Introduction to Computer Vision
TTIC 31040
Winter 2022
Image gradients
Keypoint descriptor

Lecture 16: alignment and detection

TTIC 31040: Introduction to Computer Vision

Instructor: Greg Shakhnarovich

TTI-Chicago

March 7, 2022

Roadmap

- Image formation: geometry, optics, photometry
- 3D geometry; stereo and structure from motion
- Images as 2D signals; image manipulation, filtering, matching and alignment
- Remaining lectures:
  Motion (today)
  Color (next time);
  Multi-view stereo (maybe next time)

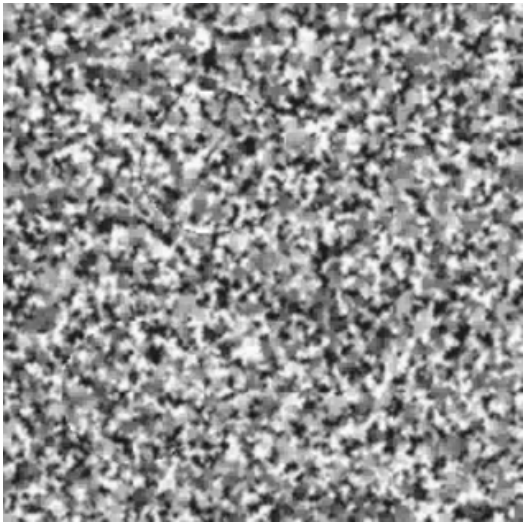
Roadmap

- Image formation: geometry, optics, photometry
- 3D geometry; stereo and structure from motion
- Images as 2D signals; image manipulation, filtering, matching and alignment
- Remaining lectures:
  Motion (today)
  Color (next time);
  Multi-view stereo (maybe next time)
- Next lecture is last (no lecture during Reading Period)
- Project descriptions out
- Quiz 2 is out (any time by Friday, same template as Quiz 1)

Motion

src: slides from L. Lazebnik, S. Seitz

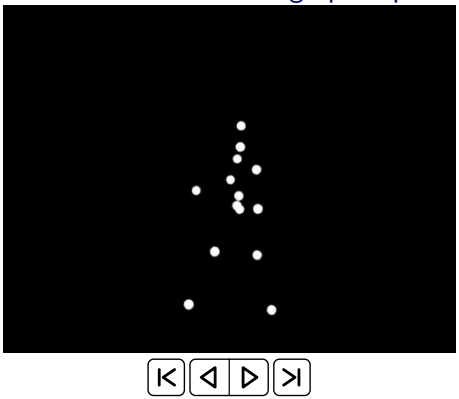
Motion is a powerful perceptual cue



- Sometimes it's the only cue!

Motion is a powerful perceptual cue

- Even impoverished motion data can evoke a strong "percept"



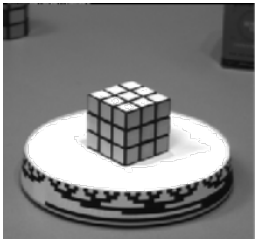
- Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis", 1973

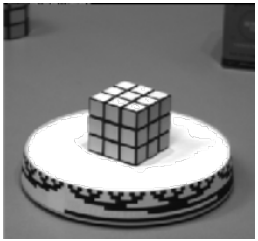
Motion in computer vision

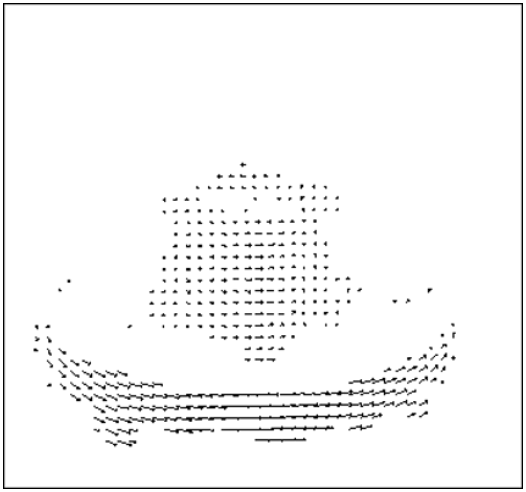
- 3D shape reconstruction from images by a moving camera (next week)
- Object segmentation in video
- Learning and tracking of *dynamical models*
- Event/activity recognition
- Self-supervised learning: representation learning driven by the objective of predicting next/subsequent frame in video

Motion field

- Motion field is the projection of the 3D scene motion onto the image
- A *vector field*: $(x, y) \in \mathbb{R}^2 \rightarrow (u, v) \in \mathbb{R}^2$

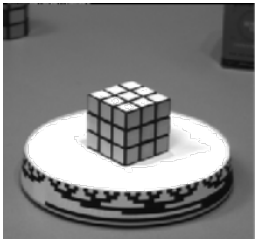


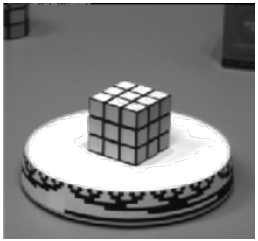


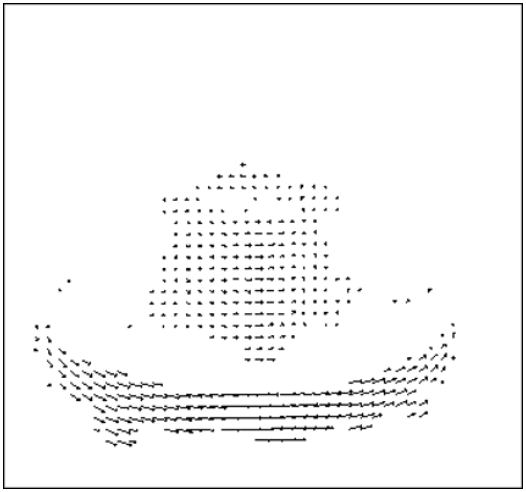


Motion field

- Motion field is the projection of the 3D scene motion onto the image
- A *vector field*: $(x, y) \in \mathbb{R}^2 \rightarrow (u, v) \in \mathbb{R}^2$



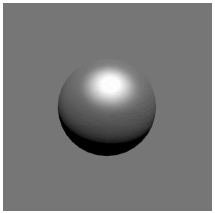




- Equivalently: "motion of the pixels" (in 2D)

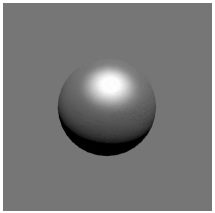
Optical flow

- Optical flow is the *apparent* motion of "brightness patterns" in the image
- Ideally, same as motion field...



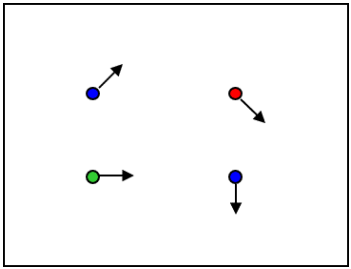
Optical flow

- Optical flow is the *apparent* motion of "brightness patterns" in the image
- Ideally, same as motion field...
- But: apparent motion may be due to moving in lighting (rather than moving scene)

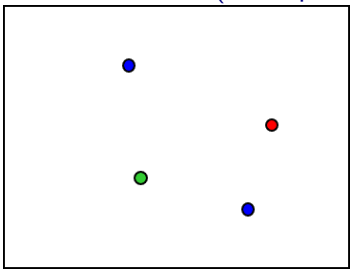


Optical flow: estimation

- How can we estimate optical flow for two *frames* (subsequent images)?



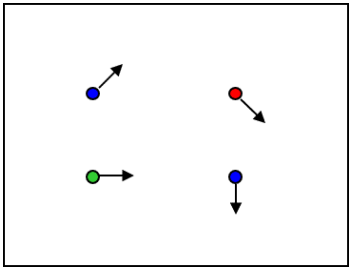
$I(x, y, t-1)$



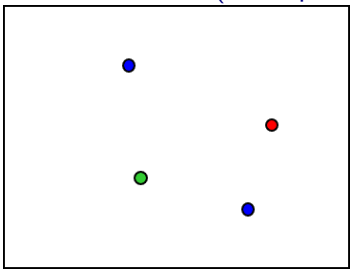
$I(x, y, t)$

Optical flow: estimation

- How can we estimate optical flow for two *frames* (subsequent images)?



$I(x, y, t-1)$

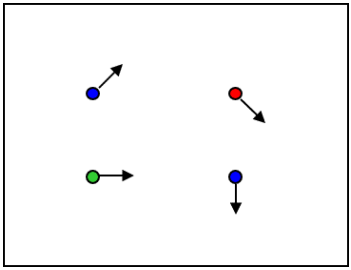


$I(x, y, t)$

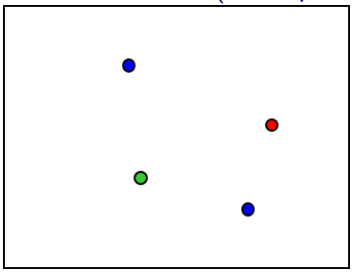
- Key assumptions we will need:

Optical flow: estimation

- How can we estimate optical flow for two *frames* (subsequent images)?



$I(x, y, t-1)$

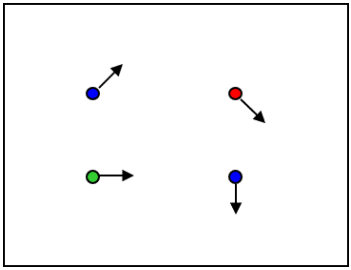


$I(x, y, t)$

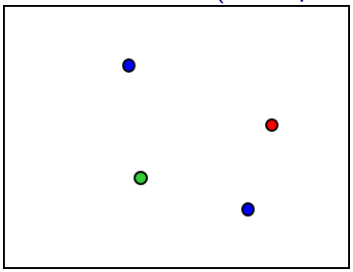
- Key assumptions we will need:
  - **Brightness constancy**: projection of the same 3D point looks the same color in every frame

Optical flow: estimation

- How can we estimate optical flow for two *frames* (subsequent images)?



$I(x, y, t-1)$

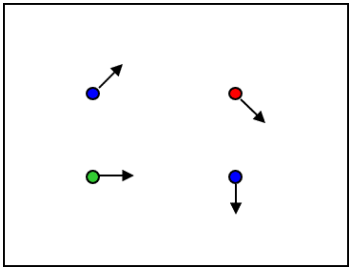


$I(x, y, t)$

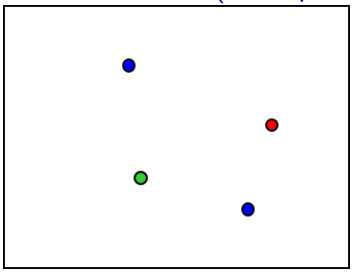
- Key assumptions we will need:
  - **Brightness constancy**: projection of the same 3D point looks the same color in every frame
  - **Small motion**

Optical flow: estimation

- How can we estimate optical flow for two *frames* (subsequent images)?



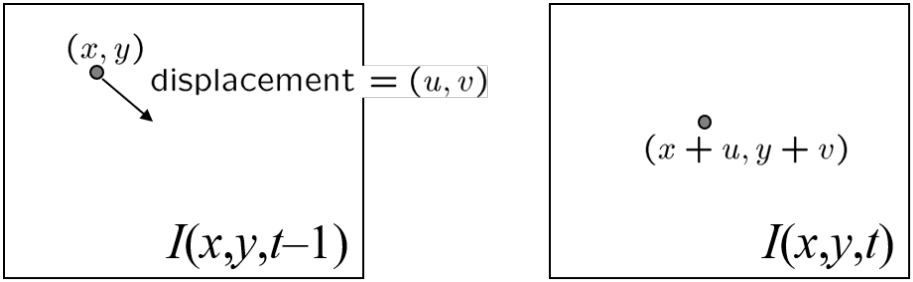
$I(x, y, t-1)$



$I(x, y, t)$

- Key assumptions we will need:
  - **Brightness constancy**: projection of the same 3D point looks the same color in every frame
  - **Small motion**
  - **Spatial coherence**: points move like their neighbors

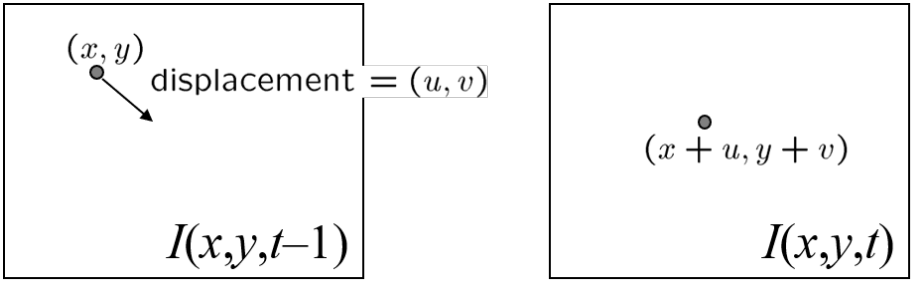
Brightness constancy



- The brightness constancy equation:

$$I(x, y, t-1) \;=\; I(x + u(x, y),\; y + v(x, y), t)$$

Brightness constancy



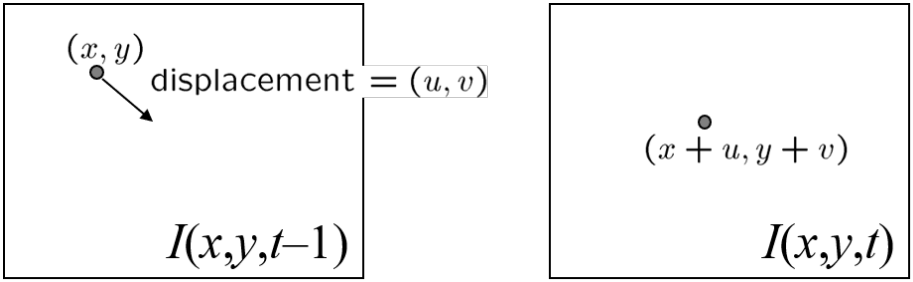
- The brightness constancy equation:

$$I(x, y, t-1) \; = \; I(x + u(x, y), \; y + v(x, y), t)$$

- Applying the Taylor expansion to the RHS (linearizing):

$$I(x, y, t-1) \; \approx \; I(x, y, t) + I_x u(x, y) \; + I_y v(x, y)$$

Brightness constancy



- The brightness constancy equation:

$$I(x, y, t-1) \;=\; I(x + u(x, y),\; y + v(x, y), t)$$

- Applying the Taylor expansion to the RHS (linearizing):

$$I(x, y, t-1) \;\approx\; I(x, y, t) + I_x u(x, y) \;+ I_y v(x, y)$$

So: $I_x u \;+ I_y v + I_t \;\approx\; 0$

The brightness constancy constraint

$$I_x u + I_y v + I_t = 0$$

- How many equations/unknowns?

The brightness constancy constraint

$$I_x u + I_y v + I_t = 0$$

- How many equations/unknowns? one equation, two unknowns...
- What does this mean? Recall that $(I_x, I_y) = \nabla I$

$$\nabla I \cdot (u, v) + I_t = 0$$

i.e., the component of the flow perpendicular to the gradient is unknown!

The brightness constancy constraint

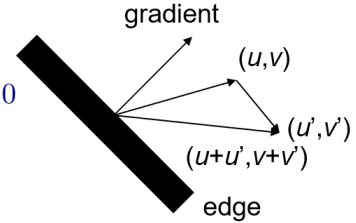
$$I_x u + I_y v + I_t = 0$$

- How many equations/unknowns? one equation, two unknowns...
- What does this mean? Recall that $(I_x, I_y) = \nabla I$

$$\nabla I \cdot (u, v) + I_t = 0$$

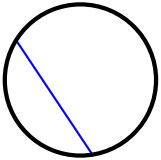
i.e., the component of the flow perpendicular to the gradient is unknown!

If (u, v) satisfies the constraint and $\nabla I \cdot (u', v') = 0$ then $(u + u', v + v')$ also does!



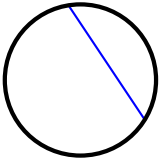
The aperture problem

- What is the motion here?



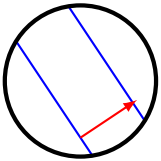
The aperture problem

- What is the motion here?



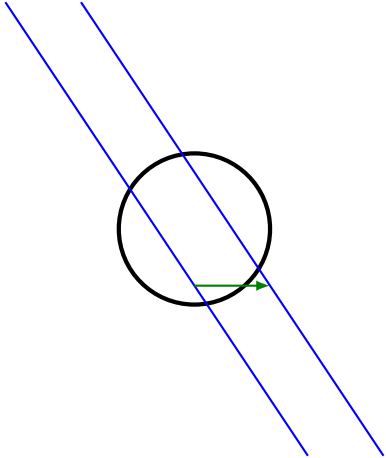
The aperture problem

- What is the motion here?



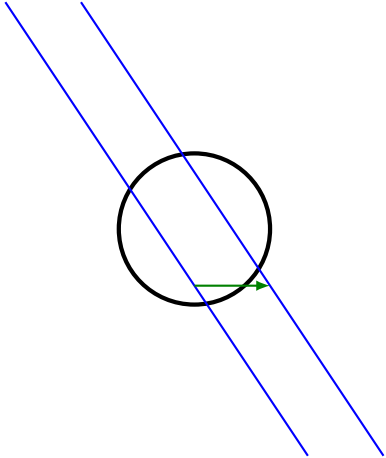
The aperture problem

- What is the motion here?



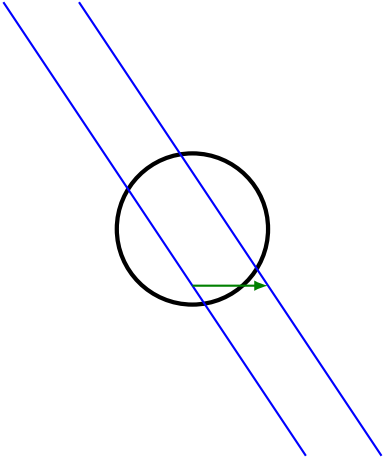
The aperture problem

- What is the motion here?

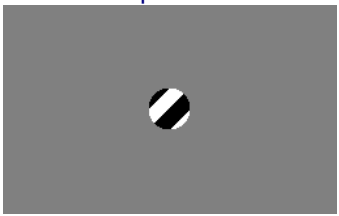


The aperture problem

- What is the motion here?



- Barber pole illusion:



Solving the aperture problem

- When we look at just the vicinity of the pixel we always suffer from the aperture problem
- Solution: get more equations for a pixel!
- **Spatial coherence** constraint: assume nearby pixels have the same motion (u, v)
  E.g., for 5×5 window ($n = 25$ equations per pixel):

$$\nabla I(\mathbf{x}_i) \cdot (u, v) + I_t(\mathbf{x}_i) = 0$$

(Lucas & Kanade, 1981)

Solving the aperture problem

- When we look at just the vicinity of the pixel we always suffer from the aperture problem
- Solution: get more equations for a pixel!
- **Spatial coherence** constraint: assume nearby pixels have the same motion (u, v)
  E.g., for 5×5 window ($n = 25$ equations per pixel):

$$\nabla I(\mathbf{x}_i) \cdot (u, v) + I_t(\mathbf{x}_i) = 0$$

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

(Lucas & Kanade, 1981)

Lucas-Kanade flow

- Linear least squares:

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

Lucas-Kanade flow

- Linear least squares:

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

$$\begin{array}{ccc} \mathbf{A} & \mathbf{d} = & \mathbf{b} \\ n \times 2 & 2 \times 1 & n \times 1 \end{array}$$

- Solution: $\mathbf{A}^T\mathbf{A}\mathbf{d} = \mathbf{A}\mathbf{b}$

Lucas-Kanade flow

- Linear least squares:

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

$$\begin{array}{ccc} \mathbf{A} & \mathbf{d} = & \mathbf{b} \\ n \times 2 & 2 \times 1 & n \times 1 \end{array}$$

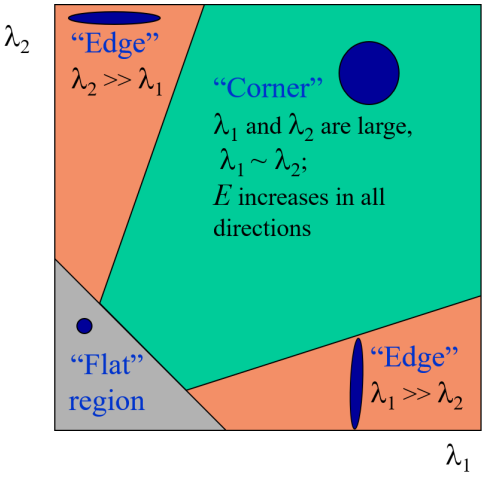
- Solution: $\mathbf{A}^T\mathbf{A}\mathbf{d} = \mathbf{A}\mathbf{b}$
- What is $\mathbf{A}^T\mathbf{A}$? a 2×2 matrix

$$\begin{bmatrix} \sum_i I_x I_x & \sum_i I_x I_y \\ \sum_i I_x I_y & \sum_i I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_i I_x I_t \\ \sum_i I_y I_t \end{bmatrix}$$

the second moment matrix

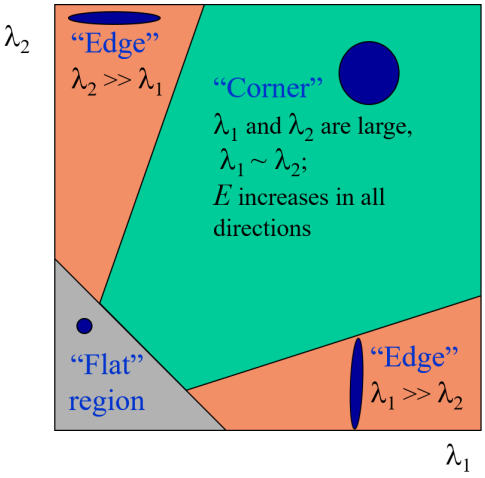
Recall: corner detection

- Estimation of Lucas-Kanade flow is well conditioned in regions with high "cornerness"

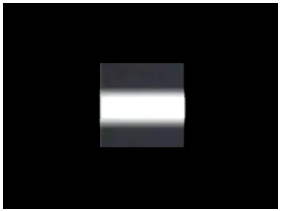


Recall: corner detection

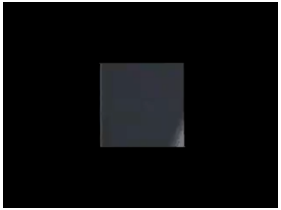
- Estimation of Lucas-Kanade flow is well conditioned in regions with high "cornerness"



Bad:



Good:



Lucas-Kanade flow example

Estimated flow

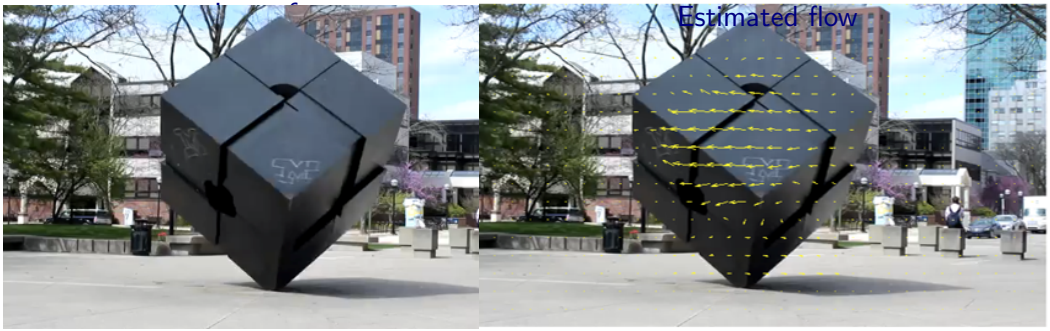
src: MathWorks

Lucas-Kanade flow example

Estimated flow

src: MathWorks

Lucas-Kanade flow example



src: MathWorks

Errors in Lucas-Kanade: large motion

- Large motion (larger than a pixel)
- Solution A: iterative refinement. Estimate velocity per pixel;

Errors in Lucas-Kanade: large motion

- Large motion (larger than a pixel)
- Solution A: iterative refinement. Estimate velocity per pixel;

Errors in Lucas-Kanade: large motion

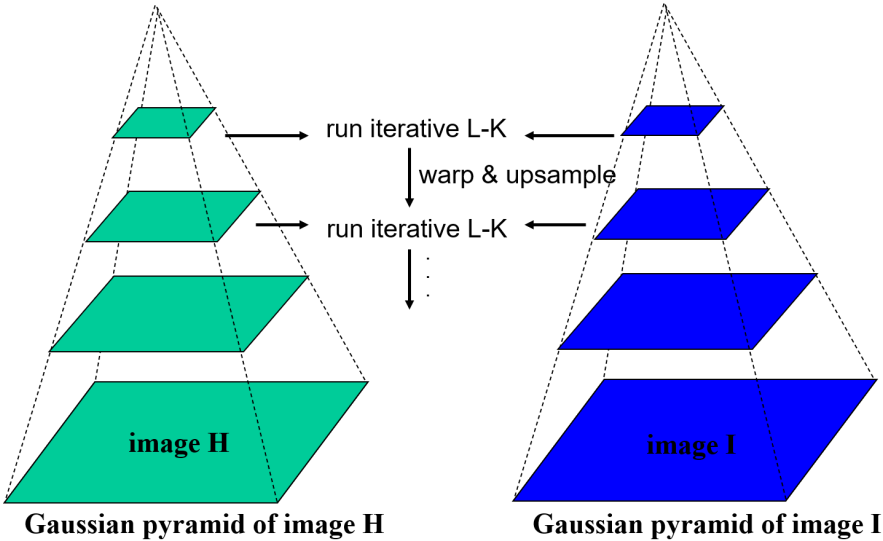
- Large motion (larger than a pixel)
- Solution A: iterative refinement. Estimate velocity per pixel; Warp I_{t-1} towards I_t using the estimated flow;

Errors in Lucas-Kanade: large motion

- Large motion (larger than a pixel)
- Solution A: iterative refinement. Estimate velocity per pixel;
  Warp I_{t-1} towards I_t using the estimated flow;
  Repeat until convergence
- Solution B: multi-resolution estimation
  large motion in high resolution is small motion in low resolution

Multi-resolution Lucas-Kanade

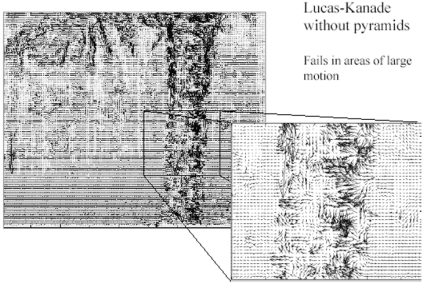
- Run on increasingly fine scales, using coarser scale flow as initialization



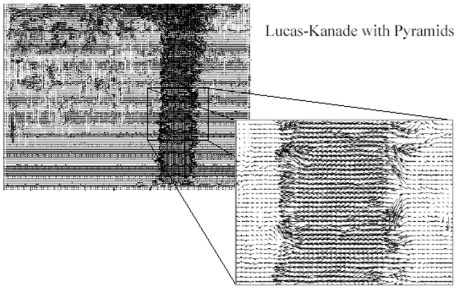
src: K. Hassan-Shafique

- An example of a general idea: *coarse-to-fine processing*

Multi-resolution Lucas-Kanade



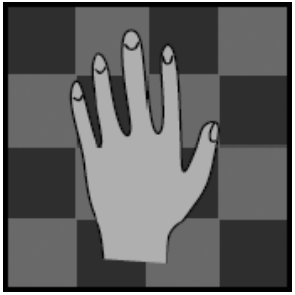
single res



multi-res

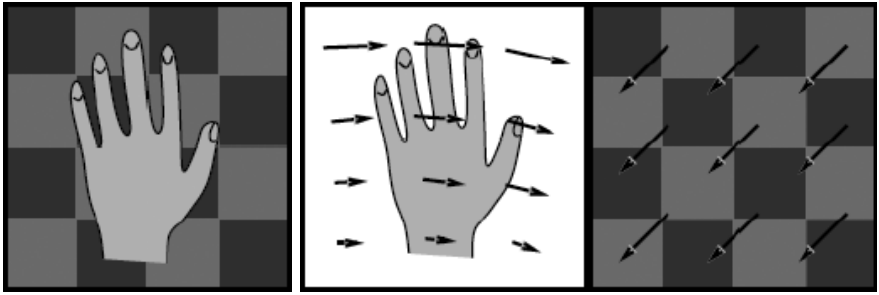
Heterogeneous motion

- Another assumption violation: neighbors that do not move similarly
- Example: foreground and background moving separately



Heterogeneous motion

- Another assumption violation: neighbors that do not move similarly
- Example: foreground and background moving separately



- Solution: motion *layers*

Motion segmentation

- Wang & Adelson, Represention Moving Images with Layers, 1994
- Group pixels with coherent motion

Figure 11: (a) The optic flow from multi-scale gradient method. (b) Segmentation obtained by clustering optic flow into affine motion regions. (c) Segmentation from consistency checking by image warping. Representing moving images with layers.

- This is *image segmentation*: partitioning of image into regions
- Now we need a *model* for the motion

Affine motion model

- A simple assumption: local flow estimates should be explained as a (global) affine function of the coordinates

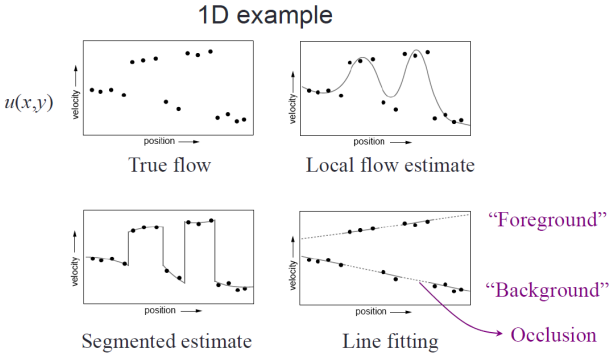
$$u(x,y) = a_u + b_u x + c_u y, \qquad v(x,y) = a_v + b_v x + c_v y$$

Affine motion model

- A simple assumption: local flow estimates should be explained as a (global) affine function of the coordinates

$$u(x,y) = a_u + b_u x + c_u y, \qquad v(x,y) = a_v + b_v x + c_v y$$

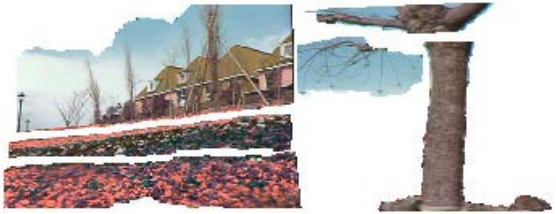
- Fit different affine functions to different subsets of pixels
- Basically, a mixture of experts!



Motion segmentation

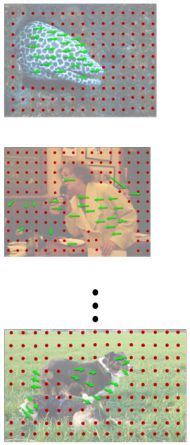
- Example:

- Can separate the layers and fill in (with enough motion)

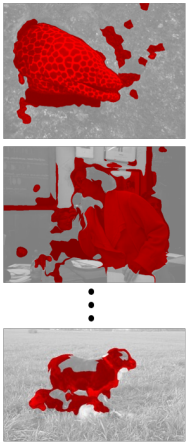


src: Wang & Adelson

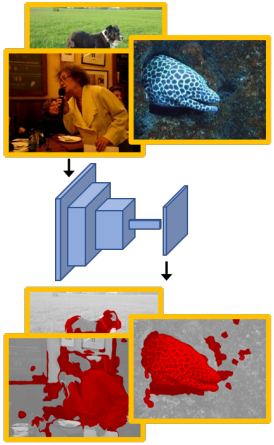
Motion as cue for learning



1. Collect videos



2. Segment using motion



3. Train ConvNet

Brightness constancy violations

- What if the brightness pattern of the projection of a 3D point changes? (e.g., changes in lighting; changing in object orientation with motion)
- Solution: use descriptors (SIFT) instead of color
- Leads to *feature tracking*

Horn-Shunck flow

- In Lucas-Kanade, the assumption about neighbors moving similarly is implicit (via solving for neighborhood)
- We can make it explicitly part of the optimization

Horn-Shunck flow

- In Lucas-Kanade, the assumption about neighbors moving similarly is implicit (via solving for neighborhood)
- We can make it explicitly part of the optimization
- Think of $u(x, y)$, $v(x, y)$ as functions of (x, y)
- Smoothness term: penalize fast-varying $\mathbf{u}, \mathbf{v}$ by $\|\nabla u\|^2 + \|\nabla v\|^2$

Horn-Shunck flow

- In Lucas-Kanade, the assumption about neighbors moving similarly is implicit (via solving for neighborhood)
- We can make it explicitly part of the optimization
- Think of $u(x, y)$, $v(x, y)$ as functions of (x, y)
- Smoothness term: penalize fast-varying $\mathbf{u}, \mathbf{v}$ by $\|\nabla u\|^2 + \|\nabla v\|^2$
- Horn-Shunck objective:

$$\min_{\mathbf{u},\mathbf{v}} \sum_{x,y} \left(I_x(x,y)u(x,y) + I_y(x,y)v(x,y) + I_t(x,y)\right)^2 + \alpha \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right)$$

Horn-Shunck flow

- In Lucas-Kanade, the assumption about neighbors moving similarly is implicit (via solving for neighborhood)
- We can make it explicitly part of the optimization
- Think of $u(x, y)$, $v(x, y)$ as functions of (x, y)
- Smoothness term: penalize fast-varying $\mathbf{u}, \mathbf{v}$ by $\|\nabla u\|^2 + \|\nabla v\|^2$
- Horn-Shunck objective:

$$\min_{\mathbf{u},\mathbf{v}} \sum_{x,y} \left(I_x(x,y)u(x,y) + I_y(x,y)v(x,y) + I_t(x,y)\right)^2 + \alpha \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right)$$

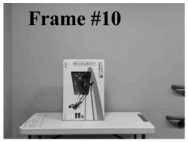
- Larger α: smoother flow
- A variational problem (the objective is a function of I, u, v and their derivatives). Can solve this iteratively using Euler-Lagrange equations (look it up)

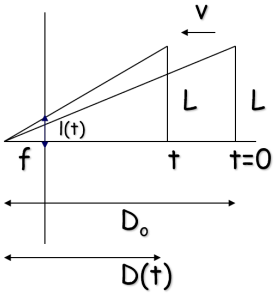
Flow with moving camera

Flow with moving camera

Time to collision

- We observe an object increase in size with time

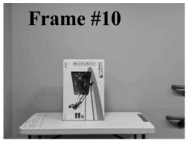


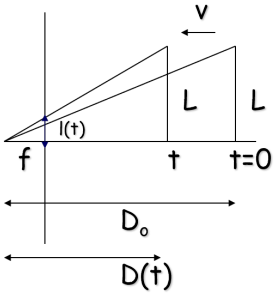


- Object of height L, moving with velocity v
- Time $t = 0$: distance $D(0) = D_0$

Time to collision

- We observe an object increase in size with time
- It is getting closer
- When will we hit it?

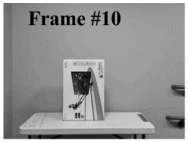




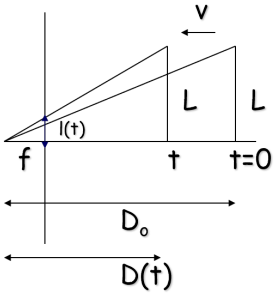
- Object of height L, moving with velocity v
- Time $t = 0$: distance $D(0) = D_0$
- Time $t > 0$: distance $D(t)$

Time to collision

- We observe an object increase in size with time
- It is getting closer
- When will we hit it?



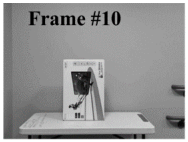
- Surprising: (?) do not need to know the speed or the size!



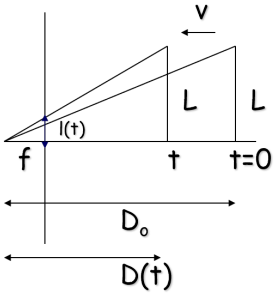
- Object of height L, moving with velocity v
- Time $t = 0$: distance $D(0) = D_0$
- Time $t > 0$: distance $D(t) = D_0 - vt$

Time to collision

- We observe an object increase in size with time
- It is getting closer
- When will we hit it?



- Surprising: (?) do not need to know the speed or the size!



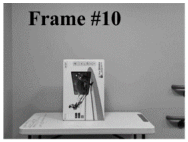
- Object of height L, moving with velocity v
- Time $t = 0$: distance $D(0) = D_0$
- Time $t > 0$: distance $D(t) = D_0 - vt$

- It will crash (reach camera center) at time τ s.t.

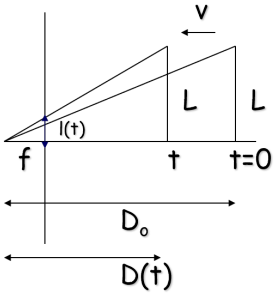
$$D(\tau) = D_0 - v\tau = 0$$

Time to collision

- We observe an object increase in size with time
- It is getting closer
- When will we hit it?



- Surprising: (?) do not need to know the speed or the size!

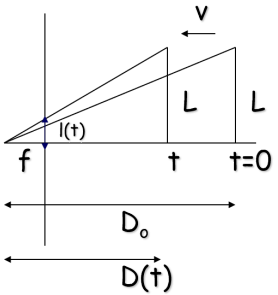


- Object of height L, moving with velocity v
- Time $t = 0$: distance $D(0) = D_0$
- Time $t > 0$: distance $D(t) = D_0 - vt$

- It will crash (reach camera center) at time τ s.t.

$$D(\tau) = D_0 - v\tau = 0 \quad \Rightarrow \quad \tau = D_0/v$$

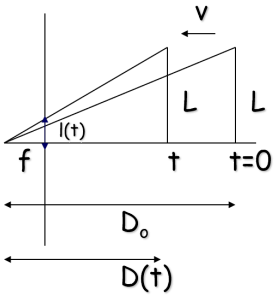
Time to collision



- Image size of the object at time t

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

Time to collision



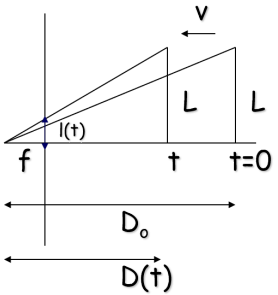
- Image size of the object at time t

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

- Derivative w.r.t. time:

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

Time to collision



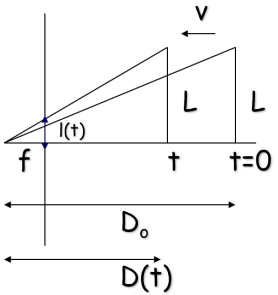
- Image size of the object at time t

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

- Derivative w.r.t. time:

$$\begin{aligned} l'(t) &= fL\frac{d\left(1/D(t)\right)}{dt} \\ &= fL\frac{-1}{D^2(t)}\frac{dD(t)}{dt} \end{aligned}$$

Time to collision



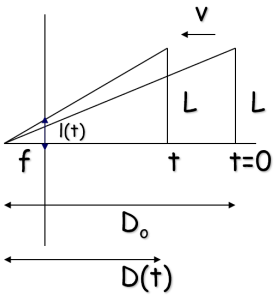
- Image size of the object at time t

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

- Derivative w.r.t. time:

$$\begin{aligned} l'(t) &= fL\frac{d\left(1/D(t)\right)}{dt} \\ &= fL\frac{-1}{D^2(t)}\frac{dD(t)}{dt} \qquad \left[\, v = -\frac{dD(t)}{dt} \,\right] \\ &= fL\frac{v}{D^2(t)} \end{aligned}$$

Time to collision

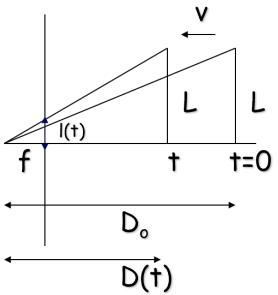


- So, we have

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

- Note: both $l(t)$ and $l'(t)$ can be measured from the image!

Time to collision



- So, we have

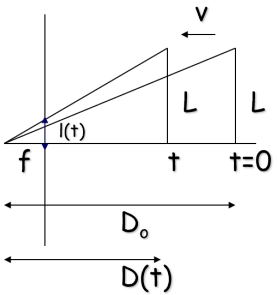
$$l(t) = \frac{fL}{D(t)}, \quad l'(t) = \frac{fLv}{D^2(t)}$$

- Note: both $l(t)$ and $l'(t)$ can be measured from the image!

- The ratio gives us:

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

Time to collision



- So, we have

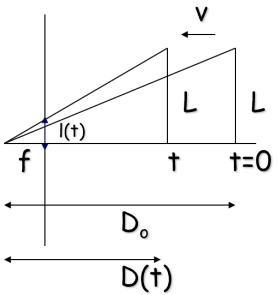
$$l(t) = \frac{fL}{D(t)}, \quad l'(t) = \frac{fLv}{D^2(t)}$$

- Note: both $l(t)$ and $l'(t)$ can be measured from the image!

- The ratio gives us:

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

Time to collision



- So, we have

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

- Note: both $l(t)$ and $l'(t)$ can be measured from the image!

- The ratio gives us:

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

- So we can find time to collision without knowing D_0 or L or v!