Announcements

- Next Wed team presentations start
- Please select the paper you want to present
- P2 submission deadline has been postponed to Friday 16th
Optical flow and tracking

- Introduction
- Optical flow & KLT tracker
- Motion segmentation
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 $$(t)$$.
Tracking features

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

Vector field function of the spatio-temporal image brightness variations

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

Vector field function of the spatio-temporal image brightness variations

http://www.youtube.com/watch?v=JlLkkom6tWw
Uses of motion

- Improving video quality
  - Motion stabilization
  - Super resolution
- Segmenting objects based on motion cues
- Tracking objects
- Recognizing events and activities
Super-resolution


- Fast and Robust Multiframe Super Resolution, Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar, EEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 10, OCTOBER 2004

Example: A set of low quality images
Super-resolution

Each of these images looks like this:

Most of the test data of a couple of exceptions. The low-temperature solder investigated (or some manufacturing technology) nonwetting of 40In-10Sn microstructural coarse-grain cycling of 8Sn42Sb...
Super-resolution

The recovery result:

Most of the test data of a couple of exceptions. The low-temperature solder investigated (or some of the manufacturing technologies) nonwetting of 40In40Sn microstructural coarse-grained cycling of 58Bi42Sn.
Visual SLAM

Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology
Segmenting objects based on motion cues

- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static *background* from the moving *foreground*
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 06
Tracking objects

• Facing tracking on openCV

OpenCV's face tracker uses an algorithm called Camshift (based on the meanshift algorithm)

http://www.youtube.com/watch?v=HTk_UwAYzVk
Tracking objects

Real-Time Facial Feature Tracking on a Mobile Device
P. A. Tresadern, M. C. Ionita, T. F. Cootes in IJCV (2012)

Fig. 1 Facial feature tracking running in real-time on the Nokia N900 smartphone. A video is available from http://www.youtube.com/watch?v=Y86rOh1Y_kk
FaceHugger: The ALIEN Tracker

**Object Tracking by Oversampling Local Features.** Del Bimbo, and F. Pernici, IEEE Transaction On Pattern Analysis And Machine Intelligence, 2014

- Use Scale Invariant Feature Transform (SIFT) when applied to (flat) objects

http://www.micc.unifi.it/pernici/#alien

DOWNLOAD http://www.micc.unifi.it/pernici/
Joint tracking and 3D localization

W. Choi & K. Shahid & S. Savarese WMC 2009
W. Choi & S. Savarese, ECCV, 2010
Tracking body parts

Cascaded Models for Articulated Pose Estimation, B Sapp, A Toshev, B Taskar, Computer Vision–ECCV 2010, 406-420

Courtesy of Benjamin Sapp
Recognizing events and activities

Recognizing group activities
Crossing – Talking – Queuing – Dancing – jogging

Choi & Savarese, CVPR 11
Choi & Savarese, ECCV 2012

X: Crossing, S: Waiting, Q: Queuing, W: Walking, T: Talking, D: Dancing
Motion estimation techniques

• Optical flow
  – Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

• Feature-tracking
  – Extract visual features (corners, textured areas) and “track” them over multiple frames
Optical flow

Definition: optical flow is the *apparent* motion of brightness patterns in the image

**GOAL:** Recover image motion at each pixel by optical flow

Note: apparent motion can be caused by lighting changes without any actual motion
Estimating optical flow

Given two subsequent frames, estimate the apparent motion field \( u(x,y), v(x,y) \) between them

- **Key assumptions**
  - **Brightness constancy**: projection of the same point looks the same in every frame
  - **Small motion**: points do not move very far
  - **Spatial coherence**: points move like their neighbors

Given two subsequent frames, estimate the apparent motion field \( u(x,y), v(x,y) \) between them.
The brightness constancy constraint

\[ (x, y)^\text{v} \quad \text{displacement} = (u, v) \]
\[ I(x,y,t-1) \]

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:

Image derivative along \( x \)

\[ I(x + u, y + u, t) \approx I(x, y, t - 1) + \left[ I_x \cdot u(x, y) + I_y \cdot v(x, y) + I_t \right] \]

\[ I(x + u, y + u, t) - I(x, y, t - 1) = I_x \cdot u(x, y) + I_y \cdot v(x, y) + I_t \]

Hence,

\[ I_x \cdot u + I_y \cdot v + I_t \approx 0 \quad \rightarrow \nabla I \cdot [u \ v]^T + I_t = 0 \]
The brightness constancy constraint

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

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

How many equations and unknowns per pixel?

- One equation (this is a scalar equation!), two unknowns \((u,v)\)
Adding constraints....


How to get more equations for a pixel?

Spatial coherence constraint:
Assume the pixel’s neighbors have the same \((u,v)\)

- If we use a 5x5 window, that gives us 25 equations per pixel

\[
0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v] \quad \quad p_i = (x_i, y_i)
\]

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
I_x(p_2) & I_y(p_2) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25}) \\
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
=
\begin{bmatrix}
I_t(p_1) \\
I_t(p_2) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]
Lucas-Kanade flow

Overconstrained linear system:

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

\[
A \quad d = b
\]

25x2 2x1 25x1
Lucas-Kanade flow

Overconstrained linear system

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

\[
A \begin{bmatrix}
u \\
v
\end{bmatrix} = b
\]

Least squares solution for \(d\) given by

\[
(A^T A) \begin{bmatrix}
u \\
v
\end{bmatrix} = A^T b
\]

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} =
- \begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

\[
A^T A \quad A^T b
\]

The summations are over all pixels in the K x K window.
Conditions for solvability

• Optimal \((u, v)\) satisfies Lucas-Kanade equation

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

\(A^T A\)

\(A^T b\)

When is this solvable?

• \(A^T A\) should be invertible
• Eigenvalues \(\lambda_1\) and \(\lambda_2\) of \(A^T A\) should not be too small
• \(A^T A\) should be well-conditioned
  – \(\lambda_1 / \lambda_2\) should not be too large (\(\lambda_1 =\) larger eigenvalue)

Does this remind anything to you?
$M = A^T A$ is the second moment matrix!
(Harris corner detector…)

$$M = A^T A = \begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}$$

- Eigenvectors and eigenvalues of $A^T A$ 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
Interpreting the eigenvalues

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

- **Corner**: \( \lambda_1 \) and \( \lambda_2 \) are large,
  - \( \lambda_1 \sim \lambda_2 \)

- **Edge**: \( \lambda_1 \) and \( \lambda_2 \) are large,
  - \( \lambda_1 \gg \lambda_2 \)

- **Flat** region:
  - \( \lambda_1 \) and \( \lambda_2 \) are small
Edge

\[ \sum \nabla I \left( \nabla I \right)^T \]
- gradients very large or very small
- large \( \lambda_1 \), small \( \lambda_2 \)
Low-texture region

\[ \sum \nabla I(\nabla I)^T \]

- gradients have small magnitude
- small \( \lambda_1 \), small \( \lambda_2 \)
High-texture region

\[ \sum \nabla I (\nabla I)^T \]

- gradients are different, large magnitudes
- large \( \lambda_1 \), large \( \lambda_2 \)
What are good features to track?

Can we measure “quality” of features from just a single image

Good features to track:
- Harris corners (guarantee small error sensitivity)

Bad features to track:
- Image points when either $\lambda_1$ or $\lambda_2$ (or both) is small (i.e., edges or uniform textured regions)
Ambiguities in tracking a point on a line

The component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be measured.

This equation $\nabla I \cdot [u' \ v']^T = 0$ is always satisfied when $(u', v')$ is perpendicular to the image gradient.
The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion
The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion
Aperture problem cont’d

* From Marc Pollefeys COMP 256 2003
Motion estimation techniques

Optical flow
- Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

Feature-tracking
- Extract visual features (corners, textured areas) and “track” them over multiple frames
  - Shi-Tomasi feature tracker
  - Tracking with dynamics

- Implemented in Open CV
Shi-Tomasi feature tracker


Find good features using eigenvalues of second-moment matrix

- Key idea: “good” features to track are the ones that can be tracked reliably

From frame to frame, track with Lucas-Kanade and a pure translation model

- More robust for small displacements, can be estimated from smaller neighborhoods

Check consistency of tracks by affine registration to the first observed instance of the feature

- Affine model is more accurate for larger displacements
- Comparing to the first frame helps to minimize drift
Tracking example

Figure 1: Three frame details from Woody Allen’s *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1, 6, 11, 16, 21 as tracked (top), and warped by the computed deformation matrices (bottom).
Recap

- **Key assumptions (Errors in Lucas-Kanade)**
  - **Small motion**: points do not move very far
  - **Brightness constancy**: projection of the same point looks the same in every frame
  - **Spatial coherence**: points move like their neighbors
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?
Reduce the resolution!

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Coarse-to-fine optical flow estimation

Gaussian pyramid of image 1 (t)  
- $u=1.25$ pixels
- $u=2.5$ pixels
- $u=5$ pixels
- $u=10$ pixels

Gaussian pyramid of image 2 (t+1)
Coarse-to-fine optical flow estimation

Gaussian pyramid of image 1 (t)  Gaussian pyramid of image 2 (t+1)

run L-K

image 1

image 2
Multi-resolution Lucas Kanade Algorithm

- Compute ‘simple’ LK at highest level
- At level $i$
  - Take flow $u_{i-1}$, $v_{i-1}$ from level $i-1$
  - bilinear interpolate it to create $u_i^*$, $v_i^*$ matrices of twice resolution for level $i$
  - multiply $u_i^*$, $v_i^*$ by 2
  - compute $f_t$ from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections $u_i'$, $v_i'$, i.e. $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$. 
Optical Flow Results

Lucas-Kanade without pyramids
Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Optical Flow Results

- http://www.ces.clemson.edu/~stb/klt/
- OpenCV

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Recap

- **Key assumptions** (Errors in Lucas-Kanade)

  - **Small motion**: points do not move very far
  - **Brightness constancy**: projection of the same point looks the same in every frame
  - **Spatial coherence**: points move like their neighbors
Motion segmentation

How do we represent the motion in this scene?
Motion segmentation


Break image sequence into “layers” each of which has a coherent (affine) motion
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:

\[ I_x \cdot u + I_y \cdot v + I_t \approx 0 \]
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:

\[
I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0
\]

- Each pixel provides 1 linear constraint in 6 unknowns
- If we have at least 6 pixels in a neighborhood, \( a_1 \ldots a_6 \) can be found by least squares minimization:

\[
Err(\bar{a}) = \sum \left[ I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2
\]
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

2. Map into motion parameter space

3. 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

2. Map into motion parameter space

3. 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

4. Assign each pixel to best hypothesis --- iterate
Next lecture:

Recognition & classification