Computer Vision 2
WS 2018/19
Part 4 – Template-based Tracking II
23.10.2018

Prof. Dr. Bastian Leibe
RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Course Outline
• Single-Object Tracking
  – Background modeling
  – Template based tracking
  – Tracking by online classification
  – Tracking-by-detection
• Bayesian Filtering
• Multi-Object Tracking
• Visual Odometry
• Visual SLAM & 3D Reconstruction
• Deep Learning for Video Analysis

Recap: Estimating Optical Flow

• 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.

Recap: Lucas-Kanade Optical Flow

• Use all pixels in a \( K \times K \) window to get more equations.
• Least squares problem:

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
I_x(p_2) & I_y(p_2) \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix}
I_x(p_1) \\
I_y(p_2) \\
I_x(p_{25})
\end{bmatrix}, \quad A \cdot d = b
\]

• Minimum least squares solution given by solution of

\[
A^T A \cdot \hat{d} = A^T b
\]

Recall the Harris detector!

Recap: Iterative LK Refinement

• Estimate velocity at each pixel using one iteration of LK estimation.
• Warp one image toward the other using the estimated flow field.
• Refine estimate by repeating the process.

• Iterative procedure
  – Results in subpixel accurate localization.
  – Converges for small displacements.
Recap: Coarse-to-fine Optical Flow Estimation

Gaussian pyramid of image 1
Gaussian pyramid of image 2

Recap: Shi-Tomasi Feature Tracker (→KLT)

- Idea
  - Find good features using eigenvalues of second-moment matrix
  - Key idea: "good" features to track are the ones that can be tracked reliably.
- Frame-to-frame tracking
  - Track with LK and a pure translation motion model.
  - More robust for small displacements, can be estimated from smaller neighborhoods (e.g., 5×5 pixels).
- Checking consistency of tracks
  - Affine registration to the first observed feature instance.
  - Affine model is more accurate for larger displacements.
  - Comparing to the first frame helps to minimize drift.

J. Shi and C. Tomasi, Good Features to Track. CVPR 1994.

Lucas-Kanade Template Tracking

- Traditional LK
  - Typically run on small, corner-like features (e.g., 5×5 patches) to compute optical flow (→ KLT).
  - However, there is no reason why we can’t use the same approach on a larger window around the tracked object.

Basic LK Derivation for Templates

\[ E(u, v) = \sum \left[I(x + u, y + v) - T(x, y)\right]^2 \]

Template model

Current frame

(u, v) = hypothesized location of template in current frame
We assume that an initial estimate of $p$ is known and iteratively solve for increments to the parameters $\Delta p$:

$$\arg\min_{\Delta p} \sum_x [I(W(x; p + \Delta p)) - T(x)]^2$$
Step-by-Step Derivation

- Key to the derivation
  - Taylor expansion around $\Delta p$
    $$I(W(x; p + \Delta p)) \approx I(W(x; p)) + \nabla I(W(x; p)) \Delta p + O(\Delta p^2)$$
  - Using pixel coordinates $x = [x, y]$
    $$I(W([x, y]; p + \Delta p)) \approx I(W([x, y]; p)) + \nabla I(W([x, y]; p)) \Delta p$$
  - Taking partial derivatives
    $$\nabla I(W([x, y]; p)) = \frac{\partial I(W([x, y]; p))}{\partial p} = \begin{bmatrix} \frac{\partial I}{\partial p_1} \\ \frac{\partial I}{\partial p_2} \\ \vdots \\ \frac{\partial I}{\partial p_n} \end{bmatrix}$$
    $$\nabla^2 I(W([x, y]; p)) = \frac{\partial^2 I(W([x, y]; p))}{\partial p \partial p'} = \begin{bmatrix} \frac{\partial^2 I}{\partial p_1 \partial p_1} & \frac{\partial^2 I}{\partial p_1 \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_1 \partial p_n} \\ \frac{\partial^2 I}{\partial p_2 \partial p_1} & \frac{\partial^2 I}{\partial p_2 \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_2 \partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 I}{\partial p_n \partial p_1} & \frac{\partial^2 I}{\partial p_n \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_n \partial p_n} \end{bmatrix}$$

Step-by-Step Derivation

- And further collecting the derivative terms
  $$I(W([x, y]; p + \Delta p)) \approx I(W([x, y]; p)) + \nabla I(W([x, y]; p)) \Delta p$$
  $$+ \begin{bmatrix} \frac{\partial I}{\partial p_1} \\ \frac{\partial I}{\partial p_2} \\ \vdots \\ \frac{\partial I}{\partial p_n} \end{bmatrix} \Delta p_1$$
  $$+ \begin{bmatrix} \frac{\partial^2 I}{\partial p_2 \partial p_1} & \frac{\partial^2 I}{\partial p_2 \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_2 \partial p_n} \\ \frac{\partial^2 I}{\partial p_3 \partial p_1} & \frac{\partial^2 I}{\partial p_3 \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_3 \partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 I}{\partial p_n \partial p_1} & \frac{\partial^2 I}{\partial p_n \partial p_2} & \cdots & \frac{\partial^2 I}{\partial p_n \partial p_n} \end{bmatrix} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_n \end{bmatrix}$$

Example: Jacobian of Affine Warp

- General equation of Jacobian
  $$\frac{\partial W}{\partial p} = \begin{bmatrix} \frac{\partial W_{x_1}}{\partial p_1} & \frac{\partial W_{x_2}}{\partial p_1} & \cdots & \frac{\partial W_{x_n}}{\partial p_1} \\ \frac{\partial W_{x_1}}{\partial p_2} & \frac{\partial W_{x_2}}{\partial p_2} & \cdots & \frac{\partial W_{x_n}}{\partial p_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial W_{x_1}}{\partial p_n} & \frac{\partial W_{x_2}}{\partial p_n} & \cdots & \frac{\partial W_{x_n}}{\partial p_n} \end{bmatrix}$$

- Affine warp function (6 parameters)
  $$W([x, y]; p) = \begin{bmatrix} 1 + p_1 & p_2 & p_3 \\ 1 & 1 + p_4 & p_5 \\ 1 & 1 & 1 + p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Result
  $$\frac{\partial W}{\partial p} = \begin{bmatrix} x + p_1 x + p_3 y + p_5 \\ y + p_2 y + p_4 y + p_6 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step-by-Step Derivation

- Rewriting this in matrix notation
  $$I(W([x, y]; p + \Delta p)) \approx I(W([x, y]; p)) + \nabla I(W([x, y]; p)) \Delta p$$

Minimizing the Registration Error

- Optimization function after Taylor expansion
  $$\arg \min_{\Delta p} \sum_x \left[ I(W(x; p)) + \nabla I(W(x; p)) \Delta p - T(x) \right]^2$$

- Minimizing this function
  - How?

Minimizing the Registration Error

- Optimization function after Taylor expansion
  $$\arg \min_{\Delta p} \sum_x \left[ I(W(x; p)) + \nabla I(W(x; p)) \Delta p - T(x) \right]^2$$

- Minimizing this function
  - Taking the partial derivative and setting it to zero
    $$\frac{\partial}{\partial \Delta p} \left| \sum_x \left[ I(W(x; p)) + \nabla I(W(x; p)) \Delta p - T(x) \right] \right| = 0$$

  - Closed-form solution for $\Delta p$ (Gauss-Newton):
    $$\Delta p = H^{-1} \sum_x \begin{bmatrix} \nabla^2 I(W(x; p)) \\ \nabla I(W(x; p)) \\ I(W(x; p)) - I(W(x; p)) \end{bmatrix}$$

  - Where $H$ is the Hessian
    $$H = \sum_x \begin{bmatrix} \nabla^2 I(W(x; p)) \\ \nabla I(W(x; p)) \end{bmatrix} \begin{bmatrix} \nabla^2 I(W(x; p)) & \nabla I(W(x; p)) \\ \nabla I(W(x; p)) & I(W(x; p)) \end{bmatrix}$$
Inverse Compositional LK Algorithm

- Iterate
  - Warp \( I \) to obtain \( I(W([x, y]; p)) \)
  - Compute the error image \( T([x, y]) = I(W([x, y]; p)) \)
  - Warp the gradient \( \nabla I \) with \( W([x, y]; p) \)
  - Evaluate \( \frac{\partial W}{\partial p} \) at \( ([x, y]; p) \)
  - Compute steepest descent images
  - Compute Hessian matrix
  - Compute
  - Update the parameters \( p \leftarrow p + \Delta p \)
- Until \( \Delta p \) magnitude is negligible

\[
H = \sum_x \left[ \frac{\partial I(W)}{\partial p} \right]^T \cdot \left[ \frac{\partial I(W)}{\partial p} \right] \\
\Delta p = H^{-1} \sum_x \left[ \frac{\partial I(W)}{\partial p} \right]^T \cdot \left[ I([x, y]) - I(W([x, y]; p)) \right]
\]

Discussion LK Alignment

- Pros
  - All pixels get used in matching
  - Can get sub-pixel accuracy (important for good mosaicking)
  - Fast and simple algorithm
  - Applicable to Optical Flow estimation, stereo disparity estimation, parametric motion tracking, etc.
- Cons
  - Prone to local minima.
  - Relatively small movement.
  - Good initialization necessary

Side Note

- LK Registration needs a good initialization
  - Taylor expansion corresponds to a linearization around the initial position \( p \).
  - This linearization is only valid in a small neighborhood around \( p \).
- When tracking templates...
  - We typically use the previous frame’s result as initialization.
    - The higher the frame rate, the smaller the warp will be.
    - This means we get better results and need fewer LK iterations.
    - Tracking becomes easier (and faster!) with higher frame rates.

Discussion

- Beyond 2D Tracking/Registration
  - So far, we focused on registration between 2D images.
  - The same ideas can be used when performing registration between a 3D model and the 2D image (model-based tracking).
  - The approach can also be extended for dealing with articulated objects and for tracking in subspaces.
  - We will come back to this in later lectures when we talk about model-based 3D tracking...

Topics of This Lecture

- Lucas-Kanade Optical Flow
  - Brightness Constancy constraint
  - LK flow estimation
  - Coarse-to-fine estimation
- Feature Tracking
  - KLT feature tracking
- Template Tracking
  - LK derivation for templates
  - Warping functions
  - General LK image registration
- Applications
Example of a More Complex Warping Function

- Encode geometric constraints into region tracking
- Constrained homography transformation model
  - Translation parallel to the ground plane
  - Rotation around the ground plane normal
  - $W(x) = W_{obj}P_WW_{t\alpha}Qx$

$\Rightarrow$ Input for high-level tracker with car steering model.

References and Further Reading

- The original paper by Lucas & Kanade

- A more recent paper giving a better explanation

- The original KLT paper by Shi & Tomasi