Confidence-Based Color Modeling for Online Video Segmentation

Fan Zhong[1], Xueying Qin[2,⋆], Jiazhou Chen[1], Wei Hua[1], and Qunsheng Peng[1,⋆]

[1] State Key Lab. of CAD&CG, Zhejiang University,
Hangzhou, 310027, P.R. China
[2] Department of Computer Science, Shandong University,
Jinan, 250101, P.R. China

Abstract. High quality online video segmentation is a very challenging task. Among various cues to infer the segmentation, the foreground and background color distributions are the most important. However, previous color modeling methods are error-prone when some parts of the foreground and background have similar colors, to address this problem, we propose a novel approach of Confidence-based Color Modeling (CCM). Our approach can adaptively tune the effects of global and per-pixel color models according to the confidence of their predictions, methods of measuring the confidence of both type of models are developed. We also propose an adaptive threshold method for background subtraction that is robust against ambiguous colors. Experiments demonstrate the effectiveness and efficiency of our method in reducing the segmentation errors incurred by ambiguous colors.

1 Introduction

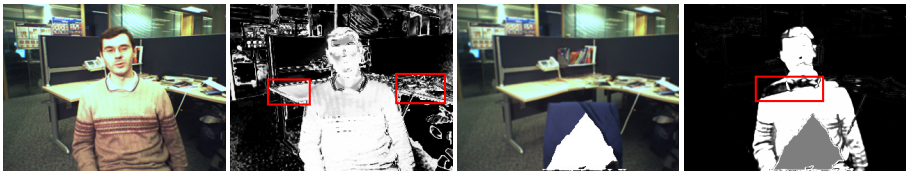
Extracting foreground object from image and video has been an active research topic for a long time[1,2,3,4,5,6]. In recent years, high quality online video segmentation has attracted more and more attention because of its potential applications in teleconferencing and augmented reality, etc. In these applications high quality segmentation that can be used for background substitution is desired.

In [3] the authors introduces an effective binocular (双目的) segmentation method, but its application is limited due to the requirement of binocular inputs. The succeeding works are all for monocular (单眼的) segmentation with stationary (静止的) background [4,5,6], which adopt color, motion and contrast as the main cues to infer segmentation. These cues are combined into an optimization framework that can be solved efficiently with max-flow/min-cut [7].

Color distribution of the foreground and background is the most important cue, which can be represented with global and per-pixel color models. The global model describes the global color distribution of foreground and background, and per-pixel model represents the background color distribution at the location of each pixel, which is in fact the background model be used for background subtraction [8]. As is well known, segmentation methods easy to produce inaccurate

⋆ These two authors are corresponding authors.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 697–706, 2010.



(a) (b) (c) (d)

Fig. 1. The error caused by ambiguous colors. (a) input image; (b) probability map produced by the global color model (a pair of GMMs learned according to the ground truth of (a)), the pixels with greater intensity are more likely to be foreground; (c) the background image. Note that the background image is incomplete (in the large white region), and the reason is explained in section 4; (d) the result of background subtraction. From (b)(d) one can find a lot of misclassified pixels due to ambiguous colors (in the red rectangles).

segmentation when foreground and background have similar colors. However, this problem gained little attention in previous works, in which it seems that the color modeling process is always safe. Fig.1 demonstrates that both global and per-pixel color models may introduce notable errors when ambiguous colors present.

In previous methods, the most often adopted global color model is the Gaussian Mixture Model (GMM). Generally, the global color model can be any classifier that can output probability, so besides GMM, other learning algorithms, including k-NN and SVM, can also be used to build the global color model (if speed is not considered). However, because there is no learning algorithm can avoid introducing errors, the output of global color model is not always trustworthy (Fig.1(b)). The same for the per-pixel color model, although many adaptive threshold methods were proposed for background subtraction, none of them is capable of dealing with ambiguous colors. Consequently, when the overlapped parts of foreground and background have similar colors, foreground may be misclassified as background (Fig.1(d)).

When multiple types of cues are jointly considered, the impact of different cues can be adjusted through their weights. In previous methods, however, the weights of each type of cues are uniform for all pixels, which implies that the predictions of color models are treated equally regardless their correctness. Since the case of every pixel may be different, with uniform weights it would be difficult to achieve the optimal combination of cues at every pixel. We therefore propose to assign each pixel an individual weight based on the confidence of color models at each pixel. In this way we can reduce the impact of incorrect predictions of color models by assigning them lower weights.

Notice that the confidence of prediction is in general not the probability of the predicted class because the latter can be seriously biased due to imperfect inductive biases [9]. A common misunderstanding about the probability is that if a color is ambiguous, a classifier would automatically assign it nearly equal probabilities of belonging foreground and background. This is not true for most

classifiers. Fig.1(b) shows the case of GMM. In fact, in this case most classifiers would classify input feature to be the class whose samples occur more often, which would definitely cause features of the other class be misclassified. In the domain of machine learning there are already some attempts to measure the confidence (or reliability) [9,10] and to design classifiers with controlled confidence [11]. Despite their solid theoretical foundation, they are yet not practical for our problem due to their large computational cost.

The main contribution of this paper can be summarized in three aspects. First, we demonstrate that traditional segmentation model based on uniform weights is error-pone in dealing with ambiguous colors, and then present an confidence-based segmentation model. Second, we propose efficient methods to measure the confidence of both global and per-pixels color models. Third, we introduce an adaptive threshold approach for background subtraction which is shown to be robust against ambiguous colors. Our work focuses on the problems caused by ambiguous colors, which have been noted for a long time but have not been solved yet.

The rest of this paper is organized as follows. Section 2 introduces our confidence-based segmentation model. Section 3 presents the proposed global (section 3.1) and per-pixel (section 3.2) color models capable of measuring confidence and estimating adaptive thresholds, as well as the method to determine the weights of each pixel (section 3.3). Section 4 presents our experimental results, and compares the proposed method with previous video segmentation methods. Finally, we conclude our method in section 5.

2 Confidence-Based Segmentation Model

Let $\mathbf{z} = (z_1, \cdots, z_i, \cdots, z_N)$ be an array of pixel color that represents the input image, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_i, \cdots, \alpha_N)$ be the corresponding segmentation result, where $\alpha_i \in \{0, 1\}$ is the state of the i-th pixel. The segmentation $\boldsymbol{\alpha}$ then can be obtained by minimizing the following energy function:

$$E(\boldsymbol{\alpha}) = \sum_i \dot{\omega}_i E_1(\alpha_i) + \lambda \sum_{(i,j)\in\mathcal{E}} E_2(\alpha_i, \alpha_j) \tag{1}$$

where E_1 is the data term measuring the cost under the assumption that the state of the i-th pixel is α_i, E_2 is the smooth term encoding our prior knowledge about the segmentation, and λ is a free parameter used to trade-off between the data and smooth terms. $\dot{\omega}_i$ is a weighting function encoding the confidence of data terms (in previous methods $\dot{\omega}_i \equiv 1$).

The smooth terms E_2 are not dependent on the color distributions, and we will focus on the data terms E_1 and the weighting function $\dot{\omega}_i$. E_1 is typically computed as the negative log of the foreground color model $p(z_i|F)$ and the background color model $p(z_i|B)$:

$$E_1(\alpha_i) = \begin{cases} -\log p(z_i|F) & \text{if} \quad \alpha_i = 1 \\ -\log p(z_i|B) & \text{if} \quad \alpha_i = 0 \end{cases} \tag{2}$$

Without loss of generality, $p(z_i|F)$ and $p(z_i|B)$ can be assumed to be normalized, that is, $p(z_i|F) + p(z_i|B) = 1$, by giving either of them we can determine both. For clarity we use $p(z_i)$ to denote the normalized (foreground) color model:

$$p(z_i) = \frac{p(z_i|F)}{p(z_i|F) + p(z_i|B)} \tag{3}$$

The color model can be used to describe the global color distribution of the foreground and background. Since the background is stationary, the background color at the location of each pixel can also be described with a distribution function. Therefore, we define $p(z_i)$ as the combination of the global and per-pixel color models:

$$p(z_i) = \ddot{\omega}_i p_*(z_i) + (1 - \ddot{\omega}_i) p_i(z_i) \tag{4}$$

where $p_*(z_i)$ is the normalized global color model, and $p_i(z_i)$ is the normalized per-pixel color model regarding the i-th pixel. $\ddot{\omega}_i$ is a weighting function to balance their effects.

The above model is an extension of the segmentation model used in [4]. The main difference is that in [4], $\dot{\omega}_i \equiv 1$ and $\ddot{\omega}_i \equiv c$ are uniform to all pixels, while in our model they may take different value at different pixels. By computing $\dot{\omega}_i$ and $\ddot{\omega}_i$ according to the confidence of corresponding terms, we can emphasize the impact of reliable cues while suppressing the impact of unreliable cues which may lead to incorrect segmentation, in this way the errors introduced by color modeling process can be greatly reduced.

3 Confidence-Based Color Modeling (CCM)

3.1 Global Color Model

We adopt Gaussian Mixture Model (GMM) to represent the global color distribution:

$$p_*(z_i|F) = \sum_{k=1}^{K^F} \pi_k^F N(z_i|\mu_k^F, \Sigma_k^F) \tag{5}$$

where $(\pi_k^F, \mu_k^F, \Sigma_k^F)$ are the parameters of the k-th component, and K^F is the number of Gaussian components. $p_*(z_i|F)$ is the global foreground color model. The global background color model $p_*(z_i|B)$ is defined similarly. $p_*(z_i|F)$ and $p_*(z_i|B)$ can be trained from the foreground and background training color set $\mathcal{S}^F$ and $\mathcal{S}^B$, respectively. After that the normalized global color model $p_*(z_i)$ can be computed easily by equation (3). The probability map in Fig.1(b) is in fact the visualization of $p_*(z_i)$ acquired in this way.

Nevertheless, the global color model obtained in the above way provides bare probability without confidence measurement. The confidence of $p_*(z_i)$ depends on both the quantity of ambiguous colors around z_i and the accuracy of GMM. Specifically, if in color space, z_i falls in the region of many ambiguous colors , or the color distribution in the neighborhood of z_i cannot be accurately described

(a) (b) (c) (d)

Fig. 2. Probability and confidence map. The input image is the same as in Fig.1. (a) detected misclassified pixels (the gray pixels); (b) confidence map of the global color model (visualized in Fig.1(b)), greater intensity implies higher confidence; (c) probability map produced with our per-pixel color model; (d) confidence map of (c).

with GMM, $p_*(z_i)$ should be of low confidence. However, these two conditions are hard to be evaluated in practice, here we propose a simple, yet effective method to measure the confidence.

Note that the training data sets $\mathcal{S}^F$ and $\mathcal{S}^B$ can be used to validate the learned global color model. A color s is misclassified by the learned model if s is a foreground sample ($s \in \mathcal{S}^F$) but $p(s|B) > p(s|F)$, or s is a background sample ($s \in \mathcal{S}^B$) but $p(s|F) > p(s|B)$. Let $\mathcal{S}^U$ denote the set of all misclassified colors in $\mathcal{S}^F$ and $\mathcal{S}^B$, then we can train an additional GMM $p_*(z_i|U)$ from $\mathcal{S}^U$. $p_*(z_i|U)$ is the probability of z_i be misclassified, larger $p_*(z_i|U)$ implies lower confidence of $p_*(z_i)$. If $p_*(z_i|U)$ is larger than both $p_*(z_i|F)$ and $p_*(z_i|B)$, z_i can be considered to be misclassified. Fig.2(a) illustrates the misclassified pixels detected in this way, which shows that our method successfully found out most misclassified pixels. Now we can compute the confidence of $p_*(z_i)$ as:

$$\mathcal{C}(\, p_*(z_i)\,) = 1 - \frac{p_*(z_i|U)}{p_*(z_i|F) + p_*(z_i|B) + p_*(z_i|U)} \tag{6}$$

where $\mathcal{C}(\,\cdot\,)$ is the confidence function. Fig.2(b) visualizes the confidence of $p_*(z_i)$. One can find that the confidence of pixels vary a lot, and the pixels of ambiguous colors are assigned much lower confidence.

3.2 Per-pixel Color Model

Per-pixel color model is in fact the background model, the maintenance of which has been studied much [8,12,13]. We don't plan to survey all of these methods due to space limitation; instead, we suppose that the background model at each pixel has available as a Gaussian distribution $N(z_i|\mu_i, \Sigma_i)$. The mean μ_i can be regarded as the background color at the location of the i-th pixel.

Given the background model, background subtraction can be accomplished by thresholding the difference of the current pixel color and corresponding background color. Specifically, the i-th pixel is regarded as background if $\parallel\ z_i - \mu_i\ \parallel < T_i$; otherwise it is regarded as foreground, where T_i is the threshold function. A popular way of computing T_i is to make it vary according to the covariance matrix:

$$T_i = \rho\sqrt{\mathrm{tr}(\Sigma_i)} \tag{7}$$

where ρ is a scale factor, and $\mathrm{tr}(\Sigma_i)$ is the trace of the covariance matrix Σ_i. This method can make T_i adaptive to system noise, but it does not consider ambiguous colors. When the overlapped parts of foreground and background have similar colors, the thresholds computed in this way may cause foreground pixels misclassified, as demonstrated in Fig.1(d). In order to solve this problem, the threshold function must take both noise and ambiguous colors into consideration.

Since the foreground object may move to anywhere, a background pixel can be occluded by any part of the foreground. To find out the safe threshold for background subtraction, we need to know the minimum distance d_i from the background color mean μ_i to all the foreground colors:

$$d_i = \min\{\parallel \mu_i - \mu_k^F \parallel | k = 1, \cdots, K^F\} \tag{8}$$

where μ_k^F is the mean of the k-th Gaussian component of the global foreground color model. We need not to check every foreground color samples to find out the minimum distance, which is not only costly but also sensitive to noise. After getting d_i we can define two threshold functions T_i^B and T_i^F:

$$T_i^B = \min(d_i/2, T_i) \quad T_i^F = \max(d_i, T_i) \tag{9}$$

and then the normalized per-pixel color model can be computed as:

$$p_i(z_i) = \begin{cases} 0 & \text{if } \parallel z_i - \mu_i \parallel < T_i^B \\ 1 & \text{if } \parallel z_i - \mu_i \parallel > T_i^F \\ \frac{\|z_i - \mu_i\| - T_i^B}{T_i^F - T_i^B} & \text{otherwise} \end{cases} \tag{10}$$

if μ_i is close to some foreground colors, d_i and T_i^B would be small, which prevents foreground pixels from being misclassified as background; on the contrary, if μ_i is far from all foreground colors, d_i and T_i^F would be large, which can suppress noise better than T_i. Fig.2(c) is the probability map produced by this method. Although it still contains some errors, it looks much better than that shown in Fig.1(d), which is produced with the threshold function T_i as in (7).

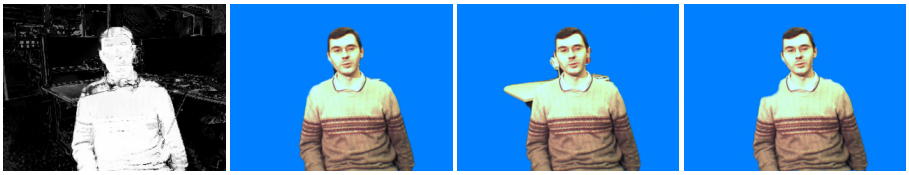
The confidence of the probability $p_i(z_i)$ is dependent on both its magnitude and the reliability of the background model $N(z_i|\mu_i, \Sigma_i)$, so we compute it as:

$$\mathcal{C}(\, p_i(z_i)\,) = \sqrt{e^{-\beta \mathrm{tr}(\Sigma_i)} * |2p_i(z_i) - 1|} \tag{11}$$

where β is chosen to be $(2 < \mathrm{tr}(\Sigma_i) >)^{-1}$, in which $< \cdot >$ denotes the expectation over all pixels. The background model becomes unreliable if it is polluted by foreground colors, in which case $\mathrm{tr}(\Sigma_i)$ is large and $p_i(z_i)$ would be assigned lower confidence. $|2p_i(z_i) - 1|$ would be 0 if $p_i(z_i) = 0.5$, which implies z_i has equal probability to be both foreground and background. Fig.2(d) is the confidence map computed in this way.

3.3 Optimal Combination

Once the confidence of the global and per-pixel color models is known, we can combine them according to the confidence so that the color model with higher



(a) (b) (c) (d)

Fig. 3. Segmentation results. (a) the combined probability map of global and per-pixel color models; (b) foreground obtained with both confidence and adaptive thresholds; (c) foreground obtained without using confidence ($\dot{\omega} \equiv 1, \ddot{\omega} \equiv 0.5$); (d) foreground obtained without using adaptive thresholds ($T_i^F = T_i^B = T_i$).

confidence can take greater effect. Since the two confidence functions $\mathcal{C}(\,p_*(z_i)\,)$ and $\mathcal{C}(\,p_i(z_i)\,)$ are both in the range of $[0, 1]$, they do not need to be re-scaled, and the weighting functions $\dot{\omega}_i$ and $\ddot{\omega}_i$ can be simply computed as:

$$\dot{\omega}_i = \frac{1}{2}(\mathcal{C}(\,p_*(z_i)\,) + \mathcal{C}(\,p_i(z_i)\,)) \tag{12}$$

$$\ddot{\omega}_i = \frac{\mathcal{C}(\,p_*(z_i)\,)}{\mathcal{C}(\,p_*(z_i)\,) + \mathcal{C}(\,p_i(z_i)\,)} \tag{13}$$

$\dot{\omega}_i$ can be regarded as the confidence of the combined color model $p(z_i)$. If both global and per-pixel color models at pixel z_i are of low confidence, $\dot{\omega}_i$ would be small, and the corresponding data term is assigned low weights, then smooth term would dominate the state of the corresponding pixel. Fig.3(a) shows the combined probability map.

4 Experimental Results

In experiments we adopt the video segmentation data set from Microsoft I2I project[1]. The test environment is a computer with 2.2GHz CPU and 4G RAM. The algorithm is implemented in C++.

Implementation details: The data terms E_1 are computed with the proposed method, and the smooth terms E_2 are computed in the same way of [4]. Since the background image is not provided in the data set, we have to accumulate it in online phase. At the start the background model of all pixels are invalid, after segmenting a frame, the acquired background pixels are used to fill the hole of the background image, and other parts of the background image are also updated as in [4]. Henceforth, the background image we use is incomplete, as shown in Fig.1(c).

The global color model is trained in the initialization phase. In [4] the program is initialized with the background image. [5] proposes an automatic initialization

[1] http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm

Fig. 4. Visual comparison with BC [4]. *Top*: input frames; *Middle*: foreground obtained with BC; *Bottom*: foreground obtained with our method.

method, but it needs labeled videos to train the motion model. Since the background image is not available in our case, we simply initialize our program with the ground truth of the first frame. In practice the initialization method can be chosen freely according to the available information.

The segmentation result is finally obtained by minimizing equation (1) with min-cut [7], then the object boundary is smoothed to suppress flicking.

Computational cost: Our system can achieve a speed of $10 \sim 12$ fps for input image sequence of size 320×240. Most computational cost is spent on minimizing the energy equation. To measure the confidence and to compute the adaptive thresholds bring only a little more cost, which is about 12ms in the case of $K^U = K^F = 10$ (lookup table is used to accelerate the computation of the exponential function in GMM).

Effect of CCM: Fig.3 demonstrates the effectiveness of the proposed color modeling method. The input image is hard to be precisely segmented due to the large area of ambiguous colors. Fig.3(c) is the foreground obtained with uniform weights, in which the desktop is mis-segmented as foreground due to the error introduced by the global color model (Fig.1(b)). Fig.3(d) is the foreground obtained without using adaptive threshold. Since the shoulder of the person appears nearly the same as the desktop, it is misclassified as a part of background by the per-pixel color model (Fig.1(c)). By using both nonuniform weights and adaptive thresholds, our method can generate much better segmentation result (Fig.3(b)).

Comparison with other methods: Fig.4 provides some visual comparisons of our method with "Background Cut" (BC, [4]). Since the background image

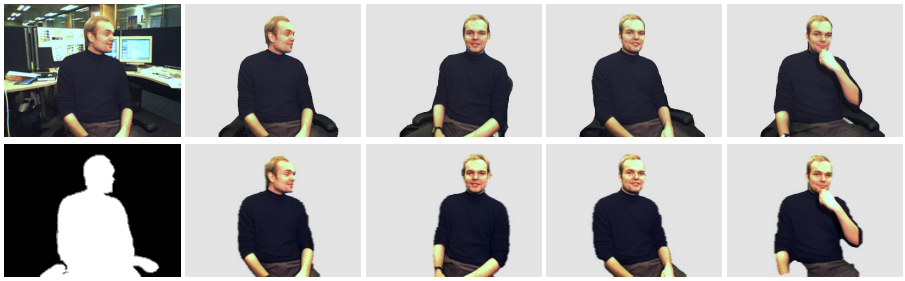


Fig. 5. Visual comparison with TM [5]. The first column is the input image and the ground truth of the frame #130.

Table 1. The error rates (%) of CCM, BC [4] and TM [5]

	JM	MS	AC	VK	50	54	56
CCM	0.13	1.40	0.47	0.68	1.12	0.39	0.33
BC	0.16	2.44	0.56	1.12	1.43	0.52	0.68
TM	0.12	2.59	0.52	-	-	-	-

is not available, our implementation of BC is not exactly the same as described in [4]. The only difference between our implementation of BC and our method exists in the modeling of color distributions, i.e. the computation of E_1 and its weights, so the comparison between them is fair.

Fig.5 is the comparison with [5] (TM), which involves Temporal and Motion priors as its cues. The results of TM are extracted from the published video, so fair comparison is not guaranteed. Tab.1 lists the error rates of CCM, BC and TM. Notice that the ground truth is available only every 5 or 10 frames, so not every frame are evaluated and the error rates may not capture all errors.

In fact, the implementation of our method in this experiment is a version of BC boosted with the proposed color modeling method. Since our color modeling method is independent of how the program is initialized and how other energy terms are computed, it can also be used to boost the performance of any other video segmentation methods that adopt color distribution as segmentation cues.

5 Conclusions

In this paper we propose a confidence-based color modeling method to improve the robustness of online video segmentation against ambiguous colors. A new confidence-based segmentation model is presented, which assigns energy terms nonuniform weights based on their confidence. We developed methods for measuring the confidence of both global and per-pixel color models, and for computing adaptive thresholds for background subtraction. The confidence is then

used to determine the weights of color models and energy terms at each pixel in order for the optimal combination of cues.

Experiments show that the proposed method can greatly enhance the segmentation result, especially for frames with large amount of ambiguous colors present. Our method to measure the confidence is very fast, and brings only a little more computational cost.

The limitation of our work is that it accounts for only ambiguous colors. Besides this, the change of lighting conditions, shadowing and camera shaking, etc. can also lead to errors in the color modeling process. Our future work is to address these problems in the confidence-based framework.

Acknowledgments. This paper is supported by 973 program of china (No. 2009CB320802) and NSF of China (No. 60870003).

References

1. Chuang, Y.-Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proceedings of CVPR, pp. 264–271 (2001)
2. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics 23, 309–314 (2004)
3. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. In: Proceedings of CVPR, pp. 407–414 (2005)
4. Sun, J., Zhang, W., Tang, X., Shum, H.Y.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
5. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proceedings of CVPR, pp. 53–60 (2006)
6. Yin, P., Criminisi, A., Winn, J., Essa, I.: Tree-based classifiers for bilayer video segmentation. In: Proceedings of CVPR (2007)
7. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 359–374 (2004)
8. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of CVPR, vol. 2, pp. 252–259 (1999)
9. Kukar, M., Kononenko, I.: Reliable classifications with machine learning. In: International Conference on Machine Learning (ICML), pp. 219–231 (2002)
10. Nouretdinov, I., Melluish, T., Vovk, V.: Ridge regression confidence machine. In: International Conference on Machine Learning (ICML), pp. 385–392 (2000)
11. Li, M., Sethi, I.K.: Svm-based classifier design with controlled confidence. In: International Conference on Pattern Recognition (ICPR), pp. 164–167 (2004)
12. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallfower: Principles and practice of background maintenance. In: International Conference on Computer Vision, pp. 255–261 (1999)
13. Mahadevan, V., Vasconcelos, N.: Background subtraction in highly dynamic scenes. In: Proceedings of CVPR (2008)