Lecture 10

Motion and Tracking

Ruta Desai, Chun-Liang Li



Administrative

- A2 is due May 2nd





Pixels Video ML Camera Segment Segmentation Convolutions Linear Models Motion Camera Clustering Edges (Conv) Neural networks Tracking 3D Geometry Detection Descriptors

Ruta Desai, Chun-Liang Li

CSE 455 Roadmap



Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005] http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



Today's agenda

• Optical flow

- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005] http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



From images to videos

- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Ruta Desai, Chun-Liang Li



Why is motion useful?



Ruta Desai, Chun-Liang Li

Lecture 10 - 7

Why is motion useful?



Ruta Desai, Chun-Liang Li

Lecture 10 - 8

Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Note: apparent motion can be caused by lighting changes without any actual motion
 - Think of a uniform rotating sphere under fixed lighting (has motion but no optical flow)
 - vesus a stationary sphere under moving illumination (no motion but has optical flow)

GOAL: Recover image motion at each pixel from optical flow



Optical flow without motion!



Ruta Desai, Chun-Liang Li

Lecture 10 - 10

Optical flow

of an image gives us the apparent motion of every pixel

It is a function of the spatio-temporal image brightness variations



Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

Ruta Desai, Chun-Liang Li

Lecture 10 - 11

Formalizing optical flow



- Given two subsequent frames,
- estimate the apparent motion field u(x,y), v(x,y) between them
- u(x, y) measuring the horizontal movement of the pixel at location (x, y).
- v(x, y) measures the vertical movement.
- Together, the pixel at (x, y, t-1) goes to (x+u, y+v, t)

Ruta Desai, Chun-Liang Li



3 assumptions when estimating optical flow

- 1. **small motions**: points do not move very far
- 2. **spatial coherence**: points move like their neighbors
- 3. **brightness constancy**: the brightness of a pixel remains constant between consecutive frames



Key Assumptions: small motions

The **small motions assumption**: Between consecutive frames the change in pixel locations is small





Ruta Desai, Chun-Liang Li



Key Assumptions: spatial coherence

The spatial coherence assumption:

Neighboring pixels typically move together because they belong to the same rigid object.



Ruta Desai, Chun-Liang Li



Key Assumptions: brightness constancy

The **brightness constancy assumption**: Average brightness of pixels in a patch stays the same across consecutive frames, although their location might change



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

Ruta Desai, Chun-Liang Li



The brightness constancy constraint



• Brightness Constancy Equation:

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

Ruta Desai, Chun-Liang Li



The brightness constancy constraint



• Brightness Constancy Equation:

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

Linearizing the right side using Taylor expansion:

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

$$I(x+u, y+v, t) - I(x, y, t-1) = I_x \cdot u(x, y) + I_y \cdot v(x, y) + I_t$$
Hence,
$$I_x \cdot u + I_y \cdot v + I_t \approx 0 \quad \rightarrow \nabla I \cdot [u \ v]^T + I_t = 0$$

Ruta Desai, Chun-Liang Li

Lecture 10 - 18

Derivative filters are now 3 dimensional

Derivative in **x direction** now has a new dimension looking at past and future frames



Ruta Desai, Chun-Liang Li



Similar for y direction



Ruta Desai, Chun-Liang Li

Lecture 10 - 20

New backward derivative in the time t dimension



Ruta Desai, Chun-Liang Li

Lecture 10 - 21

The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

$$\nabla I \cdot \left[u \ v \right]^T + I_t = 0$$

• Q. How many equations and unknowns per pixel?

•One equation, two unknowns (u,v)





Ruta Desai, Chun-Liang Li

Lecture 10 - 23



Ruta Desai, Chun-Liang Li

Lecture 10 - 24



Ruta Desai, Chun-Liang Li

Lecture 10 - 25





Apr 30, 2025

Ruta Desai, Chun-Liang Li

The barber pole illusion



http://en.wikipedia.org/wiki/Barberpole_illusion

Ruta Desai, Chun-Liang Li



The barber pole illusion



http://en.wikipedia.org/wiki/Barberpole_illusion

Ruta Desai, Chun-Liang Li



The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

$$\nabla I \cdot \left[u \ v \right]^T + I_t = 0$$

• Q. How many equations and unknowns per pixel?

•One equation, two unknowns (u,v)

Problem: The component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be measured $gradient \nabla I$

If (u, v) satisfies the equation, so does (u+u', v+v') if $\nabla I \cdot [u' v']^T = 0$



Ruta Desai, Chun-Liang Li

Lecture 10 - 29

Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005] http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



How to get more equations for a pixel?

- Add in the Spatial coherence constraint:
- Assume the pixel's neighbors have the same (u,v)
 - $\circ\,$ If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \ v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

Ruta Desai, Chun-Liang Li



Lucas-Kanade flow

• Overconstrained linear system:

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix} A = b$$

$$25 \times 2 = 2 \times 1 = 25 \times 1$$



Lucas-Kanade flow

• Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix} A = b$$

$$25 \times 2 = 2 \times 1 = 25 \times 1$$

Multiplying by A^{T} to solve for *d* gives us: $(A^{T}A) d = A^{T}b$

Lecture 10 - 33

Apr 30, 2025

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$
$$A^T A \qquad \qquad A^T b$$

The summations are over all pixels in the 5 x 5 window

Ruta Desai, Chun-Liang Li

Conditions for solving this Lucas-Kanade equation

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$
$$A^T A \qquad A^T b$$

When is This Solvable?

- **A^TA** should be invertible
- **A^TA** should not be too small, otherwise it is close to being non-invertible
 - eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
- **A^TA** should be well-conditioned
 - $-\lambda_1/\lambda_2$ should not be too large (λ_1 = larger eigenvalue)

Q. Does this remind anything to you?

Ruta Desai, Chun-Liang Li



$M = A^{T}A$ is the Harris corner detector!

$$A^{T}A = \begin{bmatrix} \sum I_{x}I_{x} & \sum I_{x}I_{y} \\ \sum I_{x}I_{y} & \sum I_{y}I_{y} \end{bmatrix} = \sum \begin{bmatrix} I_{x} \\ I_{y} \end{bmatrix} [I_{x} I_{y}] = \sum \nabla I(\nabla I)^{T}$$

- Eigenvectors and eigenvalues of A^TA relate to edge direction and magnitude
 - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
 - The other eigenvector is orthogonal to it

Ruta Desai, Chun-Liang Li



Interpreting the eigenvalues

Classification of image points using eigenvalues of the second moment matrix:



Ruta Desai, Chun-Liang Li

Lecture 10 - 36
Edges are harder to track



All the points on an edge look the same. It is hard to estimate where each point will move to.

 $\sum \nabla I (\nabla I)^T$ - gradients very large or very small
- large λ_1 , small λ_2

Ruta Desai, Chun-Liang Li

Lecture 10 - 37

Low-texture region



Low-texture regions have small eigenvalues. The matrix is harder to invert and get accurate estimates of optical flow

 $\sum \nabla I (\nabla I)^T$ - gradients have small magnitude
- small λ_1 , small λ_2

Ruta Desai, Chun-Liang Li

Lecture 10 - 38

High-texture region



These points are easier to estimate optical flow for.

This makes sense intuitively: You could say that corners and blobs (things that are easier to detect) are easier to track over time.

Apr 30, 2025

 $\sum \nabla I (\nabla I)^T$ - gradients are different, large magnitudes
- large λ_1 , large λ_2

Ruta Desai, Chun-Liang Li

Errors in Lucas-Kanade

What are the potential causes of errors in this procedure?

- \circ Suppose A^TA is easily invertible
- \circ Suppose there is not much noise in the image
- When our assumptions are violated
 - Brightness constancy is not satisfied
 - The motion is **not** small
 - A point does **not** move like its neighbors
 - window size is too large
 - what is the ideal window size?

Ruta Desai, Chun-Liang Li



Improving accuracy

• Recall our small motion assumption

 $I_x \cdot u + I_y \cdot v + I_t \approx 0$

- This is not exact
 - To do better, we need to add higher order terms back in:

 $I_x \cdot u + I_v \cdot v + \text{higher order terms} + I_t \approx 0$

- This is a *polynomial root finding* problem
 - Can solve using Newton's method (which is out of scope for this class)
 - Lukas-Kanade method does one iteration of Newton's method
 - Better results are obtained via more iterations

Ruta Desai, Chun-Liang Li



Iterative Lucas-Kanade Algorithm

- 1. Estimate velocity at each pixel by solving Lucas-Kanade equations
- 2. Warp I(t-1) towards I(t) using the estimated flow field *Calculate I(t) using the calculated optical flow*
- 3. Repeat until convergence

Ruta Desai, Chun-Liang Li



When do the optical flow assumptions fail?

In other words, in what situations does the displacement of pixel patches not represent physical movement of points in space?

- 1. Well, television (movies) screens appear to contain objects in motion
 - Yet our TVs and monitors are actually stationary
- 2. Motion that doesn't cause changes in pixels
 - e.g. A uniform rotating sphere. Nothing seems to move, yet it is rotating
- 3. Lighting changes can make things seem to move
 - for example, if a singular light source moves around a stationary sphere
- 4. Smaller motions might move in a direction opposite to motion
 - E.g. a cheetah's muscles move opposite direction of motion.

Ruta Desai, Chun-Liang Li



Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Ruta Desai, Chun-Liang Li

Lecture 10 - 44

Key assumptions (Errors in Lucas-Kanade)

• Small motion: points do not move very far

- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames
- **Spatial coherence:** points move like their neighbors

Ruta Desai, Chun-Liang Li



Revisiting the small motion assumption

- Is this motion small enough?
 - Probably not—it's much larger than one pixel (2nd order terms dominate)
 - How might we solve this problem?



Ruta Desai, Chun-Liang Li



Reduce the resolution so that assumption holds





5 10 15 20

Ruta Desai, Chun-Liang Li

Lecture 10 - 47

Apr 30, 2025

25

Coarse-to-fine optical flow estimation



Ruta Desai, Chun-Liang Li

Lecture 10 - 48



Optical Flow Results



Ruta Desai, Chun-Liang Li

Lecture 10 - 50

Optical Flow Results



• OpenCV

Ruta Desai, Chun-Liang Li

Lecture 10 - 51

Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005] http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



Key assumptions (Errors in Lucas-Kanade)

- Small motion: points do not move very far
- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames
- **Spatial coherence:** points move like their neighbors

Ruta Desai, Chun-Liang Li



• The flow is formulated as a global energy function which is should be minimized:

$$E=\iint ig[(I_xu+I_yv+I_t)^2+lpha^2(\|
abla u\|^2+\|
abla v\|^2)ig]\,\mathrm{d}x\mathrm{d}y$$

Ruta Desai, Chun-Liang Li



- The flow is formulated as a global energy function which is should be minimized:
- The first part of the function is the brightness constancy.

$$E= \iint ig \left(I_x u + I_y v + I_t
ight)^2 + lpha^2 (\|
abla u\|^2 + \|
abla v\|^2) ig] \,\mathrm{d}x\mathrm{d}y$$

Ruta Desai, Chun-Liang Li



- The flow is formulated as a global energy function which is should be minimized:
- The second part is the smoothness constraint. It's trying to make sure that the changes between pixels are small.

$$E= \iint ig[(I_x u+I_y v+I_t)^2+lpha^2 \left\|
abla u\|^2+\|
abla v\|^2ig]\,\mathrm{d}x\mathrm{d}y$$

Ruta Desai, Chun-Liang Li



- The flow is formulated as a global energy function which is should be minimized:
- *a* is a regularization constant. Larger values of *a* lead to smoother flows across time.

$$E= \iint ig[(I_x u+I_y v+I_t)^2+lpha^2 \|
abla u\|^2+\|
abla v\|^2)ig]\,\mathrm{d}x\mathrm{d}y$$

Ruta Desai, Chun-Liang Li



• The flow is formulated as a global energy function which is should be minimized:

$$E= \iint ig[(I_x u+I_y v+I_t)^2+lpha^2(\|
abla u\|^2+\|
abla v\|^2)ig]\,\mathrm{d}x\mathrm{d}y$$

• This minimization can be solved by taking the derivative with respect to u and v, we get the following 2 equations:

$$egin{aligned} &I_x(I_xu+I_yv+I_t)-lpha^2\Delta u=0\ &I_y(I_xu+I_yv+I_t)-lpha^2\Delta v=0 \end{aligned}$$

Ruta Desai, Chun-Liang Li



• By taking the derivative with respect to u and v, we get the following 2 equations: $I(I u + I v + L) = \alpha^2 \Delta u = 0$

$$egin{aligned} &I_x(I_xu+I_yv+I_t)-lpha^2\Delta u=0\ &I_y(I_xu+I_yv+I_t)-lpha^2\Delta v=0 \end{aligned}$$

- Where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is called the Lagrange operator. It is hard to calculate. So we estimate it using $\Delta u(x, y) = \overline{u}(x, y) u(x, y)$ Intuition: Lagrange is the second derivative. The estimation measures the deviation from the average change.
- where $\overline{u}(x,y)$ is the weighted average of u measured at (x,y) over its neighborhood of 5 x 5 pixels

Ruta Desai, Chun-Liang Li



• Now we substitute $\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$ in:

$$egin{aligned} &I_x(I_xu+I_yv+I_t)-lpha^2\Delta u=0\ &I_y(I_xu+I_yv+I_t)-lpha^2\Delta v=0 \end{aligned}$$

• To get:

Ruta Desai, Chun-Liang Li

$$egin{aligned} &(I_x^2+lpha^2)u+I_xI_yv=lpha^2\overline{u}-I_xI_t\ &I_xI_yu+(I_y^2+lpha^2)v=lpha^2\overline{v}-I_yI_t \end{aligned}$$

• Which is linear in u and v and can be solved analytically for each pixel individually.



• Analytical solution for:

$$(I_x^2+lpha^2)u+I_xI_yv=lpha^2\overline{u}-I_xI_t \ I_xI_yu+(I_y^2+lpha^2)v=lpha^2\overline{v}-I_yI_t$$

• is:

$$u = \bar{u} - \frac{I_x(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$
$$v = \bar{v} - \frac{I_y(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

Ruta Desai, Chun-Liang Li

Lecture 10 - 61

Iterative Horn-Schunk

- Similar to iterative Lucas-Kanade, there is an iterative version of Horn-Schunk algorithm.
- Since the solution depends on \overline{u} and \overline{v} , this calculation becomes more accurate as we iteratively update the average flow.
- After each calculate, re-calculate \overline{u} and \overline{v}



What we will learn today?

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Ruta Desai, Chun-Liang Li



Key assumptions

- Small motion: points do not move very far
- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames
- **Spatial coherence:** points move like their neighbors

Ruta Desai, Chun-Liang Li



Gestalt – common fate

Common Fate

Ruta Desai, Chun-Liang Li



 Use optical flow as the feature representation of each pixel



Ruta Desai, Chun-Liang Li



• Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)



Ruta Desai, Chun-Liang Li



• Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)



Ruta Desai, Chun-Liang Li



• Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)





J. Wang and E. Adelson. Layered Representation for Motion Analysis. CVPR 1993.

Ruta Desai, Chun-Liang Li



Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Ruta Desai, Chun-Liang Li



Single object tracking



Ruta Desai, Chun-Liang Li



Multiple object tracking



Ruta Desai, Chun-Liang Li


Tracking with a fixed camera





Ruta Desai, Chun-Liang Li

Lecture 10 - 73

. .

Tracking with a fixed camera



Ruta Desai, Chun-Liang Li



Tracking with a moving camera



Ruta Desai, Chun-Liang Li



Challenges in Feature tracking

- Figure out which features can be tracked
 Efficiently track across frames
- Some points may change appearance over time

 e.g., due to rotation, moving into shadows, etc.
- Drift: small errors can accumulate over time
- Points may appear or disappear.
 - \circ need to be able to add/delete tracked points.

Ruta Desai, Chun-Liang Li



What are good features to track?

- Intuitively, we want to avoid smooth regions and edges. But is there a more is principled way to define good features?
- What kinds of image regions can we detect easily and consistently?
 - SIFT blobs!
 - Harris corners!

Ruta Desai, Chun-Liang Li



Optical flow can help track features

Once we have the features we want to track, lucas-kanade or other optical flow algorithm can help track those features



Ruta Desai, Chun-Liang Li



Feature-tracking/Mapping





Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Ruta Desai, Chun-Liang Li



Feature-tracking/Mapping





Courtesy of Jean-Yves Bouguet - Vision Lab, California Institute of Technology

Ruta Desai, Chun-Liang Li



Simple KLT tracker

- 1. Find a good point to track (harris corner)
- 2. For each Harris corner compute optical flow (translation or affine) between consecutive frames.
- 3. Link motion vectors in successive frames to get a track for each Harris point
- Introduce new Harris points by applying Harris detector at every m (10 or 15) frames
- 5. Track new and old Harris points using steps 1-3

Ruta Desai, Chun-Liang Li



KLT tracker for fish



Ruta Desai, Chun-Liang Li



Tracking cars



Ruta Desai, Chun-Liang Li



Tracking movement



Ruta Desai, Chun-Liang Li



What we will learn today?

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Ruta Desai, Chun-Liang Li



Uses of motion

- Segmenting objects based on motion cues
- Tracking objects
- Learning dynamical models
- Improving video quality
 - \circ Motion stabilization
 - \circ Super resolution
- Recognizing events and activities

Ruta Desai, Chun-Liang Li Lecture 10 - 86

Estimating 3D structure



Ruta Desai, Chun-Liang Li



Segmenting objects based on motion cues

• Motion segmentation

Segment the video into multiple coherently moving objects



S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference (BMVC) 2006

Ruta Desai, Chun-Liang Li

Lecture 10 - 88

Tracking objects

Ruta Desai, Chun-Liang Li



Z.Yin and R.Collins, "On-the-fly Object Modeling while Tracking," IEEE Computer Vision and Pattern

Lecture 10 - 89

Recognizing events and activities



D. Ramanan, D. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. PAMI 2007.

Ruta Desai, Chun-Liang Li





W. Choi, K. Shahid, S. Savarese, "What are they doing? : Collective Activity Classification Using Spatio-Temporal Relationship Among People", 9th International Workshop on Visual Surveillance (VSWS09) in conjuction with ICCV 09

Ruta Desai, Chun-Liang Li



Recognizing events and activities



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, (<u>BMVC</u>), Edinburgh, 2006.

Ruta Desai, Chun-Liang Li



Recognizing events and activities Crossing – Talking – Queuing – Dancing – jogging



W. Choi & K. Shahid & S. Savarese WMC 2010

Ruta Desai, Chun-Liang Li

Lecture 10 - 93

Today's agenda

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]

http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



Next time

Geometry and cameras

Ruta Desai, Chun-Liang Li



What does the smoothness regularization doing?

- It's a sum of squared terms (a Euclidian distance measure).
- We're putting it in the expression to be minimized.
- => In texture free regions, there is no optical flow
- => On edges, points will flow to nearest points, solving the aperture problem.



Ruta Desai, Chun-Liang Li



Dense Optical Flow with Michael Black's method

- Michael Black took Horn-Schunk's method one step further, starting from the regularization constant:
- Which looks like a quadratic:

$$\|
abla u\|^2+\|
abla v\|^2$$

• And replaced it with this:



• Why does this regularization work better?

Ruta Desai, Chun-Liang Li



Affine motion

 $u(x, y) = a_1 + a_2 x + a_3 y$ $v(x, y) = a_4 + a_5 x + a_6 y$

• Substituting into the brightness constancy equation:

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$



Ruta Desai, Chun-Liang Li



Affine motion

 $u(x, y) = a_1 + a_2 x + a_3 y$ $v(x, y) = a_4 + a_5 x + a_6 y$

Substituting into the brightness constancy equation:



$$I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t \approx 0$$

- Each pixel provides 1 linear constraint in 6 unknowns
 - Least squares minimization:

$$Err(\overset{\boxtimes}{a}) = \sum \left[I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t \right]^2$$

Ruta Desai, Chun-Liang Li

Lecture 10 - 99

How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
 - Divide the image into blocks and estimate affine motion parameters in each block by least squares
 - Eliminate hypotheses with high residual error
 - Map into motion parameter space
 - Perform k-means clustering on affine motion parameters
 - -Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene



How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
 - Divide the image into blocks and estimate affine motion parameters in each block by least squares
 - Eliminate hypotheses with high residual error
 - Map into motion parameter space
 - Perform k-means clustering on affine motion parameters
 - -Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene



Synthesizing dynamic textures



Original Synthesized

Ruta Desai, Chun-Liang Li



Segmenting objects based on motion cues

- Background subtraction
 - $\circ\,$ A static camera is observing a scene
 - Goal: separate the static *background* from the moving *foreground*





Super-resolution

Example: A set of low quality images

Most of the test data o	Most of the test data o	Most of the test data of
couple of exceptions. T	couple of exceptions. T	couple of exceptions."
low-temperature solds	low-temperature solder	low-temperature solds
investigated (or some o	investigated (or some o	investigated (or some
manufacturing technol	manufacturing technol	manufacturing techno
nonwetting of 401n408	nonwetting of 40In405a	nonwetting of 401n405
microstructural coarse	microstructural coarse	microstructural coarse
mal cycling of 58Bi42S	mal cycling of 58Bi425	mal cycling of 58Bi425
Most of the test data o	Most of the test data o	Most of the test data of
couple of exceptions 'I	couple of exceptions. 3	couple of exceptions.
low-temperature solder	low-temperature solder	low-temperature solds
investigated (or some o	investigated (or some o	investigated (or some
manufacturing technol	manufacturing technol	manufacturing techno
nonwerting of 40In40S	nonwetting of 40In40Sn	nonwetting of 40in405
microstructural coarse	microstructural coarse	microstructura) coarse
mal cycling of 58Bi42S	mal cycling of 58Bs42S	mal cycling of 58Bi425
Most of the test data o	Most of the test data o	Most of the test data of
couple of exceptions. I	couple of exceptions. I	couple of exceptions,
low-temperature solde	low-temperature solder	low-temperature solds
investigated (or some o	investigated (or some o	investigated (or some
manufacturing technol	manufacturing technol	manufacturing techno
nonwetting of 40In40S	nonwetting of 40In40Sr	nonwetting of 401n405
microstructural coarse	microstructural coarse	microstructural coarse
mal eveling of 58Bi42S	mal excling of 58B42S	mail cycling of 58Bi425

Ruta Desai, Chun-Liang Li

Lecture 10 - 104

Super-resolution

Each of these images looks like this:

Most of the test data of couple of exceptions. T low-temperature solder investigated (or some c manufacturing technolnonwetting of 40In40St microstructural coarse mai cycling of 58Bi42St

Ruta Desai, Chun-Liang Li



Super-resolution

The recovery result:

Most of the test data o couple of exceptions. T low-temperature solder investigated (or some o manufacturing technol nonwetting of 40In40Sr microstructural coarse mal cycling of 58Bi42Si

Ruta Desai, Chun-Liang Li

Lecture 10 - 106

Problem statement

Image sequence



Slide credit: Yonsei Univ.

Ruta Desai, Chun-Liang Li



Problem statement

Feature point detection



Slide credit: Yonsei Univ.

Ruta Desai, Chun-Liang Li


Problem statement

Feature point tracking



Slide credit: Yonsei Univ.

Ruta Desai, Chun-Liang Li



What we will learn today?

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005] http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Ruta Desai, Chun-Liang Li



Problem setting

- Given a video sequence, find all the features and track them across the video.
- First, use Harris corner detection to find features and their location x.
- For each feature at location $x = [x \ y]^T$:
 - Choose a descriptor create an initial template for that feature: T(x).



KLT objective

Our aim is to find the *p* that minimizes the difference between the template *T(x)* and the description of the new location of *x* after undergoing the transformation.

$$\sum_{x} [I(W(\boldsymbol{x};\boldsymbol{p})) - T(x)]^2$$

- For all the features x in the image I,
 - -I(W(x; p)) is the estimate of where the features move to in the next frame after the transformation defined by W(x; p). Recall that p is our vector of parameters.
 - Sum is over an image patch around *x*.

Ruta Desai, Chun-Liang Li



KLT objective

• Since *p* may be large, minimizing this function may be difficult:

$$\sum_{x} [I(W(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x})]^2$$

- We will instead break down $oldsymbol{p} = oldsymbol{p}_0 + \, \Delta oldsymbol{p}$
 - Large + small/residual motion
 - Where p_0 is going to be fixed and we will solve for Δp , which is a small value.
 - We can initialize p_0 with our best guess of what the motion is and initialize Δp as zero.

Ruta Desai, Chun-Liang Li



A little bit of math: Taylor series

• Taylor series is defined as:

$$f(x + \Delta x) = f(x) + \Delta x \frac{\partial f}{\partial x} + \Delta x^2 \frac{\partial^2 f}{\partial x^2} + \dots$$

- Assuming that Δx is small.
- We can apply this expansion to the KLT tracker and only use the first two terms:

Ruta Desai, Chun-Liang Li



Expanded KLT objective

$$\sum_{x} [I(W(x; p_0 + \Delta p)) - T(x)]^2$$

$$\approx \sum_{x} \left[I(W(x; p_0)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

It's a good thing we have already calculated what $\frac{\partial W}{\partial p}$ would look like for affine, translations and other transformations!

Ruta Desai, Chun-Liang Li



Expanded KLT objective

• So our aim is to find the Δp that minimizes the following:

$$\operatorname{argmin}_{\Delta p} \sum_{x} \left[I(W(x; p_0)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

- Where $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}$
- Differentiate wrt Δp and setting it to zero:

$$\sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^{T} \left[I(W(\boldsymbol{x}; \boldsymbol{p}_{0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - T(\boldsymbol{x}) \right] = 0$$

Ruta Desai, Chun-Liang Li

Lecture 10 - 116

-



Solving for Δp

- Solving for $\Delta \boldsymbol{p}$ in: $\sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^{T} \left[I(W(\boldsymbol{x}; \boldsymbol{p}_{0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - T(\boldsymbol{x}) \right] = 0$
- we get:

$$\Delta \boldsymbol{p} = H^{-1} \sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^{T} \left[T(x) - I(W(\boldsymbol{x}; \boldsymbol{p}_{0})) \right]$$

where $H = \sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^{T} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]$

Ruta Desai, Chun-Liang Li



Interpreting the H matrix for translation transformations

$$H = \sum_{x} \left[\nabla I \frac{\partial W}{\partial p} \right]^{T} \left[\nabla I \frac{\partial W}{\partial p} \right]$$

Recall that

1.
$$\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}$$
 and

2. for translation motion,
$$\frac{\partial W}{\partial p}(x; p) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Therefore

$$H = \sum_{x} \begin{bmatrix} I_{x} & I_{y} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix}^{T} \begin{bmatrix} I_{x} & I_{y} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ = \sum_{x} \begin{bmatrix} I_{x}^{2} & I_{x}I_{y} \\ I_{x}I_{y} & I_{y}^{2} \end{bmatrix}$$
That's the Harris corner detector we learnt in class!!!

Ruta Desai, Chun-Liang Li

Lecture 10 - 118

Apr 30, 2025

Interpreting the H matrix for affine transformations

$$H = \sum_{\mathbf{x}} \begin{bmatrix} I_x^2 & I_x I_y & x I_x^2 & y I_x I_y & x I_x I_y & y I_x I_y \\ I_x I_y & I_y^2 & x I_x I_y & y I_y^2 & x I_y^2 & y I_y^2 \\ x I_x^2 & y I_x I_y & x^2 I_x^2 & y^2 I_x I_y & x y I_x I_y & y^2 I_x I_y \\ y I_x I_y & y I_y^2 & x y I_x I_y & y^2 I_y^2 & x y I_y^2 & y^2 I_y^2 \\ x I_x I_y & x I_y^2 & x^2 I_x I_y & x y I_y^2 & x^2 I_y^2 & x y I_y^2 \end{bmatrix}$$

Can you derive this yourself similarly to how we derived the translation transformation?

Ruta Desai, Chun-Liang Li



Overall KLT tracker algorithm $\left[T_{x} \left[T(x) - I(W(x; p_{0})) \right] \right]^{T}$

Given the features from Harris detector:

- 1. Initialize $oldsymbol{p}_{0}$ and \Deltaoldsymbol{p} .
- 2. Compute the initial templates T(x) for each feature.
- 3. Transform the features in the image I with $W(\mathbf{x}; \mathbf{p}_0)$.
- 4. Measure the error: $I(W(x; p_0)) T(x)$.
- 5. Compute the image gradients $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}$.
- 6. Evaluate the Jacobian $\frac{\partial W}{\partial p}$.
- 7. Compute steepest descent $\nabla I \frac{\partial W}{\partial p}$.
- 8. Compute Inverse Hessian H^{-1}
- 9. Calculate the change in parameters Δp
- 10. Update parameters $p_0 = p_0 + \Delta p$
- 11. Repeat 2 to 10 until Δp is small.

Ruta Desai, Chun-Liang Li



KLT over multiple frames

- Once you find a transformation for two frames, you will repeat this process for every couple of frames.
- Run Harris detector every 15-20 frames to find new features.

Ruta Desai, Chun-Liang Li



Challenges to consider

- Implementation issues
- Window size
 - Small window more sensitive to noise and may miss larger motions (without pyramid)
 - Large window more likely to cross an occlusion boundary (and it's slower)
 - \circ 15x15 to 31x31 seems typical
- Weighting the window
 - Common to apply weights so that center matters more (e.g., with Gaussian)

Ruta Desai, Chun-Liang Li

