


Computer Vision 2 – Lecture 2

Template-based Tracking (18.04.2016)


Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
leibe@vision.rwth-aachen.de, stueckler@vision.rwth-aachen.de

RWTH Aachen University, Computer Vision Group
<http://www.vision.rwth-aachen.de>




Content of the Lecture


- Single-Object Tracking
 - Background modeling
 - **Template based tracking**
 - Color based tracking
 - Contour based tracking
 - Tracking by online classification
 - Tracking-by-detection
- Bayesian Filtering
- Multi-Object Tracking
- Visual Odometry
- Visual SLAM & 3D Reconstruction



2 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler




Today: Template based Tracking



3 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler


Image source: Robert Collins, Shi & Tomasi



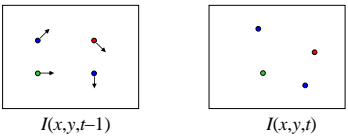
Topics of This Lecture

- **Recap: 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

4 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler




Recap: Estimating Optical Flow



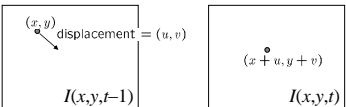
- **Optical Flow**
 - Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them.
- **Key assumptions**
 - **Brightness constancy:** projection of the same point looks the same in every frame.
 - **Small motion:** points do not move very far.
 - **Spatial coherence:** points move like their neighbors.

5 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Slide credit: Svetlana Lazebnik



Recap: 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, $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Spatial derivatives Temporal derivative

6 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Slide credit: Svetlana Lazebnik



Recap: 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 5×5 window, that gives us 25 equations per pixel

$$0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v]$$

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

B. Lucas and T. Kanade, [An iterative image registration technique with an application to stereo vision](#), In *Proc. IJCAI/81*, pp. 674–679, 1981.

13 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Recap: Solving the Aperture Problem

- Least squares problem:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \quad A \quad d = b$$

$\begin{matrix} 25 \times 2 & 2 \times 1 & 25 \times 1 \end{matrix}$
- Minimum least squares solution given by solution of

$$(A^T A) d = A^T b$$

$\begin{matrix} 2 \times 2 & 2 \times 1 & 2 \times 1 \end{matrix}$

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

$\begin{matrix} A^T A & A^T b \end{matrix}$

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

14 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Recap: Conditions for Solvability

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

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

$\begin{matrix} A^T A & A^T b \end{matrix}$
- 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.

⇒ Looking for cases where A has two large eigenvalues (i.e., corners and highly textured areas).

15 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Recap: Iterative LK Refinement

1. Estimate velocity at each pixel using one iteration of LK estimation.

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

$\begin{matrix} A^T A & A^T b \end{matrix}$
2. Warp one image toward the other using the estimated flow field.
 - (Easier said than done)
3. Refine estimate by repeating the process.

16 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide adapted from Steve Seitz

Recap: Iterative LK Refinement

(using d for displacement here instead of u)

17 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Recap: Iterative LK Refinement

(using d for displacement here instead of u)

18 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Recap: Iterative LK Refinement

Initial guess: d_2
Estimate: $d_3 = d_2 + \tilde{d}$

(using d for displacement here instead of u)

19 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Recap: Iterative LK Refinement

(using d for displacement here instead of u)

20 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Problem Case: Large Motions

21 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

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.

22 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Idea: Reduce the Resolution!

23 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Recap: Coarse-to-fine Optical Flow Estimation

Gaussian pyramid of image 1
Gaussian pyramid of image 2

24 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Recap: Coarse-to-fine Optical Flow Estimation

Run iterative LK

Warp & upsample

Run iterative LK

Image 1

Gaussian pyramid of image 1

Image 2

Gaussian pyramid of image 2

25 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Steve Seitz

Topics of This Lecture

- Recap: 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

26 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

KLT Feature Tracking

GPU_KLT:

A GPU-based Implementation of the
Kanade-Lucas-Tomasi Feature Tracker

http://www.cs.unc.edu/~ssinha/Research/GPU_KLT/

27 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Shi-Tomasi Feature Tracker

- 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.

28 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Tracking Example

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).

J. Shi and C. Tomasi. [Good Features to Track](#). CVPR 1994.

29 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
Slide credit: Svetlana Lazebnik

Real-Time GPU Implementations

- This basic feature tracking framework (Lucas-Kanade + Shi-Tomasi) is commonly referred to as "KLT tracking".
 - Used as preprocessing step for many applications
 - Lends itself to easy parallelization
- Very fast GPU implementations available, e.g.,
 - C. Zach, D. Gallup, J.-M. Frahm, [Fast Gain-Adaptive KLT tracking on the GPU](#). In CVGPU'08 Workshop, Anchorage, USA, 2008
 - 216 fps with automatic gain adaptation
 - 260 fps without gain adaptation

http://www.cs.unc.edu/~ssinha/Research/GPU_KLT/
<http://www.inf.ethz.ch/personal/chzach/opensource.html>


30 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Topics of This Lecture

- Recap: 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

31 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 RWTH AACHEN UNIVERSITY


Lucas-Kanade Template Tracking



- Traditional LK
 - Typically run on small, corner-like features (e.g., 5x5 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.

32 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins
 RWTH AACHEN UNIVERSITY

Basic LK Derivation for Templates

$$E(u, v) = \sum_{\mathbf{x}} [I(x + u, y + v) - T(x, y)]^2$$


(u, v) = hypothesized location of template in current frame

33 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins
 RWTH AACHEN UNIVERSITY

Basic LK Derivation for Templates

- Taylor expansion

$$E(u, v) = \sum_{\mathbf{x}} [I(x + u, y + v) - T(x, y)]^2$$

$$\approx \sum_{\mathbf{x}} [I(x, y) + uI_x(x, y) + vI_y(x, y) - T(x, y)]^2$$

$$= \sum_{\mathbf{x}} [uI_x(x, y) + vI_y(x, y) + D(x, y)]^2 \text{ with } D = I - T$$
- Taking partial derivatives

$$\frac{\partial E}{\partial u} = 2 \sum_{\mathbf{x}} [uI_x(x, y) + vI_y(x, y) + D(x, y)] I_x(x, y) \stackrel{!}{=} 0$$


$$\frac{\partial E}{\partial v} = 2 \sum_{\mathbf{x}} [uI_x(x, y) + vI_y(x, y) + D(x, y)] I_y(x, y) \stackrel{!}{=} 0$$
- Equation in matrix form

$$\sum_{\mathbf{x}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{\mathbf{x}} \begin{bmatrix} I_x D \\ I_y D \end{bmatrix} \Rightarrow \text{Solve via least-squares}$$

34 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins
 RWTH AACHEN UNIVERSITY

One Problem With This...

- Problematic Assumption
 - Assumption of constant flow (pure translation) for all pixels in a larger window is unreasonable for long periods of time.



- However...
 - We can easily generalize the LK approach to other 2D parametric motion models (like affine or projective) by introducing a "warp" function **W** with parameters **p**.

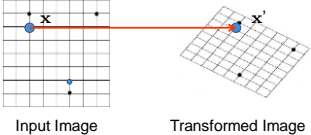
$$E(u, v) = \sum_{\mathbf{x}} [I(x + u, y + v) - T(x, y)]^2$$

$$E(\mathbf{p}) = \sum_{\mathbf{x}} [I(\mathbf{W}([x, y]; \mathbf{p})) - T(x, y)]^2$$

35 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins
 RWTH AACHEN UNIVERSITY

Geometric Image Warping

- The warp **W(x; p)** describes the geometric relationship between two images



$$\mathbf{x}' = \mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{bmatrix} W_x(\mathbf{x}; \mathbf{p}) \\ W_y(\mathbf{x}; \mathbf{p}) \end{bmatrix}$$

Parameters of the warp

36 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Jingpan Chai
 RWTH AACHEN UNIVERSITY

Example Warping Functions

37 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Steve Seitz

Example Warping Functions

- Translation

$$\mathbf{W}([x, y]; \mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
- Affine

$$\mathbf{W}([x, y]; \mathbf{p}) = \begin{bmatrix} x + p_1x + p_3y + p_5 \\ y + p_2x + p_4y + p_6 \end{bmatrix} = \begin{bmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
- Perspective

$$\mathbf{W}([x, y]; \mathbf{p}) = \frac{1}{p_7x + p_8y + 1} \begin{bmatrix} x + p_1x + p_3y + p_5 \\ y + p_2x + p_4y + p_6 \end{bmatrix}$$

– Note: Other parametrizations are possible; the above ones are just particularly convenient here.

38 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Irina Ivanova

General LK Image Registration

- Goal
 - Find the warping parameters \mathbf{p} that minimize the sum-of-squares intensity difference between the template image and the warped input image.
- LK formulation
 - Formulate this as an optimization problem

$$\arg \min_{\mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2$$
 - We assume that an initial estimate of \mathbf{p} is known and iteratively solve for increments to the parameters $\Delta \mathbf{p}$:

$$\arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - T(\mathbf{x})]^2$$

39 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Step-by-Step Derivation

- Key to the derivation
 - Taylor expansion around $\Delta \mathbf{p}$

$$I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) \approx I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} + \mathcal{O}(\Delta \mathbf{p}^2)$$
 - Using pixel coordinates $\mathbf{x} = [x, y]$

$$I(\mathbf{W}([x, y]; \mathbf{p} + \Delta \mathbf{p})) \approx I(\mathbf{W}([x, y]; p_1, \dots, p_n)) + \left[\frac{\partial I}{\partial x} \frac{\partial W_x}{\partial p_1} + \frac{\partial I}{\partial y} \frac{\partial W_y}{\partial p_1} \right]_{p_1} \Delta p_1 + \left[\frac{\partial I}{\partial x} \frac{\partial W_x}{\partial p_2} + \frac{\partial I}{\partial y} \frac{\partial W_y}{\partial p_2} \right]_{p_1} \Delta p_2 + \dots + \left[\frac{\partial I}{\partial x} \frac{\partial W_x}{\partial p_n} + \frac{\partial I}{\partial y} \frac{\partial W_y}{\partial p_n} \right]_{p_n} \Delta p_n$$

40 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins

Step-by-Step Derivation

- Rewriting this in matrix notation

$$I(\mathbf{W}([x, y]; \mathbf{p} + \Delta \mathbf{p})) \approx I(\mathbf{W}([x, y]; p_1, \dots, p_n)) + \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right] \begin{bmatrix} \frac{\partial W_x}{\partial p_1} \\ \frac{\partial W_y}{\partial p_1} \end{bmatrix}_{p_1} \Delta p_1 + \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right] \begin{bmatrix} \frac{\partial W_x}{\partial p_2} \\ \frac{\partial W_y}{\partial p_2} \end{bmatrix}_{p_2} \Delta p_2 + \dots + \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right] \begin{bmatrix} \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_n} \end{bmatrix}_{p_n} \Delta p_n$$

41 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins

Step-by-Step Derivation

- And further collecting the derivative terms

$$I(\mathbf{W}([x, y]; \mathbf{p} + \Delta \mathbf{p})) \approx I(\mathbf{W}([x, y]; p_1, \dots, p_n)) + \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right] \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \dots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \dots & \frac{\partial W_y}{\partial p_n} \end{bmatrix} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_n \end{bmatrix}$$

∇I
 $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
Increment parameters to solve for $\Delta \mathbf{p}$
- Written in matrix form

$$I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) \approx I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}$$

42 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins

Example: Jacobian of Affine Warp

- General equation of Jacobian

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \dots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \dots & \frac{\partial W_y}{\partial p_n} \end{bmatrix}$$
- Affine warp function (6 parameters)

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

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \frac{\partial}{\partial \mathbf{p}} \begin{bmatrix} x + p_1x + p_3y + p_5 \\ p_2x + y + p_4y + p_6 \end{bmatrix} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix}$$

43 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler
 Slide credit: Robert Collins

Minimizing the Registration Error

- Optimization function after Taylor expansion

$$\arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x})]^2$$
- Minimizing this function
 - How?

44 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Minimizing the Registration Error

- Optimization function after Taylor expansion

$$\arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x})]^2$$
- Minimizing this function
 - Taking the partial derivative and setting it to zero

$$\frac{\partial}{\partial \Delta \mathbf{p}} \stackrel{!}{=} 0 \rightarrow 2 \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x})] \stackrel{!}{=} 0$$
 - Closed-form solution for $\Delta \mathbf{p}$ (Gauss-Newton):

$$\Delta \mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]$$
 - where \mathbf{H} is the Hessian $\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]$

45 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Summary: LK Algorithm

- Iterate
 - Warp I to obtain $I(\mathbf{W}([x, y]; \mathbf{p}))$
 - Compute the error image $T([x, y]) - I(\mathbf{W}([x, y]; \mathbf{p}))$
 - Warp the gradient ∇I with $\mathbf{W}([x, y]; \mathbf{p})$
 - Evaluate $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $([x, y]; \mathbf{p})$ (Jacobian)
 - Compute steepest descent images $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
 - Compute Hessian matrix $\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]$
 - Compute $\sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T([x, y]) - I(\mathbf{W}([x, y]; \mathbf{p}))]$
 - Compute $\Delta \mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T([x, y]) - I(\mathbf{W}([x, y]; \mathbf{p}))]$
 - Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$
- Until $\Delta \mathbf{p}$ magnitude is negligible

46 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

LK Algorithm Visualization

[S. Baker, I. Matthews, IJCV04]

47 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Discussion LK Alignment

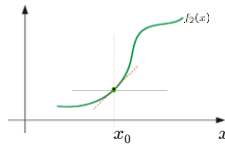
- Pros
 - All pixels get used in matching
 - Can get sub-pixel accuracy (important for good mosaicking)
 - Fast and simple algorithm
 - Applicable to Optical Flow estimation, stereo disparity estimation, parametric motion tracking, etc.
- Cons
 - Prone to local minima.
 - Relatively small movement.
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

48 Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
 Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Side Note

• LK Registration needs a good initialization

- Taylor expansion corresponds to a linearization around the initial position \mathbf{p} .
- This linearization is only valid in a small neighborhood around \mathbf{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.*

49

Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

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...

50

Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Slide credit: Irina Ivanova



Topics of This Lecture

• Recap: Lucas-Kanade Optical Flow

- Brightness Constancy constraint
- LK flow estimation
- Coarse-to-fine estimation

• Feature Tracking

- KLT feature tracking

• Template Tracking

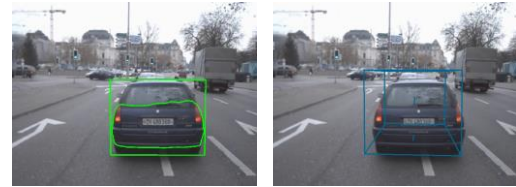
- LK derivation for templates
- Warping functions
- General LK image registration

• Applications

51

Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

Example of a More Complex Warping Function

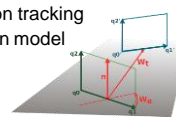


• Encode geometric constraints into region tracking

• Constrained homography transformation model

- Translation parallel to the ground plane
- Rotation around the ground plane normal
- $\mathbf{W}(\mathbf{x}) = \mathbf{W}_{obj} \mathbf{P} \mathbf{W}_t \mathbf{W}_r \mathbf{Q} \mathbf{x}$

⇒ Input for high-level tracker with car steering model.



52

Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler

References and Further Reading

• The original paper by Lucas & Kanade

- B. Lucas and T. Kanade. [An iterative image registration technique with an application to stereo vision](#). In *Proc. IJCAI*, pp. 674–679, 1981.

• A more recent paper giving a better explanation

- S. Baker, I. Matthews. [Lucas-Kanade 20 Years On: A Unifying Framework](#). In *IJCV*, Vol. 56(3), pp. 221–255, 2004.

• The original KLT paper by Shi & Tomasi

- J. Shi and C. Tomasi. [Good Features to Track](#). CVPR 1994.

54

Lecture: Computer Vision 2 (SS 2016) – Template-based Tracking
Prof. Dr. Bastian Leibe, Dr. Jörg Stückler