A New Scale and Orientation Adaptive Object Tracking System Using Kalman Filter and Expected Likelihood Kernel

Hamd Ait Abdelali, Fedwa Essannouni, Leila Essannouni and Driss Aboutajdine

Abstract This paper presents a new scale and orientation adaptive object tracking system using Kalman filter in a video sequence. This object tracking is an important task in many vision applications. The main steps in video analysis are two: detection of interesting moving objects and tracking of such objects from frame to frame. We use an efficient local search scheme (based on expected likelihood kernel) to find the image region with a histogram most similar to the histogram of the tracked object. In this paper, we address the problem of scale adaptation. The proposed approach tracker with scale selection is compared with recent state-of-the-art algorithms. Experimental results have been presented to show the effectiveness of our proposed system.

Keywords Object tracking · Computer vision · Integral image · Expected likelihood kernel · Kalman filter

1 Introduction

Real-time object tracking is a critical task in computer vision, and many algorithms have been proposed to overcome the difficulties arising from noise, occlusions, clutters, pose, and changes in the foreground object and/or background environment. Many different algorithms [1–3], have been proposed for object tracking, including mean-shift tracking [4], optical flow, and feature matching. Each algorithm has strengths in certain environments and weaknesses in others. This project aims to combine several such algorithms as inputs or "measurements" to a single Kalman filter [5], for robust object tracking. The filter can favor the algorithm that is most applicable to the current environment by decreasing its measurement noise

H. Ait Abdelali (✉) · F. Essannouni · L. Essannouni · D. Aboutajdine
Faculty of Sciences Rabat GSCM-LRIT Laboratory Associate, Unit to CNRST (URAC 29),
Mohammed V University, B.P. 1014, Rabat, Morocco
e-mail: hamd.abdelali@gmail.com

A. El Oualkadi et al. (eds.), *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, Lecture Notes in Electrical Engineering 380, DOI 10.1007/978-3-319-30301-7_13

variance, and similarly ignore less suitable algorithms by increasing their measurement variances.

Two major components can be distinguished in a typical visual tracker. Target Representation and Localization is mostly a bottom-up process which has also to cope with the changes in the appearance of the target. Filtering and Data Association is mostly a top-down process dealing with the dynamics of the tracked object, learning of scene priors, and evaluation of different hypotheses [4, 6, 7]. The way the two components are combined [8, 9], and weighted is application dependent and plays a decisive role in the robustness and efficiency of the tracker. In real-time applications, only a small percentage of the system resources can be allocated for tracking, the rest being required for the preprocessing stages or to high-level tasks such as recognition, trajectory interpretation, and reasoning. Therefore, it is desirable to keep the computational complexity of a tracker as low as possible.

The goal of this paper is dedicated to improve the similarity measure for the target representation in the Kalman filter. We derive a similarity measure by combining between the expected likelihood kernel [10–12], and the integral image [13], as a similarity measure between target and estimated scale/shape regions in the frames of video sequence. In this paper we analyzes and compares between our system with: Firstly in [7], the efficient local search framework for real-time tracking of complex non-rigid objects. The shape of the object is approximated by an ellipse and its appearance by histogram based features derived from local image properties based on mean-shift and Kalman filter. Secondly the Kalman filter is used as in [4]. Where the shape of the tracked object is approximated by an ellipse and the appearance within the ellipse is described by a histogram based model. The obvious advantage of such a model is its simplicity and general applicability. Another advantage, that made this observation model rather popular, is the existence of efficient local search schemes to find the image region with a histogram most similar to the histogram of the tracked object. Experimental results show that the proposed approach has superior discriminative power and achieves good tracking performance.

The rest of the paper is organized as follows: Sect. 2, introduces basic Kalman filter for object tracking. Section 3, present the expected likelihood kernel. Section 4, present scale estimation. And then Sect. 5 the proposed approach. Section 6 the experiment result. Section 7 concludes the paper.

2 Kalman Filter

The Kalman filter is a framework for predicting a process state, and using measurements to correct or “update” these predictions.

2.1 State Prediction

For each time step k, a Kalman filter first makes a prediction $\hat{s}_k$ of the state at this time step:

$$\hat{s}_k = A \times s_{k-1} \tag{1}$$

where s_{k-1} is a vector representing process state at time $k - 1$ and A is a process transition matrix. The Kalman filter concludes the state prediction steps by projecting estimate error covariance P_k^- forward one time step:

$$P_k^- = A \times P_{k-1} \times A^t + W \tag{2}$$

where P_{k-1} is a matrix representing error covariance in the state prediction at time $k - 1$, and W is the process noise covariance.

2.2 State Correction

After predicting the state $\hat{s}_k$ (and its error covariance) at time k using the state prediction steps, the Kalman filter next uses measurements to "correct" its prediction during the measurement update steps. First, the Kalman filter computes a Kalman gain K_k, which is later used to correct the state estimate $\hat{s}_k$:

$$K_k = P_k^- \times (P_k^- + R_k)^{-1} \tag{3}$$

where R is measurement noise covariance. Determining R_k for a set of measurements is often difficult. In our implementations we calculated R dynamically from the measurement algorithms state. Using Kalman gain K_k and measurements z_k from time step k, we can update the state estimate:

$$\hat{s}_k = \hat{s}_k + K_k \times (z_k - \hat{s}_k) \tag{4}$$

Conventionally, the measurements Z_k are often derived from sensors. In our approach, measurements Z_k are instead the output of various tracking algorithm given the same input: one frame of a streaming video, and the most likely x and y coordinates of the target object in this frame (taken the first two dimensions of $\hat{s}_k$).

The final step of the Kalman filter iteration is to update the error covariance P_k^- into P_k:

$$P_k = (I - K_k) \times P_k^- \tag{5}$$

The updated error covariance will be significantly decreased if the measurements are accurate (some entries in R_k are low), or only slightly decreased if the measurements are noise (all of R_k is high). For more details, see [8, 9, 5].

3 Expected Likelihood Kernel

Let p and q be probability distributions on a space χ and ρ be a positive constant. In this work, we are using the probability product kernels ($K_\rho : \chi \times \chi \rightarrow R$ on the space of normalized discrete distributions over some indexs set Ω) as the similarity measures for comparing two discrete distributions $p_1, p_2, p_3, \ldots, p_N \in \chi$ and $q_1, q_2, q_3, \ldots, q_N \in \chi$. The probability product kernel between distributions $\{p\}_{1..N} \varepsilon \chi$ and $\{q\}_{1..N} \in \chi$ is defined as:

$$K_\rho(p, q) = \sum_{k=1}^{N} p(k)^\rho q(k)^\rho \tag{6}$$

It is easy to show that such a similarity measure is a valid kernel, since for any $p_1, p_2, p_3, \ldots, p_N \in \chi$, the Gram matrix K consisting of elements $K_{ij} = K_\rho(p_i, q_j)$ is positive semi-definite:

$$\sum_i \sum_j \alpha_i \alpha_j K_\rho(p_i, p_j) = \sum_k (\sum_i \alpha_i p_i(k)^\rho)^2 \geq 0 \tag{7}$$

for $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_N \in IR$. Different ρ values are corresponded to different types of probability product kernels. For $\rho = 1$, we have:

$$K_1(p, q) = \sum_k p(k) q(k) = IE_p[q(k)] = IE_q[p(k)] \tag{8}$$

We call this the Expected Likelihood Kernel, is defined by $K(p, q) = \sum_k p(k)q(k)$. We denote the histogram of Target of object tracking T as h_T, and the number of pixels inside T as $|T|$, which is also equal to the sum over bins, $|T| = \sum_k h_T(k)$. Let q be the normalized version of h_T given by $q = \frac{h_T}{|T|}$, so we can consider q as a discrete distribution, with $\sum_k q(k) = 1$. Let p the normalized histogram obtained in the farms of video sequence. For the k-bin of h_T, its value is obtained by counting the pixels that are mapped to the index k:

$$h_T(k) = \sum_{x \varepsilon T} \delta[b(x) - k] \tag{9}$$

where $\delta[t]$ is the Kronecker delta, with $\delta[t] = 1$ if $t = 0$, and $\delta[t] = 0$ otherwise. The mapping function $b(x)$ maps a pixel x to its corresponding bin index. The computation of the expected likelihood kernel can be expressed as:

$$
\begin{aligned}
K(p,q) &= \sum_k p(k)q(k) = \sum_k p(k)\left(\frac{1}{|T|}\sum_{x\varepsilon R}\delta[b(x)-k]\right) \\
&= \frac{1}{|T|}\sum_{x\varepsilon T}\sum_k p(k)\delta[b(x)-k] = \frac{1}{|T|}\sum_{x\varepsilon T} p(b(x))
\end{aligned}
\tag{10}
$$

Therefore, the computation of the expected likelihood kernel can be done by taking the sum of values $p(b(x))$ within candidate target T. The output of the following algorithm is a support map using integral image to compute the similarity measure between target and candidate region from each frame of the video sequence.

4 Adaptive Scale of Target Model

A target is represented by an ellipsoidal region in the image. To eliminate the influence of different target dimensions. Let x_i denote the pixel locations of target model and i are all the pixels that belong to the object tracker, and θ is the location of the center of the object tracker in the frame to frame in sequence video. Suppose we are given an arbitrary shape S in an image specified by a set of pixel locations x_i, i.e., $S = \{x_i\}$. The original shape S we have been initially selected manually. The covariance matrix can be used to approximate the shape of the object:

$$
\theta = \frac{1}{N_s}\sum_{x_i \varepsilon S} x_i, \quad and, \quad V = \frac{1}{N_s}\sum_{x_i \varepsilon S}(x_i - \theta)(x_i - \theta)^t \tag{11}
$$

where N_s pixels that belong to the object of interest, and V describe an arbitrary elliptical region. We use here the following parametrization $s = [\theta^t, scale_x, scale_y, skew]^t$ where $scale_x$ and $scale_y$ are the scaling and $skew$ is the skew transformation obtained from V using the unique Cholesky factorization:

$$
V = \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix}^t \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix} \tag{12}
$$

We will refer to the state S as $s = (\theta, V)$ to explicitly highlight the dependence on θ and V. Similarly, $S(s)$ will denote the elliptical shape defined by s.

The appearance of an object is described by a set of k scalar features $r_1, \ldots, r_k$ that are extracted from the local area of an image I defined by $S(s)$. We view each r_k as a "bin" of a histogram. Let Γ be the set of pixel values $I(x_i)$, for example $\Gamma = [0, 255]^3$ for RGB images. We define a quantization function $b : \Gamma \rightarrow [1, \ldots, k]$, that associates with each observed pixel value a particular bin index k. The value r_k of the k-th bin is calculated from the elliptical image region $S(s = (\theta, V))$ using:

$$r_k(I, s) = |V|^{\frac{\gamma}{2}} \sum_{x_i \varepsilon S(s)} N(x_i; \theta, V)\delta[b(I(x_i)) - k] \tag{13}$$

where δ is the Kronecker delta function. The kernel function N is chosen such that pixels in the middle of the object have higher weights than pixels at the borders of the objects. A natural choice is a Gaussian kernel defined by:

$$N(x; \theta, V) = \frac{1}{|2\pi V|} \exp(-\frac{1}{2}(x - \theta)^t V^{-1}(x - \theta)) \tag{14}$$

The prefactor $|V|^{\frac{\gamma}{2}}$ in (13) discounts for the fact that in practice we use only the N_s pixels from a finite neighborhood of the kernel center. We disregard samples further than 1.6-sigma and it is easy to show that one should use $\gamma \approx 1.6$ in this case. The smooth kernel function will suppress the influence of the (arguably less reliable) pixels near the borders.

5 Proposed Approach

To ensure good organization the progress of work, we used the benefits of modular design in our approach implemented using MATLAB. The goal of an object tracking is to generate the trajectory of an object over time by discovering its exact position in every frame of the video sequence. We have implemented several object tracking algorithms (Kalman filter, Expected likelihood kernel, Adaptive scale) with different processing methods. The step of object tracking system are shown in Fig. 1.

The proposed approach for object tracking is composed of four blocks named as: Block processing, Block prediction, Block Tracking, Block correction and Block result. The functions of these blocks are as follows:

Block Processing In block processing, we start video sequence and converting video into images processing for extracting color information of images and target of object tracking.

Block Prediction Block Prediction step attempts to evaluate how the state of the target will change by feeding it through a state prediction of Kalman filter. The state prediction serves two purposes: The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimate for the next time step.

Block Tracking In this block we combine between the expected likelihood kernel, and the integral image to compute similarity measure, and the histograms of all possible target regions of object tracking in video sequence. And we based of state predicted to estimate shape and orientation of object tracker.

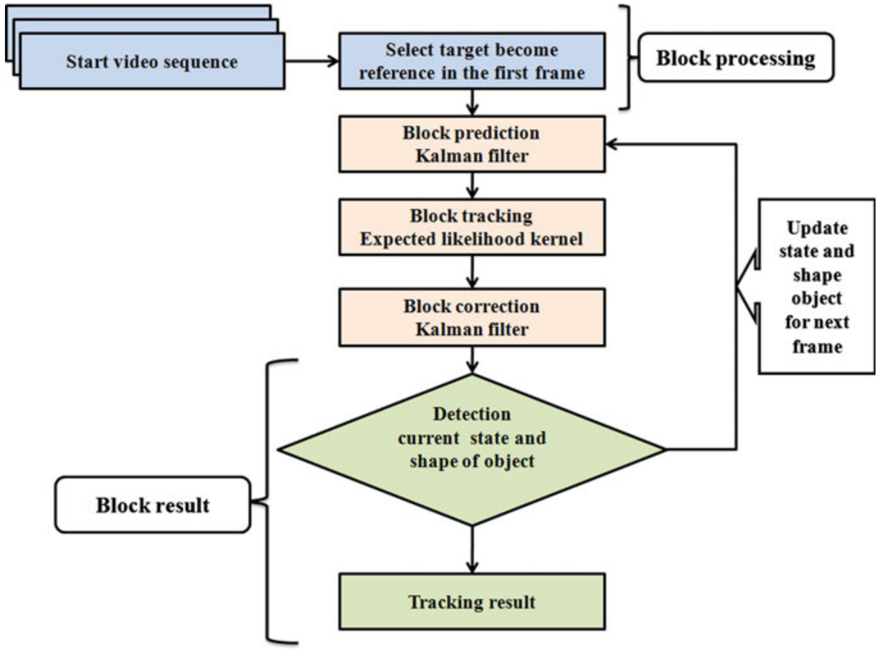
Start video sequence
Select target become reference in the first frame
Block processing
Block prediction Kalman filter
Block tracking Expected likelihood kernel
Block correction Kalman filter
Update state and shape object for next frame
Detection current state and shape of object
Block result
Tracking result

Fig. 1 Basic block diagram for proposed algorithm

Block Correction The Block correction update equations are responsible for the feedback. That is used for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations based on Block Tracking.

Block Result Tracking trajectory of object is done on the basis of the region properties of the object such as, shape, centroid, etc.

The Algorithm of the proposed approach can be explained as follows:
1. Start video sequence and select the target of object tracker in the first frame
2. Prediction using State prediction of Kalman filter to estimate how the state of an target will change by feeding it through a the current state and error covariance estimates to obtain the a priori estimate for the next time step. Using Eqs. (1) and (2)
3. Calculate similarity measure between target model and candidate regions and estimate shape and orientation of object tracker using Eqs. (10) and (13)
4. Correction and update equations into the a priori estimate to obtain an improved a posteriori estimate, using Eqs. (3)–(5), and state of similarity measure, which calculates the new position of the object
5. Draw trajectory by line joining each stored position has been drawn in every frame which shows the trajectory of the selected moving object. And go to step 2 in the next frame

6 Experiment Result

To verify the efficiency of the proposed approach (PA), we compared our system with two existing algorithms MKF [7] and MS [4], the experimental results show that, the PA system achieves good estimation accuracy of the scale and orientation of object in the sequences videos. We used different sequences, each has its own characteristics but the use of a single object in movement is a commonality between these different sequences, and we set up experiments to listed the estimated width, height, trajectory, and orientation of object. In this work, we selected RGB color space as the feature space and it was quantised into 16 × 16 × 16 bins for a fair comparison between different algorithms. One synthetic video sequence and two real videos sequences are used in the experiments:

We first use a Synthetic Ellipse sequence to verify the efficiency of the proposed approach. As shown in Fig. 2. The external ellipses represent the target candidate regions, which are used to estimate the real targets, that is, the inner ellipses. The experimental results show that the proposed approach could reliably track the trajectory of ellipse with scale and orientation changes. Meanwhile, the experimental results by the MKF and MS are not good because of significant scale and orientation changes of the object.

The second video is a Occlusion sequence is on a more complex sequence. As can be seen in Fig. 3, both proposed approach and MKF [7], algorithm can track the target over the whole sequence, and MS [4], does not estimate the trajectory of target orientation change and has bad tracking results. However, the proposed approach system works much better in estimating the scale and orientation of the target, especially when occlusion occurs.

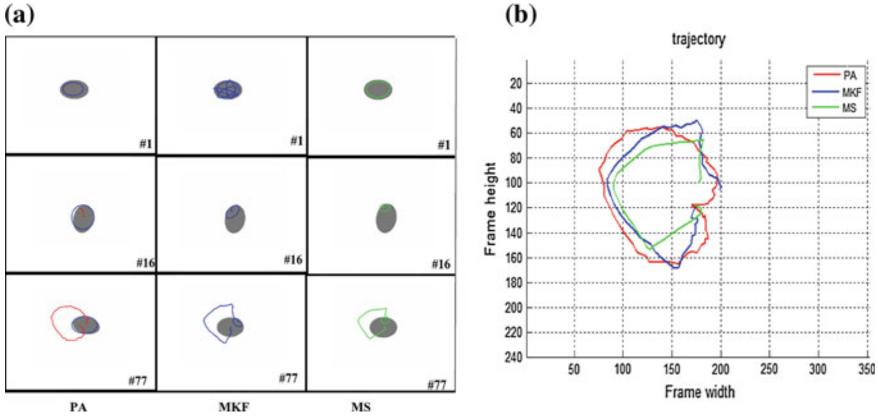

Fig. 2 **a** Tracking results of the synthetic ellipse sequence by different tracking algorithms. The frames 1, 16 and 77 are displayed. **b** Trajectory results of the synthetic ellipse video sequence by different tracking algorithms

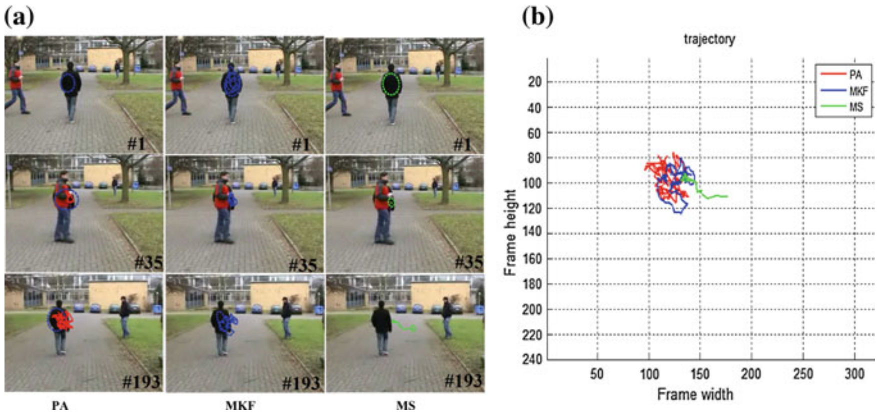

Fig. 3 **a** Tracking results of the occlusion sequence by different tracking algorithms. The frames 1, 35 and 193 are displayed. **b** Trajectory results of the occlusion video sequence by different tracking algorithms

The last video is a Player sequence where the scale of the object increases gradually as shown Fig. 4. The experimental results show that the proposed approach estimates more accurately the scale changes and good trajectory of target region than the MKF and MS algorithms.

Table 1 lists the average time by different methods on the videos sequences. We notice that our proposed approach (PA) has an average time of execution better than MKF and MS algorithms.

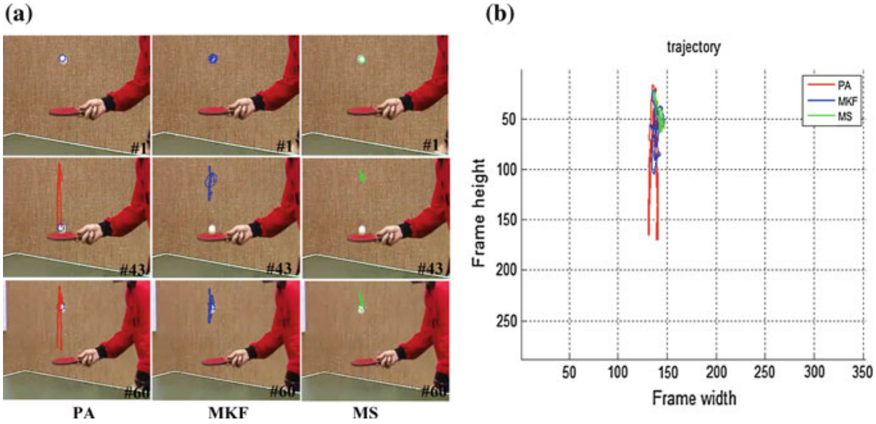

Fig. 4 **a** Tracking results of the player sequence by different tracking algorithms. The frames 1, 43 and 60 are displayed. **b** Trajectory results of the player video sequence by different tracking algorithms

Table 1 The average time by different methods on the videos sequences

Methods/sequences	PA (s)	MKF (s) [7]	MS (s) [4]
Ellipse sequence (77 frames)	0.07	0.48	0.19
Occlusion sequence (193 frames)	0.08	0.56	0.17
Player sequence (60 frames)	0.08	0.40	0.15

The experimental results demonstrate that the proposed approach is robust to track the trajectory of objects in different situations (scale variation, pose, rotation, and occlusion). It can be seen that the proposed approach achieves good estimation accuracy in real-time of the scale and orientation of the target.

7 Conclusion

In this paper, the proposed approach has been presented for tracking a single moving object in the video sequence using color information. In this approach we combine between Kalman filter and expected likelihood kernel as a similarity measure using integral image to compute the histograms of all possible target regions of object tracking in video sequence. The newly proposed approach has been compared with the state-of-the-art algorithms on a very large dataset of tracking sequences and it outperforms in the processing speed. The extensive experiments are performed to testify the proposed approach and validate its robustness to the scale and orientation changes of the target in real-time. This implemented system can be applied to any computer vision application for moving object detection and tracking.

References

1. Ning, J., Zhang, L., Zhang, D., Wu, C.: Scale and orientation adaptive mean shift tracking. IET Comput. Vis. **6**(1), 52–61 (2012)
2. Klein, D.A., Schulz, D., Frintrop, S., Cremers, A.B.: Adaptive real-time video-tracking for arbitrary objects. In: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, pp. 772–777 (2010)
3. Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking. Pattern Recogn. Lett. **49**, 250–258 (2014)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 564–577 (2003)
5. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Fluids Eng. **82** (1), 35–45 (1960)
6. Zivkovic, Z., Krose, B.: An EM-like algorithm for color-histogram-based object tracking. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, vol. 1, pp. I–798 (2004)
7. Zivkovic, Z., Cemgil, A.T., Krose, B.: Approximate Bayesian methods for kernel-based object tracking. Comput. Vis. Image Underst. **113**(6), 743–749 (2009)

8. Xu, S., Chang, A.: Robust Object Tracking Using Kalman Filters with Dynamic Covariance. Cornell University
9. Salhi, A., Jammoussi, A.Y.: Object tracking system using camshift, meanshift and Kalman filter. World Acad. Sci. Eng. Technol. **64**, 674–679 (2012)
10. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. J. Mach. Learn. Res. **5**, 819–844 (2004)
11. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. Learning Theory and Kernel Machines, pp. 57–71. Springer, Berlin (2003)
12. Chang, H.W., Chen, H.T.: A square-root sampling approach to fast histogram-based search. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3043–3049 (2010)
13. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision **57**(2), 137–154 (2004)