Readings

• Szeliski, R. Ch. 7
• Bergen et al. ECCV 92, pp. 237-252.
• Shi, J. and Tomasi, C. CVPR 94, pp.593-600.

• Slide Credits: Szeliski, Shah and B. Freeman
Recap: Estimating Optical Flow

• Assume the image intensity $I$ is constant

\[ I(x, y, t) = I(x + dx, y + dy, t + dt) \]
**First Assumption: Brightness Constraint**

\[ I(x, y, t) \approx I(x + dx, y + dy, t + dt) \]

\[ I(x(t) + u.\Delta t, y(t) + v.\Delta t) - I(x(t), y(t), t) \approx 0 \]

Assuming $I$ is differentiable function, and expand the first term using Taylor’s series:

\[ \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \]

Compact representation

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

Brightness constancy constraint
Second Assumption: Gradient Constraint

Velocity vector is constant within a small neighborhood (LUCAS AND KANADE)

\[
E(u, v) = \int_{x,y} (I_x u + I_y v + I_t)^2 \, dx \, dy
\]

\[
\frac{\partial E(u, v)}{\partial u} = \frac{\partial E(u, v)}{\partial v} = 0
\]

\[
2(I_x u + I_y v + I_t)I_x = 0
\]

\[
2(I_x u + I_y v + I_t)I_y = 0
\]
Recap: Lucas-Kanade

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

Structural
Tensor
representation

\[
\begin{bmatrix}
T_{xx} & T_{xy} \\
T_{xy} & T_{yy}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
T_{xt} \\
T_{yt}
\end{bmatrix}
\]

\[
u = \frac{T_{yt} T_{xy} - T_{xt} T_{yy}}{T_{xx} T_{yy} - T_{xy}^2}
\quad \text{and} \quad
v = \frac{T_{xt} T_{xy} - T_{yt} T_{xx}}{T_{xx} T_{yy} - T_{xy}^2}
\]
Pitfalls & Alternatives

• Brightness constancy is not satisfied
  – Correlation based method could be used
• A point may not move like its neighbors
  – Regularization based methods
• The motion may not be small (Taylor does not hold!)
  – Multi-scale estimation could be used
Multi-Scale Flow Estimation

Gaussian pyramid of image $I_t$

$u=10$ pixels

$u=5$ pixels

$u=2.5$ pixels

$u=1.25$ pixels

Gaussian pyramid of image $I_{t+1}$

image $I_t$

image $I_{t+1}$
Recap: Horn & Schunck

- Global method with smoothness constraint to solve aperture problem
- Minimize a global energy function

\[ E(u, v) = \int_{x,y} \left( (I_x u + I_y v + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2) \right) dx dy \]

- Take partial derivatives w.r.t. \( u \) and \( v \):

\[ (I_x u + I_y v + I_t) I_x - \alpha^2 \nabla u = 0 \]
\[ (I_x u + I_y v + I_t) I_y - \alpha^2 \nabla v = 0 \]
Global Motion Models (Parametric)

All pixels are considered to summarize global motion!

• **2D Models**
  – Affine
  – Quadratic
  – Planar projective (homography)

• **3D Models**
  – Inst. Camera motion models
  – Homography + epipole
  – Plane + parallax
Motion Models

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

Estimate motion using all pixels in the image
Global Motion

- Estimate motion using all pixels in the image

Global Motion can be used to
- Remove camera motion
- Object-based segmentation
- generate mosaics
Global Motion

Estimate motion using all pixels in the image

Global Motion can be used to
- Remove camera motion
- Object-based segmentation
- generate mosaics
Object Tracking

• Track an object over a sequence of images
Challenges in Object Tracking
Challenges in Object Tracking

• Which features to track?
Challenges in Object Tracking

- Which features to track?
- Efficient tracking
Challenges in Object Tracking

- Which features to track?
- Efficient tracking
- Appearance constraint violation
- ...
Challenges in Object Tracking

• Which features to track?
• Efficient tracking
• Appearance constraint violation
• ...

20
Shi-Tomasi Feature Tracker

• Good Features to Track
Shi-Tomasi Feature Tracker

• Good Features to Track
  – Find good features using eigenvalues of Hessian matrix (threshold on the smallest eigenvalue when computing Harris corner detection)
Shi-Tomasi Feature Tracker

• Good Features to Track
  – Find good features using eigenvalues of Hessian matrix (threshold on the smallest eigenvalue when computing Harris corner detection)
  – Track from frame to frame with LK
Shi-Tomasi Feature Tracker

• Good Features to Track
  – Find good features using eigenvalues of Hessian matrix (threshold on the smallest eigenvalue when computing Harris corner detection)
  – Track from frame to frame with LK
  – Check consistency of tracks by “affine registration” to the first observed instance of the feature
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).
KLT Tracking

- KLT: Kanade-Lucas-Tomasi
KLT Tracking

• KLT: Kanade-Lucas-Tomasi
• Tracking deals with estimating the trajectory of an object in the image plane as it moves around a scene
KLT Tracking

• KLT: Kanade-Lucas-Tomasi
• Tracking deals with estimating the trajectory of an object in the image plane as it moves around a scene
• Object tracking (car, airplane, person)
KLT Tracking

- KLT: Kanade-Lucas-Tomasi
- **Tracking** deals with estimating the trajectory of an object in the image plane as it moves around a scene
- Object tracking (car, airplane, person)
- Feature tracking (Harris corners)
KLT Tracking

• KLT: Kanade-Lucas-Tomasi
• Tracking deals with estimating the trajectory of an object in the image plane as it moves around a scene
• Object tracking (car, airplane, person)
• Feature tracking (Harris corners)
• Multiple object tracking
KLT Tracking

- KLT: Kanade-Lucas-Tomasi
- **Tracking** deals with estimating the trajectory of an object in the image plane as it moves around a scene
- Object tracking (car, airplane, person)
- Feature tracking (Harris corners)
- Multiple object tracking
- Tracking in single/multiple camera(s)
KLT Tracking

- KLT: Kanade-Lucas-Tomasi
- **Tracking** deals with estimating the trajectory of an object in the image plane as it moves around a scene
- Object tracking (car, airplane, person)
- Feature tracking (Harris corners)
- Multiple object tracking
- Tracking in single/multiple camera(s)
- Tracking in fixed/moving camera
KLT Tracking Algorithm

1) Find **GoodFeaturesToTrack**
   Harris Corners (thresholded on smallest eigenvalues)

2) Use LK algorithm to find optical flows

3) Use Coarse-to-Fine strategy to deal with large movements

4) When creating long tracks, check appearance of registered patch against appearance of initial patch to find points that have drifted
Recent Developments at Optical Flow

- Start with LK or similar methods
  + Gradient consistency
  + Energy minimization with smoothing term
  + Region matching
  + KeyPoint matching

Large displacement optical flow, Brox et al., CVPR 2009
Recent Developments at Optical Flow

- Use of Machine Learning
  - Deep Learning (ICCV 2015, Fischer et al., FlowNet)
DeepFlow (Large Displacement Optical Flow)

• Basically it is a matching algorithm with variational approach [Weinzaepfel et al., ICCV 2013].

• Dense correspondence (matching)
• Self-smooth matching
• Large displacement optical flow
  – https://www.youtube.com/watch?v=k_wkDLJ8IJE
• Can we use **SIFT features** for tracking?
Ex: SIFT Tracking

Frame 0 → Frame 100
How to evaluate correctness of optical flows?
Optical Flow - Quantitative Evaluation

\[ E_{ep2} = \sqrt{(u - u^*)^2 + (v - v^*)^2} \]

\[ E_{ep1} = |u - u^*| + |v - v^*| \]

- Where \( u=(u,v) \) is computed, \( u=(u^*,v^*) \) ground truth velocity vectors.

\[ E_{ang} = \arccos\left( \frac{u^T u^*}{||u|| ||u^*||} \right) \]
Interpretation of Optical Flow Fields
Interpretation of Optical Flow Fields

Object features all have Zero velocity.
Interpretation of Optical Flow Fields
Interpretation of Optical Flow Fields

Object is moving to the Right.
Interpretation of Optical Flow Fields
Interpretation of Optical Flow Fields

Object is moving directly toward the camera that is stationary.
Interpretation of Optical Flow Fields
Interpretation of Optical Flow Fields

Camera is moving into the scene, and an object moving passed the camera
Interpretation of Optical Flow Fields
Interpretation of Optical Flow Fields

Object is rotating about the line of sight to the camera
Interpretation of Optical Flow Fields
Object is rotating about an axis perpendicular to the line of sight.
Application in Image Alignment

- Motion can be used for image alignment

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  a & b \\
  c & d
\end{bmatrix} \begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

Pixel locations at time t+1

Pixel locations at time t
Practice: Homogenous Coordinates

Q: How can we represent translation as a 3x3 matrix?

\[ x' = x + t_x \]
\[ y' = y + t_y \]
Practice: Homogenous Coordinates

Q: How can we represent translation as a 3x3 matrix?

\[ x' = x + t_x \]
\[ y' = y + t_y \]

Translation matrix:

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\]
Practice: Homogenous Coordinates

Q: How can we represent translation as a 3x3 matrix?

\[ x' = x + t_x \]
\[ y' = y + t_y \]

Translation matrix:

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} =
\begin{bmatrix}
x + t_x \\
y + t_y \\
1
\end{bmatrix}
\]
Practice: Basic 2D Transformations

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & t_x \\
  0 & 1 & t_y \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

Translate

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  \cos \Theta & -\sin \Theta & 0 \\
  \sin \Theta & \cos \Theta & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

Rotate

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  s_x & 0 & 0 \\
  0 & s_y & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

Scale

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  1 & sh_x & 0 \\
  sh_y & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

Shear
Affine Transformation

• Affine transformations are combinations of ...
  – Linear transformations, and
  – Translations

• Properties of affine transformations:
  – Origin does not necessarily map to origin
  – Lines map to lines
  – Parallel lines remain parallel
  – Ratios are preserved
  – Closed under composition
  – Models change of basis

\[
\begin{bmatrix}
  x' \\
  y' \\
  w
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x' \\
  y' \\
  w'
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

projective
Affine Transformation

\[
\begin{bmatrix}
  x' \\
  y' \\
  w
\end{bmatrix}
= \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

Affine matrix decomposition
Translation+rotation+scaling

\[
\begin{bmatrix}
  x' \\
  y' \\
  w'
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & tx \\
  0 & 1 & ty \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  \cos \Theta & -\sin \Theta & 0 \\
  \sin \Theta & \cos \Theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  sx & 0 & 0 \\
  0 & sy & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

\[p' = T(t_x,t_y) \quad R(\Theta) \quad S(s_x,s_y) \quad p\]
Questions?