Optical Flow and Correspondences

CS 6384 Computer Vision
Professor Yu Xiang
The University of Texas at Dallas
**Human Motion Perception**

- Separate moving figure from a stationary background

- Motion for 3D perception
  - Look at a fruit by rotating it around

- Guide actions
  - Walking down the street or hammering a nail
Motion from Eye Movement

\[ \omega_y, \omega_x, \omega_z \]
Motion from Eye Movement

Closer pixel, larger displacement

$lateral$ $\mathbf{v}_x$

$vertical$ $\mathbf{v}_y$

$forward/backward$ $\mathbf{v}_z$
Motion from Object Movement
Optical Flow

• The pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer and a scene

• Velocity field

\[(v_x, v_y) = \left( \frac{dx}{dt}, \frac{dy}{dt} \right)\]

\[(x, y) \mapsto (-1, 0)\]

\[(x, y) \mapsto (x + y, x + y)\]
Brightness Constancy Constraint

\[ I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \]

Taylor series

\[ I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + \text{higher-order terms} \]

\[ f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \cdots \]
Brightness Constancy Constraint

\[
\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0
\]

\[
\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0
\]

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

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

\[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \] (spatial gradient; we can compute this!)

\[ \frac{dx}{dt}, \frac{dy}{dt} = (u, v) \] (optical flow, what we want to find)

\[ \frac{\partial I}{\partial t} \] (derivative across frames. Also known, e.g. frame difference)
Image Gradient

• Derivative of a function

\[ f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h} \]

• Central difference is more accurate

\[ f'(x) = \lim_{h \to 0} \frac{f(x + 0.5h) - f(x - 0.5h)}{h} \]

• Image gradient with central difference
  • Applying a filter

\[
\begin{array}{c}
1 \\
0 \\
-1
\end{array}
\quad
\begin{array}{c}
1 \\
0 \\
-1
\end{array}
\]

X derivative

Y derivative
Image Gradient

• Sobel Filter

\[
S_x = \begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{bmatrix}
\]

\[
S_y = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
\]

\[
\frac{\partial f}{\partial x} = S_x \otimes f
\]

\[
\frac{\partial f}{\partial y} = S_y \otimes f
\]

\[
\nabla f = \begin{bmatrix}
\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}
\end{bmatrix}
\]

x-derivative

weighted average and scaling
Brightness Constancy Constraint

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

- For each pixel, there are two unknowns

Known (spatial and temporal gradients)

Unknown (optical flow)

https://sites.math.washington.edu/~king/coursedir/m445w04/notes/vector/normals-planes.html
Brightness Constancy Constraint

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

- The component of the flow vector in the gradient direction is determined (called normal flow) (Recall vector projection geometry)

\[
\frac{1}{\sqrt{I_x^2 + I_y^2}} (I_x, I_y) \cdot (u, v) = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}} \quad \text{Projection}
\]

- The component of the flow vector orthogonal to this direction cannot be determined.

Lucas-Kanade Method

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

- Assumption: the flow is constant in a local neighborhood of a pixel under consideration
- Use two or more pixels to compute optical flow

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

\[
A_{25 \times 2} 
\begin{bmatrix}
u \\
v
\end{bmatrix} = b_{25 \times 1}
\]

5x5 window
Lucas-Kanade Method

• Solve the least squares problem

\[
A \quad d = b \\
\begin{array}{c}
25 \times 2 \\
2 \times 1 \\
25 \times 1
\end{array}
\rightarrow \text{minimize } \|Ad - b\|^2
\]

\[
\|Ad - b\|^2 = (Ad - b)^T(Ad - b) = (d^T A^T - b^T)(Ad - b)
\]

\[
= d^T A^T Ad - d^T A^T b - b^T Ad + b^T b
\]

Scalar

Scalar

\[
= d^T A^T Ad - 2d^T A^T b + b^T b
\]

Take derivate with respect to d, and set to 0

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

Lucas-Kanade Method

- Solve the least squares problem

\[
\begin{align*}
A \quad d &= b \\
25 \times 2 & \quad 2 \times 1 & \quad 25 \times 1 \\
\rightarrow \text{minimize} & \quad \|Ad - b\|^2 \\
(ATA) \quad d &= ATb \\
2 \times 2 & \quad 2 \times 1 & \quad 2 \times 1 \\
d &= (ATA)^{-1}ATb \\
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

https://en.wikipedia.org/wiki/Proofs_involving_ordinary_least_squares#Least_squares_estimator_for_\text{CE.B2}
Optical Flow Example
FlowNet

FlowNetSimple

Stack two images

Learnable Up-sampling: Deconvolution

Input: 2 x 2

Output: 4 x 4

3 x 3 “deconvolution”, stride 2, pad 1
Learnable Up-sampling: Deconvolution

Input: 2 x 2

3 x 3 “deconvolution”, stride 2, pad 1

Output: 4 x 4

Input gives weight for filter
Learnable Up-sampling: Deconvolution

Input: 2 x 2

Output: 4 x 4

3 x 3 “deconvolution”, stride 2, pad 1

Input gives weight for filter

Sum where output overlaps
FlowNet

• Refinement

FlowNet

FlowNet

- Correlation layer: multiplicative patch comparison between two feature maps

\[ c(x_1, x_2) = \sum_{o \in [-k, k] \times [-k, k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle \]

- Two patches centered at \(x_1\) and \(x_2\), with size \(K = 2k + 1\)
- Convolve data with another data
- Limit the patches for comparison with maximum displacement \(d\)
- Only compare patches in a neighborhood with size \(D = 2d + 1\)
- Output size \((w \times h \times D^2)\)

Correspondences

Optical flow

SIFT matching

Semantic keypoints
Universal Correspondences Network

• Learn pixel-wise features for matching

• Fully-convolutional network

• Contrastive loss function for feature learning

• Convolutional spatial transformer

Universal Correspondence Network. Choy et al., NuerIPS, 2016
Universal Correspondences Network

Universal Correspondence Network. Choy et al., NeurIPS, 2016
Universal Correspondences Network

• Correspondence contrastive loss

\[ L = \frac{1}{2N} \sum_{i}^{N} s_i \| \mathcal{F}_\mathcal{I}(x_i) - \mathcal{F}_{\mathcal{I}'}(x_i') \|^2 + (1 - s_i) \max(0, m - \| \mathcal{F}_\mathcal{I}(x) - \mathcal{F}_{\mathcal{I}'}(x_i') \|)^2 \]

positive pair

negative pair

Universal Correspondence Network. Choy et al., NuerIPS, 2016
Spatial Transformer Network

Affine transformation
\[
\begin{pmatrix}
  x_i^s \\
  y_i^s
\end{pmatrix}
= T_\theta(G_i) = A_\theta
\begin{pmatrix}
  x_i^t \\
  y_i^t
\end{pmatrix}
= \begin{bmatrix}
  \theta_{11} & \theta_{12} & \theta_{13} \\
  \theta_{21} & \theta_{22} & \theta_{23}
\end{bmatrix}
\begin{pmatrix}
  x_i^t \\
  y_i^t
\end{pmatrix}
\]

Spatial Transformer Networks. Jaderberg et al., NeurIPS, 2015
Universal Correspondences Network

Universal Correspondence Network. Choy et al., NuerIPS, 2016
Universal Correspondences Network

Universal Correspondence Network. Choy et al., NuerIPS, 2016
Self-supervised Correspondences Learning

- Use 3D reconstruction techniques to find pixel correspondences

**Correspondences from DynamicFusion**

**KinectFusion**

Positive pairs and negative pairs

Contrastive loss

\[
L(I_a, I_b, u_a, u_b, M_a, M_b) = \begin{cases} 
D(I_a, I_b, u_a, u_b)^2, & M_a(u) = M_b(u) \\
\max(0, M - D(I_a, I_b, u_a, u_b))^2, & \text{otherwise}
\end{cases}
\]

3D model coordinate

Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimdt et al., RA-L, 2017
Self-supervised Correspondences Learning

Training videos

Testing videos

Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimdt et al., RA-L, 2017
Self-supervised Correspondences Learning

https://youtu.be/jfXyAypAQWk

Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimidt et al., RA-L, 2017
Further Reading

• Lucas–Kanade method  
  https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method

  https://arxiv.org/abs/1504.06852

• Universal Correspondence Network, 2016  
  https://arxiv.org/abs/1606.03558

• Self-Supervised Visual Descriptor Learning for Dense Correspondence, 2017  