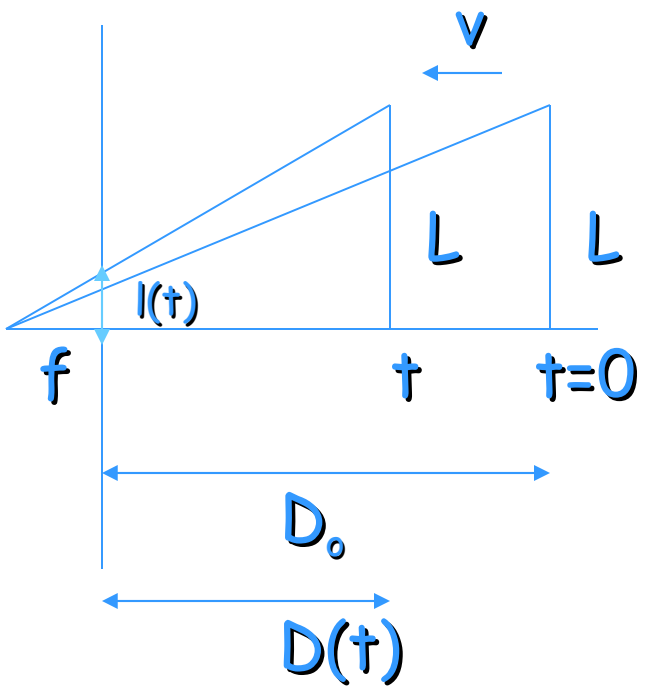
- Basic assumption
  - Illumination conditions do not change.
  - The image changes between frames are caused by the camera or the scene motion.

Visual Motion

- Allows us to compute useful properties of the 3D world, with very little knowledge.

- Example: Time to collision

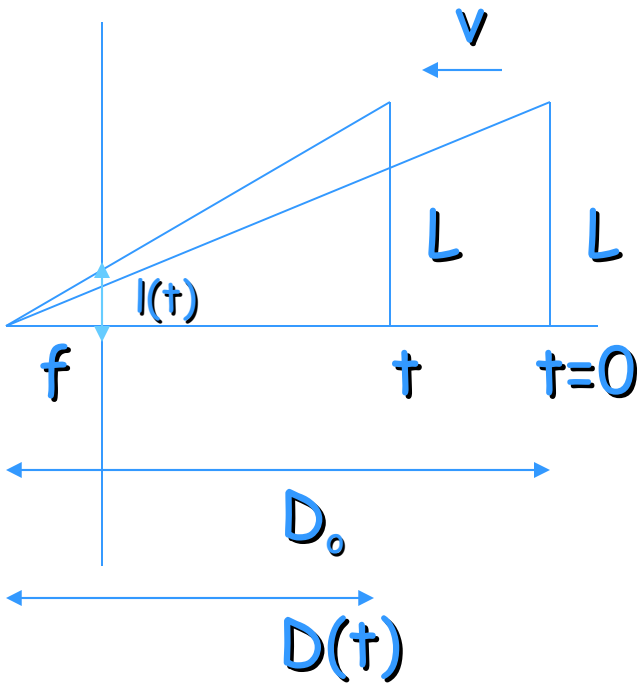
Time to Collision



An object of height L moves with constant velocity v:

- At time t=0 the object is at:
  - $D(0) = D_o$
- At time t it is at
  - $D(t) = D_o - vt$
- It will crash with the camera at time:
  - $D(\square) = D_o - v\blacksquare = 0$
  - $\blacksquare = D_o/v$

Time to Collision



The image of the object has size l(t):

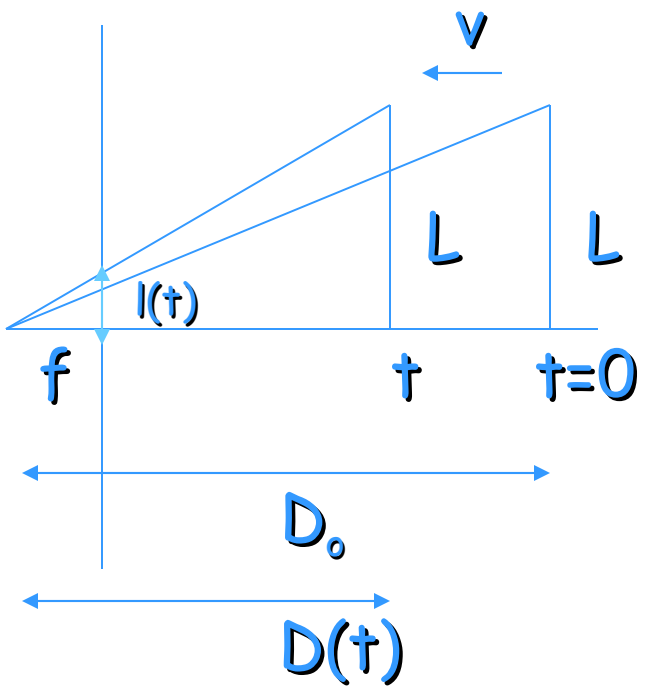
$$l(t) = \frac{fL}{D(t)}$$

Taking derivative wrt time:

$$l'(t) = \frac{dl(t)}{dt} = fL\frac{d(1/D(t))}{dt}$$

$$l'(t) = fL\frac{-1}{D^2(t)}\frac{d(D(t))}{dt}$$

Time to Collision



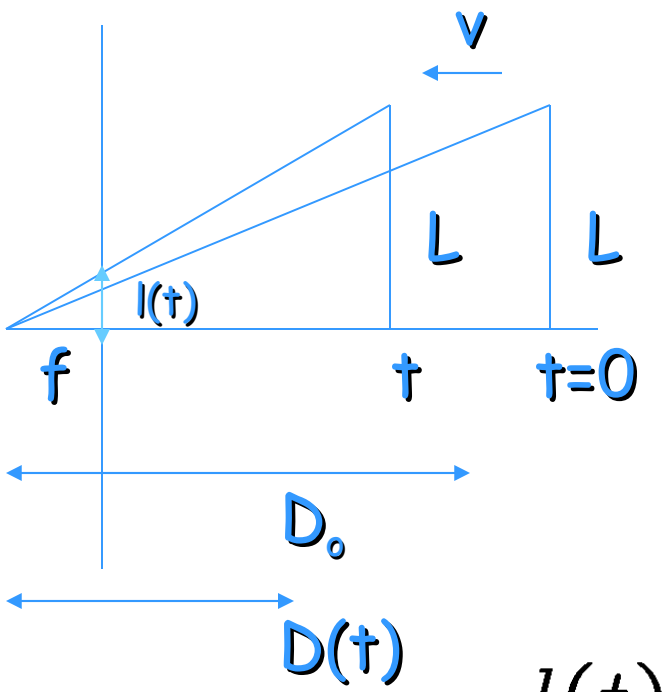
$$l'(t) = fL\frac{-1}{D^2(t)}\frac{d(D(t))}{dt}$$

$$D(t) = D_o - vt$$

$$\frac{d(D(t))}{dt} = -v$$

$$l'(t) = fL\frac{v}{D^2(t)}$$

Time to Collision



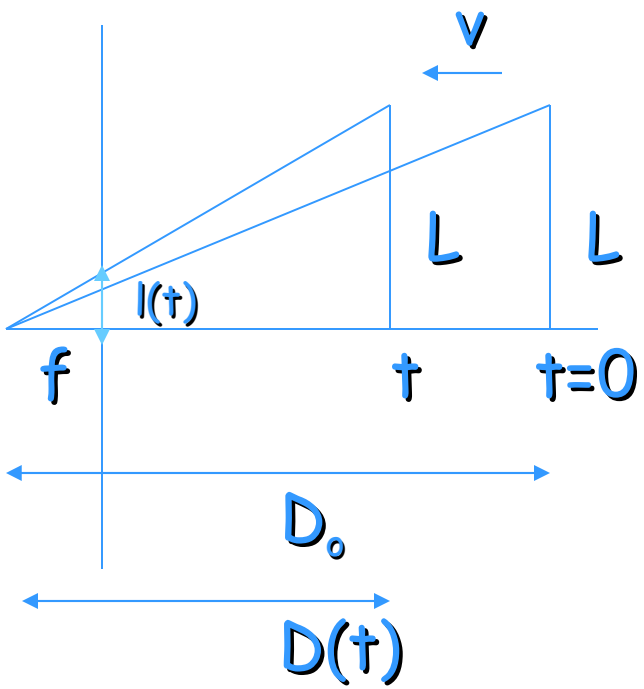
$$l'(t) = fL\frac{v}{D^2(t)}$$

$$l(t) = \frac{fL}{D(t)}$$

And their ratio is:

$$\frac{l(t)}{l'(t)} = \frac{fL}{D(t)}\frac{D^2(t)}{fLv} = \frac{D(t)}{v} = \tau$$

Time to Collision



$$l'(t) = fL\frac{v}{D^2(t)}$$

$$l(t) = \frac{fL}{D(t)}$$

Can be directly measured from image

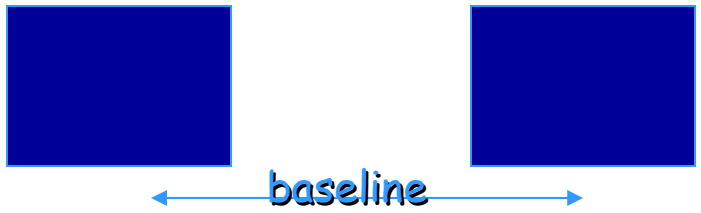
And time to collision:

$$\tau = \frac{l(t)}{l'(t)}$$

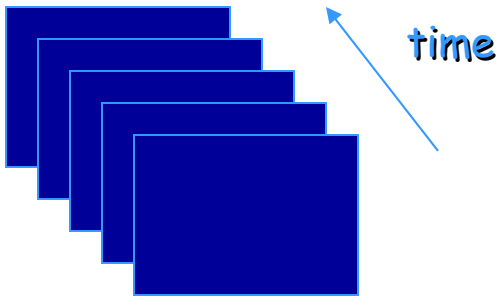
Can be found, without knowing L or D_0 or v !!

Comparison between Motion Analysis and Stereo

- Stereo: Two or more frames



- Motion: N frames



- The baseline is usually larger in stereo than in motion:
  - Motion disparities tend to be smaller
- Stereo images are taken at the same time:
  - Motion disparities can be due to scene motion
  - There can be more than 1 transformation btw frames

Motion Analysis Problems

- Correspondence Problem
  - Track corresponding elements across frames
- Reconstruction Problem
  - Given a number of corresponding elements, and camera parameters, what can we say about the 3D motion and structure of the observed scene?
- Segmentation Problem
  - What are the regions of the image plane which correspond to *different* moving objects?

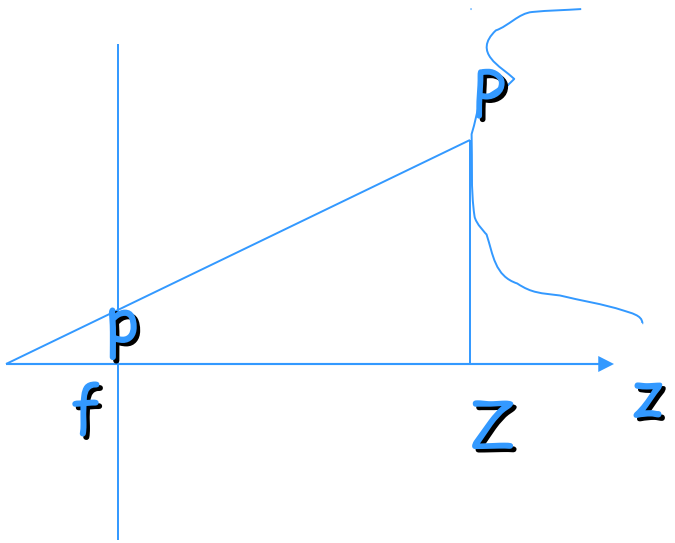
Motion Analysis Problems

- Two Subproblems
  - Correspondence:
    - Differential Methods - >dense measure (optical flow)
    - Matching Methods -> sparse measure
  - Reconstruction : More difficult than stereo since
    - Motion (3D transformation betw. Frames) as well as structure needs to be recovered
    - Small baseline causes large errors
- The Third Subproblem
  - Motion Segmentation: Chicken and Egg problem
    - Which should be solved first? Matching or Segmentation
      - Segmentation for matching elements
      - Matching for Segmentation

Motion Field (MF)

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

Consider a 3D point P and its image:



$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

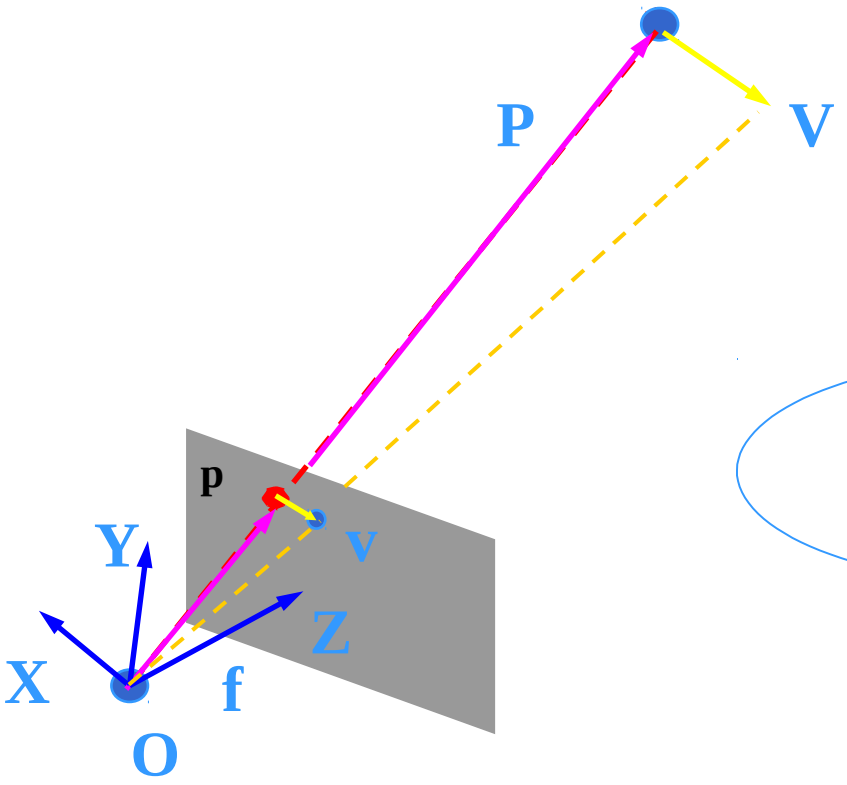
$$p = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$

Using pinhole camera equation:

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

We will work in the camera reference frame

When the things move



The ***relative velocity*** of P wrt camera:

$$V = -T - \omega \times P$$

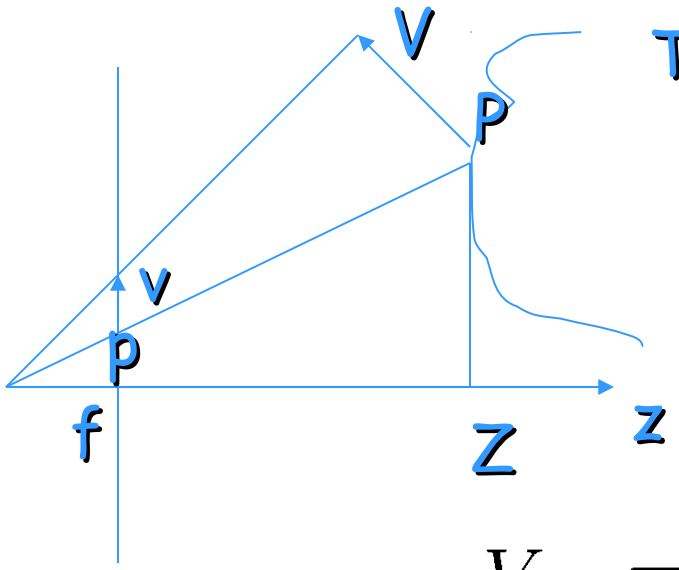
Translation velocity

Rotation angular velocity

– How to connect this with stereo geometry (with R, T)?

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

3D Relative Velocity:



The relative velocity of P wrt camera:

$$V = -T - \omega \times P$$

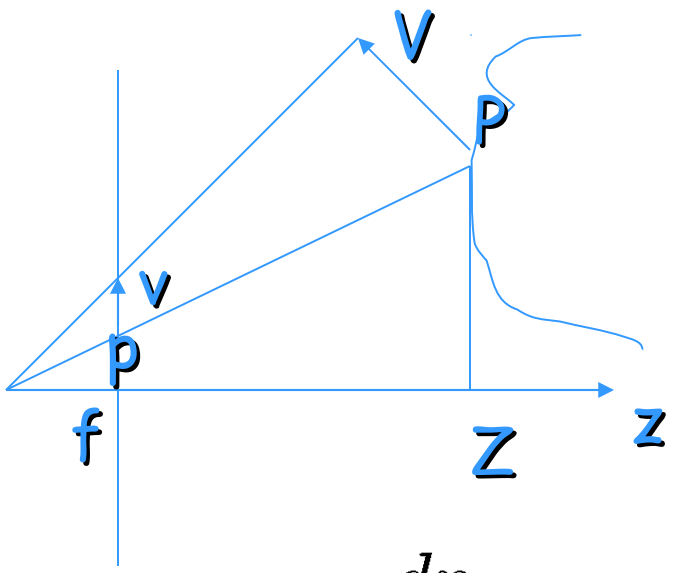
$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

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

How do we connect relative velocity with the motion field? Hint →

$$p = \frac{fP}{Z}$$

Motion Field: the velocity of p



$$p = \frac{fP}{Z}$$

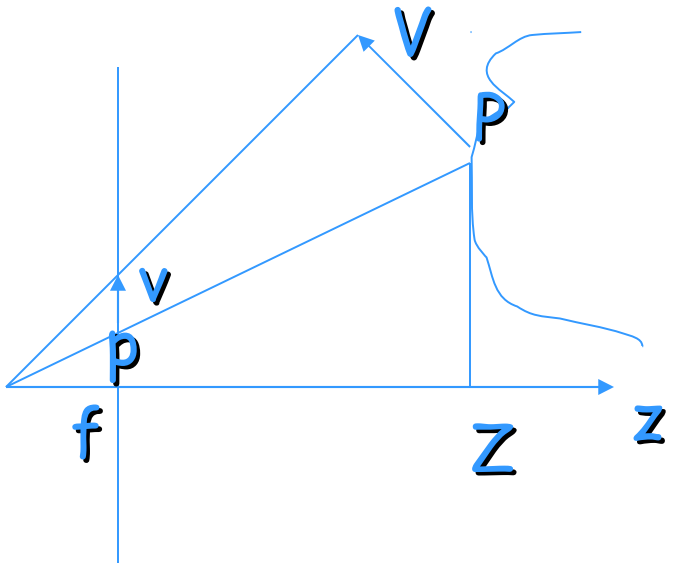
Taking derivative wrt time:

$$\frac{dp}{dt} = v = \frac{d\frac{fP}{Z}}{dt}$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[\frac{dP}{dt}.Z - P.\frac{dZ}{dt}\right]$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[V.Z - P.V_z\right]$$

Motion Field: the velocity of p

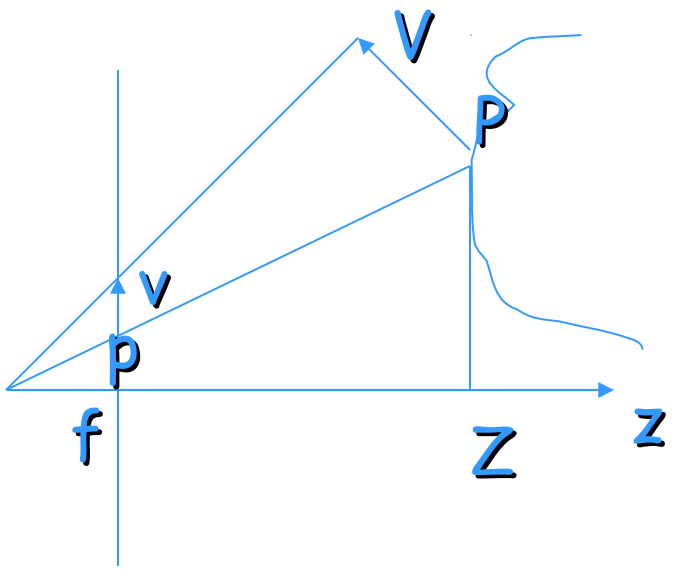


$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[V.Z - P.V_z\right]$$

$$p = \frac{fP}{Z} \qquad P = \frac{pZ}{f}$$

$$v = f\frac{V}{Z} - p\frac{V_z}{Z}$$

Motion Field: the velocity of p



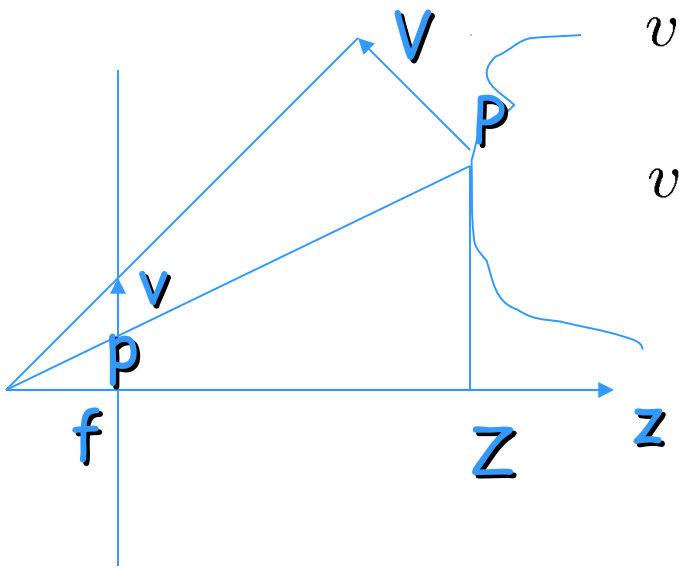
$$v = f\frac{V}{Z} - p\frac{V_z}{Z}$$

$$v_x = f\frac{V_x}{Z} - x\frac{V_z}{Z}$$

$$v_y = f\frac{V_y}{Z} - y\frac{V_z}{Z}$$

$$v_z = f\frac{V_z}{Z} - f\frac{V_z}{Z} = 0$$

Motion Field: the velocity of p



$$v_x = f\frac{V_x}{Z} - x\frac{V_z}{Z}$$

$$v_y = f\frac{V_y}{Z} - y\frac{V_z}{Z}$$

$$V_x = -T_x - \omega_y Z + \omega_z Y$$

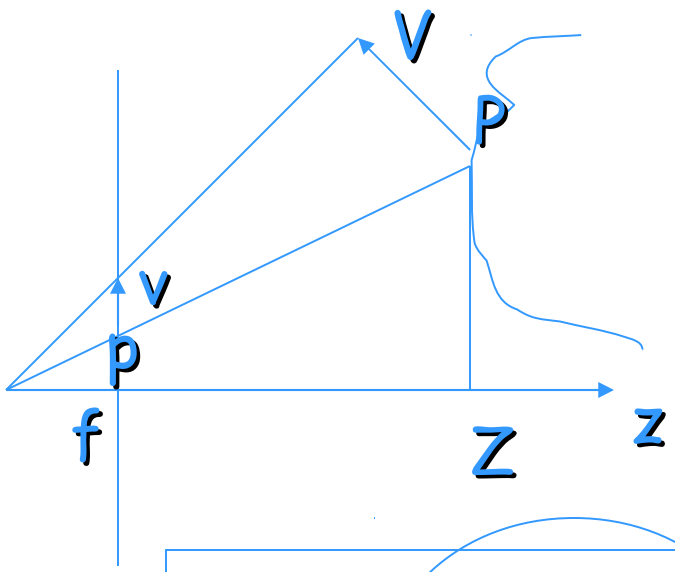
$$V_y = -T_y - \omega_z X + \omega_x Z$$

$$V_z = -T_z - \omega_x Y + \omega_y X$$

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

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

Motion Field: the velocity of p

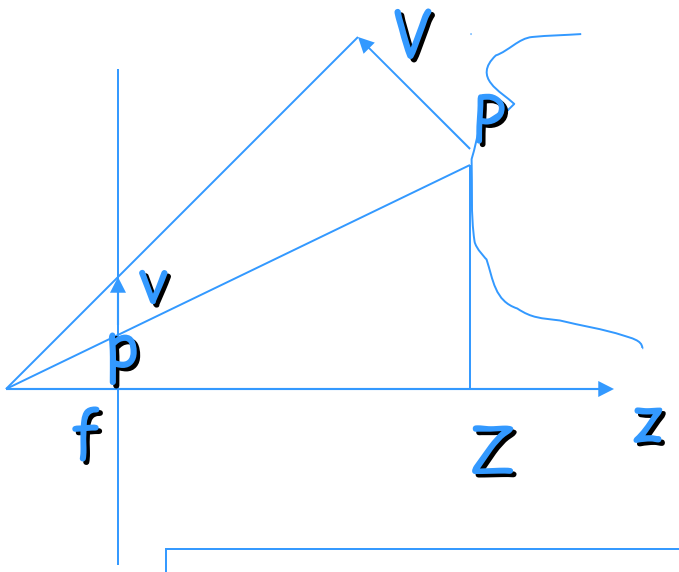


Translational component

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

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

Motion Field: the velocity of p



Rotational component

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

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

NOTE: The rotational component is independent of depth Z !

Motion Field: the velocity of p

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

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

Another way of expressing this relation

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{f} \begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} + \frac{1}{Z} \begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix} \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}$$

Rotation part: no depth information

Translation part: depth Z

Special Case I: Pure Translation

$$\omega = 0 \quad \Rightarrow \quad v_x = \frac{T_z x - T_x f}{Z}, \qquad v_y = \frac{T_z y - T_y f}{Z}$$

Assume $T_z \neq 0$

Define: $p_o = \begin{bmatrix} x_o \\ y_o \\ f \end{bmatrix} = \begin{bmatrix} \frac{fT_x}{T_z} \\ \frac{fT_y}{T_z} \\ f \end{bmatrix}$ ⇒

$$v_x = \frac{T_z x - T_z x_o}{Z} = (x - x_o)\frac{T_z}{Z} \qquad Z = \frac{T_z}{|\mathbf{v}|}\sqrt{(x-x_0)^2 + (y-y_0)^2}$$

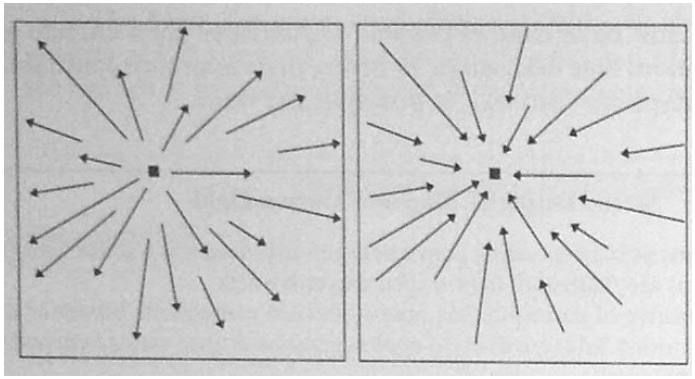
$$v_y = \frac{T_z y - T_z y_o}{Z} = (y - y_o)\frac{T_z}{Z}$$

What does this equation mean?

Special Case I: Pure Translation

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

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



$T_z > 0$ $T_z < 0$

The motion field in this case is RADIAL:

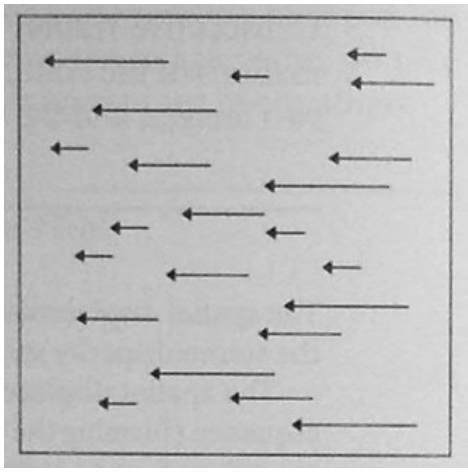
- It consists of vectors passing through $p_o = (x_o, y_o)$
- If:
  - $T_z > 0$, (camera moving towards object)
    - the vectors point away from p_o
    - p_o is the POINT OF EXPANSION
  - $T_z < 0$, (camera moving away from object)
    - the vectors point towards p_o
    - p_o is the POINT OF CONTRACTION

Special Case I: Pure Translation

$$\omega = 0 \quad \Rightarrow \quad v_x = \frac{T_z x - T_x f}{Z}, \qquad v_y = \frac{T_z y - T_y f}{Z}$$

What if $T_z = 0$?

$$v_x = -f\frac{T_x}{Z}, \qquad v_y = -f\frac{T_y}{Z} \quad \Rightarrow \quad Z = \frac{f}{|\mathbf{v}|}\sqrt{{T_x}^2 + {T_y}^2}$$



All motion field vectors are parallel to each other and inversely proportional to depth !

Pure Translation: Properties of the MF

- If $T_z \neq 0$ the MF is RADIAL with all vectors pointing towards (or away from) a single point p_0. If $T_z = 0$ the MF is PARALLEL.

- The length of the MF vectors is inversely proportional to depth Z. If $T_z \neq 0$ it is also directly proportional to the distance between p and p_0.

Pure Translation: Properties of the MF

- p_0 is the vanishing point of the direction of translation.

- p_0 is the intersection of the ray parallel to the translation vector and the image plane.

Special Cases: A Summary

- Pure Translation
  - Vanishing point and FOE (focus of expansion)
  - Only translation contributes to depth estimation
- Pure Rotation (We didn’t study this)
  - Does not carry 3D information
  - Motion field: a quadratic polynomial in image
  - Image mosaicing from a rotating camera
- Moving Plane (We didn't study this either)
  - Motion field is a quadratic polynomial in image
  - Image mosaicing for a planar scene

Estimating the MF

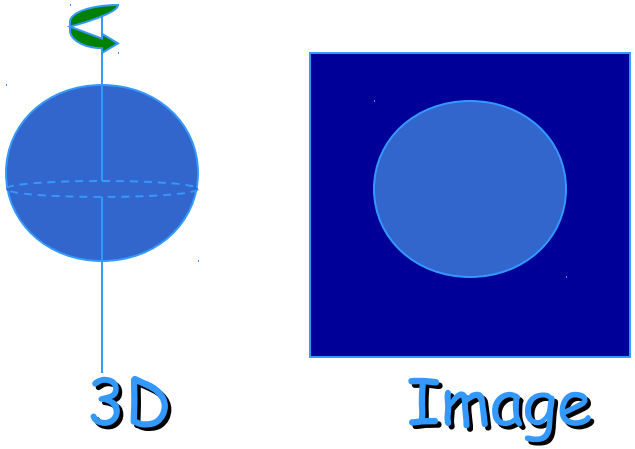
Optical Flow

MF Estimation

We will use the apparent motion of brightness patterns observed in an image sequence. This motion is called OPTICAL FLOW (OF)

MF ≠ OF

Consider a smooth, lambertian, uniform sphere rotating around a diameter, in front of a camera:

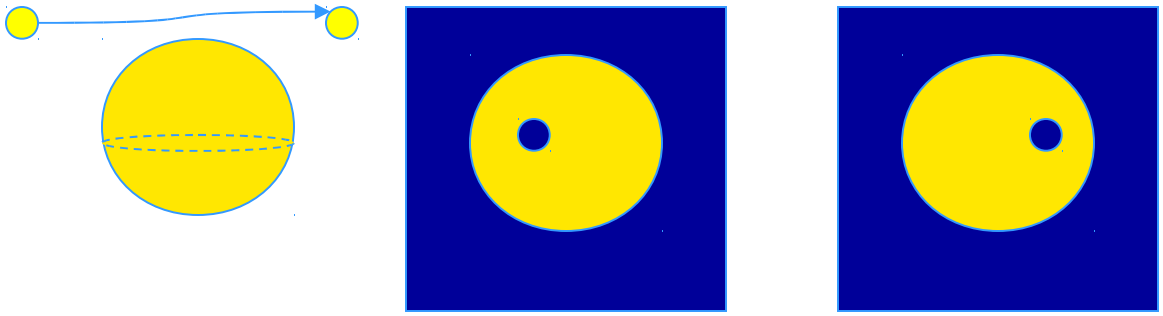


-MF ≠ 0 since the points on the sphere are moving

-OF = 0 since there are no moving patterns in the images

MF ≠ OF

Consider a still, smooth, specular, uniform sphere, in front of a stationary camera and a moving light source:



- MF = 0 since the points on the sphere are not moving
- OF ≠ 0 since there is a moving pattern in the images

Approximation of the MF

Never the less, keeping in mind that MF ≠ OF, we will assume that the apparent brightness of moving objects remain constant and hence we will estimate OF instead (since MF cannot really be observed!)

Brightness Constancy Equation

- Let P be a moving point in 3D:
  - At time t, P has coords (X(t),Y(t),Z(t))
  - Let p=(x(t),y(t)) be the coords. of its image at time t.
  - Let E(x(t),y(t),t) be the brightness at p at time t.
- Brightness Constancy Assumption:
  - As P moves over time, E(x(t),y(t),t) remains constant.

Brightness Constancy Equation

$$E(x(t), y(t), t) = Constant$$

Taking derivative wrt time:

$$\frac{dE(x(t), y(t), t)}{dt} = 0$$

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

Brightness Constancy Equation

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

Let

$$\nabla E = \begin{bmatrix} \frac{\partial E}{\partial x} \\ \frac{\partial E}{\partial y} \end{bmatrix}$$

(Frame spatial gradient)

$$v = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$$

(optical flow)

and

$$E_t = \frac{\partial E}{\partial t}$$

(derivative across frames)

Brightness Constancy Equation

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

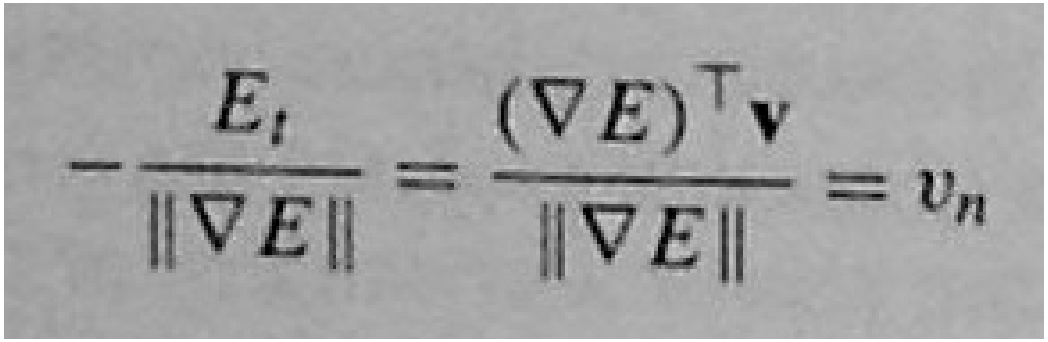
Becomes:

$$(\nabla E)^T . v + E_t = 0$$

Optic Flow

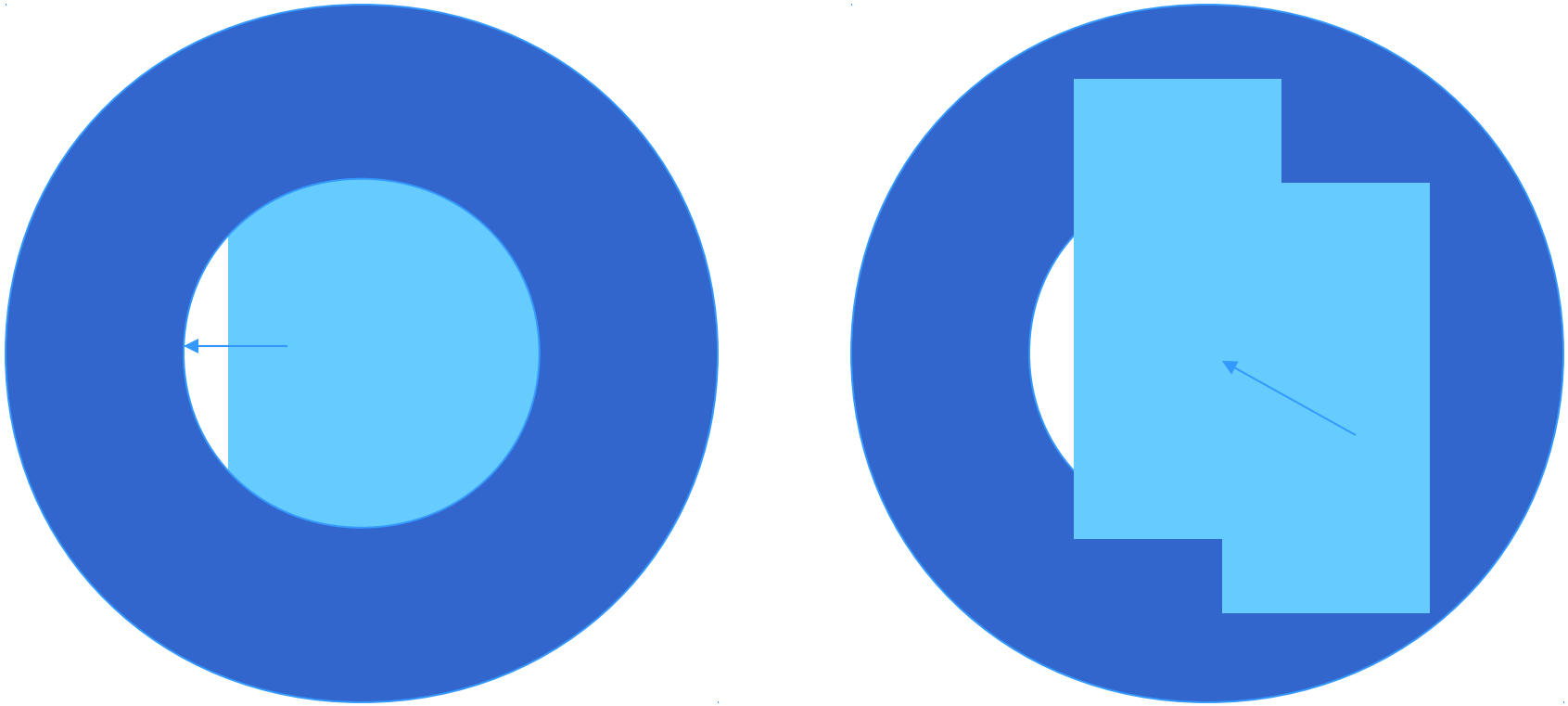
How much of the MF can be estimated by brightness constancy equation?

$$(\nabla E)^T . v + E_t = 0$$



Only the MF component in the direction of spatial image gradient can be estimated by optic flow.

The aperture problem



The Image Brightness Constancy Assumption only provides the OF component in the direction of the spatial image gradient

Optic Flow versus Motion Field

Definition: Optical Flow

The *optical flow* is a vector field subject to the constraint (8.17), and loosely defined as the *apparent motion* of the image brightness pattern. $(\nabla E)^T.v + E_t = 0$

Optical Flow and Motion Field

The optical flow is the *approximation of the motion field* which can be computed from time-varying image sequences. Under the simplifying assumptions of

- Lambertian surfaces
- pointwise light source at infinity
- no photometric distortion

the *error* of this approximation is

- *small* at points with high spatial gradient
- *exactly zero* only for translational motion or for any rigid motion such that the illumination direction is parallel to the angular velocity

Estimating OF

- The brightness constancy assumption is not enough to compute OF:
  - it only provides its normal component.
  - Need some other assumption or knowledge.
- Algorithms computing OF are of two types:
  - Differential Techniques
    - Based on spatial and temporal variations of the image brightness at all pixels.
    - Used to compute DENSE OF.
  - Matching Techniques
    - Similar to stereo feature matching, compute disparities.
    - Used to compute SPARSE OF.

A differential technique

A Differential OF Algorithm

Assumptions:

- Brightness Constancy:

  $(\nabla E)^T v + E_t = \varepsilon$

  where ε accounts for noise and model errors

- OF is constant in small patches:

  - Let Q be a small NxN patch (say N=5)
  - If the OF v is the same at each pixel $p_i \in Q$:

    $[(\nabla E(p_i))^T v + E_t(p_i)]^2 = \varepsilon_i^2$

  - Summing over all pixels in Q:

    $\Psi(v) = \sum_{p_i} [(\nabla E(p_i))^T v + E_t(p_i)]^2$

  - We need to find OF v s.t. $\Psi(v)$ is as small as possible

$$\min_v \{\sum_{pi} [(\nabla E(p_i))^T v + E_t(p_i)]^2\}$$

- This is a Least Square Error (LSE) Problem.
- It can be solved by finding v s.t. :

$$(\nabla E(p_1))^T v = -E_t(p_1)$$

$$(\nabla E(p_2))^T v = -E_t(p_2)$$

$$\vdots$$

$$(\nabla E(p_№))^T v = -E_t(p_№)$$

Expanding the equations:

$$(E_x(p_1)\ E_y(p_1)) \cdot (v_x\ v_y) \Im -E_t(p_1)$$

$$(E_x(p_2)\ E_y(p_2)) \cdot (v_x\ v_y) \Im -E_t(p_2)$$

$$\vdots$$

$$(E_x(p_№)\ E_y(p_№)) \cdot (v_x\ v_y) \Im -E_t(p_№)$$

Expanding the equations:

$$E_x(p_1)\, v_x + E_y(p_1)\, v_y = -E_t(p_1)$$
$$E_x(p_2)\, v_x + E_y(p_2)\, v_y = -E_t(p_2)$$
$$\vdots$$
$$E_x(p_{N^2})\, v_x + E_y(p_{N^2})\, v_y = -E_t(p_{N^2})$$

This a system of N^2 equations with 2 unknowns: v_x and v_y

Usually, $N^2 \gg 2$

Writing it in matrix format:

$$Av = b$$

- where:

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

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

Solving for v:

- A is N^2x2, with $N^2 > 2$:
  - it is not invertible!
- Multiply both sides of Av = b by A^T:

$$A^T A v = A^T b$$

- $A^T A$ is square (2x2):
  - $(A^T A)^{-1}$ exists if $\det(A^T A) \neq 0$
- Assuming that $(A^T A)^{-1}$ does exists:

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

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

Taking a closer look at (A^TA)

$$A = \begin{bmatrix} E_x(p_1) & E_y(p_1) \\ E_x(p_2) & E_y(p_2) \\ \vdots & \vdots \\ E_x(p_{N^2}) & E_y(p_{N^2}) \end{bmatrix}$$

$$A^T = \begin{bmatrix} E_x(p_1) & E_x(p_2) & \dots E_x(p_{N^2}) \\ E_y(p_1) & E_y(p_2) & \dots E_y(p_{N^2}) \end{bmatrix}$$

$$A^TA = \begin{bmatrix} \sum E_x^2 & \sum E_xE_y \\ \sum E_xE_y & \sum E_y^2 \end{bmatrix}$$

This is the matrix used for corner detection!

Taking a closer look at (A^TA)

The matrix for corner detection:

$$A^T A = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix}$$

is singular (not invertible) when $\det(A^TA) = 0$

But $\det(A^TA) = \prod \lambda_i = 0$ -> one or both e.v. are 0

One e.v. = 0 -> no corner, just an edge
Two e.v. = 0 -> no corner, homogeneous region

} Aperture Problem !

Algorithm CONSTANT_FLOW

- **Inputs**:
  - A time varying sequence of n images: E_1,E_2,…,E_n
- **Parameters**:
  - N: dimension of region Q
    - typically N=5
  - σ_s: stdev for Gaussian filter along x and y
    - typically 1.4 pixels-> w=2k+1=5\sigma_s =7 -> k = 3
    - discard borders of width k
  - σ_t: stdev for Gaussian filter along t
    - typically 1.4 frames-> w=2k+1=5\sigma_s =7 -> k = 3
    - discard first and last k frames

Algorithm CONSTANT_FLOW

- For each image:
  - Filter along x and y with a Gaussian filter
  - Filter along t with a Gaussian filter
  - Compute E_x,E_y and E_t
- For each pixel of each image:
  - Compute the matrix A and vector b
  - Compute OF $v = (A^T A)^{-1} A^T b$
  - Output $v=(v_x,v_y)^T$ for each pixel

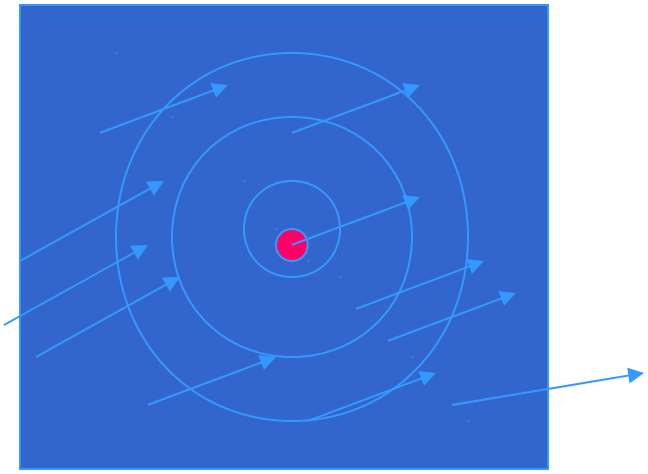
Feature-based Techniques

Feature Matching

- Estimates OF only at localized features
- Produces SPARSE OF mappings
- Two approaches:
  - Two-Frames methods
    - Similar to stereo
  - Multiple-Frames methods
    - Use previous frames to PREDICT location in next frame

An improvement …

- NOTE:
  - The assumption of constant OF is more likely to be wrong as we move away from the point of interest (the center point of Q)



Use weights to control the influence of the points: the farther from p, the less weight

Solving for v with weights:

- Let W be a diagonal matrix with weights
- Multiply both sides of Av = b by W:

$$W A v = W b$$

- Multiply both sides of WAv = Wb by $(WA)^T$:

$$A^T WWA v = A^T WWb$$

- $A^T W^2 A$ is square (2x2):
  - $(A^T W^2 A)^{-1}$ exists if $\det(A^T W^2 A) \neq 0$
- Assuming that $(A^T W^2 A)^{-1}$ does exists:

$$(A^T W^2 A)^{-1} (A^T W^2 A) v = (A^T W^2 A)^{-1} A^T W^2 b$$

$$v = (A^T W^2 A)^{-1} A^T W^2 b$$

Motion Parallax

Two points $\mathbf{P} = [X, Y, Z]^\top$ and $\bar{\mathbf{P}} = [\bar{X}, \bar{Y}, \bar{Z}]^\top$

are coincident at time t. So their projected image points are the same. $\mathbf{p} = \bar{\mathbf{p}} = [x, y]^\top$.

Their corresponding motion field vectors are

$$v_x = v_x^T + v_x^\omega$$
$$v_y = v_y^T + v_y^\omega$$

$$\bar{v}_x = \bar{v}_x^T + \bar{v}_x^\omega$$
$$\bar{v}_y = \bar{v}_y^T + \bar{v}_y^\omega.$$

Their rotational components become the same (why?)

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

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

Motion Parallax

The difference of the motion field vectors gives

$$\Delta v_x = v_x^T - \bar{v}_x^T = (T_z x - T_x f)\left(\frac{1}{Z} - \frac{1}{\bar{Z}}\right)$$

$$\Delta v_y = v_y^T - \bar{v}_y^T = (T_z y - T_y f)\left(\frac{1}{Z} - \frac{1}{\bar{Z}}\right).$$

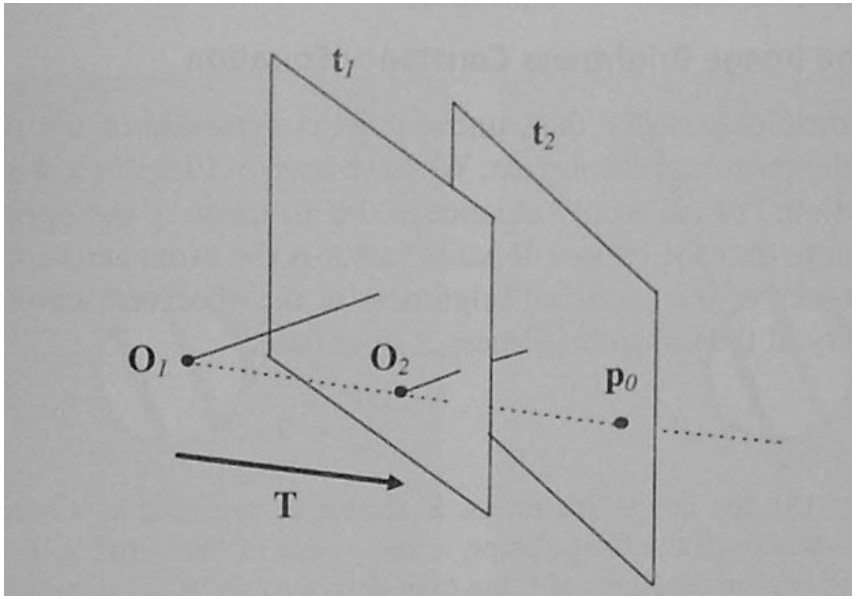
And the ratio between them is

$$\frac{\Delta v_y}{\Delta v_x} = \frac{y - y_0}{x - x_0}$$

with $[x_0, y_0]^\top$ image coordinates of $\mathbf{p}_0$, the vanishing point of the translation direction (Figure 8.5(b)).[8] Hence, *for all possible rotational motions, the vector* $(\Delta v_x^T, \Delta v_y^T)$ *points in the direction of* $\mathbf{p}_0$. Consequently, the dot product between the motion field, $\mathbf{v}$, and the vector $[y - y_0, -(x - x_0)]^\top$, which is perpendicular to $\mathbf{p} - \mathbf{p}_0$, depends neither on the 3-D structure of the scene nor on the translational component of motion, and can be written as

$$v_\perp = (y - y_0)v_x^\omega - (x - x_0)v_y^\omega.$$

Motion Parallax



Instantaneous epipole

Is the intersection of the image plane with the direction of the translation of the center of projection.

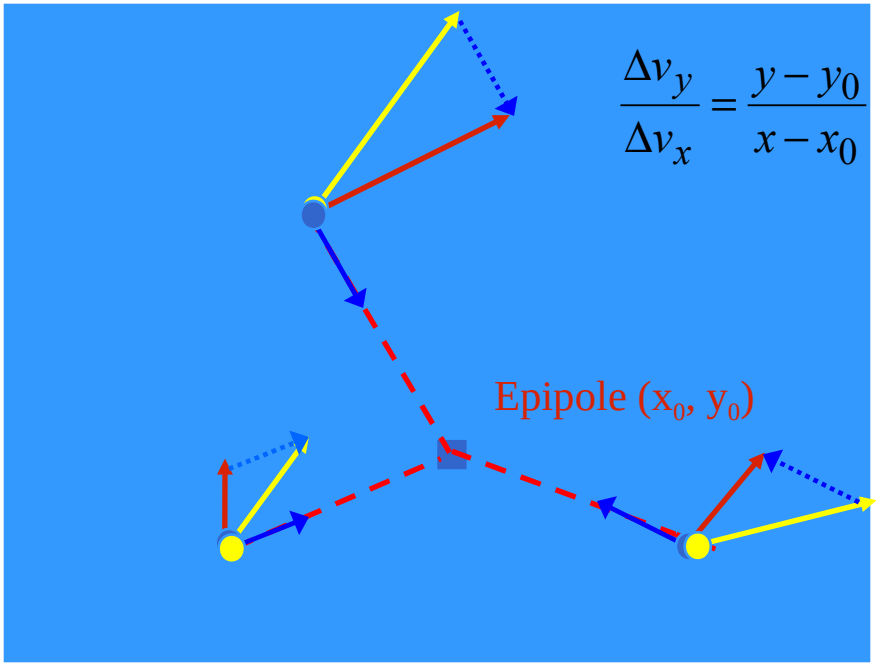
Motion Parallax

- The relative motion field of two instantaneously coincident points
  - Does not depend on the rotational component of motion
  - Points towards (away from) the vanishing point of the translation direction (the instantaneous epipole)

At instant t, three pairs of points happen to be coincident

The difference of the motion vectors of each pair cancels the rotational components

… and the relative motion field point in (towards or away from) the VP of the translational direction

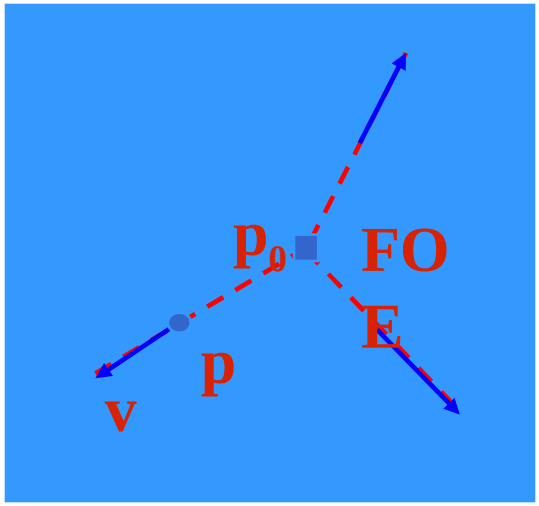


Motion Parallax

- [Observation 2] The motion field of two frames after rotation compensation

$$\frac{v_y^T}{v_x^T} = \frac{y - y_0}{x - x_0}$$

  - only includes the translation component
  - points towards (away from) the vanishing point p0 (the instantaneous epipole)
  - the length of each motion vector is inversely proportional to the depth,
  - and also proportional to the distance from point p to the vanishing point p0 of the translation direction (if Tz <> 0)

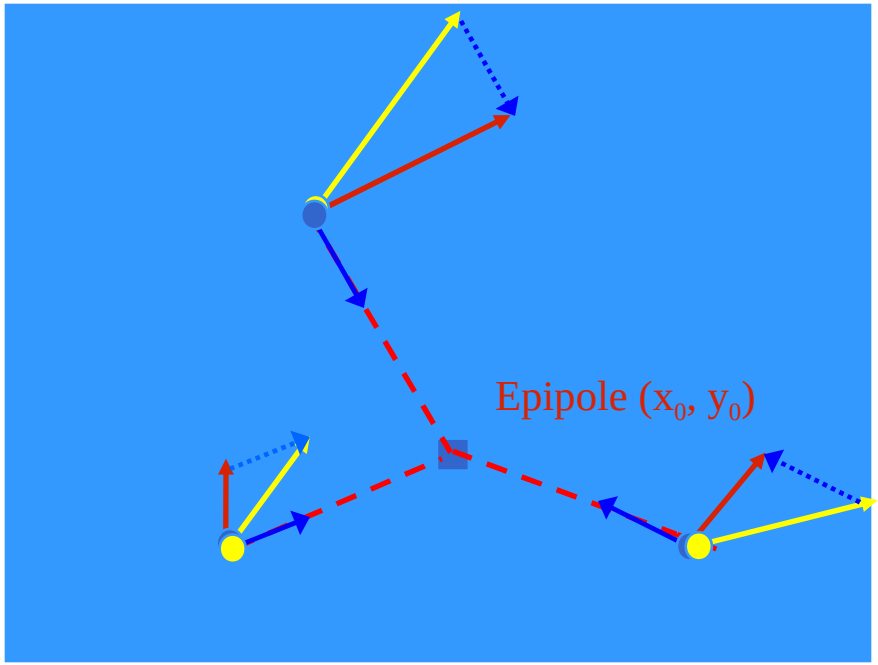


$$|\mathbf{v}| = \frac{T_z}{Z}\sqrt{(x - x_0)^2 + (y - y_0)^2}$$

How do we use MF: Structure From Motion

- 3D motion and structure from optical flow (p 208- 212)
  - Input:
    - Intrinsic camera parameters
    - dense motion field (optical flow) of single rigid motion
  - Algorithm
    - Stage 1: Translation direction
      - Epipole (x0, y0) through approximate motion parallax
      - Key: Instantaneously coincident image points
      - Approximation: estimating differences for ALMOST coincident image points

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \frac{f}{T_z}\begin{pmatrix} T_x \\ T_y \end{pmatrix}$$



$$\frac{\Delta v_y}{\Delta v_x} = \frac{y - y_0}{x - x_0}$$

Structure From Motion

- Stage 2: Rotation flow and Depth
  - Knowns: flow vector, and direction of translational component
  - One point, one equation (without depth)–
    - » Least square approximation of the rotational component of flow
  - From motion field to depth

(Figure 8.5(b)).[8] Hence, *for all possible rotational motions, the vector* $(\Delta v_x^T, \Delta v_y^T)$ *points in the direction of* $\mathbf{p}_0$. Consequently, the dot product between the motion field, $\mathbf{v}$, and the vector $[y - y_0, -(x - x_0)]^\top$, which is perpendicular to $\mathbf{p} - \mathbf{p}_0$, depends neither on the 3-D structure of the scene nor on the translational component of motion, and can be written as

$$v_\perp = (y - y_0)v_x^\omega - (x - x_0)v_y^\omega.$$

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

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

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

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

Structure From Motion

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

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

- Output
  - Direction of translation (f Tx/Tz, f Ty/Tz, f) = (x0, y0, f)
  - Angular velocity
  - 3-D coordinates of scene points

Motion Based Segmentation

- The simplest detection of motion is image differencing.

Algorithm CHANGE_DETECTION

The input is an image sequence, $I_1, I_2, \ldots, I_n$, and a positive real number, τ; for each image pair (I_k, I_{k+1})

1. Compute the pointwise image difference $\Delta_k(i, j) = I_{k+1}(i, j) - I_k(i, j)$;
2. if $|\Delta_k(i, j)| > \tau$, label pixel (i, j) of the frame k as moving.

The output is a map of the moving image regions.

When does this fail?