Pattern Recognition, 25th DAGM Symposium, Magdeburg, Germany, September 2003, pages 273-280

Illumination Insensitive Template Matching with Hyperplanes

Christoph Gräßl*, Timo Zinßer*, and Heinrich Niemann

Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg, Martensstraße 3, 91058 Erlangen, Germany
{graessl, zinsser}@informatik.uni-erlangen.de

Abstract. Data-driven object tracking is very important for many vision based applications, because it does not require any previous knowledge about the object to be tracked. In the literature, template matching techniques have successfully been used to solve this task. One promising descendant of these techniques is the hyperplane approach, which is both fast and robust. Unfortunately, like other template matching algorithms, it is inherently sensitive to illumination changes. In this paper, we describe three methods that considerably improve the illumination insensitivity of the hyperplane approach, while retaining the capability of real-time tracking. Experiments conducted on real image sequences prove the efficiency of our enhancements.

1 Introduction

In recent years, visual tracking has emerged as an important component of vision-based systems. It is used in many different application areas like medical imaging [1] and video surveillance [2]. The main purpose of visual tracking is to compute the position of a target object in each image of an image sequence. Additionally, it might be interesting to recover the orientation of the object.

The main problem of visual tracking is that the appearance of an object can change dramatically in the 2-D image sequence. This is not only caused by object motion in conjunction with projective geometry, but also by occlusions, highlight effects and changes in illumination. One way to overcome these difficulties is to use model-based tracking algorithms, which require a priori knowledge about the objects. For example, the approach presented in [3] applies lightfield models and a particle filter for estimating the pose of an object.

One shortcoming of model-based tracking approaches is that they cannot be used when dealing with unrecognized or unknown objects. In this case, data-driven tracking, for example with template matching, is the only viable alternative. The template matching algorithm proposed by Hager and Belhumeur in [4] approximates the relation between variations in intensities and variations in pose by computing the Jacobian matrix of the initial template. Recently, Jurie

* This work was funded by the European Commission 5th IST Programme - Project VAMPIRE. Only the authors are responsible for the content.

and Dhome [5] have improved the basin of convergence of Hager's algorithm by replacing the Jacobian approximation with a hyperplane approximation.

As both algorithms directly operate on image intensities, they are inherently sensitive to changes in illumination. Belhumeur and Kriegman [6] have shown that the image of an object can be reconstructed under arbitrary lighting conditions if a small number of base images is available. Hager incorporated this method into his algorithm [4], basically transforming it into a model-based algorithm and thus losing the possibility of working with unknown objects.

In this paper, we present and compare three methods for reducing the illumination sensitivity of Jurie's hyperplane tracker without using prior knowledge about the tracked objects. Our first method does not work on the original images, but on edge images created with an adapted Sobel filter. The other two methods estimate linear illumination compensation parameters, either with a least square minimization technique or by computing the mean and variance of the template intensities. Especially the last two methods do not inhibit the real-time capability of the original approach and vastly improve its illumination insensitivity.

Our paper is structured as follows. In the next section, a short review of template matching is given. We present three methods for reducing the illumination sensitivity of the hyperplane tracker in Sect. 3. In the subsequent section, the results of our experimental evaluation with real image sequences are detailed. After a summary of our work, possible future extensions are discussed in Sect. 5.

2 Template Matching for Data-Driven Tracking

Template matching algorithms for data-driven tracking work on a sequence of images, which we represent as vectors of gray-level intensities. Additionally, a *reference template* must be specified in the first image. The reference template is defined by vector $\boldsymbol{r} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^T$, which contains the 2-D coordinates of the template points. The gray-level intensity of a point $\boldsymbol{x}_i = (x_i, y_i)^T$ at time t is given by $f(\boldsymbol{x}, t)$. Consequently, vector $\boldsymbol{f}(\boldsymbol{r}, t)$ contains the intensities of template $\boldsymbol{r}$ at time t.

The transformation of the reference template $\boldsymbol{r}$ at time t can be modeled by $\boldsymbol{r}_t = \boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}(t)\right)$, where vector $\boldsymbol{\mu}(t) = \left(\mu_1(t), \mu_2(t), \ldots, \mu_n(t)\right)^T$ contains the *motion parameters*. Examples of tracking with different motion parametrizations are shown in Fig. 1. Template matching can now be described as computing the motion parameters $\boldsymbol{\mu}(t)$ that minimize the least-square intensity difference between the reference template and the current template:

$$\boldsymbol{\mu}(t) = \operatorname*{argmin}_{\boldsymbol{\mu}} \left\| \boldsymbol{f}\left(\boldsymbol{r}, t_0\right) - \boldsymbol{f}\left(\boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}\right), t\right) \right\|_2 \ . \tag{1}$$

Non-linear minimization in a high-dimensional parameter space involves extremely high computational cost and cannot be performed in real-time [7]. It is more efficient to approximate $\boldsymbol{\mu}$ by a linear system

$$\hat{\boldsymbol{\mu}}(t+1) = \hat{\boldsymbol{\mu}}(t) + \boldsymbol{A}(t+1)\left(\boldsymbol{f}\left(\boldsymbol{r}, t_0\right) - \boldsymbol{f}\left(\boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}(t)\right), t+1\right)\right) \tag{2}$$

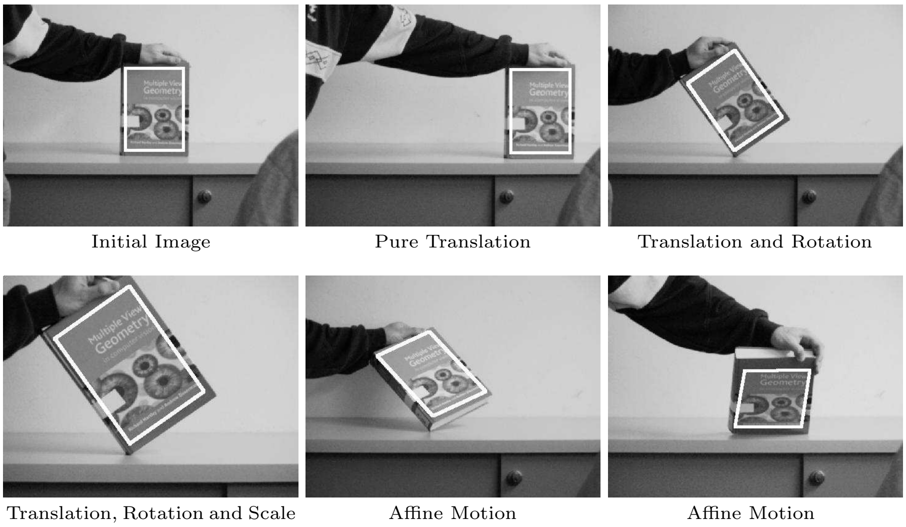


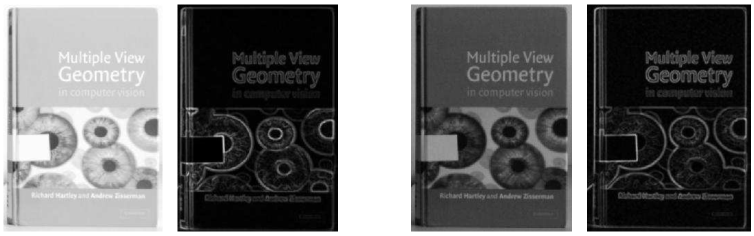
Fig. 1. Some examples of tracking with different motion parametrizations. The reference template was taken from the initial image. The templates are marked by white rectangles.

as presented in [4, 5]. There are two approaches for computing matrix $\boldsymbol{A}(t)$ from equation (2). Hager and Belhumeur [4] propose the use of a Taylor approximation. The hyperplane approach presented in [5] acquires matrix $\boldsymbol{A}$ by a least-square estimation. In the latter approach, matrix $\boldsymbol{A}$ is independent from time t, but has to be computed in a separate training step when the initial image and the reference template are available. As the hyperplane approach has a superior basin of convergence, we will use it throughout the rest of this paper.

3 Illumination Insensitive Template Matching

The template matching algorithm presented in the last section is inherently sensitive to illumination changes, because it directly uses gray-level differences of the templates to compute the motion parameters. These illumination changes are a common problem in real images; they can be caused by automatic exposure adjustments of the camera, changes of light source irradiance, appearance of shadows or movement of the tracked objects [8].

We have investigated two different methods for countering the effects of illumination changes. One possibility is to preprocess the captured images in such a way that most of the adverse lighting effects are eliminated. In the next subsection, we will present such an approach based on edge images, which were created by applying a modified Sobel filter. Another technique is to estimate illumination compensation parameters for the current template, in order to adjust its gray-level values with respect to the reference template. Two algorithms employing this approach are described below.



original image edge image
bright

original image edge image
dark

Fig. 2. Two edge images computed with the adapted Sobel filter.

3.1 Edge Images

In this subsection, we present a method for increasing illumination insensitivity by preprocessing the captured images. This approach has the advantage that there is no need to change the internal structure of the tracking system. Instead, only a preprocessing step has to be performed before passing the images to the tracker.

As it is independent of image brightness, the Sobel edge detection filter is an obvious choice for the preprocessing step. But using the Sobel filter also introduces new problems. Firstly, image noise is amplified by edge detection filters like the Sobel filter. Furthermore, if fast moving objects are temporarily blurred in the captured image, their appearance in the edge image changes considerably.

In order to counteract the problems described above, we suggest to compute the edge images according to

$$\boldsymbol{f}_{\text{edge}} = \text{blur}\left(\text{abs}\left(\text{sobel}_x\left(\text{blur}(\boldsymbol{f})\right)\right) + \text{abs}\left(\text{sobel}_y\left(\text{blur}(\boldsymbol{f})\right)\right)\right) \ , \tag{3}$$

where $\text{blur}(\cdot)$ is a 3×3 box filter operation, $\text{sobel}_x(\cdot)$ and $\text{sobel}_y(\cdot)$ are 3×3 horizontal and vertical Sobel filter operations and $\text{abs}(\cdot)$ computes the absolute values of the input image intensities. The inner blurring operation reduces the noise in the captured image. After applying both Sobel filters, the absolute values of the intensities are computed and the resulting edge images are combined. This operation ensures that similar images are obtained even if the tracked object rotates in the image plane. At last, the edge image is blurred in order to smooth the edges, thus making the final image more suitable for the hyperplane tracker. The input images of our adapted Sobel filter consist of 8 bit unsigned integer values. After internally computing with larger data types, the final values are again saturated to this range.

Figure 2 shows two images captured in varying illumination conditions. Although the brightness of the original images is clearly differing, the computed edge images look very similar. However, in areas where the intensities of the original images are saturated, edges may appear weaker in the corresponding edge image. This example clearly demonstrates that the presented approach can compensate for changes of brightness, but not for changes of contrast.

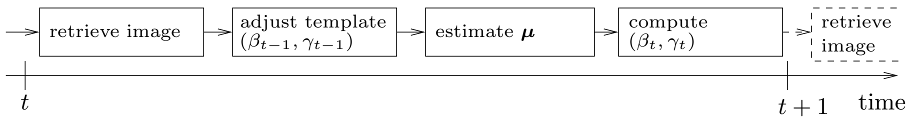


Fig. 3. One iteration cycle for the illumination insensitive hyperplane tracker using illumination compensation parameters

3.2 Intensity Difference Minimization

The influence of illumination changes on the gray-level values of an image is generally of non-linear nature. Nevertheless, previous work has shown that approximating these changes by a linear model is sufficiently accurate for our purposes [8]. Using the linear model

$$f_{\text{new}}(\boldsymbol{x}) = \beta f(\boldsymbol{x}) + \gamma \quad \forall \boldsymbol{x} \in \boldsymbol{r} \tag{4}$$

with illumination compensation parameters β and γ, we can represent variations of contrast and brightness.

When corresponding points of the initial template $\boldsymbol{f}(\boldsymbol{r}, t_0)$ and the current template $\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t)), t)$ are given, the illumination compensation parameters are a solution to the least-squares minimization problem

$$(\beta_t, \gamma_t) = \underset{(\beta,\gamma)}{\operatorname{argmin}} \sum_{\boldsymbol{x} \in \boldsymbol{r}} \left[\beta\, f(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\mu}(t)), t) + \gamma - f(\boldsymbol{x}, t_0) \right]^2 \ . \tag{5}$$

For brevity, we replace $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\mu}(t))$ with $\tilde{\boldsymbol{x}}(t)$. Differentiating equation (5) with respect to the motion compensation parameters yields the linear system

$$\left[\sum_{\boldsymbol{x} \in \boldsymbol{r}} \begin{pmatrix} f^2(\tilde{\boldsymbol{x}}(t), t) & f(\tilde{\boldsymbol{x}}(t), t) \\ f(\tilde{\boldsymbol{x}}(t), t) & 1 \end{pmatrix} \right] \begin{pmatrix} \beta_t \\ \gamma_t \end{pmatrix} = \sum_{\boldsymbol{x} \in \boldsymbol{r}} \begin{pmatrix} f(\boldsymbol{x}, t_0) f(\tilde{\boldsymbol{x}}(t), t) \\ f(\boldsymbol{x}, t_0) \end{pmatrix} \ . \tag{6}$$

When using equation (6) directly, the matrix on the left has to be computed at every time step. To avoid this time consuming operation, we swap the reference template with the current template and thus obtain the motion compensation parameters $\tilde{\beta}$ and $\tilde{\gamma}$ for adapting the reference template to the current template. Consequently, the matrix in equation (6) has to be computed only once. As we still want to adapt the current template, we revert to the original illumination compensation parameters $\beta = 1/\tilde{\beta}$ and $\gamma = \tilde{\gamma}/\tilde{\beta}$.

One iteration cycle for the illumination insensitive hyperplane tracker is shown in Fig. 3. During the initialization, the illumination compensation parameters are set to $\beta_{t_0} = 1$ and $\gamma_{t_0} = 0$.

3.3 Intensity Distribution Normalization

Another approach for illumination compensation is presented in [9]. All templates are normalized by subtracting the mean value and dividing by the standard deviation of their intensities:

$$\boldsymbol{f}_{norm}(\boldsymbol{r}) = \frac{\boldsymbol{f}(\boldsymbol{r}) - m}{\sigma}, \quad m = \frac{1}{N} \sum_{\boldsymbol{x} \in \boldsymbol{r}} f(\boldsymbol{x}), \quad \sigma^2 = \frac{1}{N} \sum_{\boldsymbol{x} \in \boldsymbol{r}} (f(\boldsymbol{x}) - m)^2 \ . \tag{7}$$

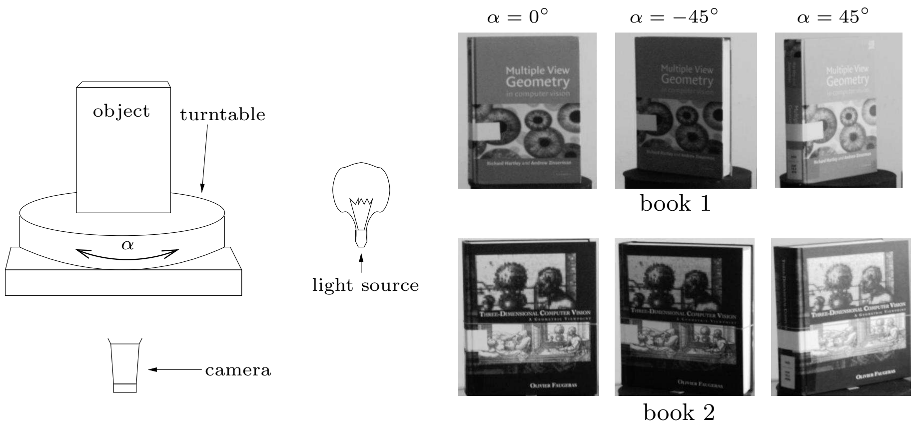


Fig. 4. The setup of the experiments is shown on the left. The camera takes images from two different books, which are placed on a turntable and illuminated from the right side. Three images for each book from different turntable positions are shown on the right to demonstrate the correlation of turntable position and illumination.

In this approach, adapting the current template to the reference template can also be written with linear illumination compensation parameters

$$\beta_t = \frac{\sigma_{t_0}}{\sigma_t} \quad \text{and} \quad \gamma_t = \frac{\sigma_{t_0}}{\sigma_t}\, m_t - m_{t_0} \ . \tag{8}$$

These parameters are applied exactly as in the previous subsection. In contrast to the intensity difference minimization approach, where the intensities of corresponding points are analyzed, the intensity distribution normalization approach considers the distribution of intensities in the templates. Consequently, we expect it to be more robust when the motion parameter estimation is slightly inaccurate, as the distribution of intensities will be affected less than the individual point intensities.

4 Experimental Results

The following experiments with real image sequences demonstrate that our proposed methods significantly reduce the illumination sensitivity of Jurie's hyperplane tracker. Our experimental setup is shown in Fig. 4. This setup is used to generate K image sequences where the turntable moves from 0° to α_k ($k = 1, 2, \ldots, K$) and back to 0°. Turntable angle α directly influences the illumination of the object. The values of α_k range from −90° to 90° in steps of 5°. The image resolution is 640×480 pixels and the turntable speed is 20.8°/sec. The frame rate of the Sony DFW-VL500 firewire camera is 30 frames per second.

The initial position of the book was obtained manually and was used for the reference template. The tracker used a general affine motion parametrization and was initialized with $N = 100$ reference template points, which were selected at

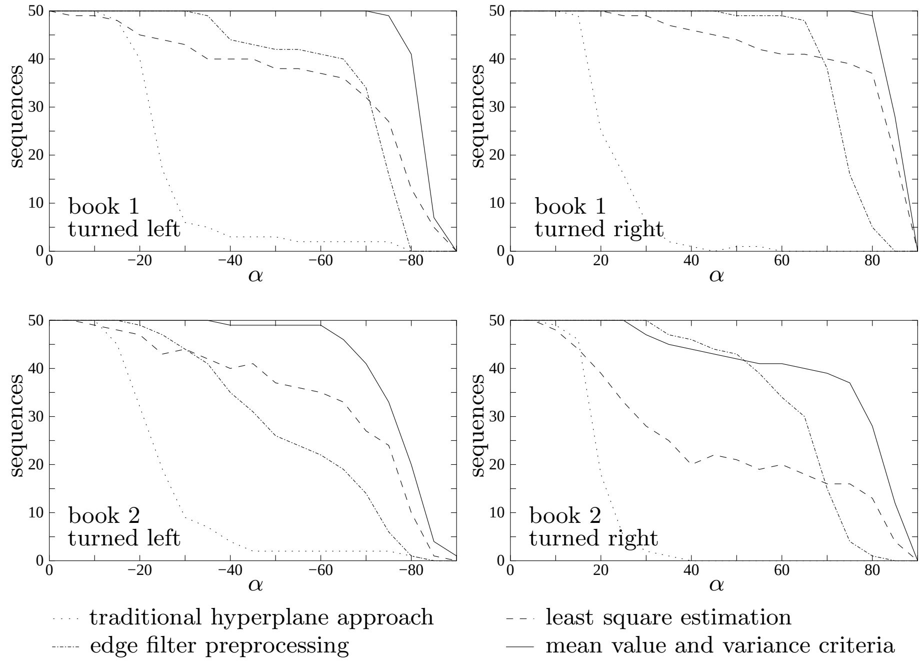


Fig. 5. The four graphs show the number of successfully tracked image sequences in dependency of the turntable angle α. The used objects (*book 1* and *book 2*) have been presented in Fig. 4.

random. An image sequence is successfully tracked if the initial motion parameter is approximately the same as the motion parameter in the last frame. With these image sequences, we tested the traditional hyperplane approach (Sec. 2), the hyperplane approach with the edge filter preprocessing step (Sec. 3.1) and both illumination compensation methods (Sec. 3.2 and 3.3).

The results of the experiments, which are presented in Fig. 5, show that all of our suggested methods significantly improve the tracker's insensitivity to illumination changes. Using intensity distribution normalization clearly leads to the best results. This is what we expected, as small errors in the estimation of the motion parameter have only a small impact on the intensity distribution of the templates, and thus the illumination parameter estimation is very stable. This is not the case for the intensity difference minimization, which generally seems to be more unstable. The edge image method performs quite respectably, but as it does not compensate for contrast changes, it cannot cope with extreme illumination changes.

All presented methods are real-time capable on an Intel Pentium 4 processor system with 2.4 GHz and 1GB of memory. Motion estimation takes about 2.3msec per frame with the traditional hyperplane approach. This value increases to 3.8msec per frame for both illumination compensation approaches. With 10.4msec per frame, the edge image method is slowest.

5 Conclusion and Further Work

We have presented three approaches for reducing the illumination sensitivity of Jurie's hyperplane tracker. The first approach uses an adapted Sobel filter for preprocessing and does not require further changes to the tracking algorithm. The two remaining approaches estimate linear illumination compensation parameters for adjusting the templates to the reference template, either by considering corresponding points or by equalizing the intensity distributions.

Experiments conducted with real image sequences prove the efficiency of the proposed algorithms, as all of them clearly enhance the illumination insensitivity of the hyperplane tracker. Among the presented algorithms, the intensity distribution normalization approach achieves the best results. Additionally, all approaches retain the tracker's real-time capability.

Our further work will concentrate on dealing with partial occlusions and highlights. For this purpose, the use of an iteratively reweighted least-square technique as shown in [4] seems to be very promising. We will also try to find a more intelligent way for choosing the points in the template, as they are currently selected at random. By doing so, we hope to reduce the number of points needed for reliable tracking.

References

1. Kang, D., Kweon, I.: Fast and Stable Snake Algorithm for Medical Images. Pattern Recognition Letters **20** (1999) 507–512
2. Ziliani, F., Reichel, J.: Effective Integration of Object Tracking in a Video Coding Scheme for Multisensor Surveillance Systems. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), Rochester, USA, IEEE Computer Society Press (2002) 521–524
3. Zobel, M., Fritz, M., Scholz, I.: Object Tracking and Pose Estimation Using Light-Field Object Models. In Greiner, G., Niemann, H., Ertl, T., Girod, B., Seidel, H., eds.: Vision, Modeling, and Visualization 2002, Erlangen, Germany, Infix (2002) 371–378
4. Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039
5. Jurie, F., Dhome, M.: Hyperplane approach for template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 996–1000
6. Belhumeur, P., Kriegman, D.: What is the set of images of an object under all possible lighting conditions. In: IEEE Confernce on Computer Vision and Pattern Recognition, IEEE Computer Society Press (1996) 270–277
7. Betke, M., Makris, N.: Fast object recognition in noisy images using simulated annealing. Technical Memo AIM-1510, Massachusetts Institute of Technology, Artificial Intelligence Laboratory (1995)
8. Jin, H., Favaro, P., Soatto, S.: Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination. In: International Conference On Computer Vision, Los Alamitos, IEEE Computer Society (2001) 684–689
9. Fusiello, A., Trucco, E., Tommasini, T., Roberto, V.: Improving Feature Tracking with Robust Statistics. Pattern Analysis and Application **2** (1999) 312–320