Efficient Combination of Histograms for Real-Time Tracking Using Mean-Shift and Trust-Region Optimization

F. Bajramovic[1], 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,
{graessl}@informatik.uni-erlangen.de,
WWW home page: http://www5.informatik.uni-erlangen.de

Abstract Histogram based real-time object tracking methods, like the Mean-Shift tracker of Comaniciu/Meer or the Trust-Region tracker of Liu/Chen, have been presented recently. The main advantage is that a suited histogram allows for very fast and accurate tracking of a moving object even in the case of partial occlusions and for a moving camera. The problem is which histogram shall be used in which situation. In this paper we extend the framework of histogram based tracking. As a consequence we are able to formulate a tracker that uses a weighted combination of histograms of different features. We compare our approach with two already proposed histogram based trackers for different historgrams on large test sequences availabe to the public. The algorithms run in real-time on standard PC hardware.

1 Introduction

Data driven, real-time object tracking is still an important and in general unsolved problem with respect to robustness in natural scenes. Obviously, for many different, high-level tasks in computer vision, there is the need for tracking a moving object in real-time without having specific knowledge about its 2D or 3D structure. In general, it is necessary in surveillance tasks, action recognition, navigation of autonomous robots, etc. Usually, tracking is initialized based on change detection in the scene. From this moment on, the position of the moving target is identified in each consecutive frame.

Several approaches for 2D data driven tracking have been presented in the past, for example feature based [1], template based [2, 3], and, most recently, histogram based methods [4, 5]. In all cases, tracking, i.e. correspondence between successive frames, is solved by defining and solving an optimization problem. The main difference between the approaches consists in the representation of the object and the way the optimization problem is solved.

⋆ 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.

In this paper, we focus on histogram based methods, where the object to be tracked is identified by a histogram of a priori defined features. One prominent example for a feature is color resulting in a color histogram used to identify the object. We present an extension of histogram based tracking where instead of a single histogram a weighted combination of several different histograms can be used. We refer to this tracker in the following as *combined histogram tracker* (CHT). For tracking, we formulate the optimization problem in a general way, such that the Mean-Shift [6] as well as the Trust-Region [7] optimization can be applied. This allows for a maximum of flexibility for the parameters that are estimated during tracking, for example, translation, rotation, and scaling. We compare the CHT with already presented histogram trackers using only one specific histogram. The results show a significantly better performance of the CHT with respect to accuracy during tracking, and at the same time without loosing its real-time capability.

The paper is structured as follows. In section 2 we introduce histogram based tracking methods together with two already presented local optimization methods, the Mean-Shift and the Trust-Region algorithm. In section 3 we present a rigorous mathematical description for the CHT. We show how the optimization problem can be solved again using the Mean-Shift and the Trust-Region algorithm. Section 4 deals with the experiments. We show results on a large set of labeled image sequences available to the public, which allows quantitative evaluation and comparison. The paper concludes with a discussion and an outlook to future work.

2 Region Based Object Tracking Using Histograms

2.1 Representation and Tracking

In general, the target is identified by an image region $R(\boldsymbol{x}(t))$, where $\boldsymbol{x}(t)$ contains the time variant parameters of the region, also referred to as the state of the region. One simple example for a region $R(\boldsymbol{x}(t))$ is a rectangle of fixed dimensions. The state of the region $\boldsymbol{x}(t) = (m_x(t), m_y(t))^T$ is the center of gravity 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 $\boldsymbol{x}(t)$, i.e. center of gravity of the rectangle, over time. If the size of the region is also included in the state, estimation of scale is possible.

The information contained within the region is used to model the moving object. The information may consists of the color, the gray value, or certain other features, like the gradient. At each time step t and for each state $\boldsymbol{x}(t)$ the representation of the moving object consists of a probability density function $p(\boldsymbol{x}(t))$ of the chosen features within the region $R(\boldsymbol{x}(t))$. In practice, this density function has to be estimated from image data. For performance reasons, a weighted histogram $\boldsymbol{q}(\boldsymbol{x}(t)) = (q_1(\boldsymbol{x}(t)), q_2(\boldsymbol{x}(t)), \ldots, q_N(\boldsymbol{x}(t)))^T$ of N bins $q_i(\boldsymbol{x}(t))$ is used as a non-parametric estimation of the true density, although it is well known that this is not the best choice from a theoretical point of view [8]. Each individual bin $q_i(\boldsymbol{x}(t))$ is computed by

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

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

with $L_{\boldsymbol{x}(t)}(\boldsymbol{u})$ being a suited weighting function introduced below, $b_t(\boldsymbol{u})$ 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 $C_{\boldsymbol{x}(t)} = 1/\sum_{\boldsymbol{u} \in R(\boldsymbol{x}(t))} L_{\boldsymbol{x}(t)}(\boldsymbol{u})$ is a normalizing constant. In other words, (1) counts all occurrences of pixels that fall into bin i, where the increment within the sum is given by the weighting function $L_{\boldsymbol{x}(t)}(\boldsymbol{u})$.

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 $t = 0$ — an initial histogram $\boldsymbol{q}(\boldsymbol{x}(0))$ can be computed. For $t > 0$ the corresponding region is defined by

$$\boldsymbol{x}(t) = \operatorname*{argmin}_{\boldsymbol{x}} D(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x})) \tag{2}$$

with $D(\cdot, \cdot)$ being a suited distance function defined on histograms. In our work we use two local optimization techniques, the Mean-Shift algorithm [4] and the Trust-Region algorithm [5]. In the context of histogram based tracking, also a global optimization using a particle filter can be applied [9].

2.2 Kernel and Distance Functions

There are two open aspects left: the choice of the weighting function $L_{\boldsymbol{x}(t)}(\boldsymbol{u})$ in equation (1) and the distance function $D(\cdot, \cdot)$. The weighting function is typically chosen as a kernel, whose support is exactly the region $R(\boldsymbol{x}(t))$. Different kernel profiles can be used, like the Epanechnikov, the biweight, or the truncated Gauss profile [10].

For the optimization problem in (2) several distance functions on histograms have been proposed, like the Bhattacharya distance, the Kulback-Leibler distance, the Euclidean distance or the scalar product distance. It is worth noting that for the following optimization no metric is necessary. The main restriction on the given distance functions in our work is the following special form

$$D(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x}(t))) = \hat{D}\left(\sum_{n=0}^{N} d(q_n(\boldsymbol{x}(0)), q_n(\boldsymbol{x}(t)))\right) \tag{3}$$

with a monotone, bijective function $\hat{D}$, and a function $d(a, b)$, which is twice differentiable for b. Now, substituting (3) into (2) we get

$$\boldsymbol{x}(t) = \operatorname*{argmax}_{\boldsymbol{x}} \left(-sgn(\hat{D}) \sum_{n=0}^{N} d(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x}))\right) \tag{4}$$

where $sgn(\hat{D}) = 1$ or $sgn(\hat{D}) = -1$ if $\hat{D}$ is monotonly increasing or decreasing, respectively. More details can be found in [10].

2.3 Optimization

This section deals with the optimization of (4) using the Mean Shift algorithm. Hints are given in the end how the optimization can be solved by Trust Region optimization.

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

The main idea is to do a Taylor series expansion of the right hand side of (4). After a couple of computations and simplifications (for details, see [10]) we get

$$\boldsymbol{x}(t) \approx \underset{\boldsymbol{x}}{\operatorname{argmax}} \left(C_0 \sum_{\boldsymbol{u} \in R(\boldsymbol{x})} L_{\boldsymbol{x}}(\boldsymbol{u}) \underbrace{\sum_{n=1}^{N} \delta(b_t(\boldsymbol{u}) - n) \tilde{w}_t(n)}_{\tilde{w}_t(b_t(\boldsymbol{u}))} \right) \tag{5}$$

with the weights

$$\tilde{w}_t(b_t(\boldsymbol{u})) = -sgn(\hat{D}) \frac{\partial d(a,b)}{\partial b} \Big|_{(a,b)=\left(q_{b_t(\boldsymbol{u})}(\boldsymbol{x}(0)), q_{b_t(\boldsymbol{u})}(\boldsymbol{x})\right)} \tag{6}$$

This special reformulation allows us to interpret the weights $\tilde{w}_t(b_t(\boldsymbol{u}))$ as weights on the pixel coordinates $\boldsymbol{u}$. For a certain distance function $D(\cdot, \cdot)$ we need to calculate the corresponding pixel weights. Finally, we can apply the Mean-Shift algorithm for the optimization of (5), since (5) is a weighted kernel density estimation. Due to lack of space, for details, on how the Mean-Shift algorithm is applied, the reader is referred to [11, 10]. Alternatively, the Trust-Region optimization algorithm can be applied. In this case, we need the gradient and Hessian matrix of the right hand side of (4). Both quantities can be derived in closed form [10]. The advantage of the Trust Region method is, that — besides estimation of translation of the target region — also rotation and scale can be integrated in the optimization problem [10].

2.4 Example

Now we give an example for the equations and quantities presented above. Using the Bhattacharyya distance between histograms (as in [4]), defined as

$$D(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x}(t))) = \sqrt{1 - B(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x}(t)))} \tag{7}$$

with

$$B(\boldsymbol{q}(\boldsymbol{x}(0)), \boldsymbol{q}(\boldsymbol{x}(t))) = \sum_{n=1}^{N} \sqrt{q_n(\boldsymbol{x}(0)) \cdot q_n(\boldsymbol{x}(t))} \tag{8}$$

we have $\hat{D}(a) = \sqrt{1-a}$, $d(a,b) = \sqrt{a \cdot b}$ and

$$\tilde{w}_t(n) = \frac{1}{2} \sqrt{\frac{q_n(\boldsymbol{x}(0))}{q_n(\boldsymbol{x}(t)}} \tag{9}$$

3 Combination of Histograms

Up to now, the formulation of histogram based tracking relies on a certain histogram of n-dimensional features, defined a priori for the tracking task at hand. Some examples are gray value histograms ($n = 1$), edge histograms ($n = 1$) or RGB color histograms

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

($n = 3$). Certainly, using several different features for representing the object to be tracked will result in better tracking performance, especially, if the different features are weighted dynamically according to the situation in the scene. For example, color might be a problem, if illumination changes. In this case, information on the edges might be more useful. On the other side, a unique color of the moving object in a highly textured environment will favour for color and against edges. One idea would now be, to combine several features into one histogram of larger dimension. The problem with that idea is the curse of dimensionality: higher dimensional features result in very sparse histograms so that the estimation of the true, underlying density becomes very inaccurate. This problem prevents us from just combining different features to a bigger feature vector with larger dimension.

We propose now a different solution for combining histograms of different feature for object tracking. The key idea is to use a weighted combination of several histograms with low dimensions instead of one weighted histogram with high dimension. Let $\mathcal{H} = \{1, \ldots, H\}$ be the set of features used for representing the object. For each feature $h \in \mathcal{H}$ we define a separate function $b_t^{(h)}(\boldsymbol{u})$. The number of bins in histogram h is N_h and might differ between the histograms. Also, for each histogram a different weighting function $L_{\boldsymbol{x}(t)}^{(h)}(\boldsymbol{u})$ can be applied, i.e. different kernels for each individual histogram are possible if necessary. This results in H different weighted histograms $\boldsymbol{q}^{(h)}(\boldsymbol{x}(t))$ with the bins

$$\boldsymbol{q}_i^{(h)}(\boldsymbol{x}(t)) = C_{\boldsymbol{x}(t)}^{(h)} \sum_{\boldsymbol{u} \in R(\boldsymbol{x}(t))} L_{\boldsymbol{x}(t)}^{(h)}(\boldsymbol{u}) \delta(b_t^{(h)}(\boldsymbol{u}) - i), h \in \mathcal{H}, i = 1, \ldots, N_h \quad (10)$$

We now define a combined representation of the object by $\phi(\boldsymbol{x}(t)) = \left(\boldsymbol{q}^{(h)}(\boldsymbol{x}(t))\right)_{h \in \mathcal{H}}$ and a new distance function (compare (2)), based on the weighted sum of the distances for the individual histograms

$$D^* = \sum_{h \in \mathcal{H}} \beta_h D_h(\boldsymbol{q}^{(h)}(\boldsymbol{x}(0)), \boldsymbol{q}^{(h)}(\boldsymbol{x}(t))) \quad (11)$$

where $\beta_h \geq 0$ being the contribution of the individual histogram h to the object resprentation. The quantities β_h can be adjusted to best model the object in the current context of tracking. Currently, we set these parameters empirically. In future work we plan to find the optimal values automatically and to dynamically adjust them during tracking.

As before, for the Mean-Shift as well as for the Trust-Region method we can formulate a corresponding optimization problem. If we use the same weighting function $L_{\boldsymbol{x}(t)}(\boldsymbol{u})$ for all histograms and as state $\boldsymbol{x} = (m_x, m_y)^T$ the position of the moving object in the image plane, we get

$$\boldsymbol{x}(t) \approx \operatorname*{argmax}_{\boldsymbol{x}} C_0 \sum_{\boldsymbol{u} \in R(\boldsymbol{x})} L_{\boldsymbol{x}}(\boldsymbol{u}) \underbrace{\sum_{h \in \mathcal{H}} \tilde{w}_{h,t}(b_t^{(h)}(\boldsymbol{u}))}_{w_{h,t}(\boldsymbol{u})} \quad (12)$$

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

which is again a weighted kernel density estimation. The constant C_0 can be shown to be independent of $\boldsymbol{x}$. The corresponding pixel weights are

$$w_{h,t}(\boldsymbol{u}) = \sum_{h \in \mathcal{H}} \tilde{w}_{h,t}(b_t^{(h)}(\boldsymbol{u})) \tag{13}$$

$$= \sum_{h \in \mathcal{H}} -\beta_h sgn(D_h) \frac{\partial d_h(a,b)}{\partial b}\bigg|_{(a,b)=(\boldsymbol{q}^{(h)}(\boldsymbol{x}(0)),\boldsymbol{q}^{(h)}(\boldsymbol{x}))} \tag{14}$$

where $d_h(a,b)$ is defined as in (3) for each individual feature h. For the Trust-Region optimization again gradient and Hessian matrix have to be derived. Details can be found in [10].

4 Experiments

We will now show that a weighted combination of different histograms is suited to improve tracking performance. In the experiments we use the test videos of the CAVIAR project [12], originally recorded for action and behaviour recognition experiments. Although, we do not have this kind of application in mind, the videos are prefectly suited, since they are recorded in natural environment, with change in illumination and scale of the moving persons as well as partial occlusions. Most important, the moving persons are hand-labelled, i.e. for each frame a reference rectangle is stored.

To evaluate the results of the original Mean-Shift and Trust-Region tracker as well as our proposed CHT we used an area based criterion. We measure the difference e of the returned region A and the ground-truth region B by

$$e(A,B) = 1 - \frac{|A \cap B|}{\frac{1}{2}(|A| + |B|)} \tag{15}$$

This error metric is zero, if the two regions are identical, and one if they do not overlap. If the two regions have the same size, the error increases with increasing distance between the center of both regions. Also, equal center but different size if taken care of.

In the experiments we combined three different histograms. The first is the standard color histogram consisting of the RGB channels, abbreviated in the figures as *rgb*. The second histogram is computed from a sobel edge strength image (*gradn*), with the edge strength normalized to fit the gray value range from 0 to 255. The third histogram is computed from a corner interest map (*minev*). This interest map is based on the interest operator returning the smallest eigenvalue of the structure matrix from a 5×5 window around the respective pixel [13]. Thus, high values in the interest map correspond to corners in the image. For simplicity reasons, we call this histogram a corner histogram.

In Figure 1 for the Mean-Shift the accuracy of tracking is documented using the error percentile. For a certain percentile pz (x–axis) we measure the largest error $e(A,B)$ (y–axis, compare (15) taking into account the $pz\%$ best images only. In the left figure, we only evaluated images until object lost, in the right figure all images are considered. The reader can verify, that a combination of RGB with a gradient histogram leads to a significant improvement of tracking stability compared to a pure RGB histogram

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

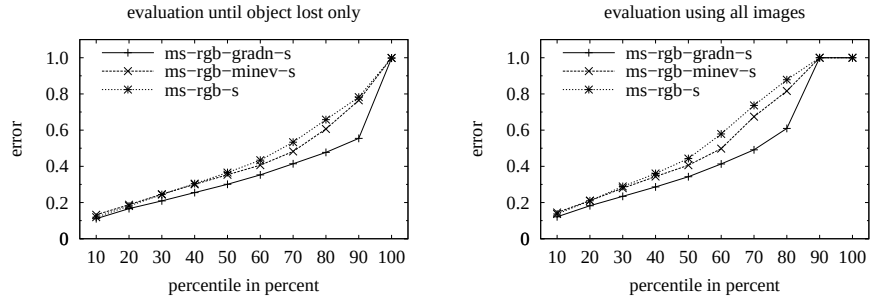

Figure 1. Error percentile using CHT of RGB and gradient (rgb-gradn) and RGB and corner histogram (rgb-minev) as well as pure RGB histogram tracker (rgb). Results are give for the Mean-Shift-Tracker with scale estimation, Biweight-Kernel, Kullback-Leibler distance for all individual histograms

tracker as well as a tracker with a combination of RGB and corner histogram. We got similar results for the corresponding Trust-Region tracker and our extension to combined histograms. The weights β_h for combining RGB with corner and edge histogram (compare (11)) has been empirically set to 0.8 and 0.2, respectively. To automatically find these weights for a certain object and to adjust them dynamically during tracking is one of the focus of our future work.

The computation time for one images is on average less than 2 msec on a PIV, 3.2 GHz compared to approximately 1 msec for a tracker using one histogram only. One example of a successful tracking including correct scale estimation is shown in Figure 2.

5 Conclusion

In our paper we have presented a mathematically consistent extension of histogram based tracking, which we call combined histogram tracker. We could show that the corresponding optimization problems can still be solved using the Mean-Shift as well as the Trust-Region algorithm without loosing real-time capability. The formulation allows the combination of an arbitrary number of histograms with varying dimensions as wells as individual distance functions between two histograms. This allows for a maximum of flexibility in the application of the method. In the experiments we have shown for three different feature histograms that a combination of two of them can improve tracking accuracy and stability. The improvement of course depends on the chosen histogram and on the object to be tracked itself. One important result is, that tracking can still be performed in real-time on standard PC hardware. In the end we like to stress again, that similar results are achieved using the Trust-Region algorithm, although the presentation in this paper was focused on the Mean-Shift algorithm. For more details, the reader is referred to [10].

In our future work we will investigate the adaptive combination of histograms during tracking such that the weights of the histograms are dynamically adjusted depending

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

on the context of tracking, the objects, and background. Also, we are going to compare systematically the CHT with state of the art 2-d tracker, like the tracker of Perez [9].

Figure 2. Tracking result for one of the images sequences from the CAVIAR test bed (first and last image of the sucessfully tracked person). Ground truth and computed region of the moving person (ellipses) are almost the same, even in the case of change in scale.

References

1. Denzler, J., Niemann, H.: Active rays: Polar–transformed active contours for real–time contour tracking. Journal on Real–Time Imaging **5** (1999) 203–213
2. Hager, G.D., Belhumeur, P.N.: Efficient Region Tracking With Parametric Models of Geometry and Illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039
3. Jurie, F., Dhome, M.: Hyperplane Approximation for Template Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 996–1000
4. Comaniciu, D.: Bayesian Kernel Tracking. In: Annual Conference of the German Society for Pattern Recognition. (2002) 438–445
5. 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
6. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 790–799
7. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. SIAM (2000)
8. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall (1995)
9. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: 7th European Conference on Computer Vision. Volume 1. (2002) 661–675
10. Bajramovic, F.: Kernel-basierte Objektverfolgung. Technical Report Masters thesis at Computer Vision Group, Department of Mathematics and Computer Science, University of Passau (2004)
11. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 603–619
12. CAVIAR: Ec funded caviar project, ist 2001 37540, url: http://homepages.inf.ed.ac.uk/rbf/caviar/ (2004)
13. Trucco, E., Verri, E.: Introductory Techniques for 3-D Computer Vision. Prentice Hall (1998)

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