Computer Vision - Lecture 20

Motion and Optical Flow

22.01.2015

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de
leibe@vision.rwth-aachen.de

Many slides adapted from K. Grauman, S. Seitz, R. Szeliski, M. Pollefeys, S. Lazebnik
Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Uncalibrated Reconstruction
  - Active Stereo
- Motion
  - Motion and Optical Flow
- 3D Reconstruction (Reprise)
  - Structure-from-Motion
Recap: Epipolar Geometry - Calibrated Case

\[ x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T Ex' = 0 \quad \text{with} \quad E = [t \times]R \]

Essential Matrix (Longuet-Higgins, 1981)
Recap: Epipolar Geometry - Uncalibrated Case

\[
\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}
\]

\[
x = K \hat{x}
\]

\[
x' = K' \hat{x}'
\]

Fundamental Matrix
(Faugeras and Luong, 1992)
Recap: The Eight-Point Algorithm

\[ x = (u, v, 1)^T, \quad x' = (u', v', 1)^T \]

\[
(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0
\]

This minimizes:

\[
\sum_{i=1}^{N} (x_i^T F x_i')^2
\]

Solve using… SVD!

This minimizes:

\[
A \mathbf{f} = 0
\]

Slide adapted from Svetlana Lazebnik

B. Leibe
Recap: Normalized Eight-Point Algorithm

1. Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels.

2. Use the eight-point algorithm to compute $F$ from the normalized points.

3. Enforce the rank-2 constraint using SVD.

\[
\text{SVD} \quad F = U D V^T = U \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T
\]

Set $d_{33}$ to zero and reconstruct $F$

4. Transform fundamental matrix back to original units: if $T$ and $T'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T F T'$.

[Hartley, 1995]
Practical Considerations

1. Role of the baseline
   - Small baseline: large depth error
   - Large baseline: difficult search problem

• Solution
  - Track features between frames until baseline is sufficient.

Slide adapted from Steve Seitz
Topics of This Lecture

• Introduction to Motion
  ➢ Applications, uses

• Motion Field
  ➢ Derivation

• Optical Flow
  ➢ Brightness constancy constraint
  ➢ Aperture problem
  ➢ Lucas-Kanade flow
  ➢ Iterative refinement
  ➢ Global parametric motion
  ➢ Coarse-to-fine estimation
  ➢ Motion segmentation

• KLT Feature Tracking

B. Leibe
Video

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

- Sometimes, motion is the only cue...
Motion and Perceptual Organization

- Sometimes, motion is foremost cue
Motion and Perceptual Organization

- Even “impoverished” motion data can evoke a strong percept

Slide credit: Svetlana Lazebnik
Motion and Perceptual Organization

- Even “impoverished” motion data can evoke a strong percept
Uses of Motion

- Estimating 3D structure
  - Directly from optic flow
  - Indirectly to create correspondences for SfM
- Segmenting objects based on motion cues
- Learning dynamical models
- Recognizing events and activities
- Improving video quality (motion stabilization)
Motion Estimation Techniques

• Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

• Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)
Topics of This Lecture

• Introduction to Motion
  ➢ Applications, uses

• Motion Field
  ➢ Derivation

• Optical Flow
  ➢ Brightness constancy constraint
  ➢ Aperture problem
  ➢ Lucas-Kanade flow
  ➢ Iterative refinement
  ➢ Global parametric motion
  ➢ Coarse-to-fine estimation
  ➢ Motion segmentation

• KLT Feature Tracking
Motion Field

- The motion field is the projection of the 3D scene motion into the image

Slide credit: Svetlana Lazebnik
Motion Field and Parallallax

- \( P(t) \) is a moving 3D point
- Velocity of 3D scene point: \( V = \frac{dP}{dt} \)
- \( p(t) = (x(t), y(t)) \) is the projection of \( P \) in the image.
- Apparent velocity \( v \) in the image: given by components \( v_x = \frac{dx}{dt} \) and \( v_y = \frac{dy}{dt} \)
- These components are known as the \textit{motion field} of the image.

Slide credit: Svetlana Lazebnik
Motion Field and Parallax

\[
V = [V_x, V_y, V_z] \quad p = f \frac{P}{Z}
\]

To find image velocity \( v \), differentiate \( p \) with respect to \( t \) (using quotient rule):

\[
v = f \frac{ZV - V_z p}{Z^2} = \frac{fV - V_z p}{Z}
\]

- Image motion is a function of both the 3D motion \( V \) and the depth of the 3D point \( Z \).

Slide credit: Svetlana Lazebnik
**Motion Field and Parallax**

- Pure translation: \( V \) is constant everywhere

\[
\begin{align*}
v_x &= \frac{fV_x - V_z x}{Z} \\
v_y &= \frac{fV_y - V_z y}{Z}
\end{align*}
\]

\[
v = \frac{1}{Z} (v_0 - V_z p),
\]

\[
v_0 = (fV_x, fV_y)
\]
Motion Field and Parallax

• Pure translation: \( V \) is constant everywhere
  \[
  v = \frac{1}{Z} (v_0 - V_z p),
  \]
  \[
  v_0 = (fV_x, fV_y)
  \]

• \( V_z \) is nonzero:
  
  Every motion vector points toward (or away from) \( v_0 \), the vanishing point of the translation direction.
Motion Field and Parallax

- Pure translation: $V$ is constant everywhere
  \[ v = \frac{1}{Z} (v_0 - V_z p), \]
  \[ v_0 = (fV_x, fV_y) \]

- $V_z$ is nonzero:
  - Every motion vector points toward (or away from) $v_0$, the vanishing point of the translation direction.

- $V_z$ is zero:
  - Motion is parallel to the image plane, all the motion vectors are parallel.

- The length of the motion vectors is inversely proportional to the depth $Z$. 

Slide credit: Svetlana Lazebnik
Topics of This Lecture

• Introduction to Motion
  ➢ Applications, uses

• Motion Field
  ➢ Derivation

• Optical Flow
  ➢ Brightness constancy constraint
  ➢ Aperture problem
  ➢ Lucas-Kanade flow
  ➢ Iterative refinement
  ➢ Global parametric motion
  ➢ Coarse-to-fine estimation
  ➢ Motion segmentation

• KLT Feature Tracking
Optical Flow

• Definition: optical flow is the \textit{apparent} motion of brightness patterns in the image.

• Ideally, optical flow would be the same as the motion field.

• Have to be careful: apparent motion can be caused by lighting changes without any actual motion.
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination.
Apparent Motion ≠ Motion Field

Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.
Estimating Optical Flow

Given two subsequent frames, estimate the apparent motion field \( u(x,y) \) and \( v(x,y) \) 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.
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 hand side using Taylor expansion:**
  \[ I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y) \]

- **Hence,**
  \[ (I_x \cdot u + I_y \cdot v + I_t) \approx 0 \]
The Brightness Constancy Constraint

\[ I_x \cdot u + I_y \cdot v + I_t = 0 \]

- How many equations and unknowns per pixel?
  - One equation, two unknowns

- Intuitively, what does this constraint mean?
  \[ \nabla I \cdot (u, v) + I_t = 0 \]

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

If \((u, v)\) satisfies the equation, so does \((u+u', v+v')\) if \(\nabla I \cdot (u', v') = 0\).

Slide credit: Svetlana Lazebnik
The Aperture Problem

Perceived motion
The Aperture Problem

Actual motion
The Barber Pole Illusion

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

Slide credit: Svetlana Lazebnik
The Barber Pole Illusion

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

Slide credit: Svetlana Lazebnik
The Barber Pole Illusion

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

Slide credit: Svetlana Lazebnik

B. Leibe
Solving the Aperture Problem

• How to get more equations for a pixel?
• **Spatial coherence constraint**: pretend the pixel’s neighbors have the same \((u, v)\)
  
  - If we use a 5x5 window, that gives us 25 equations per pixel
  
  \[
  0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v]
  \]

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


Slide credit: Svetlana Lazebnik
Solving the Aperture Problem

• Least squares problem:

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

\[
A \quad d = b
\]

25x2 2x1 25x1

• Minimum least squares solution given by solution of

\[
(A^T A) \quad d = A^T b
\]

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

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

(The summations are over all pixels in the \(K \times K\) window)

Slide adapted from Svetlana Lazebnik
Conditions for Solvability

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

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

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

- When is this solvable?
  - \(A^T A\) should be invertible.
  - \(A^T A\) entries should not be too small (noise).
  - \(A^T A\) should be well-conditioned.
Eigenvectors of $A^T A$

\[
A^T A = \begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T
\]

- Haven’t we seen an equation like this before?
- Recall the Harris corner detector: $M = A^T A$ is the second moment matrix.
- The eigenvectors and eigenvalues of $M$ 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:

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

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

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

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

- Gradients very large or very small
- Large \( \lambda_1 \), small \( \lambda_2 \)

Slide credit: Svetlana Lazebnik
Low-Texture Region

$$\sum \nabla I(\nabla I)^T$$

- Gradients have small magnitude
- Small $\lambda_1$, small $\lambda_2$

Slide credit: Svetlana Lazebnik
High-Texture Region

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

- Gradients are different, large magnitude
- Large \( \lambda_1 \), large \( \lambda_2 \)

Slide credit: Svetlana Lazebnik
Per-Pixel Estimation Procedure

• Let \( M = \sum (\nabla I)(\nabla I)^T \) and \( b = \begin{bmatrix} -\sum I_xI_t \\ -\sum I_yI_t \end{bmatrix} \)

• Algorithm: At each pixel compute \( U \) by solving \( MU = b \)

• \( M \) is singular if all gradient vectors point in the same direction
  - E.g., along an edge
  - Trivially singular if the summation is over a single pixel or if there is no texture
  - I.e., only normal flow is available (aperture problem)

• Corners and textured areas are OK
Iterative Refinement

1. Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation.

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

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

2. Warp one image toward the other using the estimated flow field.

- (Easier said than done)

3. Refine estimate by repeating the process.
Optical Flow: Iterative Refinement

Initial guess: $d_0 = 0$
Estimate: $d_1 = d_0 + \tilde{d}$

(using $d$ for displacement here instead of $u$)
Optical Flow: Iterative Refinement

Initial guess: $d_1$

Estimate: $d_2 = d_1 + \hat{d}$

(Using $d$ for displacement here instead of $u$)
Optical Flow: Iterative Refinement

Initial guess: \( d_2 \)

Estimate: \( d_3 = d_2 + \tilde{d} \)

(\textit{using } d \textit{ for displacement here instead of } u \textit{)
Optical Flow: Iterative Refinement

\[ f_1(x - d_3) \approx f_2(x) \]

(using \( d \) for displacement here instead of \( u \))
Optic Flow: Iterative Refinement

- Some Implementation Issues:
  - Warping is not easy (ensure that errors in warping are smaller than the estimate refinement).
  - Warp one image, take derivatives of the other so you don’t need to re-compute the gradient after each iteration.
  - Often useful to low-pass filter the images before motion estimation (for better derivative estimation, and linear approximations to image intensity).
Extension: Global Parametric Motion Models

Translation: 2 unknowns
Affine: 6 unknowns
Perspective: 8 unknowns
3D rotation: 3 unknowns

Slide credit: Steve Seitz
Example: 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:

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

• Least squares minimization:

\[ \text{Err}(\vec{a}) = \sum [ I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t ]^2 \]
Problem Cases in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Iterative refinement, coarse-to-fine estimation
- A point does not move like its neighbors
  - Motion segmentation
- Brightness constancy does not hold
  - Do exhaustive neighborhood search with normalized correlation.
Dealing with Large Motions
Temporal Aliasing

- Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.
- I.e., how do we know which ‘correspondence’ is correct?

- To overcome aliasing: coarse-to-fine estimation.

![Diagram showing nearest match is correct (no aliasing) and nearest match is incorrect (aliasing)]
Idea: Reduce the Resolution!
Coarse-to-fine Optical Flow Estimation

Gaussian pyramid of image 1

$u=10$ pixels

$u=5$ pixels

$u=2.5$ pixels

$u=1.25$ pixels

Image 1

Gaussian pyramid of image 2

Image 2

Slide credit: Steve Seitz
Coarse-to-fine Optical Flow Estimation

Image 1: Gaussian pyramid of image 1
Run iterative L-K
Warp & upsample
Run iterative L-K

Image 2: Gaussian pyramid of image 2

Slide credit: Steve Seitz
Dense Optical Flow

- Dense measurements can be obtained by adding smoothness constraints.

T. Brox, C. Bregler, J. Malik, Large displacement optical flow, CVPR‘09, Miami, USA, June 2009.
Summary

- **Motion field**: 3D motions projected to 2D images; dependency on depth.
- **Solving for motion with**
  - Sparse feature matches
  - Dense optical flow
- **Optical flow**
  - Brightness constancy assumption
  - Aperture problem
  - Solution with spatial coherence assumption
References and Further Reading

• Here is the original paper by Lucas & Kanade

• And the original paper by Shi & Tomasi

• Read the story how optical flow was used for special effects in a number of recent movies
  - [http://www.fxguide.com/article333.html](http://www.fxguide.com/article333.html)