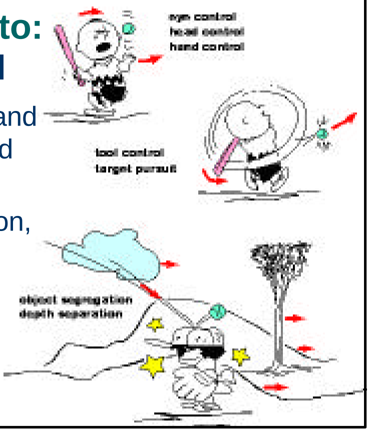
Detecting Motion and Optic Flow

Cmput 306
Lecture 28
Martin Jagersand

Image motion

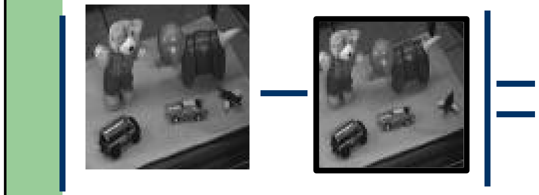
- Somehow quantify differences in image sequences.
- Image differences.
- Optic flow
- 3-6 dim motion

Motion is used to:

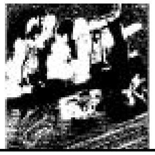


- **Attention:** Detect and direct using eye and head motions
- **Control:** Locomotion, manipulation, tools
- **Vision:** Segment, depth, trajectory

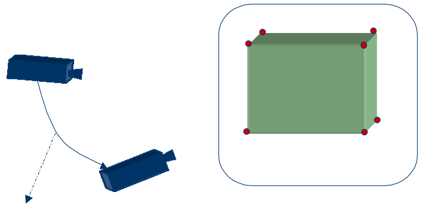
Small camera re-orientation



Note: Almost all pixels change!



MOVING CAMERAS ARE LIKE STEREO

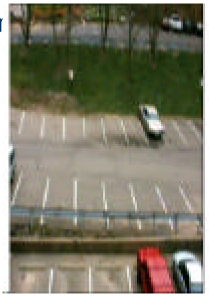


The change in spatial location between the two cameras (the “motion”)

Locations of points on the object (the “structure”)

Classes of motion

- Still camera, single moving object
- Still can
- Moving
- Moving

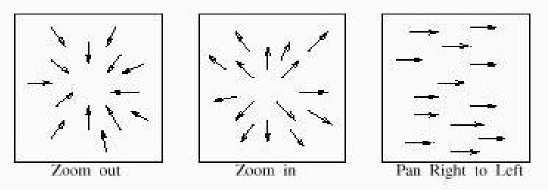


The motion field

- Vector field over the image:

  [u,v] = f(x,y), u,v = Vel vector, x,y = Im pos

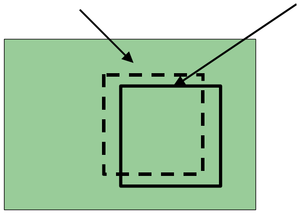
- **FOE, FOC** Focus of Expansion, Contraction



Optic/image flow

- Assumption: Image intensities from object points remain constant over time.

Im(**x** + **îx**; **y** + **îy**; **t** + **ît**) = Im(**x**; **y**; **t**)



Taylor expansion of intensity variation

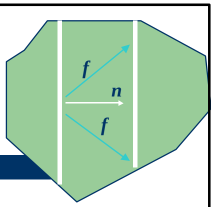
$$Im(x + îx; y + îy; t + ît) = \underbrace{Im(x; y; t) + \frac{@Im}{@x}îx + \frac{@Im}{@y}îy + \frac{@Im}{@t}ît}_{\text{Keep linear terms}} + h:o:$$

- Use constancy assumption and rewrite:

$$0 = \frac{@Im}{@x}îx + \frac{@Im}{@y}îy + \frac{@Im}{@t}ît$$

- Notice: Linear constraint, but no unique solution

Aperture problem



- Rewrite as dot product

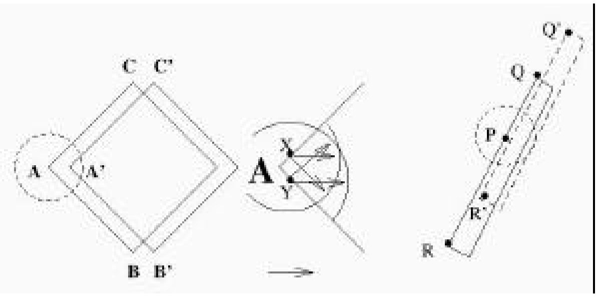
$$à\frac{@Im}{@t} = \left(\frac{@Im}{@x}; \frac{@Im}{@y}\right) á \begin{pmatrix} îx \\ îy \end{pmatrix} = r Im á \begin{pmatrix} îx \\ îy \end{pmatrix}$$

- One equation, two unknowns:

  f . ***n*** = k

- Notice: Can only detect vectors normal to gradient direction
- The motion of a line cannot be recovered using only local information

Aperture problem 2



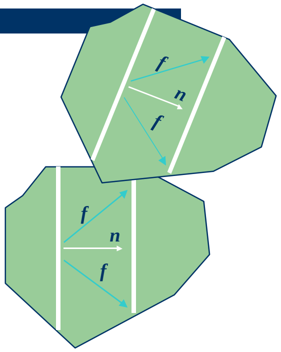
The flow continuity constraint

- Flows of nearby patches are nearly equal
- Two equations, two unknowns:

  f . $\boldsymbol{n}_1$ = k_1

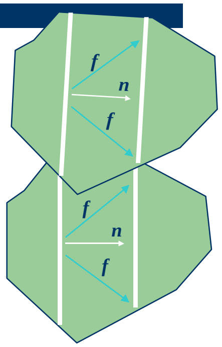
  f . $\boldsymbol{n}_2$ = k_2

- Solution exists, provided $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ not parallel
- Inverse optics gets expensive fast

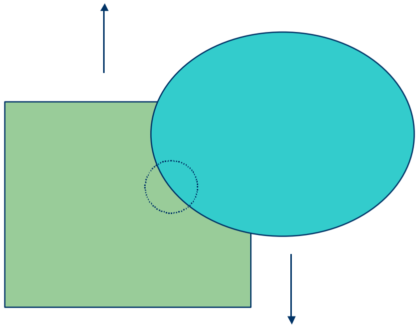


Sensitivity to error

- $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ might be ***almost*** parallel
- Tiny errors in estimates of *k*'s or ***n***'s can lead to huge errors in the estimate of ***f***



Sometimes the continuity constraint is false



Using several points

- Over determined equation system

$$@\ \grave{a}\ \frac{@Im}{@t}\ A\ =\ @\ \frac{@Im}{@x}\quad \frac{@Im}{@y}\ A\ \grave{o}\ \begin{matrix}\hat{\imath}x\\ \hat{\imath}y\end{matrix}\ \acute{o}$$

$$Im = Mu$$

- Can be solved in e.g. least squares sense using matlab u = M\Im

3-6D Optic flow

- Generalize to many freedooms (DOFs)

$$\delta I = \|I - I_t\|$$

$$\delta I = M\delta\mu$$

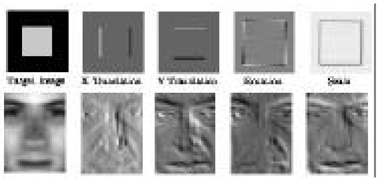
$$M = [I_x|I_y|I_r|I_s]$$

$$I_x = I(x,y) - I(x-1,y)$$

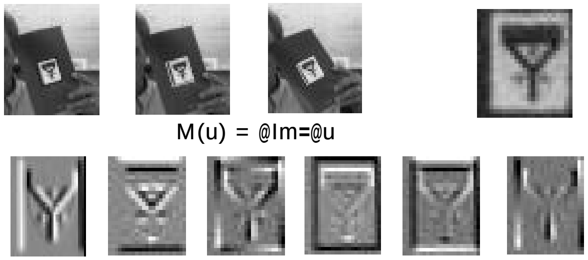
$$I_y = I(x,y) - I(x,y-1)$$

$$I_r = -yI_x + xI_y$$

$$I_s = \frac{1}{\sqrt{x^2+y^2}}(xI_x + yI_y)$$



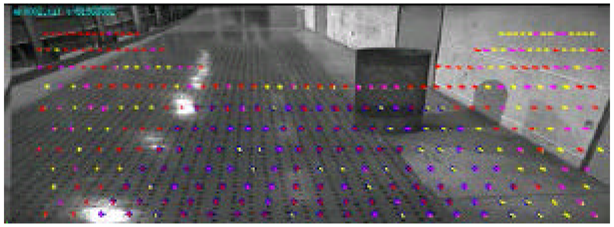
All 6 freedoms



M(u) = @Im=@u

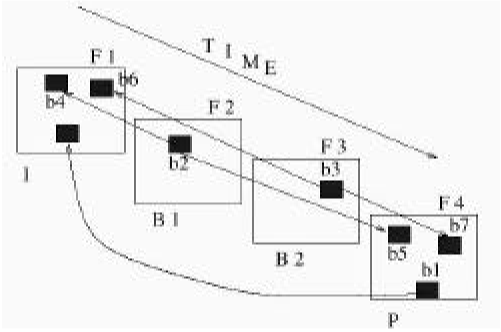
X Y Rotation Scale Aspect Shear

Flow vectors



- Norbert’s trick: Use an mpeg-card to speed up motion computation

Application: mpeg compression



Other applications:

- Recursive depth recovery: Kostas and Jane
- Motion control (we will cover)
- Segmentation
- Tracking