6th IEEE Southwest Symposium on Image Analysis and Interpretation, Lake Tahoe, 2004, IEEE Computer Society Press

Efficient Hyperplane Tracking by Intelligent Region Selection

Christoph Gräßl* Timo Zinßer* Heinrich Niemann
Chair for Pattern Recognition, University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
graessl@informatik.uni-erlangen.de

Abstract

The main aim of this work is to improve the accuracy of Jurie's hyperplane tracker for real-time template matching. As the computation time of the initialization of the algorithm depends on the number of points used for estimating the motion of the template, only a subset of points in the tracked template is considered. Traditionally, this subset is determined by random. We present three different methods for selecting points better suited for the hyperplane tracker. We also propose to incorporate color information by working with eigenintensities instead of gray-level intensities, which can greatly improve the estimation accuracy, but only entails a slight increase in computation time. We have carefully evaluated the performance of the proposed methods in experiments with real image sequences.

1. Introduction

In current research, template matching approaches are widely used for tracking objects in video sequences. They are very robust and have the ability to estimate different transformations like translation, rotation, scaling or perspective transformation. Black and Jepson presented an approach based on the eigenspace representation of an image, which can handle partial occlusions [2]. One disadvantage of this approach is the high computational cost, as an eigenspace with at least 50 dimensions has to be applied. Another problem is that the computation of the eigenspace takes very long and has to be done in an offline step. Zobel et al. proposed a different solution, combining a condensation-based approach with lightfield object models [8].

One shortcoming of such model-based approaches is that they cannot be used when dealing with unknown or unrecognized objects. In this case, data-driven tracking is the only viable alternative. The template matching algorithm proposed by Hager and Belhumeur approximates the relation between variations in intensities and variations in pose by computing the Jacobian matrix of the initial template [5]. Recently, Jurie and Dhome have improved the basin of convergence of Hager's algorithm by replacing the Jacobian approximation with a hyperplane approximation [6]. As both algorithms directly operate on image intensities, they are inherently sensitive to changes in illumination. Belhumeur and Kriegman have shown that the image of an object can be reconstructed under arbitrary lighting conditions if a small number of base images is available [1]. Hager incorporated this method into his algorithm, basically transforming it into a model-based algorithm and thus losing the possibility of working with unknown objects. In [4], the robustness of the hyperplane tracker against illumination changes is improved by using a linear illumination model, while preserving the data-driven nature of the algorithm.

One disadvantage of the hyperplane tracker is that a short training has to be performed after an object has been selected. The duration of this training strongly depends on the number of pixels which have to be taken into account. In order to reduce the training time, Jurie and Dhome randomly selected a small number of points in the template [6]. In this paper, we present three different methods for the selection of points which are better suited for hyperplane tracking than selecting by random. We also show how to incorporate color information without changing the basic algorithm by using eigenintensities. In experiments with real images, we demonstrate that the tracking accuracy is improved by our enhancements.

Our paper is structured as follows. In the next section, we shortly summarize the basic principles of the hyperplane tracker. We present three different methods for intelligently selecting points in the template in Sect. 3. Using eigenintensities for incorporating color information is detailed in Sect. 4. The experiments we have conducted are presented in Sect. 5. After a summary of our work, possible future extensions are discussed in Sect. 6.

* This work was partially funded by the European Commission's 5th IST Programme under grant IST-2001-34401 (project VAMPIRE). Only the authors are responsible for the content.

2. Template matching with hyperplanes

Template matching algorithms for data-driven tracking work on a sequence of images. After the specification of a *reference template* in the first image of the sequence, the pose of this template is successively computed in the following images. We represent the images as vectors of gray-level intensities, and define the reference template by a vector $\boldsymbol{r} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$, which contains the 2-D coordinates $\boldsymbol{x}_i = (x_i, y_i)^T$ of the template points. The gray-level intensity of a point $\boldsymbol{x}_i$ at time t is given by $f(\boldsymbol{x}_i, t)$. 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 $\boldsymbol{r}_t = \boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t))$, where vector $\boldsymbol{\mu}(t) = (\mu_1(t), \ldots, \mu_n(t))^T$ contains the *motion parameters*. It is possible to parameterize different kinds of motions in the image plane like pure translation, rotation, scale, affine and perspective deformation. Examples for these motion types are shown in Fig. 1. Consequently, template matching can be described as computing the motion parameters $\boldsymbol{\mu}(t)$ that minimize the least-squares intensity difference between the reference template and the current template.

Because non-linear minimization in a high-dimensional parameter space involves extremely high computational cost, it is more efficient to use a first order approximation

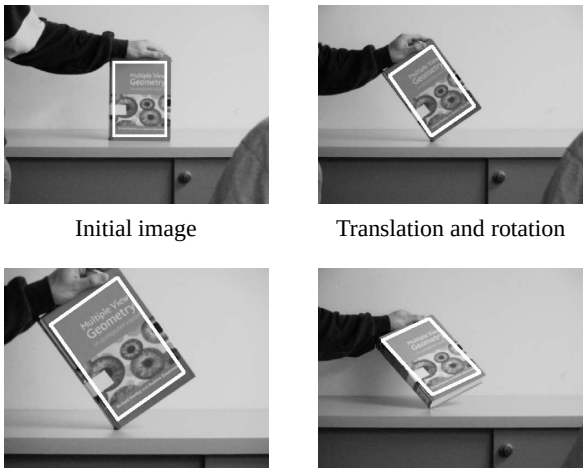
$$\boldsymbol{\mu}(t+1) = \boldsymbol{\mu}(t) + \boldsymbol{A}(t+1)\boldsymbol{e}(t+1) \quad (1)$$

$$\boldsymbol{e}(t+1) = \boldsymbol{f}(\boldsymbol{r}, t_0) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t)), t+1) \quad (2)$$

as presented in [5, 6]. There are two approaches for computing matrix $\boldsymbol{A}(t)$ in Equ. (1). Hager and Belhumeur proposed using a Taylor approximation [5]. For the hyperplane approach presented in [6], matrix $\boldsymbol{A}$ can be made independent of time t. Accordingly, it has to be estimated only once in an initial training stage where a number of random motions are simulated and are used to calculate matrix $\boldsymbol{A}$ by a least-squares estimation. As the hyperplane approach has a superior basin of convergence, we will use it throughout the rest of this paper.

3. Intelligent selection of regions

In the last section, we showed in Equ. (2) that an error vector $\boldsymbol{e}(t)$ is used for the estimation of the motion parameters. Obviously, pixels which lie in regions without perceivable intensity variations contribute almost no information for the motion estimation, because the value of the according component of the error vector contains mainly noise. Examples of some of those regions are illustrated in Fig. 2. As [6] selects the points of region $\boldsymbol{r}$ by random — the pixels are only restricted to lie inside a user-defined area — $\boldsymbol{r}$ may contain a lot of *useless* points.



Initial image | Translation and rotation

Translation, rotation and scale | Affine motion

Figure 1. Three examples for tracking with different motion parameterizations. The reference template was taken from the initial image.

To find out which points of an object should be used, we define $q(\boldsymbol{x})$ as the measure of the quality of point $\boldsymbol{x}$. For estimating $q(\boldsymbol{x})$ only a quadratic region $\boldsymbol{w}(\boldsymbol{x})$ with the center $\boldsymbol{x}$ is taken into account. The size of this square can be chosen according to the strength of the movements during the training.

We analyzed three different criteria for rating potential region points based on their contribution to the motion estimation.

- **Variance criterion (v):** As areas with no or little intensity variation do not contain much information, the variance of intensities in the neighborhood is used for rating the points. Therefore, the quality of a point using the variance criterion is defined as

$$q_v(\boldsymbol{x}) = \text{var}(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots), \quad \boldsymbol{x}_i \in \boldsymbol{w}(\boldsymbol{x}). \quad (3)$$

- **Corner criterion (c):** The Shi-Tomasi-Kanade point tracker [7] uses the condition of its estimation matrix for determining the quality of a feature point. It is conceivable that points which can be tracked well by a point tracker can also be tracked well by a region tracker. Consequently, the first step for calculating the quality of a point using the corner criterion is computing the matrix sum

$$\boldsymbol{Z}(\boldsymbol{x}) = \sum_{\tilde{\boldsymbol{x}} \in \boldsymbol{w}(\boldsymbol{x})} \begin{pmatrix} (f_x(\tilde{\boldsymbol{x}}))^2 & f_x(\tilde{\boldsymbol{x}}) f_y(\tilde{\boldsymbol{x}}) \\ f_x(\tilde{\boldsymbol{x}}) f_y(\tilde{\boldsymbol{x}}) & (f_y(\tilde{\boldsymbol{x}}))^2 \end{pmatrix},$$

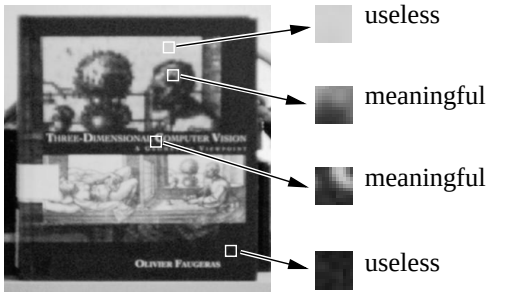


Figure 2. Examples of regions which are well suited or ineligible for template matching with hyperplanes

where $f_x(\tilde{\boldsymbol{x}})$ and $f_y(\tilde{\boldsymbol{x}})$ are the derivatives in x and y direction respectively. The quality is defined as

$$q_c(\boldsymbol{x}) = \min(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x})) \;, \tag{4}$$

where $\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$ are the eigenvalues of $\boldsymbol{Z}(\boldsymbol{x})$.

- **Gradient criterion (g):** Another way of finding areas with adequate intensity variation is to consider the absolute value of the first derivative. This criterion is not well suited for a point tracker, because of the aperture problem. However, in a region tracking approach, the used points are not tracked independently. Particularly, a planar appearance of the object is assumed in the hyperplane approach. Instead of using a Sobel filter, we generate a filter kernel by derivating a Gaussian kernel, which allows for a finer control of the mask size by parameter d_φ:

$$\begin{aligned} G(x,y) &= \exp\left(-\frac{x^2+y^2}{2d_\varphi}\right) \;, \\ G_x(x,y) &= -\frac{x}{d_\varphi} G(x,y) \;, \\ G_y(x,y) &= -\frac{y}{d_\varphi} G(x,y) \;. \end{aligned}$$

The quality of a point using the gradient is calculated by the convolutions

$$q_g(\boldsymbol{x}) = |f(\boldsymbol{w}) \otimes G_x| + |f(\boldsymbol{w}) \otimes G_y| \;. \tag{5}$$

When the v, c, or g-criterion has been computed for every pixel in the template, the points in region $\boldsymbol{r}$ can be determined. Choosing the N pixels with the highest rating is suboptimal, because the resulting region often comprises only some insular parts of the template. Our experiments show that this phenomenon will degrade the performance of the tracker. We propose to use a threshold θ for the feature quality in order to exclude *bad* areas. The region points will then be selected randomly from the points which have not been ruled out.

Figure 3. In the left picture, a standard gray-level image of a blue and a green lotion flask is shown. The right gray-level image was generated using eigenintensities. Here, the objects can be easily distinguished.

4. Including color information by eigenintensities

For performance reasons, a lot of real-time tracking systems only use gray-level images instead of color images. In this case, only one entry per pixel in $\boldsymbol{f}(\boldsymbol{r},t)$ is needed for the region-based hyperplane tracker. This technique can be devastating in scenarios where color information is the only possibility to distinguish regions (an example is shown in Fig. 3). But the incorporation of all three color intensities results in three components per pixel in $\boldsymbol{f}(\boldsymbol{r},t)$, which leads to significantly longer computation time of the approximation matrix $\boldsymbol{A}$ in Equ. (1).

If the performance penalty of using multiple color channels is too large, at least the mapping of the color values to a one-dimensional intensity can be improved. We propose to project the RGB color vector onto the axis of the highest intensity variance of the RGB distribution of the image area which includes the object. The color distribution and the principal axis of the object in Fig. 3 are presented in Fig. 4 The projection can easily be determined by a principal components analysis [3] of the color distribution, where first the mean RGB vector and the covariance matrix

$$\begin{aligned} \bar{\boldsymbol{f}}_C &= \frac{1}{N}\sum_{\boldsymbol{x}} \boldsymbol{f}_C(\boldsymbol{x}) \\ \boldsymbol{\Sigma}_C &= \frac{1}{N-1}\sum_{\boldsymbol{x}} (\boldsymbol{f}_C - \bar{\boldsymbol{f}}_C)(\boldsymbol{f}_C - \bar{\boldsymbol{f}}_C)^T \end{aligned}$$

have to be estimated. Vector $\boldsymbol{f}_C(\boldsymbol{x})$ consists of the three RGB color intensities of point $\boldsymbol{x}$. The dimension reduction of a RGB color intensity vector to a one dimensional eigenintensity value is done by

$$f_E(\boldsymbol{x}) = \boldsymbol{a}\left(\boldsymbol{f}_C(\boldsymbol{x}) - \bar{\boldsymbol{f}}_C\right) \tag{6}$$

where $\boldsymbol{a}$ is the eigenvector of $\boldsymbol{\Sigma}_C$ with the largest eigenvalue. Obviously, the calculation of an eigenintensity requires only a small amount of computational cost, because

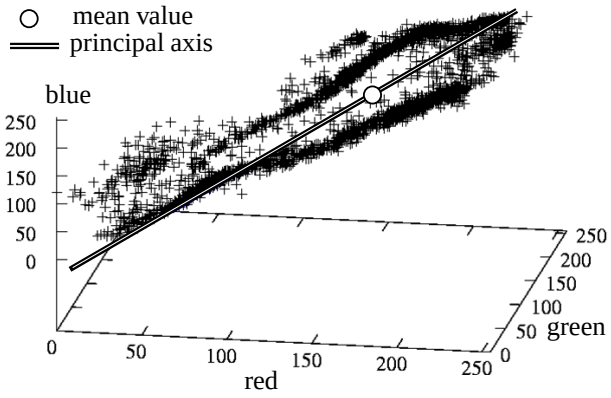


Figure 4. The color distribution of the image in Fig. 3 and its principal axis

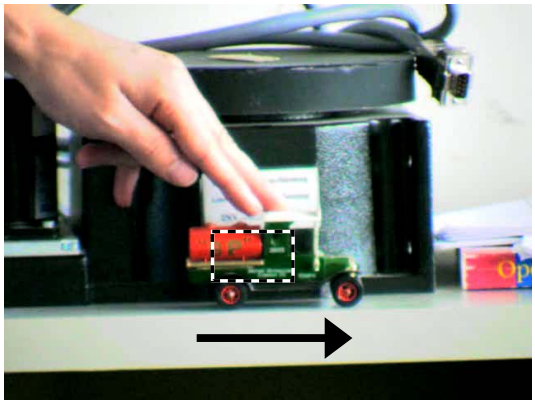


Figure 5. Experimental setup: An object is moved horizontally. The rectangle describes the area from which the points of the regions are taken which should be tracked. The environment is very natural with cluttered background

it takes three subtractions and three multiplications per pixel and only the eigenintensities of the points in $\boldsymbol{r}_t$ have to be computed. Another advantage is that the internal structure of the tracking system does not have to be changed. The effect of our technique is illustrated in Fig. 3, where a color image has been converted to a gray-level (left) and an eigenintensity image (right). Obviously, the bottles can be distinguished easily in the right image.

5. Experimental results

The following experiments with real image sequences demonstrate that our proposed methods significantly increase the approximation accuracy of Jurie's hyperplane tracker. Our experimental setup is shown in Fig. 5, where the object is moved by seven pixels horizontally. The acquired image sequence contains about 200 frames, has a resolution of 640×480 pixels and was captured with a Sony DFW-VL500 firewire camera. In order to retrieve a *ground truth* value $\mu_x^*(t)$ for the motion parameters, the hyperplane tracker has been especially configured for extremely high accuracy. We used a region consisting of 250 points, five hierarchy levels (more information about hierarchy levels can be found in [6]) and a motion parameterization which only estimates translation. For the evaluation of our proposed enhancements, a tracker has been configured for much lower accuracy (region of 100 points, affine motion parameterization, one hierarchy level, and 7 pixel translation). For every frame t in the image sequence, this tracker is newly initialized using the first image and has to estimate the current position which is denoted as $\hat{\mu}_x(t)$. The quality of a tracker can then be expressed by the mean square error

$$\epsilon = \frac{1}{N_S} \sum_{t=1}^{N_S} \left(\mu_x^*(t) - \hat{\mu}_x(t)\right)^2 \ ,$$

where N_S is the number of images in one sequence

First, we present results on experiments of tracking on eigenintensity images. We have done experiments on different objects, but due to lack of space and the fact that the results are very similar, we present the evaluation of one object only. Altogether, the object has been tracked 50 times with different initializations of the region using the traditional approach and using eigenintensities. For every tracked sequence the mean square error ϵ has been calculated, ordered ascendingly and plotted in the graph of Fig. 6. It is clearly visible that using eigenintensities leads to a much higher estimation accuracy of the hyperplane tracker. At this point it should be mentioned that noise effects can decrease the efficiency of the eigenintensities.

For comparison of the different criteria for selecting points of region $\boldsymbol{r}$, a suitable threshold θ is needed for every method. Therefore we used the same experimental framework as presented at the experiments with eigenintensities to test different thresholds. Values of $\theta_v = 144$, $\theta_c = 3$ and $\theta_g = 70$ have proven to be well suited for our purpose. We tested the v-, c- and g-criterion on various objects. The results are presented in Fig. 7, where the image sequence has been tracked 200 times for each method. It is clearly visible that the proposed methods improve the accuracy of the hyperplane tracker compared to the traditionally method, where points are selected by random. In experiments with other objects, we discovered that the corner cri-

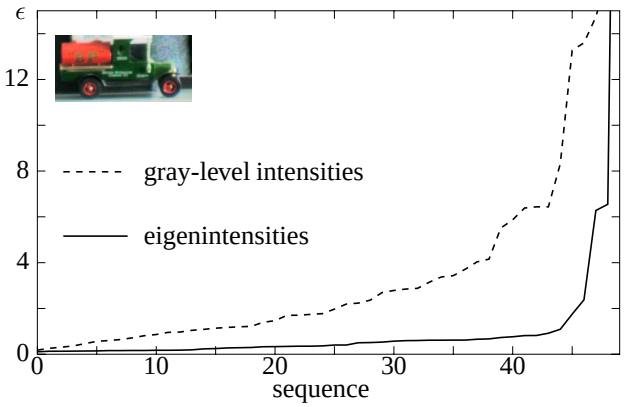


Figure 6. Comparison of tracking on eigenintensities and gray-level intensities. The mean square error is clearly higher using gray-level intensities.

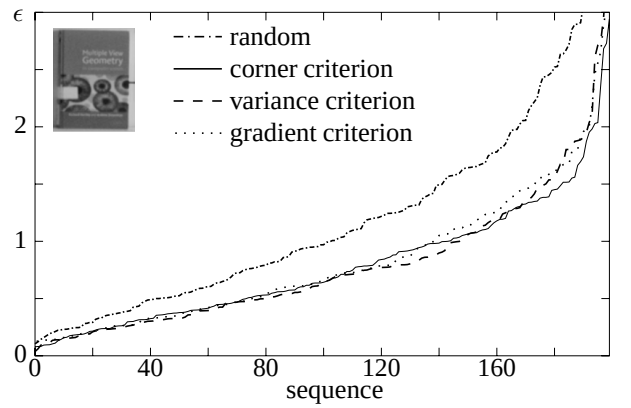


Figure 7. Comparison of the v-, g- and c-criterion for selecting points of the region. All three proposed methods beat the traditional random selection technique.

terion sometimes leads to worse results than the random criterion, especially if the object is strongly textured.

6. Conclusion and outlook

We presented three approaches for enhancing the estimation accuracy of Jurie's hyperplane tracker by a new method for selecting suitable points. Consequently, areas of high variance, areas with large gradients, and areas with corners were used for point selection. In quantitative experiments with real images, it could be shown that the best results can be achieved using the variance or the gradient criterion, which clearly outperform the traditional random selection technique.

As using gray-level intensities can lead to bad estimation accuracy, we proposed to use eigenintensities. The advantage of this approach is that important color information can be used without a significant increase of the computation time. Furthermore, the internal structure of the hyperplane tracker does not have to be changed. We experimentally verified the benefits of the eigenintensities in comparison to gray-level intensities.

Our further work will concentrate on dealing with partial occlusions and highlights. For this purpose, iteratively reweighing least-squares techniques as shown in [5] seem to be very promising.

References

[1] P. Belhumeur and D. Kriegman. What is the Set of Images of an Object Under All Possible Lighting Conditions. In *IEEE Confernce on Computer Vision and Pattern Recognition*, pages 270–277, San Francisco, USA, 1996. IEEE Computer Society Press.

[2] M. J. Black and A. D. Jepson. Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View-based Representation. In B. F. Buxton and R. Cipolla, editors, *Computer Vision - ECCV'96, 4th European Conference on Computer Vision*, pages 329–342, Cambridge, UK, 1996. Springer.

[3] D. Forsyth and J. Ponce. *Computer Vision - A Modern Approach*. Prentice Hall, Upper Saddle River, USA, 2002.

[4] C. Gräßl, T. Zinßer, and H. Niemann. Illumination Insensitive Template Matching with Hyperplanes. In *Pattern Recognition, 25th DAGM Symposium*, pages 273–280, Magdeburg, 2003. Springer.

[5] G. Hager and P. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[6] F. Jurie and M. Dhome. Hyperplane Approach for Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, 2002.

[7] C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[8] M. Zobel, M. Fritz, and I. Scholz. Object Tracking and Pose Estimation Using Light-Field Object Models. In G. Greiner, H. Niemann, T. Ertl, B. Girod, and H. Seidel, editors, *Vision, Modeling, and Visualization 2002*, pages 371–378, Erlangen, Germany, 2002. Infix.