CS231A

Computer Vision: From 3D Reconstruction to Recognition

Optical and Scene Flow
Learning Goals for Upcoming Lectures

Representations & Representation Learning

Monocular Depth Estimation, Feature Tracking

Optical & Scene Flow

Optimal Estimation

Neural Radiance Fields

A Database and Evaluation Methodology for Optical Flow.
Baker et al. IJCV. 2011
What will you learn today?

Optical Flow
  What is it and why do you care?
  Assumptions
  Formulating the optimization problem
  Solving it
Scene Flow
Learning-based Approaches to Estimating Motion
Optical Flow - What is it?

J. J. Gibson, The Ecological Approach to Visual Perception
Optical Flow - What is it?

Motion Field

Motion field = 2D motion field representing the projection of the 3D motion of points in the scene onto the image plane.

B. Horn, Robot Vision, MIT Press
Optical flow = 2D velocity field describing the apparent motion in the images.
What is the motion field? What is the apparent motion?

Lambertian (matte) ball rotating in 3D

What does the 2D motion field look like?

What does the 2D optical flow field look like?
What is the motion field? What is the apparent motion?

Stationary Lambertian (matte) ball

Moving Light Source

What does the 2D motion field look like?

What does the 2D optical flow field look like?
Optical flow - What is it?

Motion Displacement of all image pixels

Image pixel value at time $t$ and Location $x = (x, y): I(x, y, t)$

$u(x, y)$ horizontal component
$v(x, y)$ vertical component

Slide Credit: Michael Black
Optical Flow - What is it good for?

Painterly effect
Optical Flow - What is it good for?

Face morphing in matrix reloaded

Slide Credit: Michael Black
Optical Flow - What is it good for?

Optical Flow - What is it good for?

Caren Marzban and Scott Sandgathe
Optical Flow for Verification, Weather and Forecasting,
Volume 25 No. 5, October 2010

Slide Credit: Michael Black
Optical Flow - What is it good for?

![Diagram showing optical flow and its applications in RGB inputs, scene flow, and segmentation.](image)
Optical Flow - What is it good for?

$I(t), \{p_i\}$

Optical Flow

Velocity vectors $\{\vec{v}_i\}$

Slide Credit: CS223b – Sebastian Thrun
Compute Optical Flow

Goal
Compute the **apparent** 2D image motion of pixels from one image frame to the next in a video sequence.
Compute (Sparse) Optical Flow

Also see CS131a
Simple KLT Tracker

History of the Kanade-Lucas-Tomasi (KLT) Tracker

1981
An Iterative Image Registration Technique with an Application to Stereo Vision.

1991
Detection and Tracking of Feature Points.

1994
Good Features to Track.

The original KLT algorithm

16-385 Computer Vision (Kris Kitani)
Simple KLT Tracker

1. Find good points to track (Harris corners)
2. For each Harris corner compute motion (translation or affine) between consecutive frames
3. Link motion vector of successive frames to get a track for each Harris point
4. Introduce new Harris points by running detector every 10-15 frames
5. Track old and new corners using step 1-3
Computing (Sparse) Optical Flow

$I(t), \{p_i\}$

Optical Flow

Velocity vectors $\{\vec{v}_i\}$

Jean-Yves Bouguet, Ph.D., CalTech
Compute (Dense) Optical Flow

Step 1 - Assumptions

Step 2 - Objective Function

Step 3 - Optimization

Assumption 1 - Brightness Constancy

\[ I(x + u, y + v, t + 1) = I(x, y, t) \]

\( u, v = \) pixel offset \quad t = \) time \quad \( x, y = \) pixel position

Slide Credit: Michael Black
Assumption 2 - Spatial Smoothness

- Neighboring pixels in the image are likely to belong to the same surface.
- Surfaces are mostly smooth.
- Neighboring pixels will have similar flow.
Assumption 3 – Temporal Coherence

Figure 1.8: Temporal continuity assumption. A patch in the image is assumed to have the same motion (constant velocity, or acceleration) over time.
Compute Optical Flow

Step 1 - Assumptions

Step 2 - Objective Function

Objective Function – Data term - Brightness Constancy

\[ E_D(u, v) = \sum_{S = \text{all pixels}} (I(x_s + u_s, y_s + v_s, t + 1) - I(x, y, t))^2 \]

New Assumption: Quadratic error implies Gaussian noise

Alternative: Huber/L1 Loss
Objective Function – Spatial Term – Spatial Smoothness

\[ E_S(u, v) = \sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2 \]

\[ G(s) = \text{Pixel Neighborhood} \]

New Assumptions:
Flow field smooth
Gaussian Deviations
First order smoothness good enough
Flow derivative approximated by first differences
Objective Function

\[ E(u, v) = E_D(u, v) + \lambda E_S(u, v) \]

Data term

\[ E(u, v) = \sum_{s}(I(x_s + u_s, y_s + v_s, t + 1) - I(x, y, t))^2 + \lambda \left( \sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2 \right) \]

Spatial term

Nonlinear Optimization

Optimization Variables

Relative weighting term
Compute Optical Flow

Step 1 - Assumptions

Step 2 - Objective Function

Step 3 - Optimization

Linear Approximation

\[ E(u, v) = E_D(u, v) + \lambda E_S(u, v) \]

\[ E(u, v) = \sum_s (I(x_s + u_s, y_s + v_s, t + 1) - I(x, y, t))^2 + \lambda (\sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2) \]

\[ u_s = dx, v_s = dy, dt = 1 \]

Partial Derivative in x direction

Partial Derivative in y direction

\[ I(x, y, t) + dx \frac{\delta}{\delta x} I(x, y, t) + dy \frac{\delta}{\delta y} I(x, y, t) + dt \frac{\delta}{\delta t} I(x, y, t) - I(x, y, t) = 0 \]

Constraint Equation for Optical Flow
Optical Flow Constraint Equation

Linearized cost function

\[ u \frac{\delta}{\delta x} I(x, y, t) + v \frac{\delta}{\delta y} I(x, y, t) + \frac{\delta}{\delta t} I(x, y, t) = 0 \]

\[ I_x u + I_y v + I_t = 0 \quad = \text{Constraint at every pixel} \]

New Assumptions:
Flow is small
Image is differentiable
First order Taylor series is a good approximation
Optical Flow Constraint Equation

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

\[ \nabla I^T u = -I_t \]

\[ u = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \]

At a single image pixel, we get a line:

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

"Normal flow"
Aperture Problem
Aperture Problem
What are the constraint lines?
Multiple Constraints

Each pixel gives us a constraint: $I_x u + I_y v = -I_t$
Image Gradient Examples - Edge
Image Gradient Examples – Low texture
Image Gradient Examples – Low texture
How do we solve this optimization problem?

\[ E(u, v) = \sum_{x, y \in R} (I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t))^2 \]

\[
\frac{\partial E}{\partial u} = \sum_R (I_xu + I_yv + I_t)I_x = 0 \\
\frac{\partial E}{\partial v} = \sum_R (I_xu + I_yv + I_t)I_y = 0
\]

Horn-Schunk Method
How do we solve this optimization problem?

Rearrange in Matrix form

\[
\begin{bmatrix}
\sum_{R} I_x^2 & \sum_{R} I_x I_y \\
\sum_{R} I_x I_y & \sum_{R} I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= \begin{bmatrix}
-\sum_{R} I_x I_t \\
-\sum_{R} I_y I_t
\end{bmatrix}
\]
How do we solve this optimization problem?

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

\[Au = b\]
How do we solve this optimization problem?

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

\[ Au = b \]

If \( A \) was invertible

\[ A^{-1} Au = A^{-1} b \]

\[ u = A^{-1} b \]

\[ Au = b \]

\[ A^T Au = A^T b \]

\[ u = (A^T A)^{-1} A^T b \]

Pseudoinverse
Bag of tricks

Small motion assumption
Bag of tricks

Reduce Resolution

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Bag of tricks

Reduce Resolution

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Bag of tricks

Reduce Resolution

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Spatial Pyramids

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

$u=10$ pixels
$u=5$ pixels
$u=2.5$ pixels
$u=1.25$ pixels

Gaussian pyramid of image $I$

$u=10$ pixels
Scene Flow = 3D Optical Flow

A Database and Evaluation Methodology for Optical Flow. Baker et al. IJCV. 2011

What are the main challenges with this traditional formulation?

• Assumptions
  – Brightness constancy
  – Small motion
  – Etc

• Occlusions

• Large motion
Learning-based approaches

• Since 2015 - FlowNet
• Availability of synthetic data, e.g. Sintel
FlowNet - Learning Optical Flow with Convolutional Networks

Supervised Learning with Labeled Data Set

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Smagt, D. Cremers, Thomas Brox. IEEE International Conference on Computer Vision (ICCV), 2015
FlowNet - Learning Optical Flow with Convolutional Networks

1. FlowNetSimple
2. FlowNetCorr

Cross correlation

Supervised Learning
Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing

Spatial Pyramids

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

$u=10$ pixels

$u=5$ pixels

$u=2.5$ pixels

$u=1.25$ pixels

Gaussian pyramid of image $I$

$u=10$ pixels

$u=5$ pixels

$u=2.5$ pixels

$u=1.25$ pixels
Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing

Cost Volume

Original Color Image, 384 rows by 512 columns.

Template Image to Search For, 50 rows by 72 columns.

Normalized Cross Correlation Output, 433 rows by 583 columns.

Template Image Found in Original Image.

Cost volume
Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 2018
Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 2018
Scene Flow

A Database and Evaluation Methodology for Optical Flow. Baker et al. IJCV. 2011

Motion-based Object Segmentation based on Dense RGB-D Scene Flow
Motion-based Object Segmentation based on Dense RGB-D Scene Flow

Motion-based Object Segmentation based on Dense RGB-D Scene Flow
Motion-based Object Segmentation based on Dense RGB-D Scene Flow

<table>
<thead>
<tr>
<th>RGB Inputs</th>
<th>Scene Flow</th>
<th>Ground Truth</th>
<th>HOMC</th>
<th>Segmentation</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>frame $I^{t-1}$</td>
<td>PD-flow</td>
<td>OurC+V+Rig</td>
<td>Ground Truth</td>
<td>OurC+V+Rig</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>frame $I^t$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Motion-based Object Segmentation based on Dense RGB-D Scene Flow

<table>
<thead>
<tr>
<th>RGB Inputs</th>
<th>Scene Flow</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>frame $I^{t-1}$</td>
<td>PD-flow</td>
<td>HOMC</td>
</tr>
<tr>
<td>frame $I^t$</td>
<td>OurC+vL+Rig</td>
<td>OurC+vL+Rig</td>
</tr>
</tbody>
</table>

CS231
Introduction to Computer Vision

Next lecture:
Optimal Recursive Estimation