Real-Time Face Tracking using Discriminator Technique on Standard PC Hardware

Alexander Mojaev
Computer Science Dept.,Computer Architecture,
University of Tuebingen
Sand 1, D - 72076 Tuebingen, Germany
mojaev@informatik.uni-tuebingen.de

Andreas Zell
Computer Science Dept.,Computer Architecture,
University of Tuebingen
Sand 1, D - 72076 Tuebingen, Germany
zell@informatik.uni-tuebingen.de

***Abstract*— We show here that it is possible to track a face or an object in real-time using usual PC hardware, even if the object quickly changes its rotation and scale. An algorithm based on the discriminator technique (used earlier in analogue signal processing) was developed. Tracking control was realized by a scale factor and roll angle discriminator control loops and convolution based 2D cross-correlation. Experiments revealed that this method enables real-time scale and rotation invariant tracking control of an object or face template in a wide scale range and provides robustness against high frequency camera vibrations, from which cameras on mobile robots suffer.**

I. Introduction

Robots have to process a large amount of data in real-time in order to operate in the environment, therefore it is very important to develop fast algorithms to work with visual information. Visual object tracking is often applied for various applications in robotics such as visual navigation, human following, object grasping and manipulating, gesture recognition and visual surveillance. Tracking algorithms are often used in low-level preprocessing stages of robot vision and need therefore to have a low computational complexity. The purpose of our work was to find an effective method for automatic tracking of objects or structures in real-time using standard PC hardware.

II. Related Work

Many of the tracking techniques reported earlier use colour, contour or geometric templates for an object or face [1]. Isard and Blake [2] describe “condensation” algorithm using “factored sampling” for tracking curves in visual sequences. Another branch of tracking techniques utilizes well-defined object features for the estimation of object displacements. Viola and Jones [6] proposed real-time object detection framework uses an image representation (“integral image”) based on the features collected from learning images. Krüger described in [3] a network based approach (”Gabor Wavelet Network”). The network was optimized with the Levenberg-Marquardt algorithm. During face tracking this optimisation was done for each frame [4].

Unfortunately, almost all iterative optimization and tracking methods have common disadvantages - the running time increases strongly with the number of nodes (tracked features resp. hypotheses), that problem in addition to a convergence problem puts limitations on the usage of such approaches. Another problem of visual tracking is to deal with a wide range of scaling and other deformations of the tracked object.

III. Discriminator based tracking

The key idea of this work is to apply a well-known discriminator technique (e.g. used in oscillator phase/frequency control circuits) to estimate the scale factor of the tracked object in a video sequence in relation to a stored object image (template) [5]. Using this estimation method the complexity of a matching algorithm can be dramatically reduced. A new technique was developed, which combines fast cross correlation with scale and rotation discriminator control. In the following sections we describe two implemented tracking methods working with a fixed template image: a scale invariant tracking, and a combined tracking by scale factor and roll angle estimation.

A. Moving search frame

Both a template image $\hat{I}$ size and a search frame I size are selected to 64x64 pixels (independent of the input image size). The use of square frames allows to drastically reduce the computional complexity of the cross correlation calculation utilising a 2D convolution technique based on the 2D Fast Fourier Transform (FFT). The cross correlation is simply convolution with normalized images $I, \hat{I}$ and mirrored second image:

$$R(I,\hat{I}) = I \otimes \hat{I}^{*} = \mathrm{FFT}^{-1}\left(\mathrm{FFT}\,(I) \times \mathrm{FFT}(\hat{I}^{*})\right)\ , \quad (1)$$

where $\otimes$ denotes the convolution operator, $\times$ denotes the complex multiplication. The search frame moves during tracking on the original input image corresponding to the estimated $\tilde{x}$ and $\tilde{y}$ translation parameters, which denote the center of the frame. Thus the maximum translation displacement in the x or y direction of the object between two neighboring frames is limited by the search frame size (± 32 pixels).

B. Image preprocessing

The image in the search frame and the template image are normalized to minimize the contrast/brightness influ-

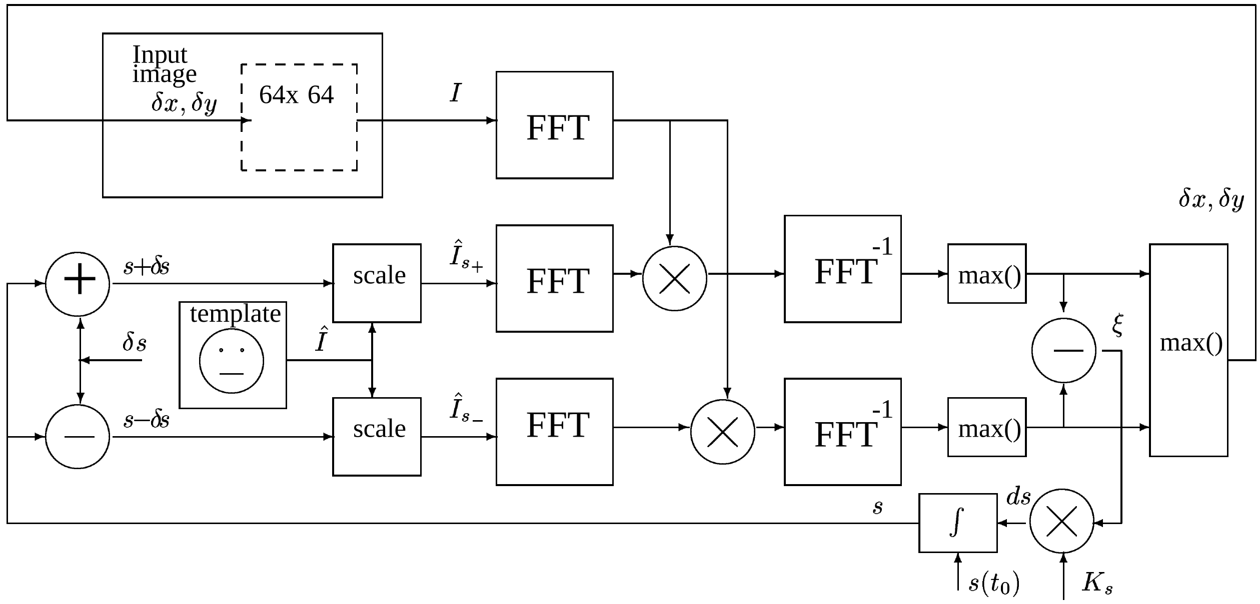

Fig. 1. Tracking by the scale factor discriminator loop

ence and to eliminate the DC component as follows:

$$I(x,y) = \frac{I_0(x,y) - \overline{I}_0}{\max(I_0) - \min(I_0)}\ , \tag{2}$$

where $\overline{I}_0$ is the mean of the image intensity.

C. Template generation

A template used for matching is a 64x64 image stored at the beginning of the tracking process. For template selection a motion detection or color based object detection algorithm can be utilized, which delivers a rough position of the object.

IV. Scale estimation discriminator

To estimate the current scale factor $s(t)$ of the object in the input sequence in comparison with the stored template we use a two channel discriminator loop (Fig. 1).

The matching images $\hat{I}_{s+}$, $\hat{I}_{s-}$ are obtained from the stored template applying a simple scaling operator T_{scale} with different symmetric scale factors $s(t) \pm \delta s$:

$$\hat{I}_{s_+}(t) = T_{\text{scale}}(s(t) + \delta s)\hat{I}, \quad \hat{I}_{s_-}(t) = T_{\text{scale}}(s(t) - \delta s)\hat{I} \tag{3}$$

For both channels the cross correlation functions with the input image are calculated using (1). The peak values of the global maxima of the correlation results are used as discriminator inputs. The output of the discriminator

$$\xi_s(t) = \max(I(t) \otimes \hat{I}^*_{s_+}(t)) - \max(I \otimes \hat{I}^*_{s_-}(t)) \tag{4}$$

is the estimation of the scale factor deviation. The gained deviation (with the gain factor $1/K_s$) is integrated and results in the new scale factor value $s(t)$, which is used for next frame analysis and closes the control loop:

$$s(t) = -\frac{1}{K_s}\int_{t_0}^{t} \xi_s(t)dt \tag{5}$$

or in iterarive form

$$s(t+\Delta t) = s(t) - \frac{1}{K_s}\xi_s(t), \tag{6}$$

where K_s is the discriminator slope. New coordinates of the search frame are obtained using the coordinates of the position of the global maximum $\delta x, \delta y$ in the convolution image with the maximal magnitude (comparing outputs of both channels). These values denote the relative translation vector of the search frame.

We have experimentally calculated the discriminator characteristic using a “face” and a “can” templates for different δs. The results are shown in Fig. 2, top. Unfortunately, it is hardly possible to determine theoretically parameters of the discriminator characteristic because it strongly depends on the object features. Some experimentally determined parameters are collected in tables I andII.

δs	linear deviation limit	discriminator slope K_S
0.015	0.02	0.35
0.03	0.05	0.3
0.05	0.1	0.25
0.09	0.08	0.225

TABLE I
Scale discriminator parameters for the “face” image

Note: except for the output magnitude there is no essential difference in the characteristic curve using completely different images. To maximize the linear deviation limit (region of input value with nearly linear output) and slope parameters, a value $\delta s = 0.03 - 0.05$ is used to achieve a smooth robust object tracking for various object images.

To determine the relative translational displacement the positive peak coordinates $(\delta x, \delta y)$ of the sum $R(I, \hat{I}_1) +$

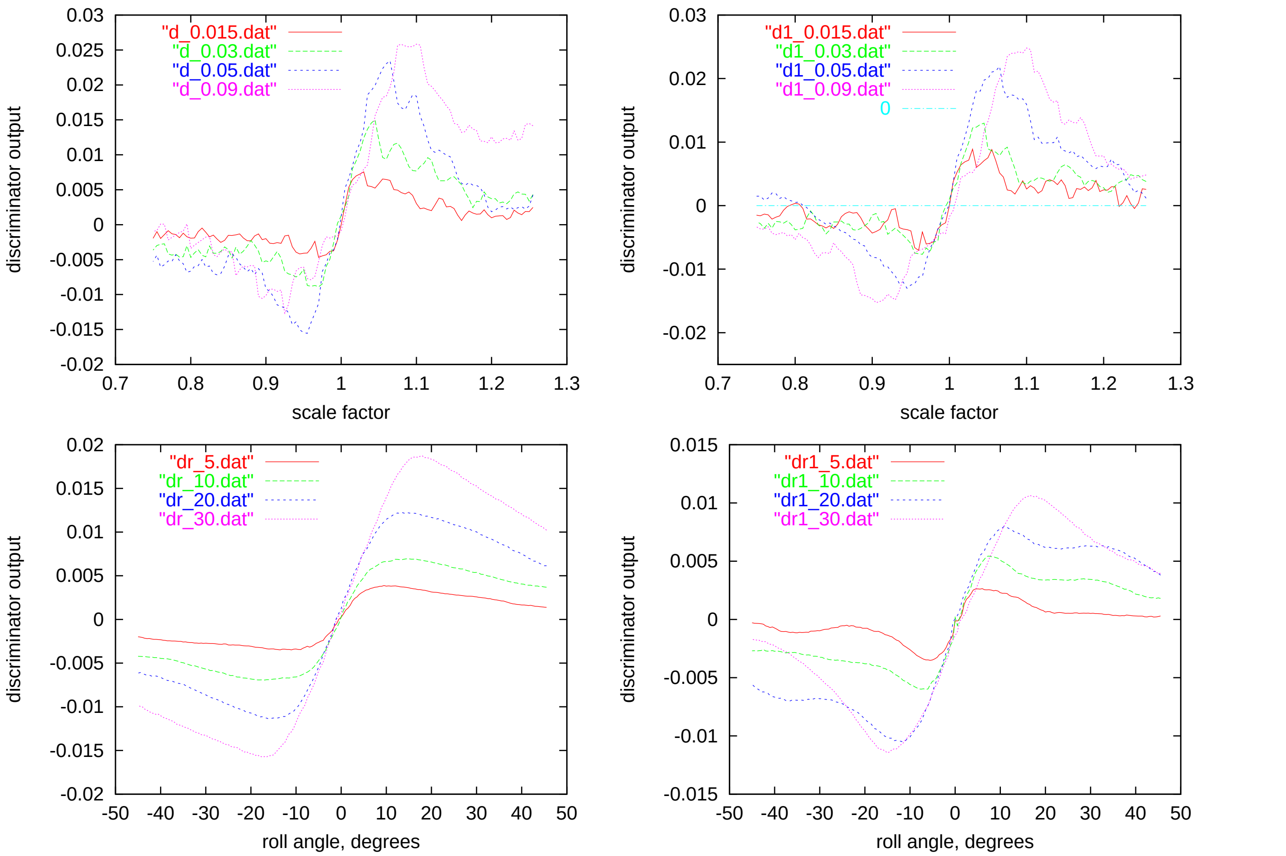

Fig. 2. Scale (top) and roll angle (bottom) discriminator characteristics for different images (left: “face”, right: “can”) with various δs: $\delta s = 0.015$, $\delta s = 0.03$, $\delta s = 0.05$ and $\delta s = 0.09$ and $\delta\theta$: $\delta\theta$=5, $\delta\theta$=10, $\delta\theta$=20 and $\delta\theta$=30 degrees

δs	linear deviation limit	discriminator slope K_S
0.015	0.03	0.37
0.03	0.04	0.37
0.05	0.08	0.31
0.09	0.08	0.2

TABLE II
SCALE ISCRIMINATOR PARAMETERS FOR THE “CAN” IMAGE

$R(I, \hat{I}_2)$ can be used. However, the larger the channel scale factor divergence δs is, the less exactly we candetermine the translation vector $(\delta x, \delta y)$, because of the higher cross-correlation error.

V. Roll Angle Estimation Discriminator

The described model based tracking is robust against moderate object deformations and small rotations about ± 10 degrees. However, it is possible to add an additional roll angle estimation loop using the template image $\hat{I}(t)$ scaled by the current $s(t)$ and the input search image $I(t)$ of the input video sequence. Similar to the estimation of the scale parameter, we implement a two channel rotation discriminator. First we generate two modified template

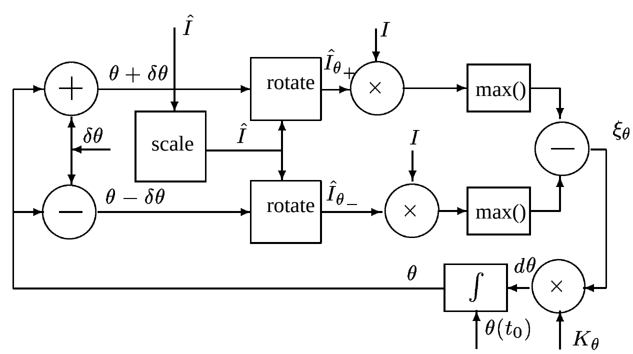

Fig. 3. Roll angle discriminator loop

images with a symmetric rotation displacement:

$$\hat{I}_{\theta_+}(t) = T_{\text{rot}}(\theta(t) + \delta\theta)\hat{I}, \quad \hat{I}_{\theta_-}(t) = T_{\text{rot}}(\theta(t) - \delta\theta)\hat{I} \tag{7}$$

where $T_{\text{rot}}(\cdot)$ is a image rotation operator. The output of the discriminator is given by

$$\xi_\theta(t) = \max(I(t) \cdot \hat{I}_{\theta_+}(t)) - \max(I(t) \cdot \hat{I}_{\theta_-}(t)) \tag{8}$$

where “$\cdot$” denotes pixel by pixel multiplikation. The estimated deviation of the current roll angle is compensated by

$$\theta(t + \Delta t) = \theta(t) - \frac{1}{K_\theta}\xi_\theta(t), \tag{9}$$

where K_θ is the roll angle discriminator slope. By integration of the rotation displacement error we can track the object roll angle.

Experimentally calculated roll angle discriminator characteristica using a "face" and a "can" templates for different $\delta\theta$ are shown in Fig. 2, bottom. The corresponding parameters are collected in tab. III and IV. Maximizing

$\delta\theta$	linear dev. limit, [degrees]	discr. slope K_θ, [degrees^{-1}]
5	10	$6 \cdot 10^{-4}$
10	16	$7.5 \cdot 10^{-4}$
20	21	$1 \cdot 11^{-3}$
30	32	$1 \cdot 10^{-3}$

TABLE III

ANGLE DISCRIMINATOR PARAMETERS FOR THE "FACE" IMAGE

$\delta\theta$	linear dev. limit, [degrees]	discr. slope K_θ, [degrees^{-1}]
5	7	$8.3 \cdot 10^{-4}$
10	16	$9 \cdot 10^{-4}$
20	24	$8.5 \cdot 10^{-4}$
30	30	$7 \cdot 10^{-4}$

TABLE IV

ANGLE DISCRIMINATOR PARAMETERS FOR THE "CAN" IMAGE

the linear deviation limit and slope parameters we obtain the following channel angle displacement $\delta s = 10 - 20$.

VI. FILTERING

To increase stability and robustness of tracking against object deformations and varying light conditions we use a simple filtering technique based on the scale factor and position prediction: If no movement takes place, the maximum of the convolution outputs (results of the cross correlation $\max(I \otimes \hat{I}_1)$ and $\max(I \otimes \hat{I}_2)$) is in the center. During the tracking process the position deviation $\delta x, \delta y$ is gaussian distributed with the dispersion σ, which depends linearly on the scaling factor. So by weighting this cross correlation output images using the 2D Gaussian function

$$\tilde{R}(I, \hat{I}) = R(I, \hat{I}) \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right), \qquad (10)$$

where $\sigma = \sigma_0 + Ks$, we can significantly increase the robustness of the tracking. Other standard filtering techniques such as Kalman filters can also be applied.

VII. COMPUTATIONAL COMPLEXITY

The computational complexity of the tracking algorithm is given by

$$C(N) \approx 5(N \log N) \qquad (11)$$

(five FFTs), where N is the number of pixels in the search frame $N = 64 \times 64$.

Fig. 4. Tracking examples using discriminator technique (translation and scale).

VIII. EXPERIMENTAL RESULTS

We tested the proposed tracking technique on various video sequences, obtained with one of the two cameras of our mobile robot "Robin", an RWI B21 robot. The tracking control runs in real-time (>25fps) with a 64x64 search frame in an 168x128 image (although the successive tracking process is independent of the image size) on a Linux PC with a Pentium 1.3GHz processor. The computation time is mostly defined by the convolution calculation (11). Adding a rotation tracking increases these values by 1-2 milliseconds and depending on grabbing and visualisation time the resulting frame rate is about 30-35 fps.

Fig. 4 shows a tracking example (additional videos can be obtained from the author's homepage [7]). Experiments demonstrate high robustness of tracking control against strong high frequency vertical/horizontal camera vibrations and a wide scale changing range of 20%-150%.

Tab. (V) presents the computation time and tracking.

IX. CONCLUSION

In this paper we present a real-time technique for scale and rotation invariant object or face tracking with standard PC hardware. Tracking control is realized by scale factor and roll angle discriminator loops and a convolution based

scale	scale & roll tracking
7.5-9 msec	8.5-13 msec

TABLE V
TRACKING COMPUTATION TIME

2D cross-correlation. At present we can handle translation, scale and orientation. It is also possible to realize multiple object tracking using the described method, but in this case we need separate discriminator loops for each object.

The implemented tracking technique shows the robustness of the system against large scale and rotation variations and extreme camera vibrations, which is very useful in robot vision.

REFERENCES

[1] S. Feyrer and A. Zell, "Detection, Tracking, and Pursuit of Humans with an Autonomous Mobile Robot," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'99)*, pp. 864–869, 1999.

[2] M. Isard and A. Blake", "Condensation – conditional density propagation for visual tracking," in *International Journal of Computer Vision*, 29(1), pp. 5–28, 1998.

[3] V. Krüger and Sommer, G.: Gabor wavelet networks for object representation. in *22. DAGM-Symposium*, Kiel, Germany, pp. 309–316, 2000.

[4] V. Krüger and R. Feris, "Wavelet Subspace Method for Real-time Face Tracking," in *Proc. Pattern Recognition, 23rd DAGM Symposium*, Munich, Germany, 2001.

[5] A. Mojaev and A. Zell, "Real-Time Object and Face Tracking with Gabor Wavelets," in *Proceedings of the IEEE International Conference on Advanced Robotics (ICAR 2003)*, Vol. 2, pp 1178–1183, 2003.

[6] P. Viola and M. Jones, "Robust Real-time Object Detection," in *Int. Journal of Computer Vision*, 2002.

[7] A. Mojaev's Homepage,
"http://www-ra.informatik.uni-tuebingen.de/mitarb/mojaev"