A Comparative Evaluation of Template and Histogram Based 2-d Tracking Algorithms

F. Bajramovic[1], B. Deutsch[2⋆], Ch. Gräßl[2⋆⋆], J. Denzler[1]

[1] Chair for Computer Vision, Friedrich-Schiller-University Jena,
{bajramov,denzler}@informatik.uni-jena.de,
WWW home page: http://www4.informatik.uni-jena.de

[2] Chair for Pattern Recognition, University of Erlangen-Nuremberg,
{deutsch,graessl}@informatik.uni-erlangen.de,
WWW home page: http://www5.informatik.uni-erlangen.de

Abstract In this paper, we compare and evaluate five contemporary, data-driven real-time 2D object tracking methods: the region tracker by Hager et al., the Hyperplane tracker, the CONDENSATION tracker, and the Mean Shift and Trust Region trackers. The first two are classical template based methods, while the latter three are from the more recently proposed class of histogram based trackers. All trackers are evaluated for the task of pure translation tracking, as well as tracking translation plus scaling. For the evaluation, we use a publically available, labeled data set consisting of surveillance videos of humans in public spaces. This data set demonstrates occlusions, changes in object appearance, and scaling.

1 Introduction

Data driven real-time 2D object tracking is a preliminary for many different computer vision tasks, like face and gesture recognition, surveillance tasks, or action recognition. Recently, two promissing classes of 2D data driven tracking methods have been proposed: template, or region based, tracking methods and histogram based methods. The idea of template based tracking is to track a moving object by defining a region of pixels belonging to that object and using local optimization methods to estimate the transformation parameters of that region between two consecutive images. In histogram based methods the idea is to represent an object by a distinctive histogram, for example a color histogram. Tracking is then performed by searching for a similar region in the image whose histogram best matches the object histgram from the first image. In this paper, we present a comparative evaluation of five different object trackers, two region based [1, 2] and three histogram based approaches [3–5]. We test the performance fo each tracker both for pure translation and for translation with scaling. Due to the rotational invariance of the histogram based methods, further motion modals, such as rotation

⋆ This work was partially funded by the German Science Foundation (DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.

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

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

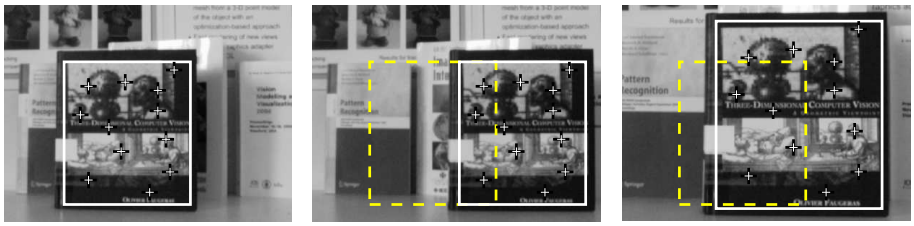

Figure 1. Example of template matching: The left image is the reference image from which the reference template was extracted. The region points are marked by crosses. In the other two images, the reference region has been transformed (center: translation, right: translation and scaling) to match with the reference template. The reference region is marked by the dashed rectangle.

or general affine motion, are not considered. In the evaluation, we focus especially on natural scenes with changing illuminations and partial occlusions basd on a publically available data set [6].

The paper is structured as follows: in Section 2.1 we give a short introduction to the mathematics of both tracking principles. Section 3 deals with the test set and evaluation criteria that we use for our comparative study. The main contribution consists of the evaluation of the different tracking algorithms in Section 4. The paper end with a short conclusion and discussion of the results.

2 Data Driven 2D Object Tracking

In the following two sections, we summarize two different classes of data driven object tracking in the image plane: template matching methods and histogram matching methods.

2.1 Template Matching

One type of algorithm for data driven object tracking is based on template-matching. During an initialization step, the intensity values are extracted from image points belonging to the object. These points form the *reference region* $\boldsymbol{r} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^T$, where $\boldsymbol{x}_i = (x_i, y_i)^T$ is a 2-D point. The gray-level intensity of a point $\boldsymbol{x}$ at time t is given by $f(\boldsymbol{x}, t)$. Consequently, the vector $\boldsymbol{f}(\boldsymbol{r}, t)$ contains the intensities of the entire region $\boldsymbol{r}$ at time t and is called a *template*. The template at the starting time t_0 is denoted as the *reference template*. 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:

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

The function $\boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}\right)$ performs a geometrical transformation of the the region, parameterized by vector $\boldsymbol{\mu}$. Several such transformations can be considered, e.g., [2] use a

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

parameterization which not only deals with translation, rotation, and scaling, but also with affine and projective transformations. In this paper, we restrict ourselves to translation and scale estimation, as illustrated in Fig. 1

The minimization in Eq. 1 is computational expensive if done by a brute-force search. It is more efficient to approximate $\boldsymbol{\mu}$ through a linear system

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

We compare two approaches for computing matrix $\boldsymbol{A}(t)$ from Eq. 2. Jurie and Dhome [2] perform a short training step, where random transformations are simulated in the reference image. Typically, on the order of 1000 transformations are executed and their motion parameters $\tilde{\boldsymbol{\mu}}_i$ and difference vectors $\boldsymbol{f}\left(\boldsymbol{r}, t_0\right) - \boldsymbol{f}\left(\boldsymbol{g}\left(\boldsymbol{r}, \tilde{\boldsymbol{\mu}}_i\right), t_0\right)$ collected. Afterwards, matrix $\boldsymbol{A}$ is derived through a least squares approach. Note that $\boldsymbol{A}$ can be made independent from t in this approach. For details, we refer to the original paper. A more analytical way is proposed by Hager et. al [1], who use a first order Taylor approximation. During initialization, the gradients of the region points used to build a Jacobian matrix. Although $\boldsymbol{A}$ cannot be made independent from t, the transformation can be performed very efficient and the approach has real-time capability.

2.2 Histogram Matching

In histogram based tracking methods, the target is again identified by an image region $\boldsymbol{r}(\boldsymbol{\mu}(t))$, where $\boldsymbol{\mu}(t)$ contains the time variant parameter of the region, also referred to as the state of the region. One simple example for a region $\boldsymbol{r}(\boldsymbol{\mu}(t))$ is a rectangle of fixed dimensions. The state of the region $\boldsymbol{\mu}(t) = (m_x(t), m_y(t))^T$ is the center of that rectangle in pixel coordinates, $m_x(t)$ and $m_y(t)$, for each time step t. With this simple model, translation of a target region can be easily described by estimating $\boldsymbol{\mu}(t)$, i.e. center of gravity of the rectangle, over time. If the size of the region is also included in the state, estimation of the scale is possible.

The information contained within the region is used to model the moving object, but instead of focusing on individual pixels and their values, the distribution of features defined at each pixel is used. The information may consist of the color, the intensity, or certain other features like the gradient. At each time step t and for each state $\boldsymbol{\mu}(t)$, the representation of the moving object consists of a probability density function $p(\boldsymbol{\mu}(t))$ of the chosen features within the region $\boldsymbol{r}(\boldsymbol{\mu}(t))$. In practice, this density function has to be estimated from image data. For performance reasons, a weighted histogram $\boldsymbol{q}(\boldsymbol{\mu}(t)) = (q_1(\boldsymbol{\mu}(t)), q_2(\boldsymbol{\mu}x(t)), \ldots, q_N(\boldsymbol{\mu}(t)))^T$ of N bins is used as a non-parametric estimation of the true density. Each individial bin $q_i(\boldsymbol{\mu}(t))$ of the histogram is computed by

$$q_i(\boldsymbol{\mu}(t)) = C_{\boldsymbol{\mu}(t)} \sum_{\boldsymbol{u} \in \boldsymbol{r}(\boldsymbol{\mu}(t))} L_{\boldsymbol{\mu}(t)}(\boldsymbol{u}) \delta(b_t(\boldsymbol{u}) - i), i = 1, \ldots, N \tag{3}$$

with $L_{\boldsymbol{\mu}(t)}(\boldsymbol{u})$ being a suitable weighting function, $b_t(\boldsymbol{u})$ the function that maps the pixel $\boldsymbol{u}$ to the number j of the bin which the feature at position $\boldsymbol{u}$ falls into ($j \in \{1, \ldots, N\}$), and δ being the Kronecker-Delta function. The value

$$C_{\boldsymbol{\mu}(t)} = \frac{1}{\sum_{\boldsymbol{u} \in \boldsymbol{r}(\boldsymbol{\mu}(t))} L_{\boldsymbol{\mu}(t)}(\boldsymbol{u})} \tag{4}$$

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

Figure 2. Stills from three of the videos used. The solid box marks the hand-labled ground truth. The dashed box is the tracker's current estimate. In the right-most image, the tracker has been distracted by a temporary occlusion from another person, and subsequently lost the real target.

is a normalizing constant. In other words, (3) counts all occurances of pixels that fall into bin i, where the increment within the sum is weighted by $L_{\boldsymbol{\mu}(t)}(\boldsymbol{u})$.

Object tracking can now be defined as an optimization problem. Starting with an initial target region — for example, manually or automatically defined in the first image at $t = t_0$ — an initial histogram $\boldsymbol{q}(\boldsymbol{\mu}(0))$ can be computed. For $t > t_0$, the corresponding region is defined by

$$\boldsymbol{\mu}(t) = \underset{\boldsymbol{\mu}}{\operatorname{argmin}}\, D(\boldsymbol{q}(\boldsymbol{\mu}(0)), \boldsymbol{q}(\boldsymbol{\mu}(t))) \tag{5}$$

with $D(\cdot, \cdot)$ being a suitable distance function defined on histograms. In our work we compare two local optimization techniques, the Mean Shift algorithm [7, 8] and the Trust Region algorithm [4, 9], as well as a global optimization technique using particle filters [5, 10].

3 Test Set and Evaluation Criteria

The experiments were performed on publically available videos from the CAVIAR [6] project. These are surveilance-type videos from a fixed camera, showing human beings performing a variety of actions. The videos come with hand-labeled ground truth information, which allows an independent evaluation of our trackers. The ground truth information describes rectangles surrounding the individual humans in each scene.

Figure 2 shows sample images from three of the used videos. The change in the tracked person's appearance, as well as the heterogeneous background, makes this a relatively difficult problem.

In each experiment, a specific person was to be tracked. The tracking system was given the frame number of the first unoccluded appearence of the person, the ground truth rectangle around the person, and the frame of the person's dissappearance. Each tracker was initialized with this enclosing rectangle. Aside from this initialization, the trackers had no access to the ground truth information.

For each frame, two measurements between the tracked region and the ground truth region were recorded. The first is defined as the fraction of non-overlapping area by the total area of both regions:

$$e_r(A, B) := \frac{|A \setminus B| + |B \setminus A|}{|A| + |B|} \tag{6}$$

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

Table 1. Timing results from the first sequence, in milliseconds. For each tracker, the time taken for initialization, and the average time per frame, are shown for scaling and non-scaling versions.

	Without scaling		With scaling	
	initial	per frame	initial	per frame
Hager & Belhumeur	5	2.33	5	2.87
Hyperplane	528	2.16	536	2.19
Mean Shift	2	1.03	2	2.74
Trust Region	9	4.01	18	8.63
CONDENSATION	11	79.71	11	109.95

where A and B are image regions, represented as sets of image points, and $|\cdot|$ is the cardinal number of a set. Identical regions have a region error of $e_r(A, A) = 0$, while non-overlapping rectangles have a region error of 1. The second measurement, denoted e_c, is the Euclidean distance between both rectangles' centers, measured in pixels.

Twelve experiments were performed on seven videos (some videos were reused, tracking a different person each time).

4 Experimental Results

The following five trackers were compared: The region tracking algorithm of Hager et al.[1], working on a three-level Gaussian image pyramid hierarchy to enlarge the basin of convergence. The Hyperplane tracker, using a 150 point region and initialized with 1000 training perturbation steps. The Mean Shift and Trust Region algorithms, using an Epanechnikov weighting kernel, the Bhattacharyya distance measure and the HSV color histogram feature from [5] for maximum comparability. Finally, the CONDENSATION based color histogram approach from Pérez et al.[5], with 400 particles diffused by a zero-mean Gaussian distribution with a variance of 5 pixels in each dimension. All trackers were tested with pure translation, and with translation and scaling.

All tests were timed on a 2.8 GHz Intel Xeon processor. The methods differ greatly in the times taken for initialization (once per sequence) and tracking (once per frame). Table 1 shows the timing results from the first sequence. Notable points are the long initialization phase of the Hyperplane tracker due to training, and the long per-frame time of the CONDENSATION tracker due to the large number of particles.

The trackers' output was compared to the ground truth with the two evaluation criteria introduced in section 3 (distance between centers e_c, and fraction of region overlap e_r). For each tracker, the errors from all sequences were concatenated and sorted.

Figure 3 shows the measured distance error e_c and the region error e_r for all trackers, both with and without scaling.

Performance varies widely between all tested trackers, showing strengths and weaknesses for each individual method. There appears to be no method which is universally "better" than others.

The structure-based region trackers, Hager and Hyperplane, are potentially very accurate, as can be seen at the left-hand side of each graph, where they display a larger

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

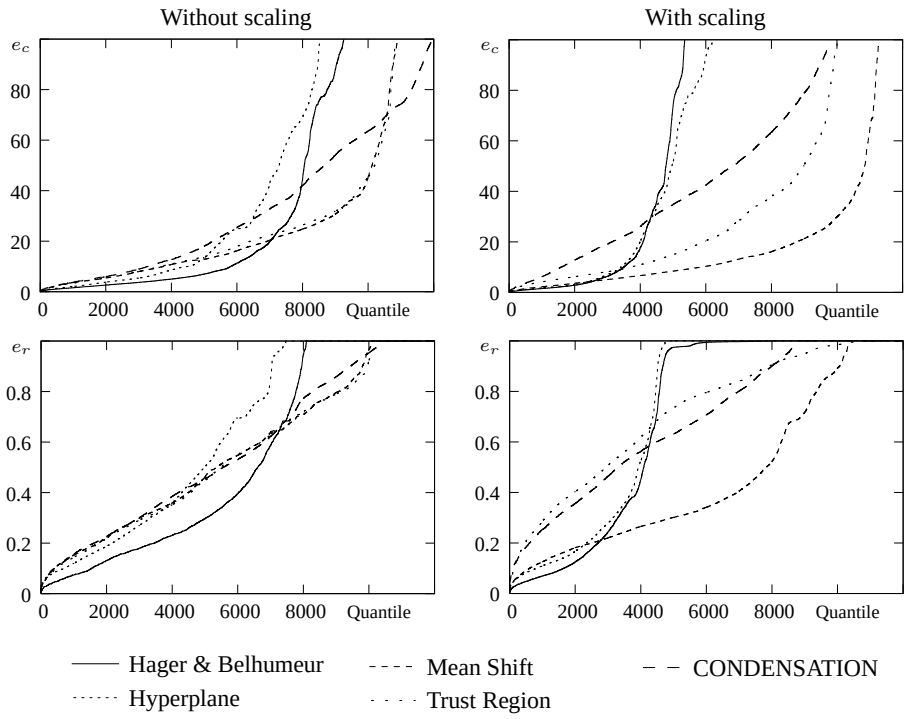

Figure 3. The result graphs. The top row shows the distance error e_c, the bottom row shows the region error e_r. The left-hand column contains the results for trackers without scaling, the right-hand column those with scaling. THe horizontal axis dois not correspond to time, but to sorted aggregation. The vertical axis for e_c has been truncadet to 100 pixels to emphasize the relevant details.

number of frames with low errors. However, both are prone to losing the target quicker, causing their errors to climb faster than the other three methods. Particularly when scaling is also estimated, the additional degree of freedom typically provide additional accuracy, but causes the estimation to diverge sooner. This is the consequence of the strong changes of appearance of the tracked regions in these image sequences.

The CONDENSATION method, for the most part, is not as accurate as the two local optimization methods, Mean Shift and Trust Region. We believe this is partly due to the fact that basic CONDENSATION does not provide intra-frame refinement, and that time constraints necessitate the use of a quickly computable particle evaluation function. However, the strength of the CONDENSATION approach lies in its robustness against local optima: it is capable of reacquiring a lost (or nearly lost) target, which shows in the flatness of the error curves towards the high end.

Figure 4 shows a direct comparison between a locally optimizing structural tracker (Hager) and the globally optimizing histogram based CONDENSATION tracker. It is clearly visible that the Hager tracker provides more accurate results, but cannot reac-

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

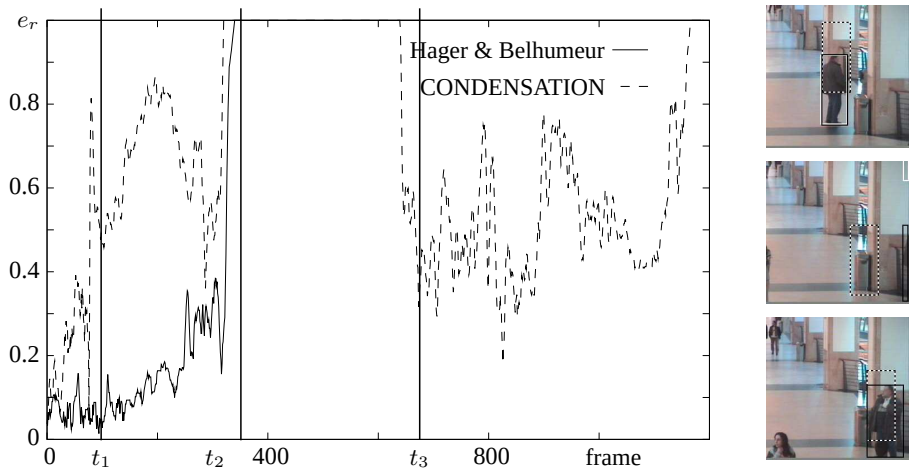

Figure 4. Comparison of the Hager and CONDENSATION trackers using the e_r error measure (cf. Sec. 3). The black rectangle shows the ground truth. The white rectangle is from the Hager tracker, the dashed rectangle from the CONDENSATION tracker. The top, middle and bottom images are from frames t_1, t_2, and t_3 respectively. The tracked person (almost) leaves the camera's field of view in the middle image, and returns near the left image. The Hager tracker is more accurate, but loses the person irretrievably, while the CONDENSATION tracker is able to reacquire the person.

quire a lost target. The CONDENSATION tracker, on the other hand, can continue to track the person after it reappears.

The Mean Shift and Trust Region trackers perform equally well and provide the overall best tracking when scaling is not estimated. When scaling is introduced, however, the Mean Shift algorithm performs noticably better than the Trust Region approach. This is especially visible when comparing the region error e_r (figure 3, bottom right), where the error in the scaling component plays an important role.

Another very interesting thing to note is that tracking translation and scaling, as opposed to tracking translation only, generally did *not* improve the results on these sequences. In fact, the performance of all trackers deteriorated, even when measuring the fraction of region non-overlap (where any changes in target scale will automatically penalize trackers which do not estimate scaling).

For the structure-based trackers, Hager and Hyperplane, the changing appearance of the tracked persons is a strong handicap. The extra degree of freedom opens up more chances to diverge towards local optima, which causes the target to be lost sooner.

The trackers using histogram features, on the other hand, suffer from the fact that the features themselves are typically rather invariant under scaling. Once the scale, and therefore the size of the region, is wrong, small translations of the target can go completely unnoticed.

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.

5 Conclusion

In this paper, we presented a comparative evaluation of five state of the art algorithms for data-driven object tracking, namely Hager's region tracking technique [1], Jurie's Hyperplane approach [2], the probabilistic color histogram tracker by Perez [5], Comaniciu's Mean Shift tracking approach [3], and the Trust Region method introduced by Chen [11]. All of those trackers have the ability to estimate the position and scale of an object in the image in real-time. For the comparison, the CAVIAR video database, which includes ground-truth data, has been employed. The results of our experiments show that, in cases of strong appearance change, the region based methods of [2, 1] tend to lose the object more often than the histogram based methods. On the other side, if the appearance change is weak, the region based methods surpass the other approaches in tracking accuracy. Comparing the the histogram based methods among each other, the Mean Shift approach [3] leads to the best results. The experiments also show that the probabilistic color histogram tracker [5] is not quite as accurate as the other techniques, but is more robust in case of occlusions and appearance changes.

References

1. 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
2. Jurie, F., Dhome, M.: Hyperplane approach for template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 996–1000
3. Comaniciu, D., Meer, P., Ramesh, V.: Kernel-Based Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2004) 564–575
4. Liu, T.L., Chen, H.T.: Real-Time Tracking Using Trust-Region Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 397–402
5. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: 7th European Conference on Computer Vision. Volume 1. (2002) 661–675
6. CAVIAR: EU funded project, IST 2001 37540, URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/ (2004)
7. Comaniciu, D.: Bayesian Kernel Tracking. In: Annual Conference of the German Society for Pattern Recognition. (2002) 438–445
8. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 790–799
9. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. SIAM (2000)
10. Isard, M., Blake, A.: Condensation – Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision **29** (1998) 5–28
11. Chen, H.T., Liu, T.L.: Trust-Region Methods for Real-Time Tracking. In: 8th IEEE International Conference on Computer Vision. Volume 2. (2001) 717–722

W. Kropatsch et al. (Eds.): DAGM 2005, LNCS 3663, pp. 254–261, 2005.