Object Tracking Using Discriminative Feature Selection

Bogdan Kwolek

Rzeszów University of Technology,
W. Pola 2, 35-959 Rzeszów, Poland
bkwolek@prz.rzeszow.pl

Abstract. This paper presents an approach for evaluating multiple color histograms during object tracking. The method adaptively selects histograms that well distinguish foreground from background. The variance ratio is utilized to measure the separability of object and background and to extract top-ranked discriminative histograms. Experimental results demonstrate how this method adapts to changing appearances of both object undergoing tracking and surrounding background. The advantages and limitations of the particle filter with embedded mechanism of histogram selection are demonstrated in comparisons with the standard CamShift tracker and a combination of CamShift with histogram selection.

1 Introduction

This work addresses the issue of on-line selection of discriminative color features during object tracking. Feature selection is a process of mapping the original data into more effective features [1]. If features with little discrimination capabilities are selected, even a good algorithm can lead to poor tracking performance. On the other side, if discriminative features are selected the tracking system can be simplified and thus a limited number of CPU cycles can be sufficient. The most tracking methods operate using only a fixed set of features that are determined in advance. As stated in [2][3], comparatively little work has been done in building tracking systems, which can select most discriminative features on-line. In their work [4], Shi and Tomasi have pointed out that discriminative features are just as equally important as good tracking algorithms.

Selecting a low-dimensional discriminative feature set can improve tracker performance. The goal of dimensionality reduction is to preserve most of the relevant information of the original data according to some optimality criteria. Methods such as principal component analysis (PCA), independent component analysis (ICA) and linear discriminant analysis (LDA) are exemplars of algorithms finding a mapping between the original feature space and a lower dimensional feature space [5]. These methods involve feature transformation and create a set of transformed features rather than a subset of the original features. In work [3] feature extraction is achieved by PCA and the number of dimensions is determined by the pre-defined proportion of eigenvalues. Weights are assigned to each pixel and the mean-shift algorithm [1][6] is utilized to perform tracking. The variance ratio is employed to evaluate the degree of the salience for the foreground in the likelihood image. The main limitation of this approach is that some visual information from the original image can be lost by the projection. The work [2] also uses the likelihood image to combine feature spaces and to select better ones. A method for evaluating several feature spaces while the tracking process proceeds is proposed. It selects the best feature space among candidates that are constructed by different linear combinations of the three color channels from the RGB color space. The method utilizes the previous frame as the training frame to perform a feature selection and then utilizes the current frame as the test frame for foreground-background classification. The features are ranked on the basis of a variance test for the distinctiveness between object and background. Improved tracking performance to standard mean-shift based tracking algorithm has been reported. However, the creation of 49 likelihood images is time consuming.

The importance of the background appearance for tracking has been emphasized in other work [7]. This algorithm maintains a pool of discriminant functions each distinguishing an object pattern against the background patterns that are currently relevant. A searching for the region that best matches the targets and simultaneously avoids background patterns seen previously is embedded in this algorithm. Combining both labeled and unlabeled data is utilized in discriminant expectation maximization (D-EM) algorithm [8] to automatically select a good color space. The basic idea of D-EM is to identify some similar samples in the unlabeled set to grow the labeled data set and then to apply a supervised technique on such enlarged labeled set. Both background and foreground are represented by mixtures of Gaussians.

In work [9] a dynamic switching between five predetermined color spaces takes place in order to improve the performance of face tracking. The selection of color space is done using the ratio of flesh probability pixels within the internal and external face windows with concentric location.

Traditional appearance based representations construct appearance models from examples in training data sets and then utilize such models to track the object of interest. Color histograms [10] that are invariant to some degree of viewpoint change are often used to construct appearance models. Appearance based representations can be very useful in construction of fast and effective tracking systems [11][12][6][13]. For example, the scale invariant feature transform (SIFT) [14] employs a histogram of gradient that is scale and rotation invariant.

Recent work on on-line selection of discriminative features for tracking as well as the success of appearance methods in tracking inspired us to base our tracking method on color histograms. We employ a selection algorithm that maintains a pool of histograms to select histograms yielding more discriminative power. A pool of histograms assigning the various number of bins to each of the color component of the utilized color space is maintained. Our contribution to on-line selection of discriminative features is a method which allows to select the most appropriate color histograms in the current context.

The rest of the paper is organized as follows. The next Section contains a description of evaluating feature discriminability. Section 3. is devoted to object tracking. In Section 4. we outline CamShift based tracking with feature selection. In Section 5. we present all ingredients of our probabilistic tracker with adaptive feature selection and report results which were obtained in experiments. Finally, some conclusions follow in the last Section.

2 Evaluating Feature Discriminability

At the beginning of this section, we show how the log likelihood ratios are computed. The feature space will be presented as the second topic. A description of feature discriminability ends this section.

2.1 Likelihood ratios of foreground and background histograms

A variety of parametric and non-parametric statistical methods can be utilized to represent color distributions of homogeneous colored areas. The histogram is the oldest and most widely applied non-parametric density estimator. It is computed by counting the number of pixels in a region of interest that have given color. The colors are quantized into bins. This operation allows similar color values to be clustered as single bin. By normalizing the histogram by the number of elements in it we form the discrete probability density representing the given object. Methods using histograms techniques are effective only when the number of bins can be kept relatively low and when sufficient data amounts are in disposal. Histogram based methods are only suitable for low dimensional data spaces because as the number of dimensions expand, the number of bins should grow exponentially.

Given a foreground histogram and a background histogram, the log-likelihood ratio for a pixel with color $\mathbf{u}$ is given by [3]:

$$L(\mathbf{u}) = \log \frac{\max(p(\mathbf{u}), \delta)}{\max(q(\mathbf{u}), \delta)}, \tag{1}$$

where δ is a very small number, whereas $p(\mathbf{u})$, $q(\mathbf{u})$ represent the discrete probability density of color pixels in the foreground and background, respectively. Colors that are shared by both foreground and background have values $L(\mathbf{u})$ which tend towards zero. The likelihood image can be computed by back-projecting the ratio for each pixel in the image. Then the salient region in object of interest can by identified by pixels with high likelihood ratios. Such regions, extracted on the basis of different features can be employed to extract a binary mask identifying the object.

2.2 Feature space

The color histograms are usually extracted through assigning to each color channel a fixed number of bits, determined a priori. Such approaches ignore the fact that both foreground and background appearance undergo changes as the target moves. The ability to distinguish between object and background can be insufficient when histograms assigning each color channel a fixed number of bits have been chosen. A color histogram with specific combination of bins for each color channel and possessing good discrimination capabilities for tracking a car in front of green background can perform poorly when colors in the background change their values.

In our approach we maintain identical number of total bins in all candidate histograms. The set of candidate histograms is composed of linear combinations of bin numbers assigned to color channels. In our implementation the RGB color space is utilized and the number of histogram bins m is set to 512. With this histogram length and assuming that the number of bins for each color channel can take the values 2^b, where $b = 0, 1, ..., 5$, we can construct a pool of candidate histograms. Table 1. presents a set of candidate histograms that was utilized in this work. Given a pixel at position x_i, the bin index of 1D histogram is computed as follows:

$$\text{idx} = c_R(x_i) + c_G(x_i) * m_R + c_B(x_i) * m_R * m_G \tag{2}$$

where the function $c_j(x) : \Re^2 \rightarrow \{1, ..., m_j\}$ associates the value of pixel at location x_i to bin number, $j \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$, whereas R, G, B denote color channels.

Table 1. Number of bins assigned to each color channel in the set of candidate histograms

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m_R	8	4	4	8	16	16	8	2	16	16	1	1	16	32	16	32
m_G	8	16	8	4	4	8	16	16	2	16	16	32	32	16	1	1
m_B	8	8	16	16	8	4	4	16	16	2	32	16	1	1	32	16

2.3 Feature discriminability

The foreground and background pixels are sampled using center-surround approach in which an internal rectangle covers the object, while a larger surrounding rectangle represents the background. Following the suggestion in [2], the grade of the salience for the foreground and the likelihood image can be expressed by the variance ratio:

$$\mathrm{VR}(L; p, q) = \frac{\mathrm{var}(L; (p+q)/2)}{\mathrm{var}(L; p) + \mathrm{var}(L; q)} \tag{3}$$

where $\mathrm{var}(L;a) = \sum_i a(i)L^2(i) - [\sum_i a(i)L(i)]^2$. The log likelihood images associated with features of high variance ratio correspond to good features in terms of foreground and background separability. On the basis of the variance ratio we extract top-ranked discriminative histograms.

3 Object Tracking

There are, generally, two types of tracking algorithms: deterministic and probabilistic. The mean-shift algorithm and CamShift are the most famous deterministic tracking algorithms. They may be trapped in local minima and generally can not recover from temporary failure. This problem can ameliorate probabilistic methods built on particle filters. They achieve robustness to clutter and occlusion by maintaining multiple hypotheses over the state space. At the beginning of this section we describe the CamShift algorithm. The second part of this section is devoted to particle filtering.

3.1 CamShift

CamShift tracking algorithm is based on a robust non-parametric technique called mean-shift to seek the nearest mode of probability distribution. The searching starts from the final location in the previous frame and proceeds iteratively to find the nearest mode. The value of each pixel in the probability image represents the probability that the pixel belongs to the object of interest. The object probability density image $P(x, y)$ is extracted through thresholding the log likelihood image.

The mean location of the distribution within the search window is computed using moments [15][12]. It is given by:

$$x_1 = \frac{\sum_x \sum_y xP(x,y)}{\sum_x \sum_y P(x,y)}, \quad y_1 = \frac{\sum_x \sum_y yP(x,y)}{\sum_x \sum_y P(x,y)} \tag{4}$$

where x, y range over the search window. The eigenvalues (major length and width) of the probability distribution are calculated as follows [15][12]:

$$l = 0.707\sqrt{(a+c) + \sqrt{b^2 + (a-c)^2}}, \;\; w = 0.707\sqrt{(a+c) - \sqrt{b^2 + (a-c)^2}} \tag{5}$$

where
$a = \frac{M_{20}}{M_{00}} - x_1^2, \;\; b = 2\frac{M_{11}}{M_{00}} - x_1 y_1, \;\; c = \frac{M_{02}}{M_{00}} - y_1^2, \;\; M_{00} = \sum_x \sum_y P(x,y),$
$M_{20} = \sum_x \sum_y x^2 P(x,y), \;\; M_{02} = \sum_x \sum_y y^2 P(x,y).$

The algorithm repeats the computation of the centroid and repositioning of the search window until the position difference converges to some predefined value, that is, changes less than some assumed value. Relying on the zero-th moment M_{00} the CamShift adjusts the size of the search window in the course of its operation. It requires the selection of the initial location and size of the search window. The algorithm outputs the position, dimensions, and orientation of object undergoing tracking. It can be summarized in the following steps [12]:

1. Set the search window at the initial location (x_0, y_0).
2. Determine the mean location in the search window (x_1, y_1).
3. Center the search window at the mean location computed in Step 2, set the window size to zero-th moment M_{00}.
4. Repeat Steps 2 and 3 until convergence.

3.2 Particle Filtering

The effectiveness of object tracking in image sequences has been greatly improved with the development of particle filtering. The particle filter is an algorithm for estimating the posterior state of a dynamic system over time where the state cannot be measured directly, but may be estimated at the current time-step t. Particle filters are attractive for nonlinear models, multi-modal, non-Gaussian or any combination of these models for several reasons. They utilize imperfect observation and motion models and incorporate noisy collection of observations through Bayes rule. The ability to represent multimodal posterior densities allows them to globally localize as well as relocalize the object of interest in case of temporal failure during tracking. Particle filters are any-time because by supervising the number of samples on-line they can adapt to the available computational resources.

Two important components of each particle filter are motion model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ describing the state propagation and observation model $p(\mathbf{z}_t \mid \mathbf{x}_t)$ describing the likelihood that a state $\mathbf{x}_t$ causes the observation $\mathbf{z}_t$. Starting with a weighted particle set $S = \left\{(\mathbf{x}_{t-1}^{(n)}, \pi_{t-1}^{(n)}) \mid n = 1...N\right\}$ approximately distributed according to $p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1})$ the particle filter operates through predicting new particles from a proposal distribution. To give a particle representation $S = \left\{(\mathbf{x}_t^{(n)}, \pi_t^{(n)}) \mid n = 1...N\right\}$ of the posterior density $p(\mathbf{x}_t \mid \mathbf{z}_{1:t})$ the weights of particles are set to $\pi_t^{(n)} \propto \pi_{t-1}^{(n)} p(\mathbf{z}_t \mid \mathbf{x}_t^{(n)}) p(\mathbf{x}_t^{(n)} \mid \mathbf{x}_{t-1}^{(n)}) / q(\mathbf{x}_t^{(n)} \mid \mathbf{x}_{t-1}^{(n)}, \mathbf{z}_t)$. When the proposal distribution from which particles are drawn is chosen as the distribution conditional on the particle state at the previous time step, the importance function reduces to $q(\mathbf{x}_t^{(n)} \mid \mathbf{x}_{t-1}^{(n)}, \mathbf{z}_t) = p(\mathbf{x}_t^{(n)} \mid \mathbf{x}_{t-1}^{(n)})$ and the weighting function takes the form $\pi_t^{(n)} \propto p(\mathbf{z}_t \mid \mathbf{x}_t^{(n)})$. This simplification leads to a variant of a particle filter, Condensation [16]. From time to time the particles should be resampled according to their weights to avoid degeneracy [17].

4 CamShift Based Tracking with Feature Selection

The tracking algorithm we present here follows the idea of selection of discriminative features on-line, which is presented in [2]. In this section we examine a selection algorithm to determine how well each histogram distinguishes object from background in the current frame. The feature selection algorithm is embedded in CamShift based tracking system.

The CamShift algorithm is utilized to find the estimate of the 2D object location of the object in the frame. Using the estimated object location as well as an object mask we extract all candidate histograms. Afterwards, we select the top-ranked discriminative histograms on the basis of the variance ratio. The best three histograms are used to extract the likelihood images for the next frame. Using such likelihood images we extract the compound likelihood image, which is a simple weighted average. After thresholding the compound image we get the binary image. The compound image is subjected to CamShift.

The algorithm iterates through frames and chooses new sets of discriminative histograms. All candidate histograms representing both background and foreground are adapted over time. To avoid model drift the histograms are adapted using linear combination of current observed histograms, the histograms from the last frame as well as histograms from the first frame. The accommodation coefficients were determined experimentally under assumption that the object appearance will not change drastically over the tracking sequence. The set of features used for tracking changes while the tracking process proceeds. Figure 1. depicts some probability images corresponding to the best and worst pair of foreground and background color histograms, in terms of foreground and background separability.

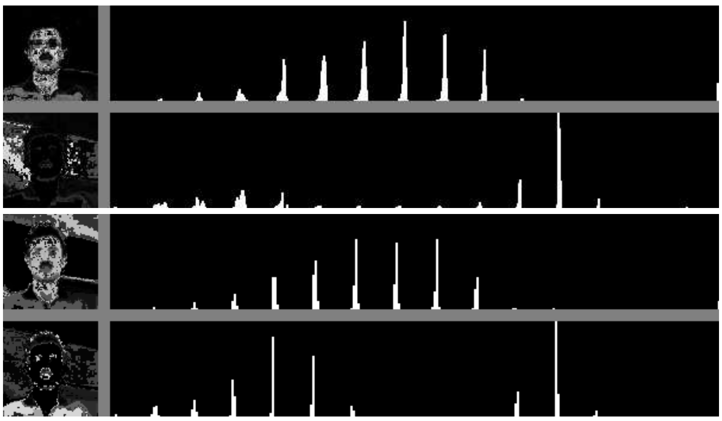
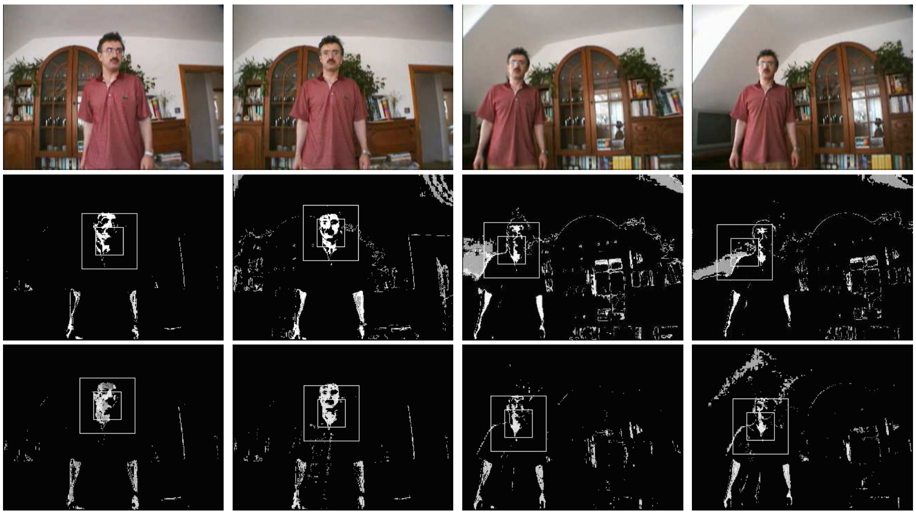


Fig. 1. Probability images of foreground/background and corresponding histograms (frame #50 in the sequence of images demonstrated in Fig. 2). The probability images and histograms for the most discriminative feature are in upper row. The images and histograms for least discriminative feature are in bottom row.

The images from middle row of Fig. 2. illustrate the failure of standard CamShift algorithm. The standard CamShift algorithms operate using only a fixed set of three histograms and do not change this pre-selected set while the tracking process proceeds. During tracking in varying illumination conditions the tracker is affected by similar background color, leading to tracking failure.



#2 #11 #165 #180

Fig. 2. Face tracking in varying illumination conditions using CamShift. Raw color images (top row), object probability images-no feature selection takes place (middle row), object probability images-feature selection (bottom row).

The tracker with histogram selection detects which colors in the model are similar to colors in background and tries to choose the histograms that allow for better foreground/background separation. This property can be observed in Fig. 3. We see that during tracking under illumination changes, frame #165 in Fig. 2., the tracker adapts to changing appearances of both tracked object and the background. Our algorithm continues the tracking whereas the standard CamShift with pre-selected histogram pool suddenly loses the object.

Figure 3. shows how the selection of the best histogram in sequence of images from Fig. 2. evolves over time. For most frames of the sequence the algorithm selects the histogram number zero, see Tab. 1., which assigns the equal number of bins to all color channels. In several frames the algorithm selects thirteen pair of histograms. The selection mechanism supports the tracking and allows the object model to adapt to current conditions and background distractions.

5 Probabilistic Tracking with Feature Selection

In our approach we consider only the location $\mathbf{d} = (x, y)$ in the image coordinate system, the window scale s and the histogram number as the state variables to

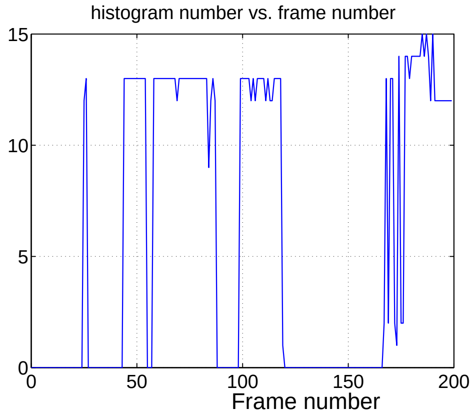


Fig. 3. Number of the best histogram for tracking versus frame number.

be estimated. One way to model the transition of the state is using a random walk which can be described by

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta. \tag{6}$$

A Gaussian noise $N(0, \nu^2)$, where ν^2 is typically learned from training sequences, has been added to the first three state variables, whereas the evolution of the histogram number in such a hybrid state particle filter was modeled using a probability distribution over possible histogram numbers. Such a choice was motivated by observation that the frame to frame position differences in our test sequences are not too large.

The observation model must favor candidate object locations close to the true object locations as well as favor histograms yielding better separability between foreground and background. We therefore need to consider the object probability in the object window given the state of the particle. An iterative mode-seeking in the form of the mean-shift algorithm can be applied to shift the particles to high weight areas [18][19].

The kernel based methods of density estimation construct an estimate of the true density distribution through placing a kernel function on each sample. The estimate of the posterior distribution $p(\mathbf{x}_t \mid \mathbf{z}_t)$ with kernel K can be formulated as follows:

$$\hat{p}(\mathbf{x}_t \mid \mathbf{z}_t) = \sum_{n=1}^{N} K_h(\mathbf{x}_t - \mathbf{s}_t^{(n)})\pi_t^{(n)} \tag{7}$$

where $K_h(\mathbf{x}_t - \mathbf{s}_t^{(n)}) = \frac{1}{Nh^d} K(\frac{\mathbf{x}_t - \mathbf{s}_t^{(n)}}{h})$, and h is the kernel bandwidth. For the radially symmetric kernel we have $K(\mathbf{x}_t - \mathbf{s}_t^{(n)}) = ck(\|\mathbf{x}_t - \mathbf{s}_t\|)$, where c is a normalization constant which makes the integral $K(\mathbf{x}_t - \mathbf{s}_t^{(n)})$ to one, and

$k(r) = k(\|\mathbf{x}_t - \mathbf{s}_t\|)$ is called the profile of the kernel K. In our particle filter we employ the Epanechnikov kernel that is defined as:

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & 0 \le \|\mathbf{x}\| \le 1 \\ 0 & \|\mathbf{x}\| > 1 \end{cases} \tag{8}$$

Given a particle set and the associated weights $\{\pi_t^{(n)}\}_{n=1}^N$, the particle mean is determined by

$$m(\mathbf{s}_t^{(n)}) = \frac{\sum_{i=1}^N H_h(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(i)})\pi_t^{(i)}\mathbf{s}_t^{(i)}}{\sum_{i=1}^N H_h(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(i)})\pi_t^{(i)}}, \tag{9}$$

where $h(r) = -k'(r)$ is in turn a profile of kernel H_h. It can be shown that the mean-shift vector $m(\mathbf{x}) - \mathbf{x}$ always points toward steepest ascent direction of the density function.

The choice of bandwidth h is of crucial importance in kernel based density estimation. A small value can generate a very ragged density approximation with many peaks, while a large value of h can produce over-smoothed density estimates. In particular, if the bandwidth of the kernel is too large, significant features of the distribution, like multi-modality can be missed.

The mode-seeking continues searching until a maximum number of iterations has been reached or until the Euclidean distance between the corresponding modes in the last two iterations is below an empirically determined threshold. We scale down the kernel bandwidth at each mean-shift iteration in order to concentrate on the most dominant modes. Following mode-seeking, the most dominant mode is extracted on the basis of weighted average over all particles within the kernel. The tracking scheme can be summarized as follows: $p(\mathbf{x}_{t-1} \mid \mathbf{z}_{t-1}) \xrightarrow{dynamics} p(\mathbf{x}_t \mid \mathbf{z}_{t-1}) \xrightarrow{measurement} p(\mathbf{x}_t \mid \mathbf{z}_t) \xrightarrow{mean-shift} \hat{p}(\mathbf{x}_t \mid \mathbf{z}_t)$. Each particle can only change its location during mean-shift iterations. The following observation model is utilized:

$$p(\mathbf{z}_t \mid \mathbf{x}_t) = (1.0 - \exp(-\lambda_1 \mathrm{VR}^2)) \times (1.0 - \exp(-\lambda_2 \mathrm{Pr}^2)) \tag{10}$$

where VR denotes the variance ratio and Pr is the mean probability in the object window.

To test our probabilistic tracker we performed experiments using various test sequences. Experimental results that are depicted in Fig. 4. indicate that due to its Monte Carlo nature, the particle filter better handles confusions that are caused by similar colors in the background. Both CamShift and probabilistic tracker were initialized with a manually selected object region of size 20x20 in frame #2799.

The algorithms were implemented in C/C++ and run on a 2.4 GHz PIV PC. The average number of mean-shift iterations per frame is 2.9. The tracker runs with 60 particles at frame rates of 12-13 Hz. All experiments were conducted on images of size 320x240.

#2800 #2810 #2866

Fig. 4. The results of tracking using CamShift (top row) and particle filter (bottom row).

6 Conclusions

We have presented an approach for evaluating multiple color histograms during object tracking. The elaborated method adaptively selects histograms that well distinguish foreground from background. It employs the variance ratio to quantify the separability of object and background and to extract top-ranked discriminative histograms. The superiority of CamShift based tracker using the histogram selection over the traditional CamShift tracking arises because the variance ratio when applied to log likelihood images, which are computed on the basis of various candidate histograms, yield very useful information. Our algorithm evaluates all candidate histograms to determine which ones provide better separability between foreground and background. By employing the histogram selection, the modified CamShift can track objects in case of dynamic background. The particle filter with the embedded selection of histograms is able to track objects reliably during varying lighting conditions. To show advantages of our approach we have conducted several experiments on real video sequences. Currently, only RGB space is used. The performance of the visual tracker could be much better if other color spaces such as HSI could be utilized within this tracking framework.

Acknowledgment

This work has been supported by MNSzW within the project 3 T11C 057 30.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2 edn. Acad. Press (1990)
2. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. In: Proc. of the Int. Conf. on Computer Vision, Nice, France (2003) 346–352
3. Han, B., Davis, L.: Object tracking by adaptive feature extraction. In: Proc. of Int. Conf. on Image Processing (ICIP), Singapore (2004) III:1501–1504
4. Shi, J., Tomasi, C.: Good features to track. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Seattle, Washington (1994) 593–600
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2 edn. John Wiley & Sons, Inc. (2001)
6. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Hilton Head, SC (2000) 142–149
7. Nguyen, H.T., Smeulders, A.: Tracking aspects of the foreground against the background. In: 8th European Conf. on Computer Vision, Prague, Czech Republic (2004) 446–456
8. Wu, Y., Huang, T.S.: Color tracking by transductive learning. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (2000) 133–138
9. Stern, H., Efros, B.: Adaptive color space switching for tracking under varying illumination. Image and Vision Computing **23** (2005) 353–364
10. Swain, M.J., Ballard, D.H.: Color indexing. Int. Journal of Computer Vision **7** (1991) 11–32
11. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: IEEE Conf. on Computer Vision and Pattern Recognition, Santa Barbara, CA (1998) 232–237
12. Bradski, G.R.: Computer vision face tracking as a component of a perceptual user interface. In: Proc. IEEE Workshop on Applications of Comp. Vision, Princeton (1998) 214–219
13. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: 7th European Conf. on Computer Vision, Copenhagen, Denmark (2002) 661–675
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Computer Vision **60** (2004) 91–110
15. Horn, B.K.P.: Robot Vision. The MIT Press (1986)
16. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: 4th European Conf. on Computer Vision, Cambridge, UK (1996) 343–356
17. Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing **10** (2000) 197–208
18. Han, B., Zhu, Y., Davis, L.: Incremental density approximation and kernel-based bayesian filtering for object tracking. In: Int. Conf. on Computer Vision and Pattern Recognition, Washington, DC (2004) 638–644
19. Fritsch, J., J., Kwolek, B.: Kernel particle filter for real-time 3d body tracking in monocular color images. In: IEEE Int. Conf. on Face and Gesture Recognition, Southampton, UK, IEEE Computer Society Press (2006) 564–567