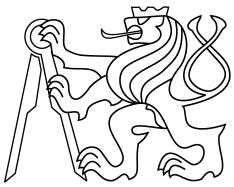
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

Lecture notes

Tracking in image sequences

Lecture notes for the course Computer Vision Methods

Tomáš Svoboda

svobodat@fel.cvut.cz

March 23, 2011

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Contents

The following chapters do not provide a complete survey of motion estimation or object tracking, see e.g. [8]. On the other hand the presented methods are widely accepted and often serve as building blocks in more complex computer vision algorithms.

The text is still a draft and the author will gladly receive any comments and criticism.

1 Patch tracking, Lucas-Kanade tracker and Normalized Cross-Correlation

This chapter introduces one of the essential cornerstones of computer vision, *image alignment*, see Figure 1.1. The problem of finding an image patch within a larger image appears in many areas of computer vision, image matching, tracking, or image based retrieval to name just a few. There are many ways how to compare two image patches here, we will measure the similarity by directly comparing the pixel intensities. The number of papers and amount of work in this area is vast. We do

Figure 1.1: *Goal* is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x}$ is a column vector containing image coordinates $[x, y]^\top$. The $I(\mathbf{x})$ could be also a small sub-window within an image.

not provide exhaustive overview the publications cited here are truly just the seminal works and partly a personal selection of the author(s).

1.1 Measuring pixel-based (di)similarity between patches

The sought criterion function measures a difference between a image patch $I(\mathbf{x})$ and a template $T(\mathbf{x})$, where $\mathbf{x}$ denotes set of spatial image coordinates. The sizes of the

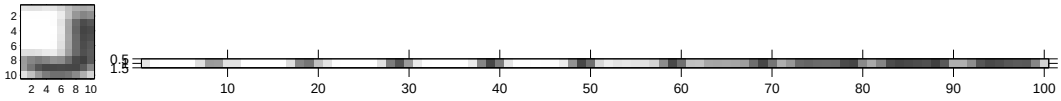


Figure 1.2: The image matrix, image of the "J" character in this case, can be arranged in to a vector.

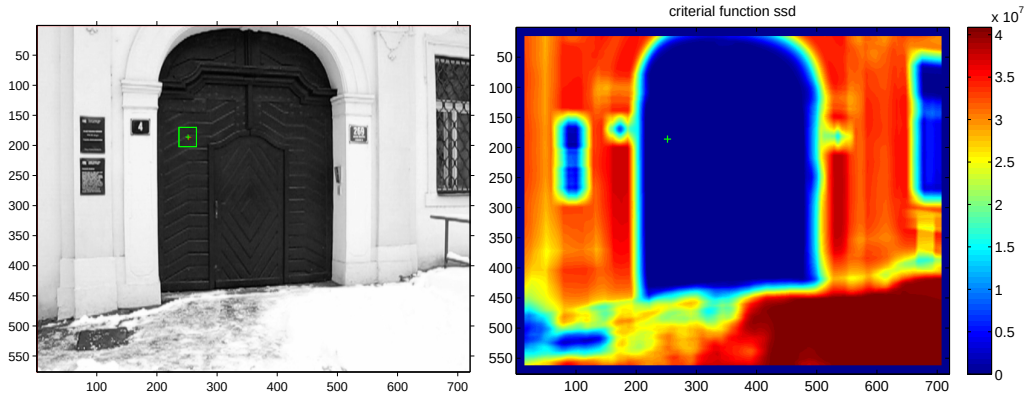


Figure 1.3: Sum of square differences

image patch and the template are assumed to be the same. We can see I and T as discrete 2D function on a regular grid with image intensities as the functional values. Matrix (algebraic) oriented view may see I and T as matrices, which may be conveniently re-arranged into vectors, see Figure 1.2. We can here work with the image as an 1D function which simplifies thinking. A simple difference obviously does not suffice. *Sum of squared differences* (SSD) is an often used similarity measure:

$$ssd(x, y) = \sum_k \sum_l (T(k, l) - I(x + k, y + l))^2 \ , \tag{1.1}$$

where x, y denote position of the patch, think about the center or one of the corners, k, l denote the local template coordinates. The equation above essentially means: compute the square difference in all pixels and sum them up. A simple experiment is visualized on Figure 1.3. The template which is delineated by the green rectangle is compared to the all possible patches of the same size in the image. One can see it as the green window slides over the whole image and in all possible positions the *ssd* measure is computed. Dark blue is the best match (small difference) and dark red is the worst. We can see that the dark patch is clearly distinguishable form the brighter areas but the fine wooden structure on the doors do not play any significant role.

A similar result can be observed for another alternative measure *Sum of absolute*

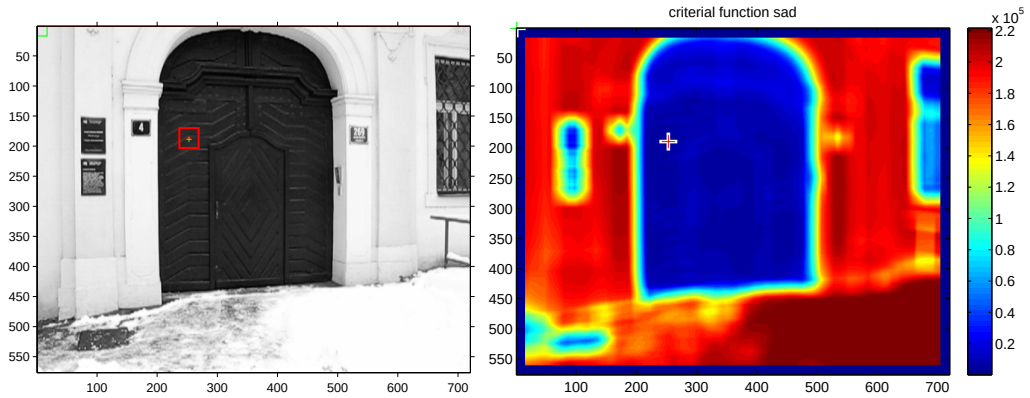


Figure 1.4: Sum of absolute differences

differences

$$sad(x,y) = \sum_k \sum_l |T(k,l) - I(x+k, y+l)| \tag{1.2}$$

see Figure 1.4. The main advantage of *sad* over the *ssd* is naturally the computation cost. Absolute difference is computed faster than the power of two. Yet the *sad* is differentiable in the whole domain.

Normalized cross-correlation is known from statistics and signal processing. It is defined as:

$$r(x,y) = \frac{\sum_k \sum_l \left(T(k,l) - \overline{T}\right)\left(I(x+k,y+l) - \overline{I(x,y)}\right)}{\sqrt{\sum_k \sum_l \left(T(k,l) - \overline{T}\right)^2 \sum_k \sum_l \left(I(x+k,y+l) - \overline{I(x,y)}\right)^2}}, \tag{1.3}$$

where $\overline{T}$ means mean value in the template, and $\overline{I(x,y)}$ mean value of the image patch at the position x, y. From the statistical point of view we may see that the normalized cross-correlation can be computed from variances and covariance.

$$r_{IT} = \frac{\sigma_{TI}}{\sqrt{\sigma_T \sigma_I}}, \tag{1.4}$$

where σ_T is the variance of the intensity signal in the template, σ_I in the image patch and σ_{IT} is the covariance (sometimes called cross-covariance) between the template and image patch. The interpretation is clear: $r(x,y) = 1$ means exactly same patch, $r(x,y) = -1$ means exact negative, $r(x,y) = 0$ just random difference (independent variables have $r = 0$). Result for the normalized cross correlation is visible on Figure 1.5. We can see that this measure is much more discriminative than *ssd* or *sad*. It is able to distinguish individual similar areas on the wooden doors. Although (1.3) may look frightening from the computational point of view several elements may be

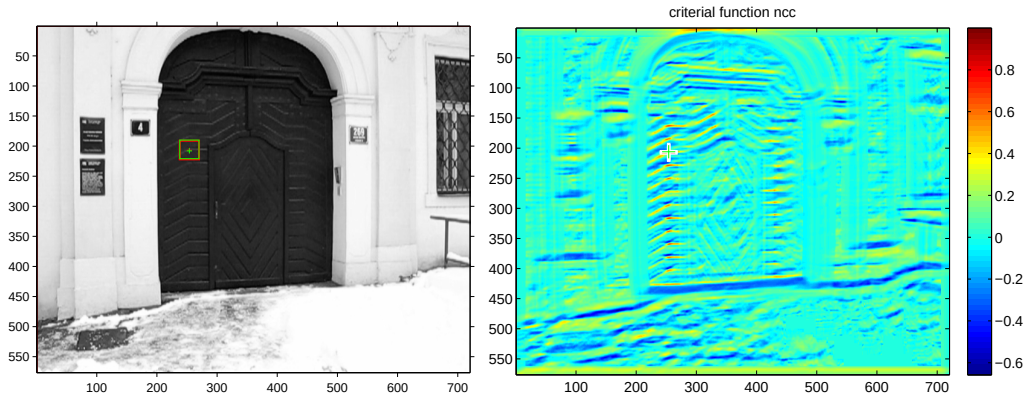


Figure 1.5: Normalized cross-correlation

pre-computed and an efficient algorithm exists [3] and implementations are widely available in many programming languages.

Although the normalized cross correlation has been replaced by more invariant feature detectors [5] it may actually outperform the advanced rivals at several circumstances [6].

We see that all the criterion functions are far from convex. One possible way for aligning the template with its proper position would be to slide over the whole image and selects the best value of the particular criterion function. Despite effective implementations computation complexity of this *sliding window* approach may by prohibitive at many applications. Thankfully, in many application we know at least approximate position and number of possible location is thus conveniently limited. Further possibility would be an optimization approach which is discussed in the next section.

1.2 Lucas-Kanade tracker, and (some) of its modifications

We explain the essentials of this widely adopted optimization technique. Our explanation mainly follows the paper [1], the original paper is [4]. Just remind, the goal is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x}$ is a column vector containing image coordinates $[x, y]^\top$. The $I(\mathbf{x})$ could be also a small sub-window within an image, see Figure 1.1. In the optimization formulation we are at some initial state–position and we search for the optimal transformation that brings us to the perfect alignment. Let define set of allowable warps (geometrical transformations)

$\mathbf{W}(\mathbf{x};\mathbf{p})$, where $\mathbf{p}$ is a vector of parameters. For translations it would be

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \tag{1.5}$$

but $\mathbf{W}(\mathbf{x};\mathbf{p})$ can be arbitrarily complex. The best *alignment*, $\mathbf{p}^*$, minimizes image dissimilarity

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2 \tag{1.6}$$

which is the already known sum of squared differences. It is important to realize that (1.6) is a highly non-linear optimization problem. Though the warp $\mathbf{W}(\mathbf{x};\mathbf{p}))$ can be linear function in $\mathbf{p}$ the pixel values $I(\mathbf{x})$ are not linear $\mathbf{x}$. Essentially, intensity values are (linearly) unrelated to their locations. We can linearize (1.6) in a point $\mathbf{p}$ by using the first order Taylor expansion and, assuming some $\mathbf{p}$ is known, search for the optimal increment

$$\Delta\mathbf{p}^* = \arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})\right]^2 , \tag{1.7}$$

where ∇I is the derivation of the image function, i.e. the row-wise oriented[1] image gradient

$$\nabla I = \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right] \tag{1.8}$$

evaluated at $\mathbf{W}(\mathbf{x};\mathbf{p})$ and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the Jacobian of the warp

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} \partial W_x/\partial p_1 & \partial W_x/\partial p_2 & \dots \\ \partial W_y/\partial p_1 & \partial W_y/\partial p_2 & \dots \end{bmatrix} . \tag{1.9}$$

For the translation warp (1.5)

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \tag{1.10}$$

The problem (1.7) *is* linear in $\Delta\mathbf{p}$ and we can find a close form solution. We are looking for an extreme of (1.7) hence, we derive w.r.t. the $\Delta\mathbf{p}$ and get:

$$2 \sum_{\mathbf{x}} \left[\nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^\top \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})\right] \tag{1.11}$$

[1] Vectors are usually column-wise oriented, here we use row vector in order to simplify notation.

We set equality to zero yields and multiply the elements in brackets, the multiplicative "2" is not important

$$0 = \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \Delta\mathbf{p} - \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} T(\mathbf{x})$$

separating $\Delta\mathbf{p}$ yields:

$$\Delta\mathbf{p} = \mathtt{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))] \tag{1.12}$$

where $\mathtt{H}$ is (Gauss-Newton) approximation of *Hessian matrix*. Is it insightful to see that the element $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ is an error image, difference between the template and warped image.

$$\mathtt{H} = \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]. \tag{1.13}$$

In order to make the equation more clear we consider the warp to be pure 2D translation. Inserting (1.8) and (1.10) into (1.13) yields:

$$\begin{aligned}
\mathtt{H} &= \textstyle\sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \\
&= \textstyle\sum_{\mathbf{x}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial x}] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \textstyle\sum_{\mathbf{x}} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}
\end{aligned}$$

Putting everything into (1.12) reveals

$$\Delta\mathbf{p} = \left(\sum_{\mathbf{x}} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}\right)^{-1} \sum_{x} \begin{bmatrix} \Delta I(\mathbf{x};\mathbf{p}) \frac{\partial I}{\partial x} \\ \Delta I(\mathbf{x};\mathbf{p}) \frac{\partial I}{\partial y} \end{bmatrix} \tag{1.14}$$

where $\Delta I(\mathbf{x};\mathbf{p})$ denote the intensity differences between the template and warped image. Update equation (1.14) illuminates the process in several aspects. Clearly, if there is no difference between warped image and the template the update is zero. The higher gradients in the image the lower is the inverse Hessian hence, the smaller update. And, perhaps most importantly, no gradients in the image, just think about a textureless wall, means very unstable update. If the Hessian (1.13) is close to a singular matrix the alignment gets unstable. This texturedness condition has been pointed out in [2] and, probably independently, in [7].

Assume, we have some initial estimate of $\mathbf{p}$, it could be also $\mathbf{p} = \mathbf{0}$. The iterative search for the best alignment can be summarized as follows:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$
2. Warp the gradient ∇I with $\mathbf{W}(\mathbf{x};\mathbf{p})$
3. Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x};\mathbf{p})$ and compute the steepest descent image $\nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
4. Compute the Hessian $\mathtt{H} = \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$
5. Compute $\Delta \mathbf{p} = \mathtt{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I \ \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\top} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
6. Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$

until $\|\Delta \mathbf{p}\| \leq \epsilon$

2 Color model for particle filtering

3 Mean-shift tracking

References

[1] Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[2] C. Harris and M. Stephen. A combined corner and edge detection. In M. M. Matthews, editor, *Proceedings of the 4th ALVEY vision conference*, pages 147–151, University of Manchaster, England, September 1988. on-line copies available on the web.

[3] J.P. Lewis. Fast template matching. In *Vision Interfaces*, pages 120–123, 1995. Extended version published on-line as "Fast Normalized Cross-Correlation" at http://www.idiom.com/~zilla/Work/nvisionInterface/nip.pdf.

[4] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Conference on Artificial Intelligence*, pages 674–679, August 1981.

[5] K Mikolajczyk, T Tuytelaars, C Schmid, A Zisserman, J Matas, F Schaffalitzky, T Kadir, and L van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(7):43 – 72, November 2005.

[6] Tomáš Petříček and Tomáš Svoboda. Matching by normalized cross-correlation—reimplementation, comparison to invariant features. Research Report CTU–CMP–2010–09, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, July 2010.

[7] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[8] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.