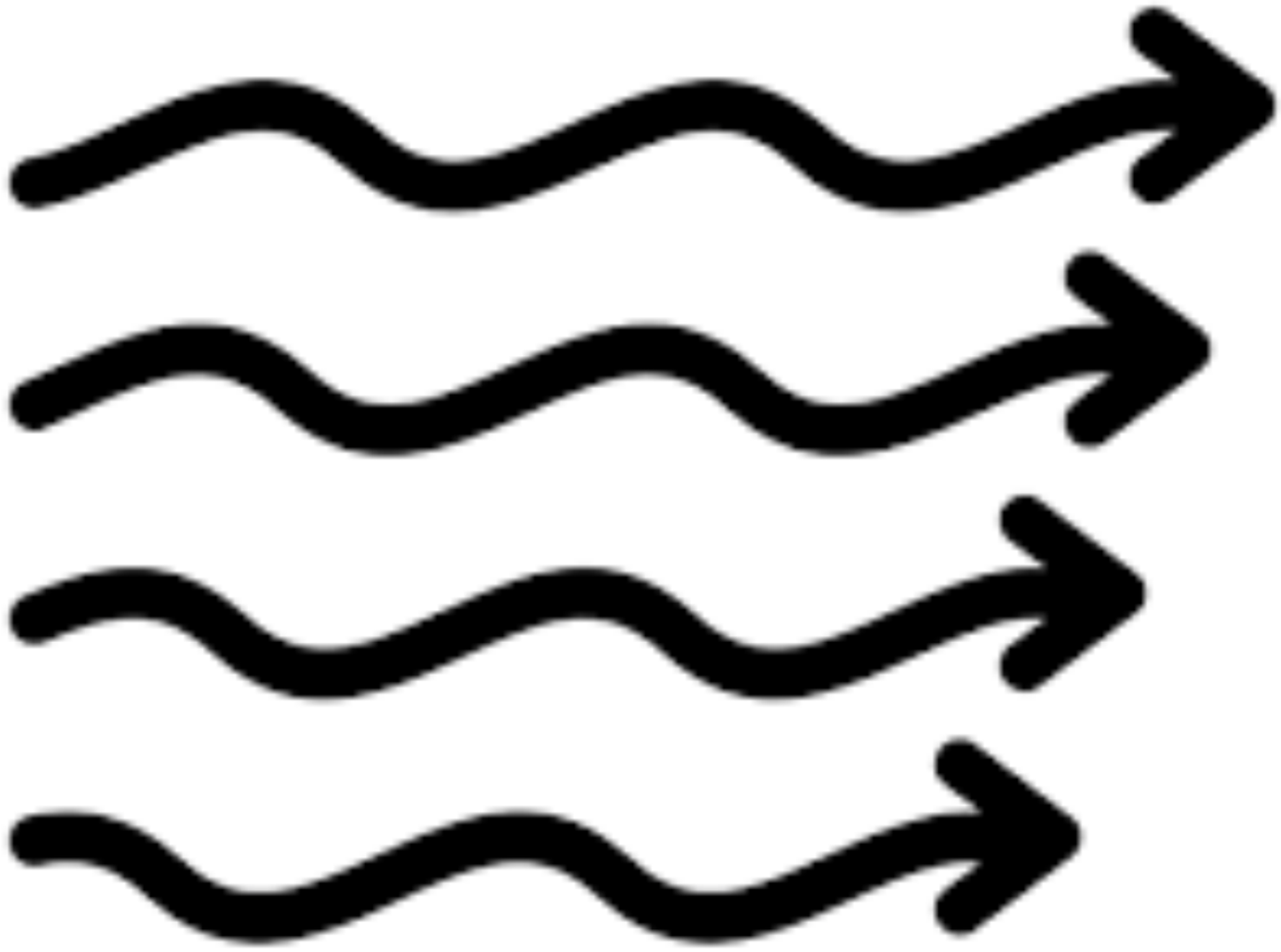
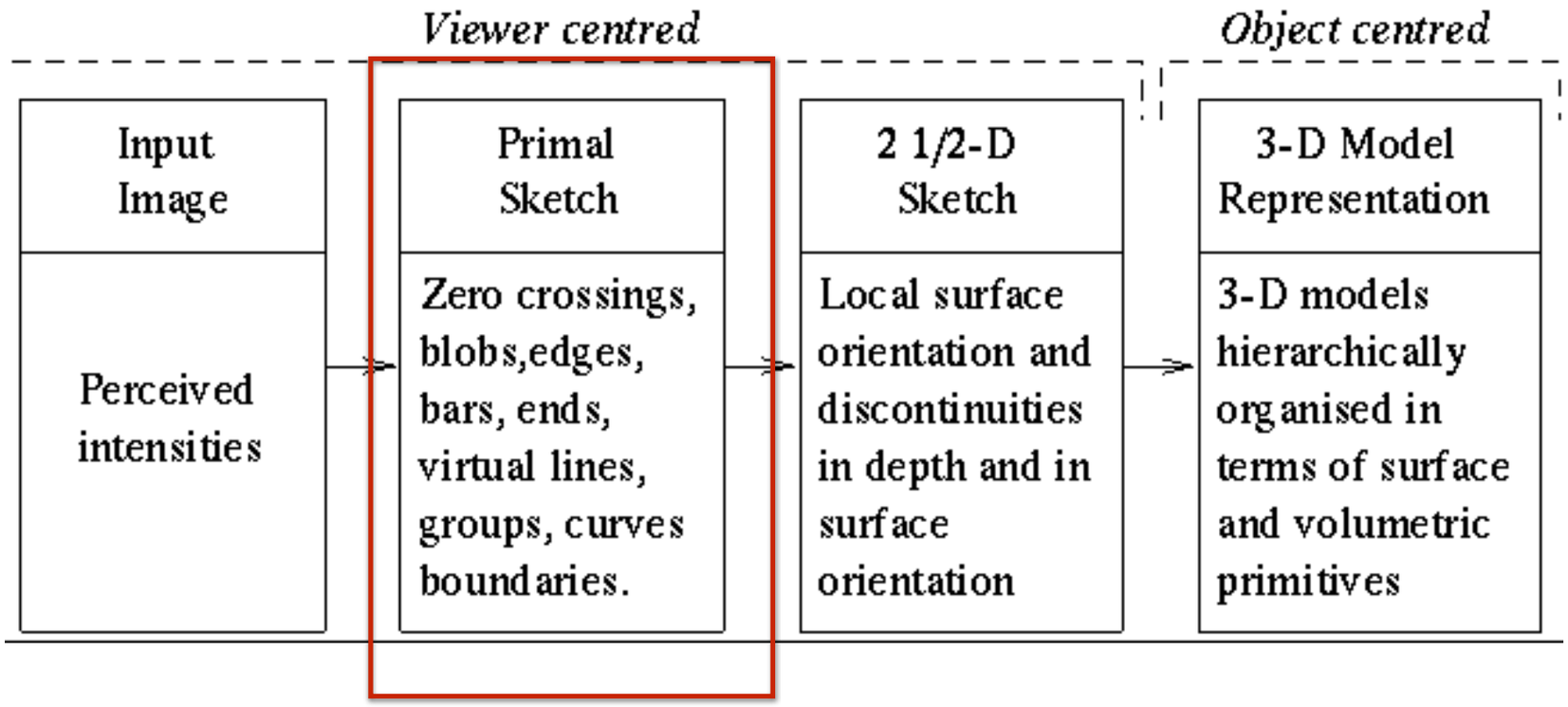
Motion and flow

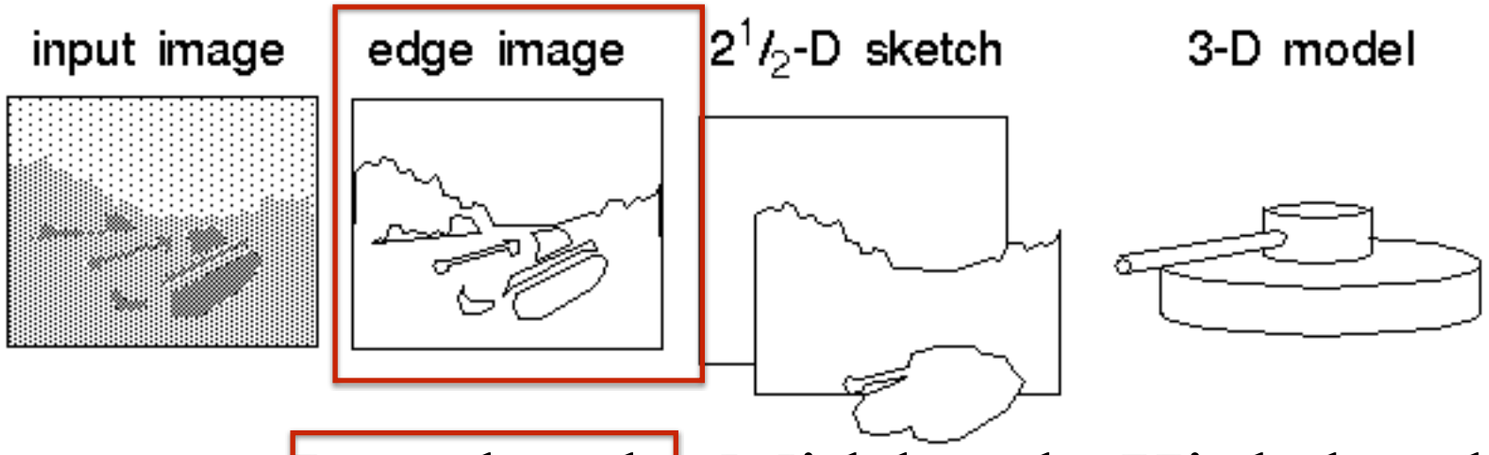


Logistics

- HW 2 has been released
- It is due on Oct 2 (in 2 weeks).

David Marr's Taxonomy of Vision





Outline

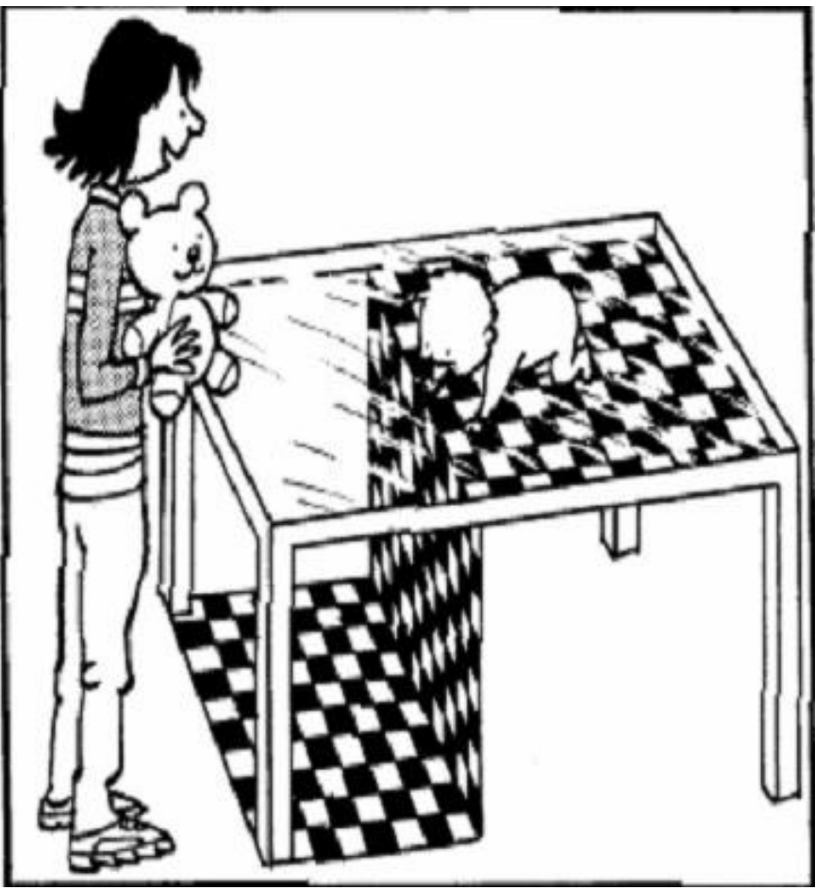
- Egomotion
  - Time-to-Contact, Parallax, Focus-of-Expansion
- Optical Flow

Moving is a part of life!

Sea squirt:

- The sea squirt moves through water and attaches itself to a rock or other stable structure.
- Once it finds a suitable place to attach itself, it “eats” its own brain (its brain is absorbed by its body).
- Being permanently attached to a home makes the sea squirt’s brain that control its locomotion unnecessary.
- This is an example of the connection between movement and intelligence in animals!

Human development:
Crawling teaches babies depth perception

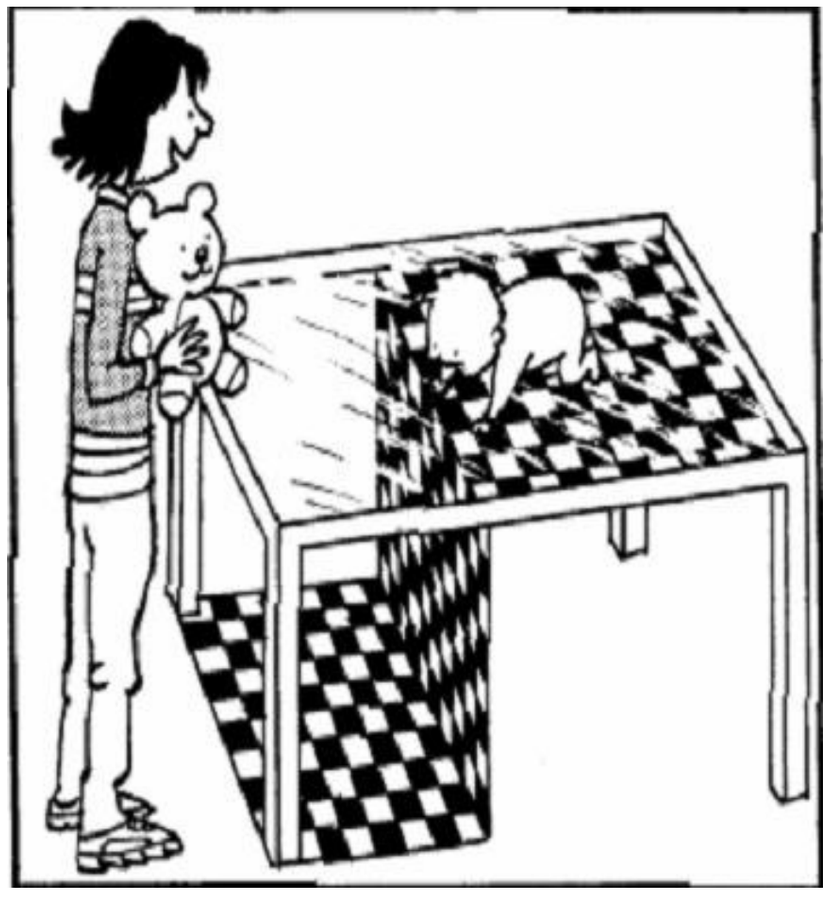


Gibson, 1960s

After 1 week of crawling:

After several weeks of crawling:

Today's goal

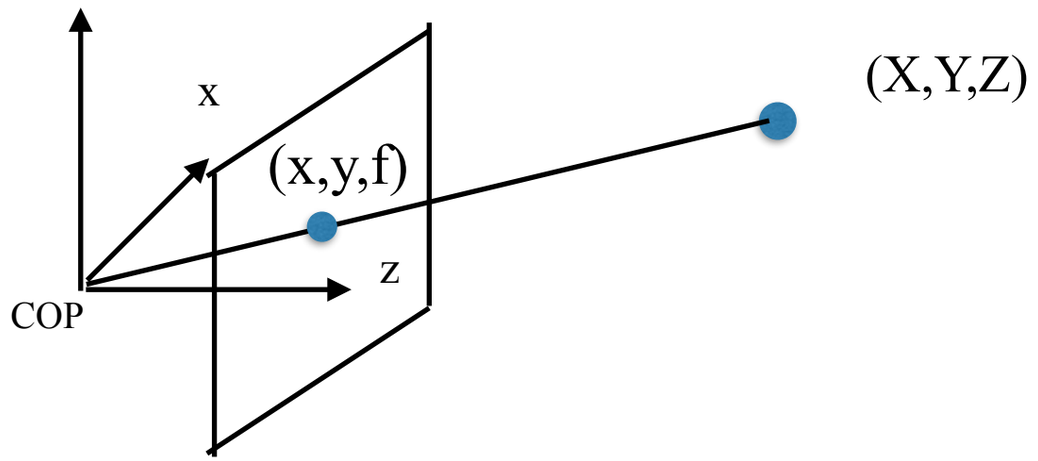


Understand the space of image transformations that we see when we move

egomotion - the task of estimating our own motion (e.g. from a video)

Optical Flow

Motion



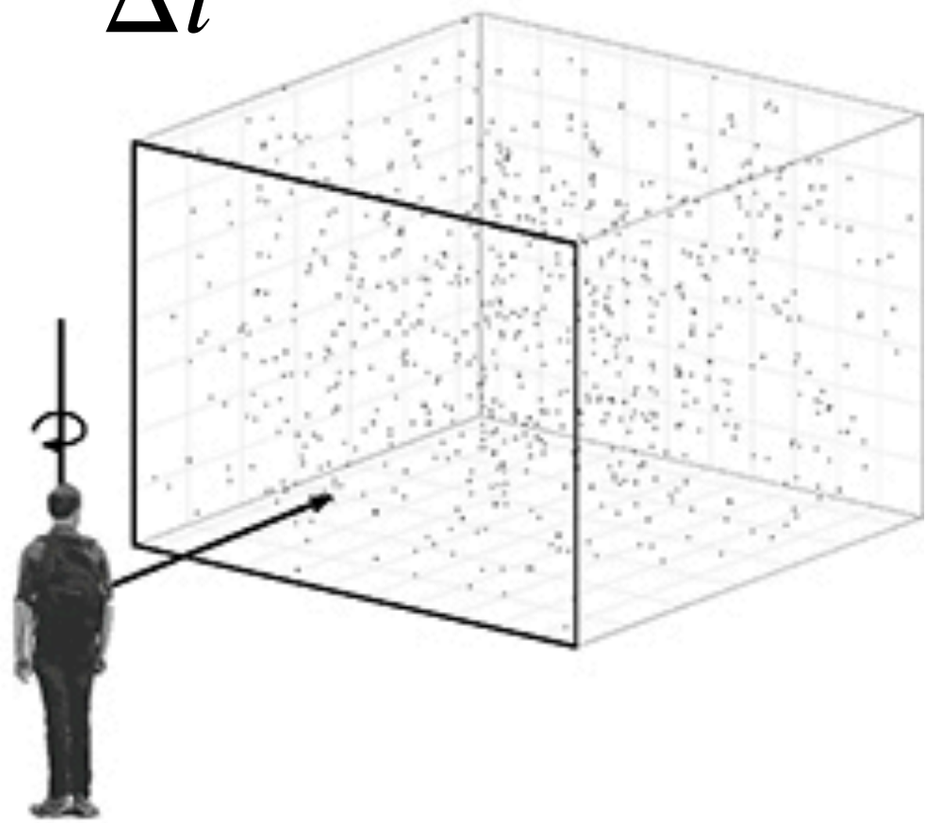
When a point (X, Y, Z) moves relative to the camera, its projection on the image (x, y) moves as well

This movement in the image plane is called "optical flow"

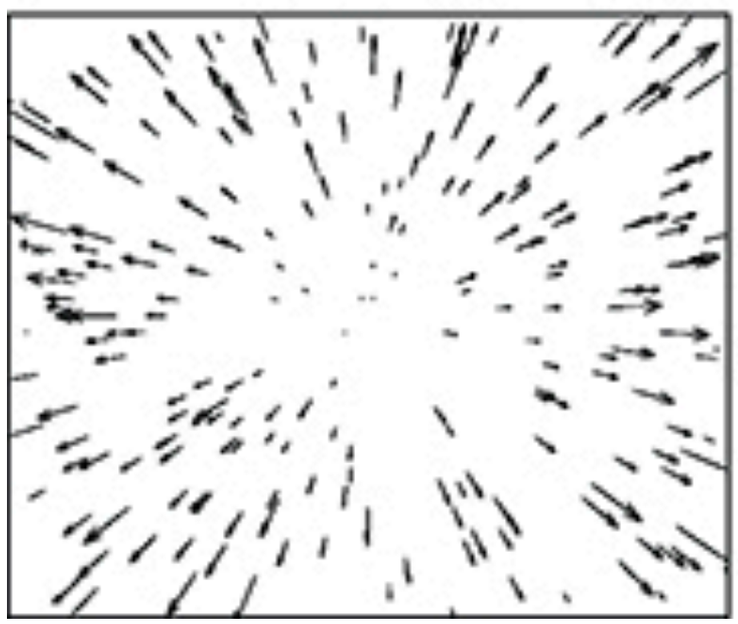
Suppose the point at pixel (x, y) moves to the pixel $(x + \Delta x, y + \Delta y)$ in time Δt, then the two components of the optical flow at (x, y) are:

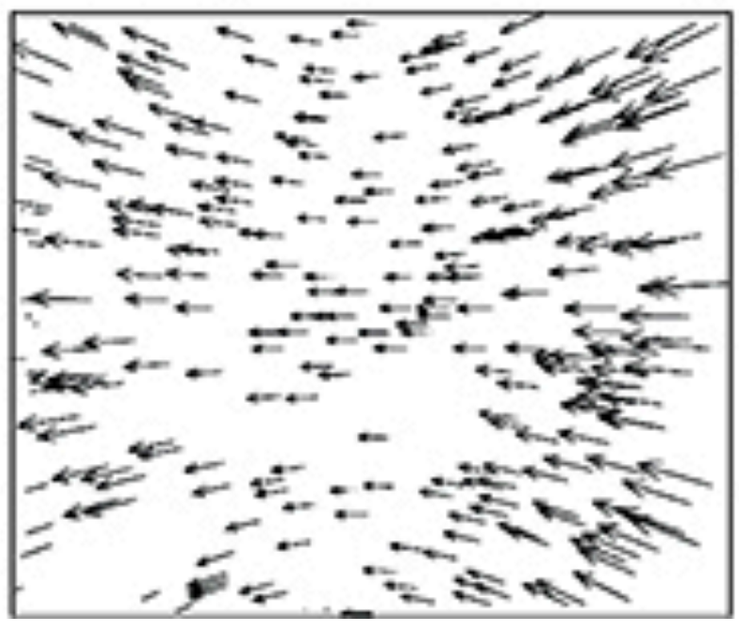
$$u = \frac{\Delta x}{\Delta t}, \quad v = \frac{\Delta y}{\Delta t}$$

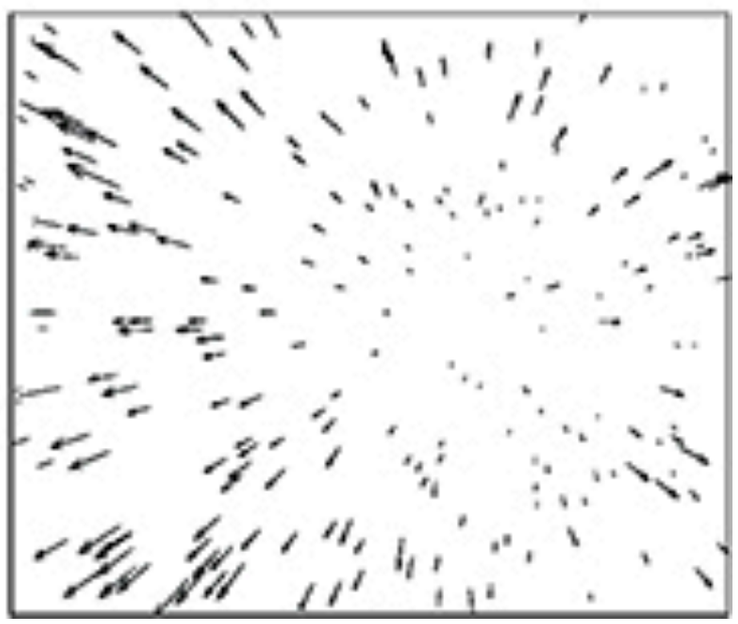
We can visualize $u = \dfrac{\Delta x}{\Delta t}, \quad v = \dfrac{\Delta y}{\Delta t}$ as a "flow field"



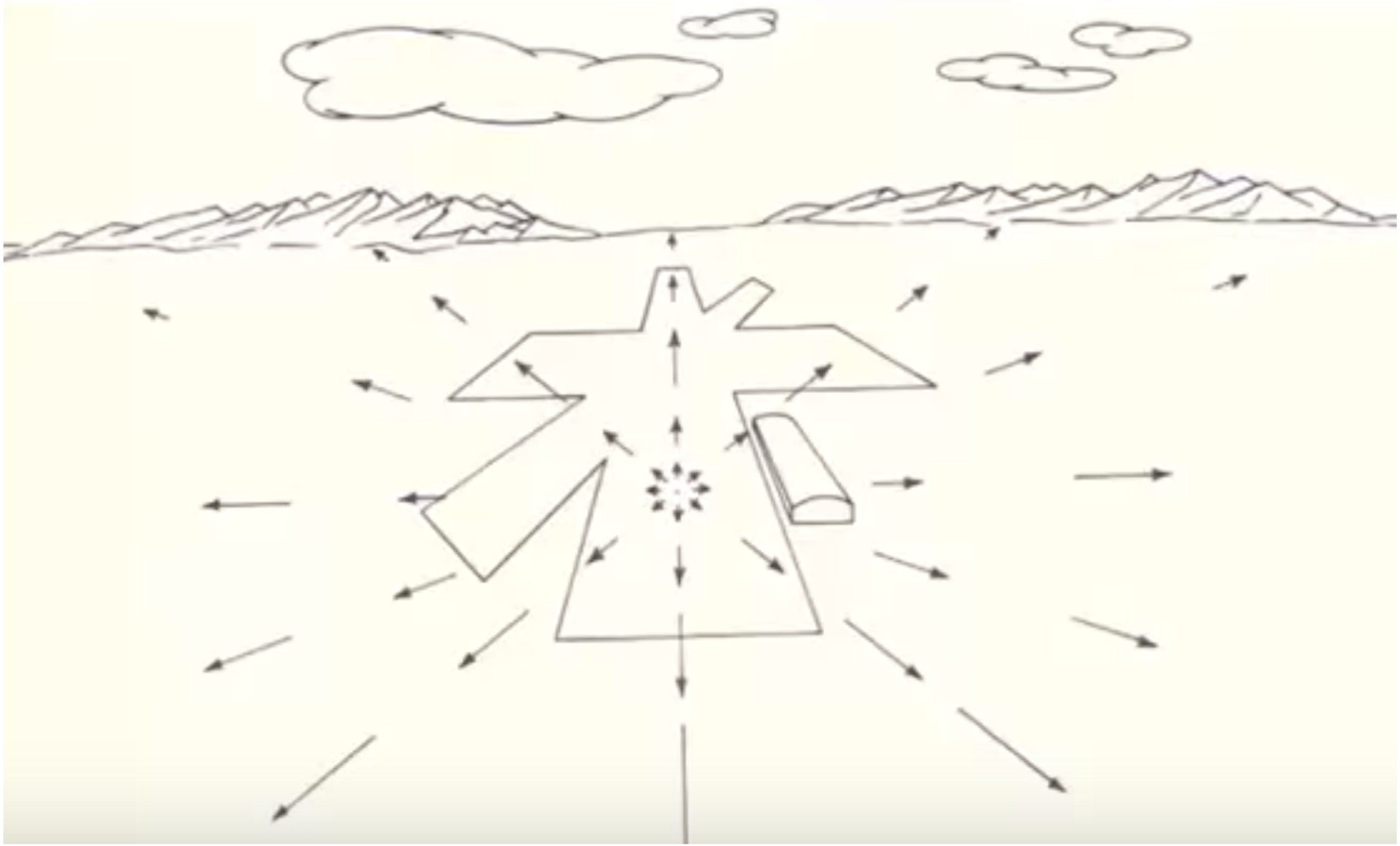
What direction do you think each of these flow fields correspond to?

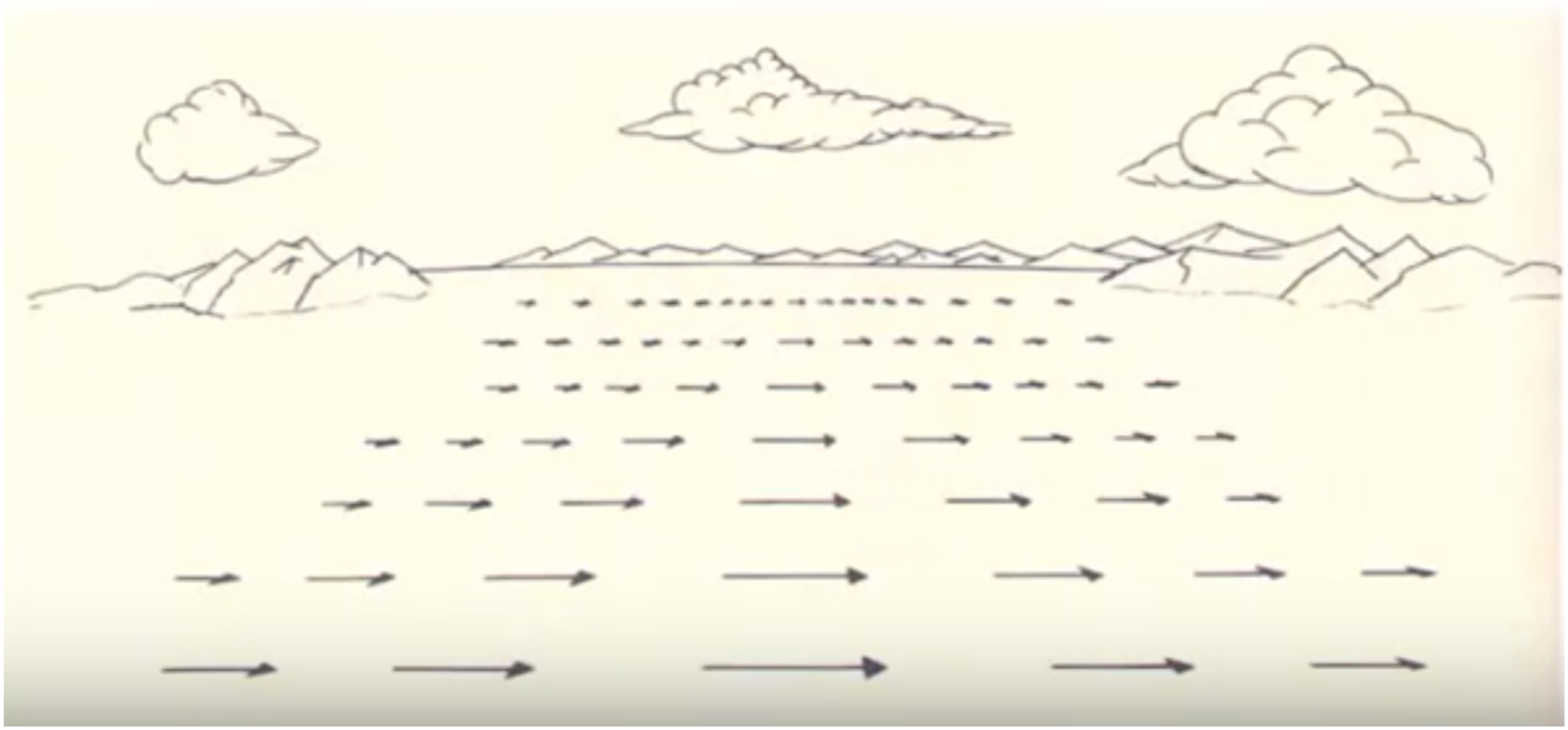






Flow field: Moving closer towards a point
We can determine the direction of motion by finding the “center of flow”





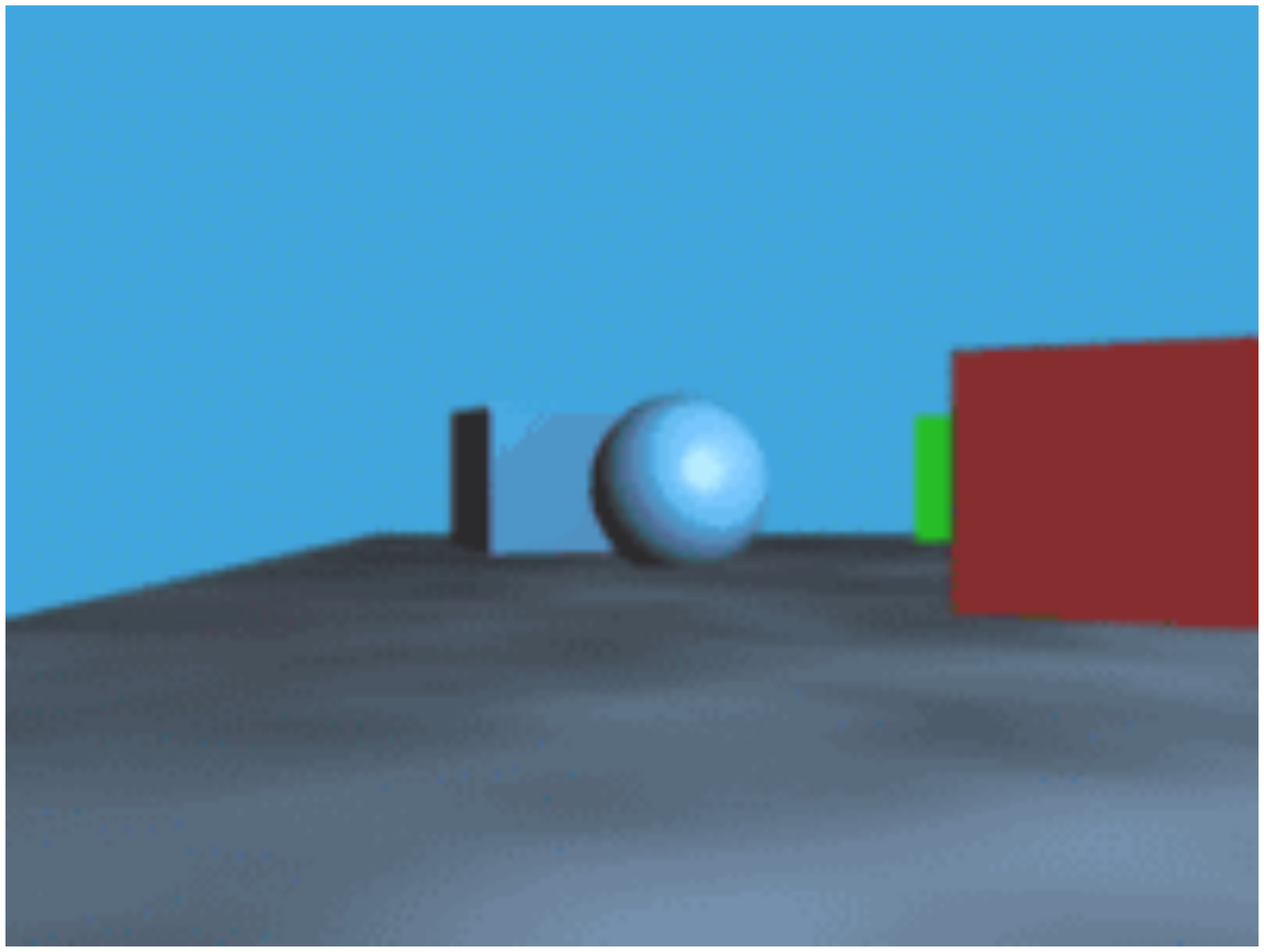
Why is the flow smaller for more distant objects?

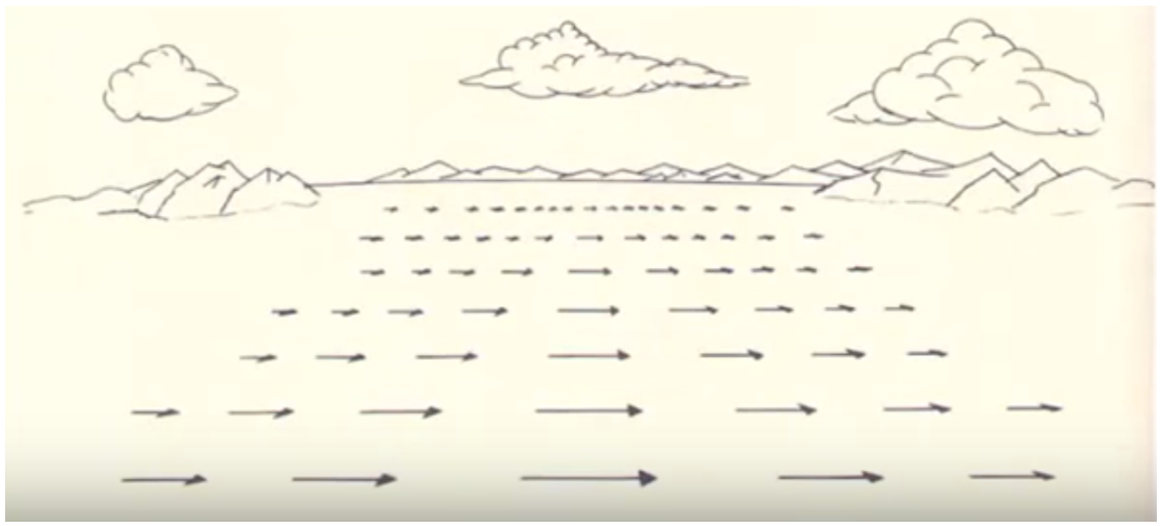
Apparent motion on the image is smaller due to perspective projection
(similar to how more distant objects appear smaller in the image)

Thus, we can use the size of the flow field of a static world / moving camera to estimate depth!

Flow due to translating cameras reveals depth about scene

This pixel motion is known an as “parallax”



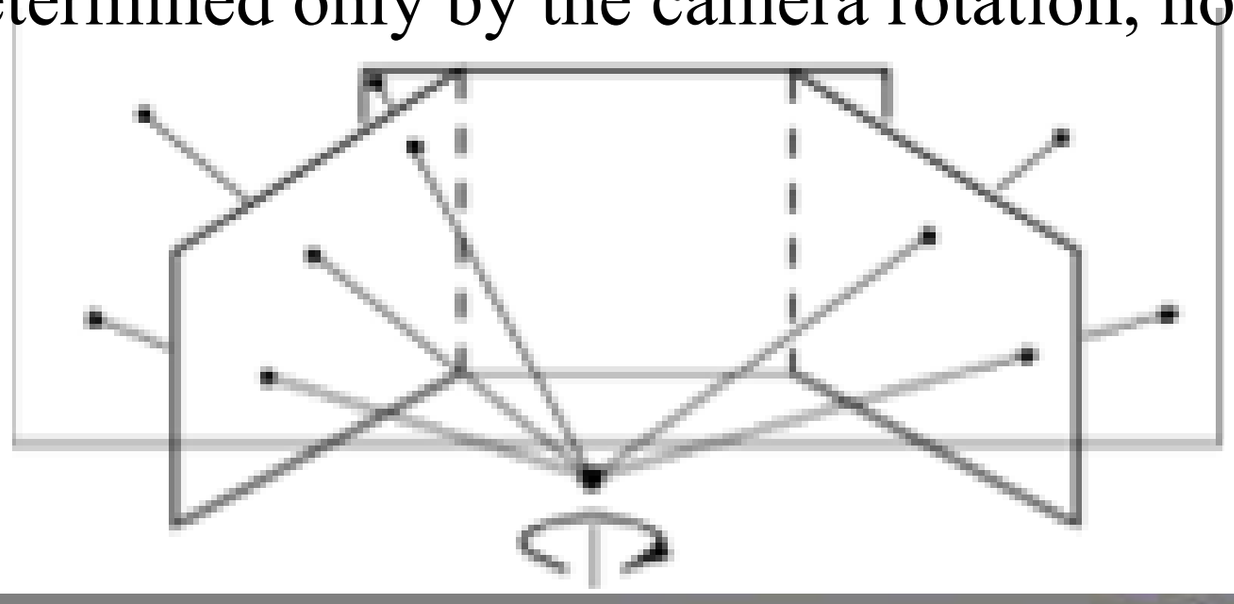


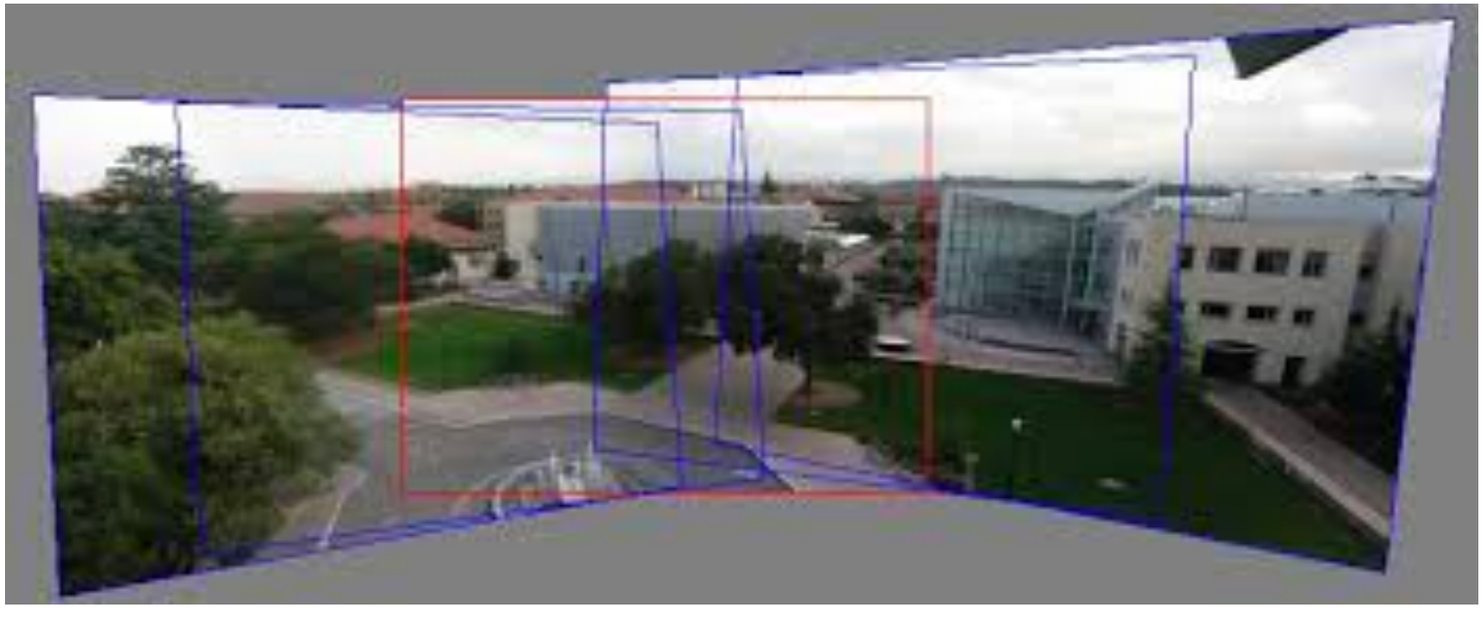
Rotating cameras

What does a rotating camera tell us about scene depth?

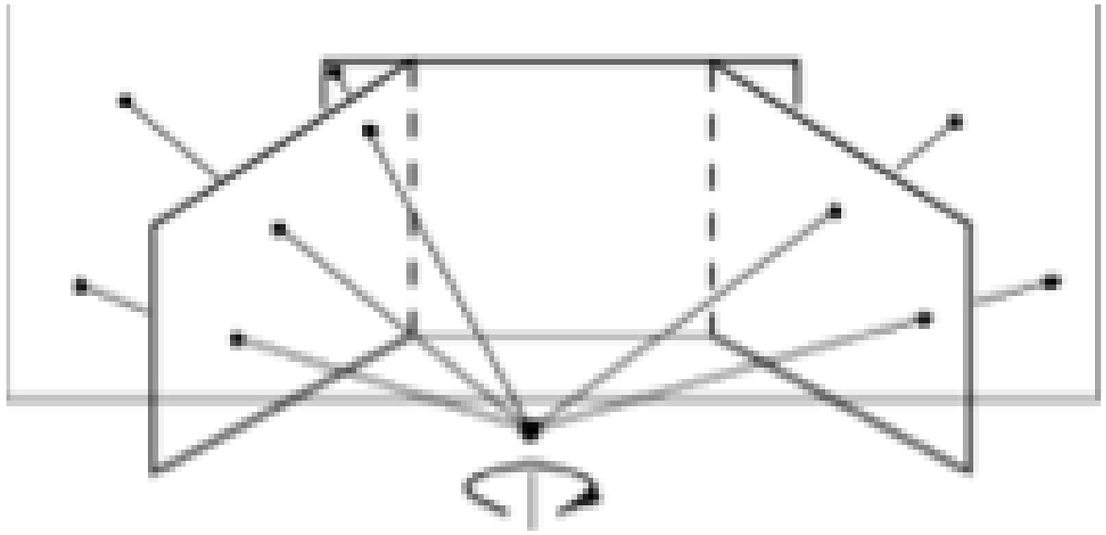
How do pixels change if we just rotate (not translate) the camera?

Since pixels are related by a homography transformation,
this reveals nothing about the scene depth!
The motion is determined only by the camera rotation, not the scene depth





Rotations about camera center



Relation between 3D camera coordinates: $\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = R \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$

Rotation matrix

3D->2D projection: $\lambda_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix}$

Instrinsic camera matrix *K*

Combining both: $\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = KRK^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$

This is a 3x3 homography!

Scene depth is not part of the equation

This is why 360 degree cameras work

Summary

- Camera rotation causes apparent object movement, **independent** of the depth of the scene
- Camera translation causes apparent object movement, **dependent** of the depth of the scene

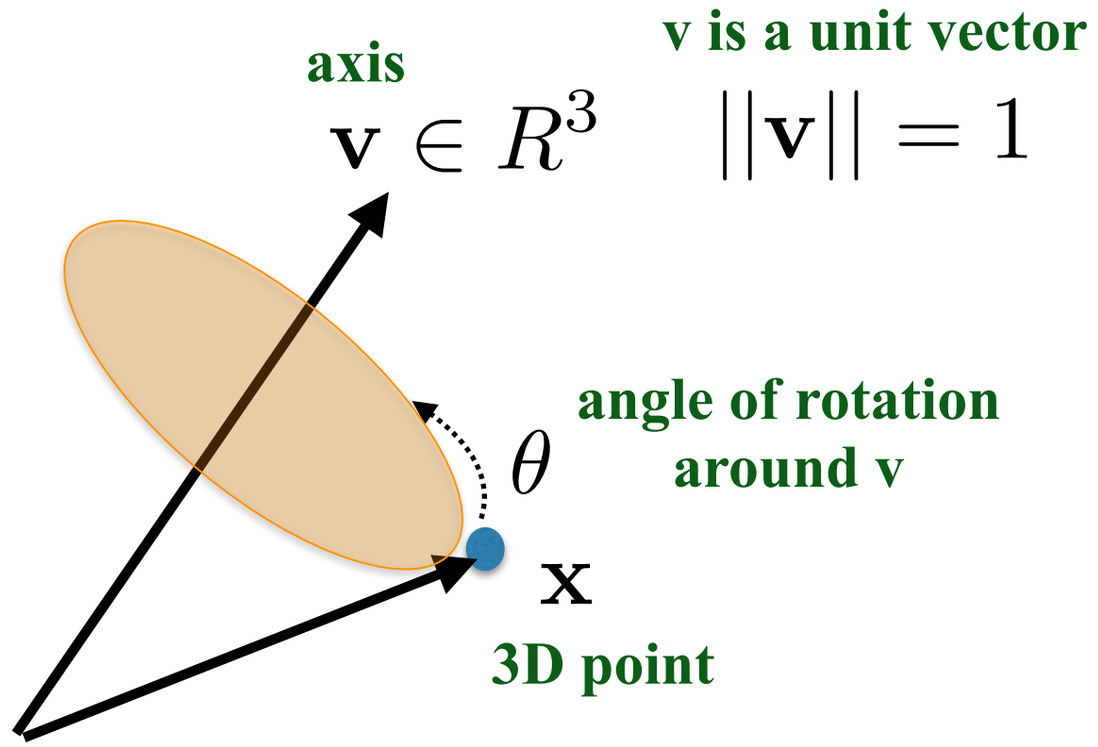
Roadmap

We want to describe the relationship between camera movement, depth, and flow.

Here's how we will do this:

- Let's parameterize rigid-body camera movement by translation **t** and a rotation
- We will look at flow-fields produced by a small camera movement:
  - Flow fields: u(x,y) and v(x,y) for every pixel (x,y)
    - u is the horizontal component of flow; v is the vertical component of flow
- We'll explicitly derive an equation that shows how flow depends on camera movement and the depth of each point

Axis-angle representation of rotations



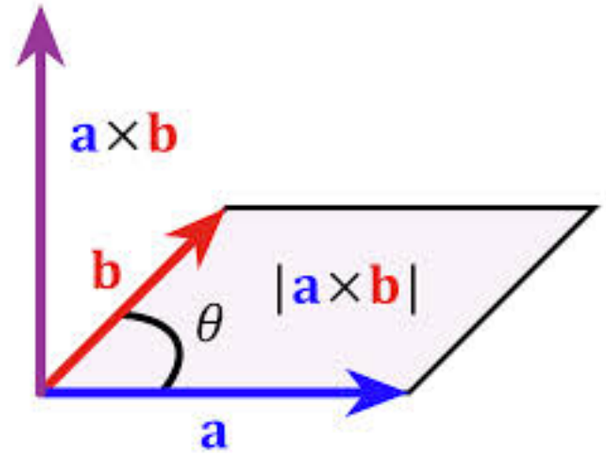
Can represent with 3 numbers as: θv

https://en.wikipedia.org/wiki/Axis-angle_representation

Review: cross products

Cross product: $\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta \, \mathbf{n}$

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ b_1 a_3 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$



(Cross product as a linear operation)

$$= \hat{\mathbf{a}}\mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Cross product *matrix*

Cross product *matrix*

Angular velocity

https://en.wikipedia.org/wiki/Angular_velocity

Notation switch: let's call the vector of the rotation axis as "angular velocity" ω

We can approximate the change in position due to a small rotation as:

$$\dot{\mathbf{X}} = \omega \times \mathbf{X} = \hat{w}\mathbf{X}$$

linear motion from angular velocity

Recall: exponentials are solutions of linear differential equations

$$\dot{x}(t) = ax(t)$$

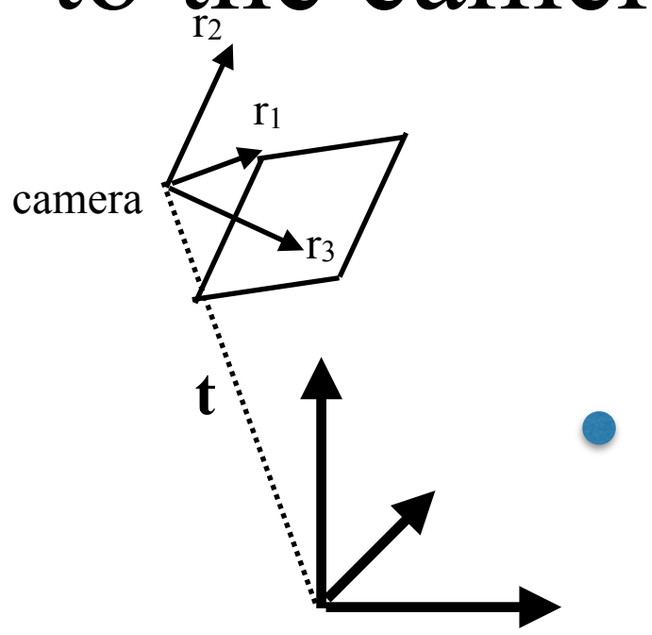
$$x(t) = e^{at}x(0)$$

Matrix exponentials are solutions of matrix linear differential equations

$$\dot{\mathbf{X}}(t) = \hat{\omega}\mathbf{X}(t)$$

$$\mathbf{X}(t) = e^{\hat{\omega}t}\mathbf{X}(0)$$

How does a fixed scene point **X** move with respect to the camera?



Suppose a camera is moving at translational velocity of **t** and rotating with angular velocity ω in the world frame

Then with respect to the camera frame, the point is moving with a linear velocity of:

$$\dot{\mathbf{X}} = -\mathbf{t} - \omega \times \mathbf{X}$$

Point's velocity in camera frame — **linear camera velocity** — **linear motion from camera's angular velocity** — **negative sign since the point appears to move in the opposite direction from the camera motion**

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} \omega_y Z - \omega_z Y \\ \omega_z X - \omega_x Z \\ \omega_x Y - \omega_y X \end{bmatrix}$$

expand out cross-product

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} \omega_y Z - \omega_z Y \\ \omega_z X - \omega_x Z \\ \omega_x Y - \omega_y X \end{bmatrix}$$

If we assume f=1, $x(t) = \frac{X(t)}{Z(t)}$ and $y(t) = \frac{Y(t)}{Z(t)}$ (Projection equations)

x image coordinate over time **y image coordinate over time**

Then what is dx/dt, dy/dt? **change in image coordinates over time**

$$\dot{x} = \frac{\dot{X}Z - Z\dot{X}}{Z^2}, \quad \dot{y} = \frac{\dot{Y}Z - Z\dot{Y}}{Z^2}$$

quotient rule

image velocity in x-direction, "u" **image velocity in y-direction, "u"**

Plug in equations for $\dot{X}, \dot{Y}, \dot{Z}, \frac{X}{Z} = x, \frac{Y}{Z} = y$

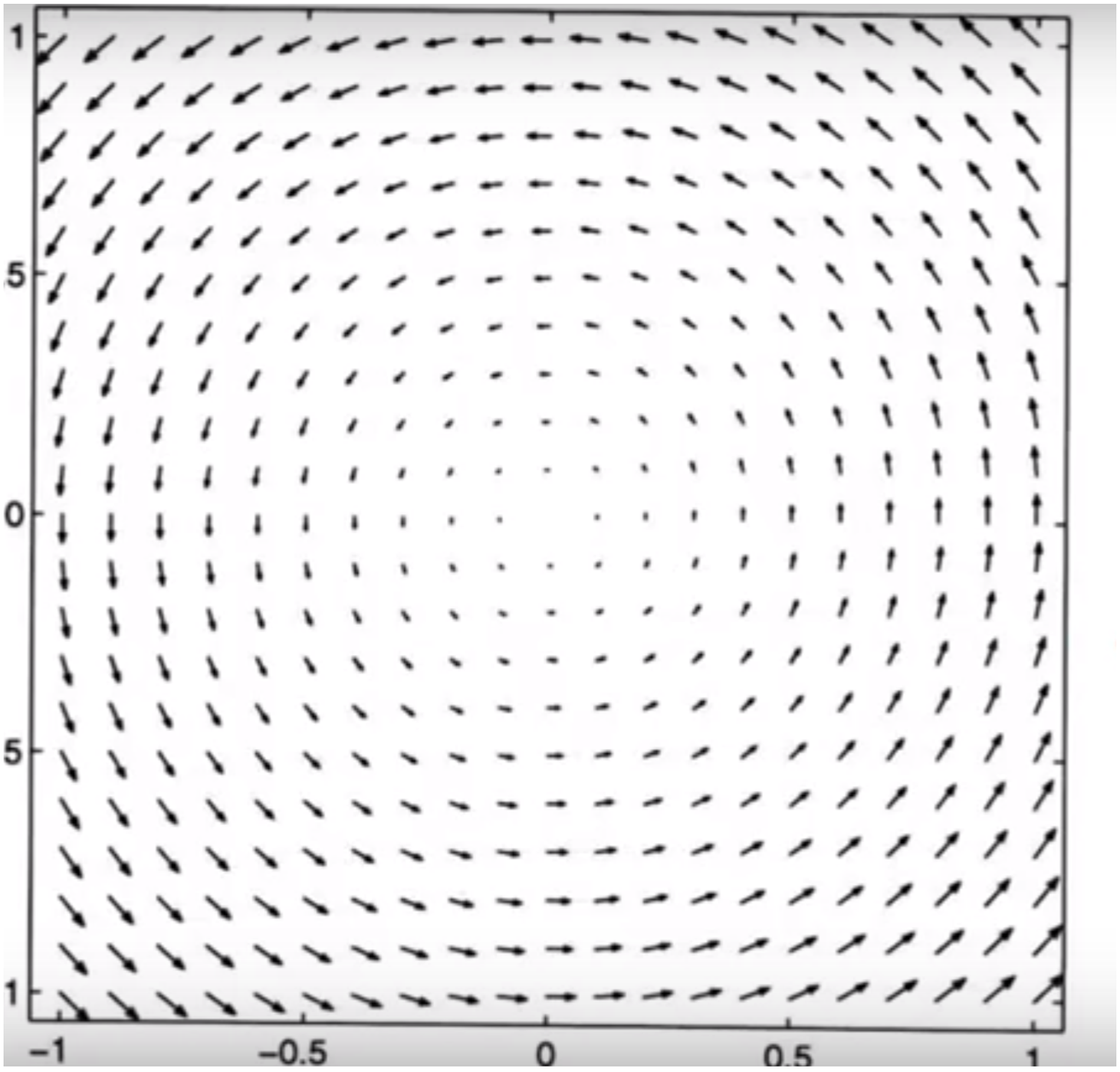
Final result:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

velocity of image coordinates **depth** **camera translation** **(no depth)** **camera rotation**

This is the important part!

Example: What does the flow look like for rotating about the z-axis?



Flow does not depend on object depth

Egomotion optical flow

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

velocity of image coordinates — **depth** — **camera translation** — **(no depth)** — **camera rotation**

velocity of image coordinates — **depth** — **translation part** — **rotation part**

$$u(x,y) = \frac{1}{Z(x,y)}(-t_x + xt_z) + xy\omega_x - (1+x^2)\omega_y + y\omega_z$$

$$v(x,y) = \frac{1}{Z(x,y)}(-t_y + yt_z) + (1+y^2)\omega_x - xy\omega_y - x\omega_z$$

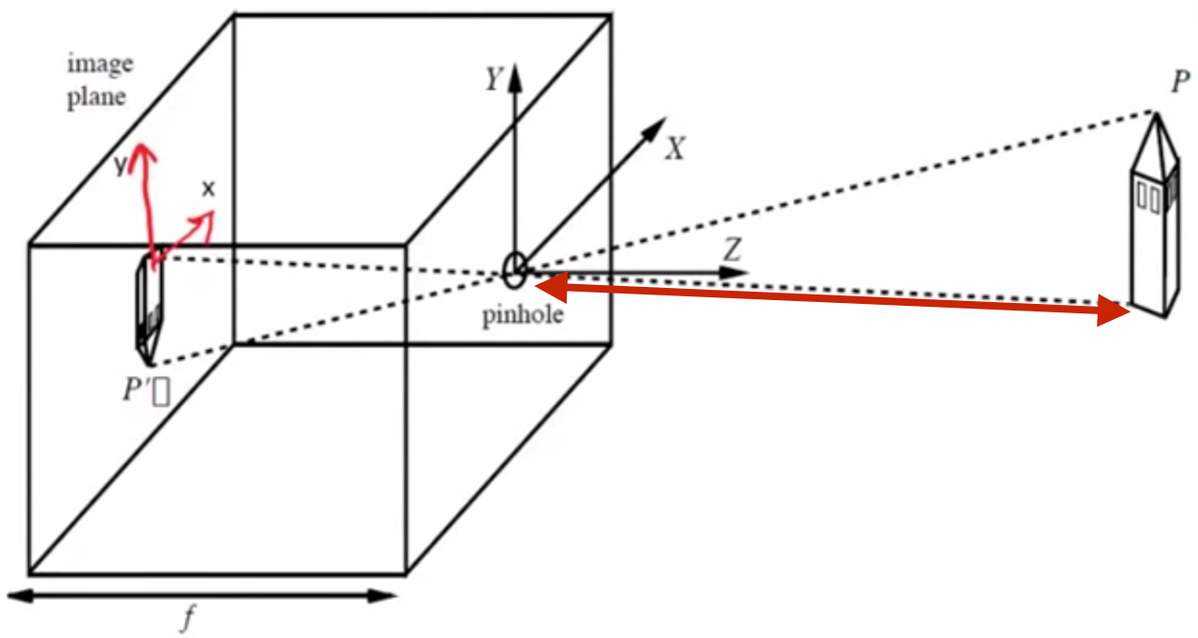
What would happen if there is no camera rotation (pure translation)?

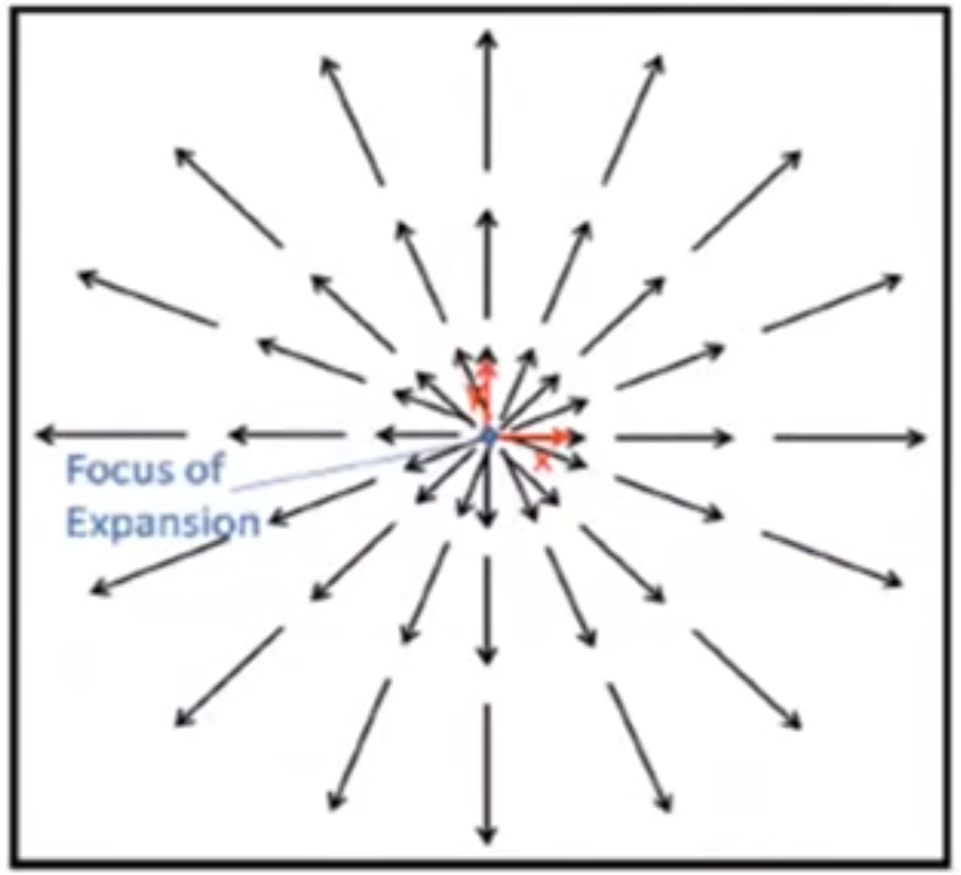
What would happen if we translate along the optical axis ($t_x = 0$, $t_y = 0$)? i.e. move camera forward

$$u(x,y) = \frac{xt_z}{Z(x,y)} \qquad v(x,y) = \frac{yt_z}{Z(x,y)} \qquad \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{t_z}{Z(x,y)} \begin{bmatrix} x \\ y \end{bmatrix}$$

What does this flow field look like?

Optical flow for translation along camera's z-axis





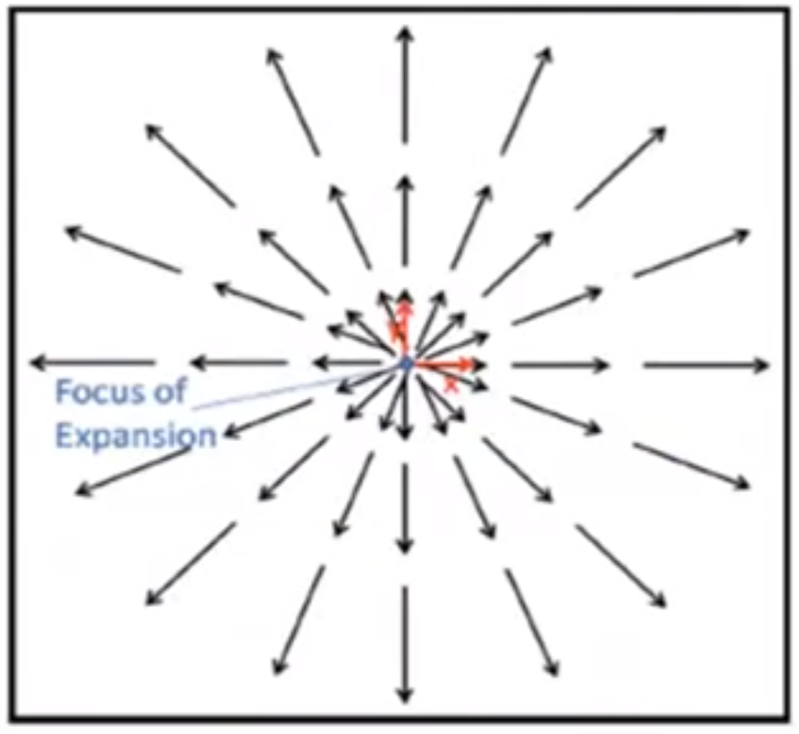
$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{t_z}{Z(x,y)} \begin{bmatrix} x \\ y \end{bmatrix}$$

Optical flow vector is a scalar multiple of the position vector
(points in the direction of x,y) - expands out!

(the scalar depends upon the speed of camera translation *and* the depth of the scene point)

What is the flow in the center of the image?

Moving toward a wall (constant distance to all points Z)



$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{t_z}{Z} \begin{bmatrix} x \\ y \end{bmatrix}$$

If speed (t_z) and distance (Z) to wall doubles, what happens to flow?

Nothing => fundamental scale ambiguity

If I am walking to a wall that is close by, or I am running towards a wall that is far away, I will get the same flow field!

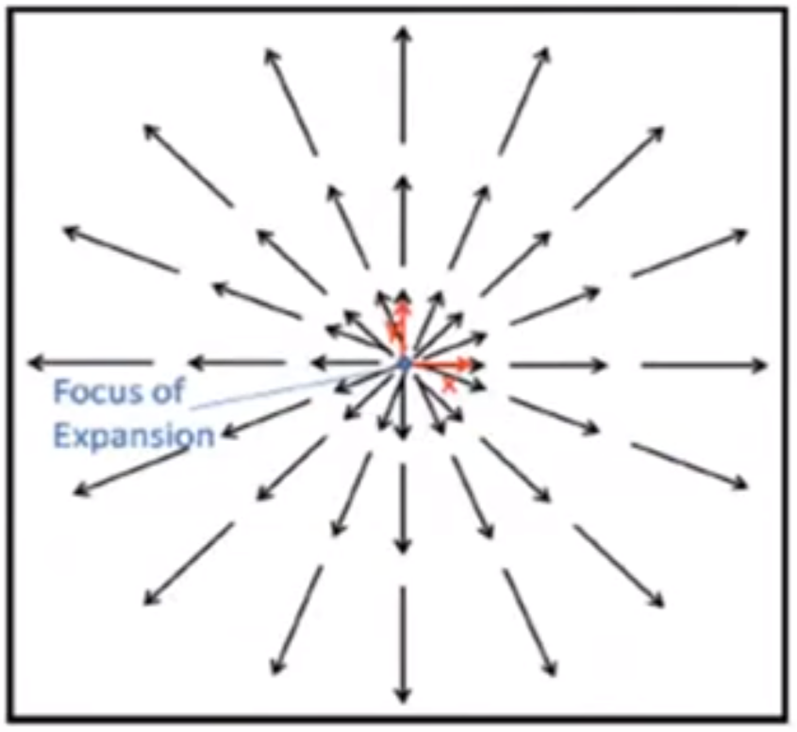
From a video, we cannot determine if the person holding the camera is walking or running if we do not know the depth of the scene!

We cannot determine the depth of the scene unless we know the speed of the camera!

Of course, with priors on depth and speed, we could guess both, but we won't know for sure

If we have a robot with a sensor to measure its own speed, we can estimate depth

Moving toward a wall (constant distance to all points Z)



$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{t_z}{Z} \begin{bmatrix} x \\ y \end{bmatrix}$$

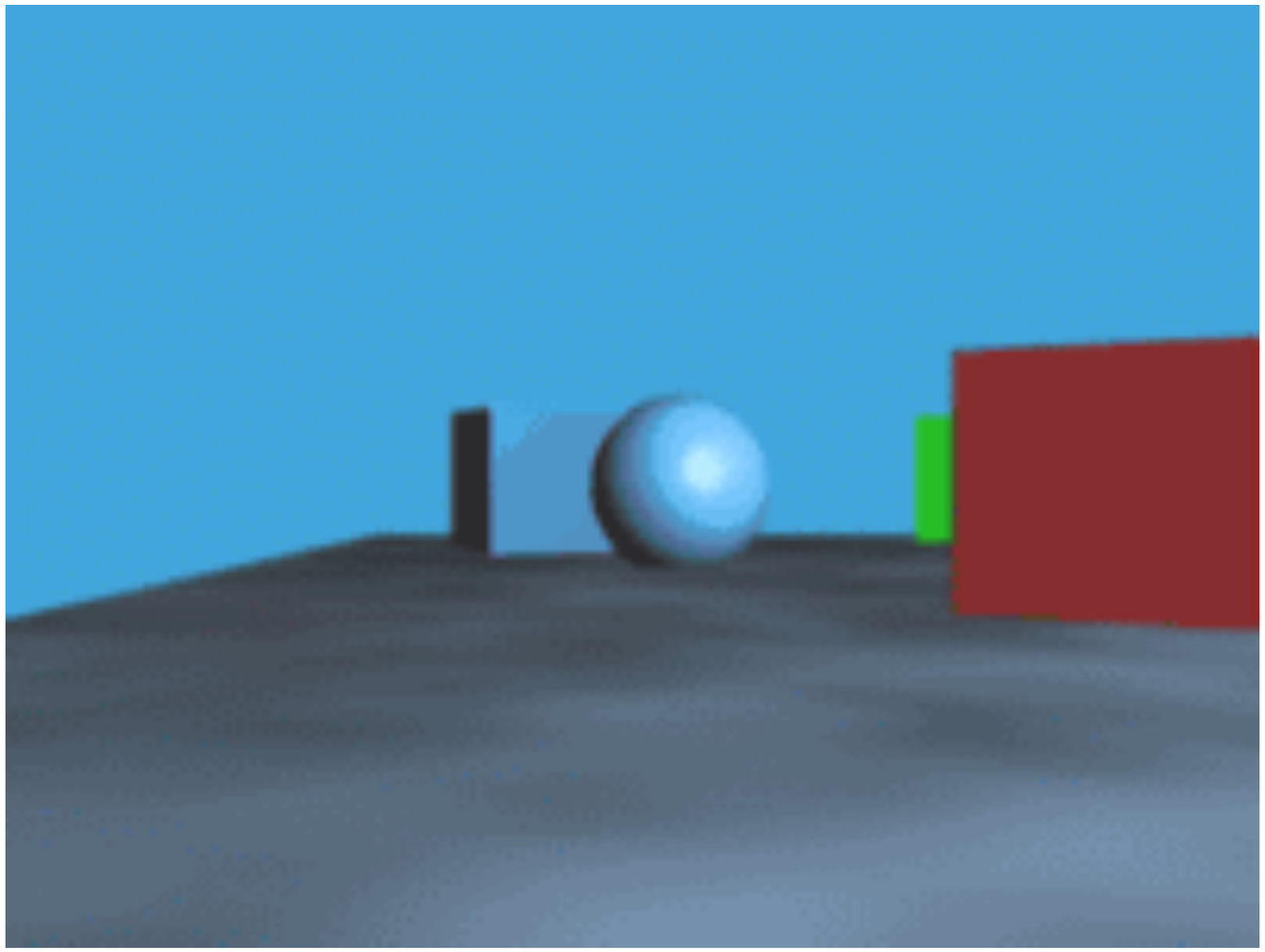
If we are traveling at 2 ft/sec, and the wall is 4 ft away, what's the time-to-contact?

Time to contact = Z / t_z

Implies that time-to-contact (assuming constant depth) can be computed from 2D flow!
(Even though we don't know the depth or speed)
Example: walking to a close all or running to a far wall

Thus we can avoid hitting things without estimating the depth of the scene,
just by estimating the flow!
(Primitive animals can do this)

Optical flow for translation along camera's x-axis ($t_y, t_z = 0$)



$$u(x,y) = \frac{-t_x + xt_z}{Z(x,y)}, v(x,y) = \frac{-t_y + yt_z}{Z(x,y)}.$$

$$v(x,y) = 0$$

$$u(x,y) = -t_x / Z(x,y)$$

Flow depends on depth and translational velocity!

Optical flow for general translation

$$u(x,y) = \frac{-t_x + xt_z}{Z(x,y)}, v(x,y) = \frac{-t_y + yt_z}{Z(x,y)}.$$

When is the flow (u,v) = (0,0) ?

$$x = \frac{t_x}{t_z} \qquad y = \frac{t_y}{t_z}$$

If the camera is translating at (t_x, t_y, t_z), then there will be some point where the flow is 0:

$$\left(\frac{t_x}{t_z}, \frac{t_y}{t_z}\right)$$

If I find the point (x,y) where the flow is 0, then I know the ratio of the flows!

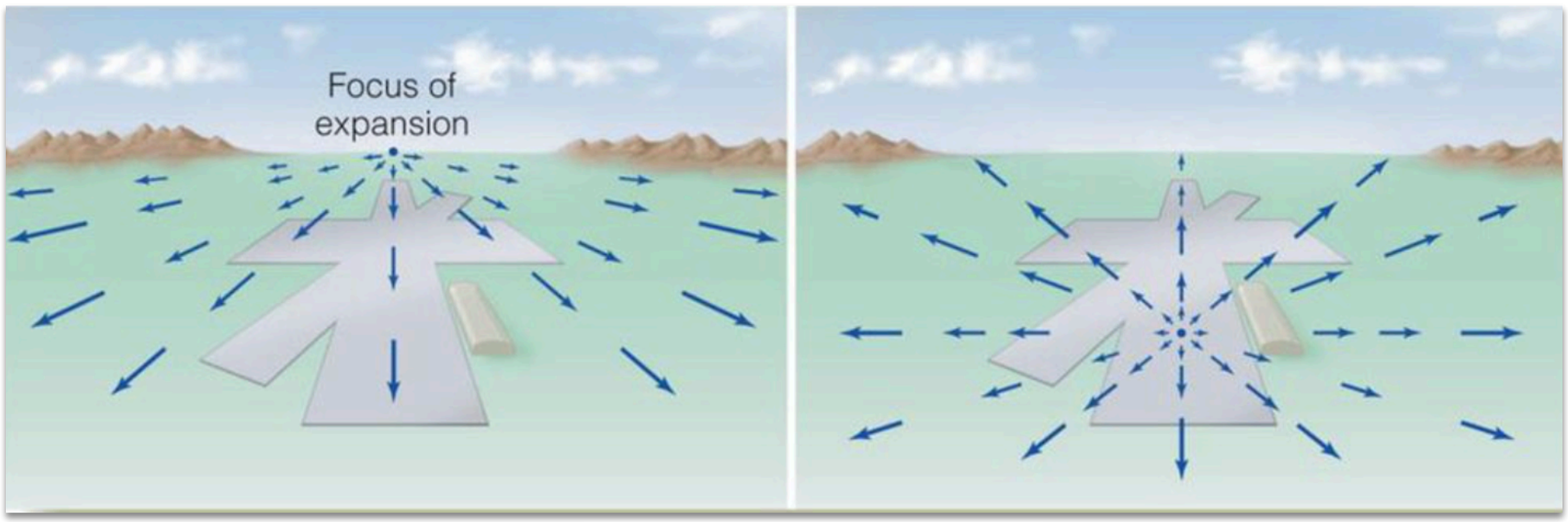
Let $t_z = \lambda$ Motion = $(\lambda x, \lambda y, \lambda) = \lambda(x, y, 1)$

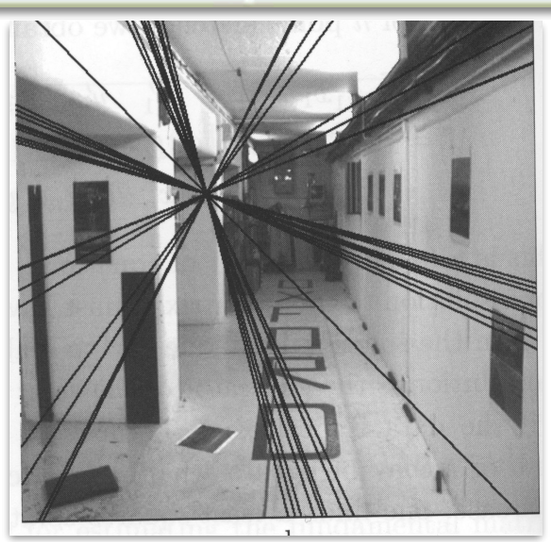
Thus, I know the direction that I am moving!

Focus of expansion

reveals the projection of camera translation velocity vector into image

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{t_x}{t_z} & \frac{t_y}{t_z} \end{bmatrix}$$

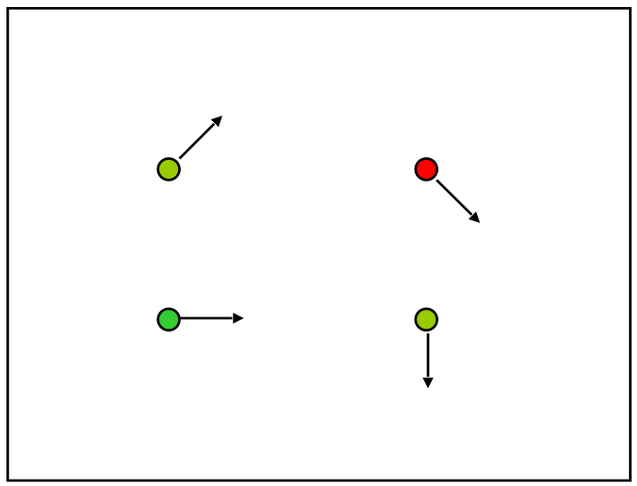




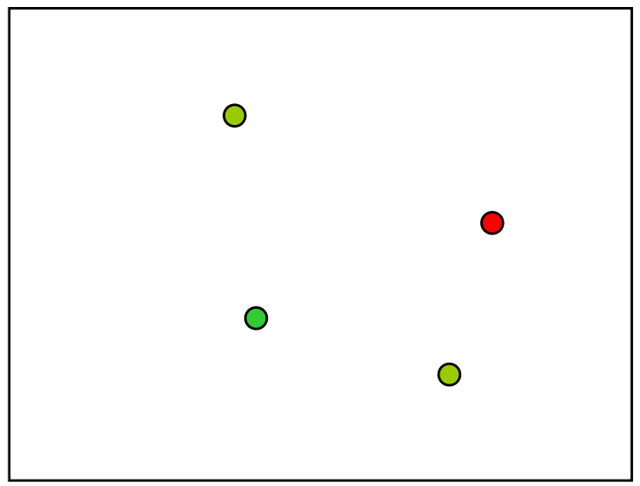
Summary

- If we can compute the 2D optical flow, then we can:
  - compute the time to contact
  - estimate the rotation of the camera
  - estimate the translation of the camera (if scene depth is known)
  - estimate the scene depth if camera translation is known

Problem Definition: Optical Flow



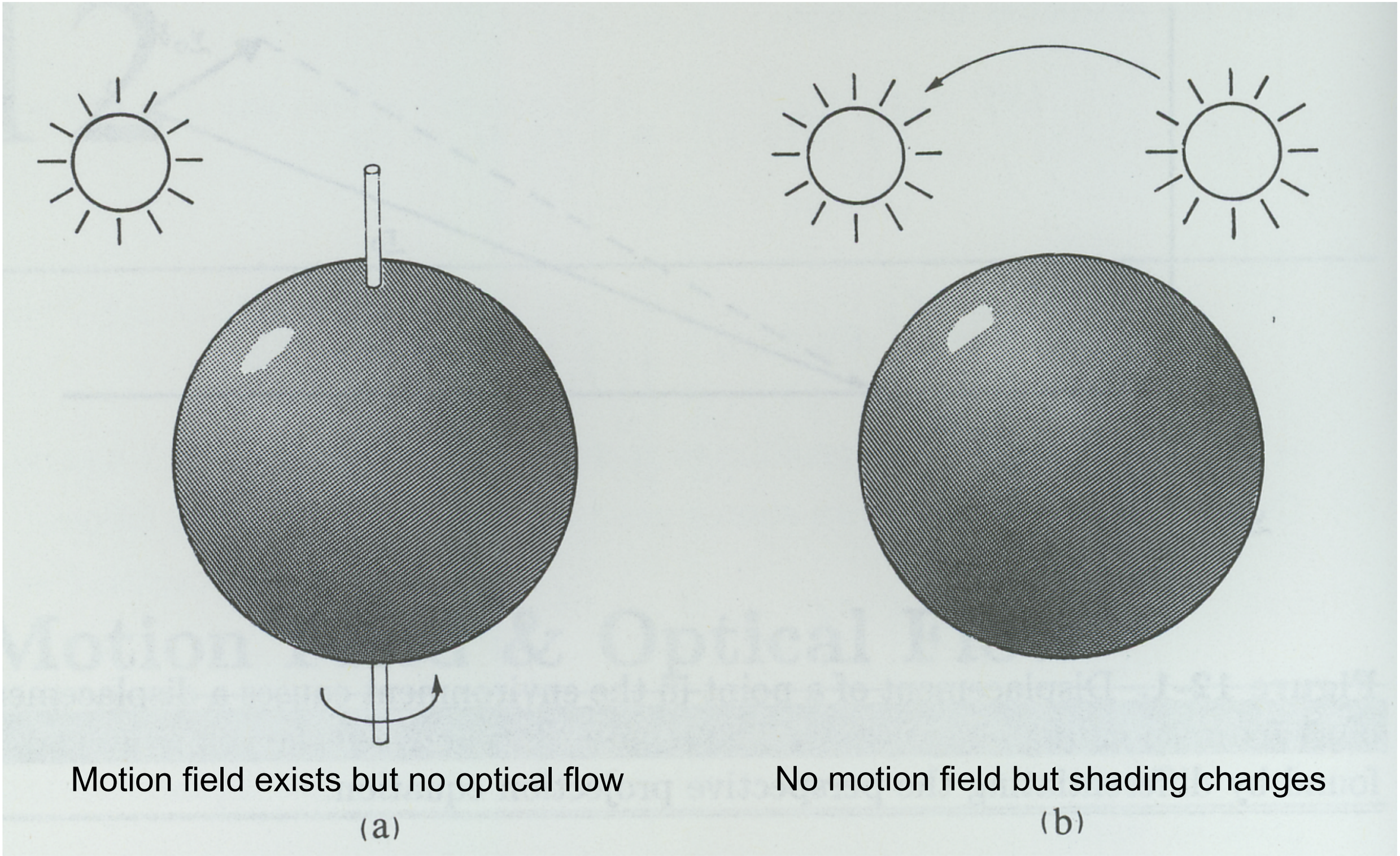
$H(x, y)$



$I(x, y)$

- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - Given a pixel in H, look for nearby pixels of the same color in I
- Key assumption
  - **color constancy**: a point in H looks "the same" in image I
    - For grayscale images, this is **brightness constancy**

Caution:
2D measured optical flow $\neq$ 3D scene flow



Motion field exists but no optical flow

(a)

No motion field but shading changes

(b)

General motion fields

Why do we want to compute motion fields beyond ego=motion?

Simple evidence that “single image” analysis won’t always suffice