LOCAL KERNEL COLOR HISTOGRAMS FOR BACKGROUND SUBTRACTION

Philippe Noriega, Benedicte Bascle, Olivier Bernier
France Telecom, Recherche & Developpement
2, av. Pierre Marzin, 22300 Lannion, France
$\{$*philippe.noriega, benedicte.bascle, olivier.bernier*$\}$*@francetelecom.com*

Keywords: Histograms, background subtraction, color quantization.

Abstract: In addition to being invariant to image rotation and translation, histograms have the advantage of being easy to compute. These advantages make histograms very popular in computer vision. However, without data quantization to reduce size, histograms are generally not suitable for realtime applications. Moreover, they are sensitive to quantization errors and lack any spatial information. This paper presents a way to keep the advantages of histograms avoiding their inherent drawbacks using local kernel histograms. This approach is tested for background subtraction using indoor and outdoor sequences.

1 INTRODUCTION

A normalized color histogram is easy to compute and is invariant to rotation and translation of image content. It is robust regarding partial occlusions of objects of interest in the scene. These advantages explain why histograms are widely used in computer vision. Examples of applications are: content based image retrieval (CBIR) (Han and Ma, 2002; Pass and Zabih, 1996; H. Yamamoto, 1999), tracking (B. Han and Davis, 2005; M. Mason, 2001), background subtraction (A. Elgammal and Davis, 2000; K. Toyama and Meyers, 1999)...

However, histograms have some drawbacks. First, they lack any spatial information: two images can have the same histogram and be dissimilar due to a different ordering of the pixels in the images. A second drawback occurs when histogrammed data is insufficiently quantized. This problem generally implies large histograms (several thousands of bins) requiring important computation costs and preventing real-time computation. Histograms are also sensitive to image noise and to quantization errors that may cause bin changes even though image variation is small. So, bin by bin comparison measure can lead to important dissimilarities between histograms from similar pictures.

The goal of local kernel histograms is to deal with these drawbacks while keeping the advantages of histograms. This technique is applied on background subtraction using local kernel color histograms to demonstrate its efficiency. The next section presents the related works, section 3 describes the local kernel histograms taking example on color feature extraction, section 4 explains how to apply them to background subtraction, experimental results are presented in section 5 and section 6 concludes this paper.

2 RELATED WORKS

Some histogram techniques permit to recover missing spatial information. The color cooccurrence histogram (Chang and Krumm, 1999; Huang et al., 1997) is an elegant solution where a histogram bin b is associated with two colors c_1, c_2 and a distance d. The histogram bin $b(c_1, c_2, d)$ records the number of (c_1, c_2) colored pixel pairs wich are d distant. A variant consist in only considering pixels belonging to contours (Crandall and Luo, 2004). Color cooccurrence histograms tend to have a huge number of bins making real time computation difficult. Another solution is to split the histogram bins in two classes to classify coherent and incoherent pixels of the same color (Pass and Zabih, 1996). A pixel is considered as coherent if it is part of a homogeneously colored zone. Otherwise, the pixel is considered as incoherent. This method needs clustering algorithms to define the homogeneous zones. A last solution for this problem consists in dividing the image in regions and computing a histogram for each one. Each histogram is associated with a local zone in the image providing spatial information. A variant consists in dividing the image in equal squares and compute one histogram for each square (M. Mason, 2001). The Multi-Scale Histogram Intersection Representation (MSHIR) (Gargi and Kasturi, 1999) is another variant. It is a global to local representation where the image is divided into decreasing scale blocks. Another similar approach consists in recursively dividing the image into regions until each region has a homogeneous feature distribution or until the size of each region becomes smaller than a given threshold value (H. Yamamoto, 1999).

To reach real-time performance, it is necessary to reduce the amount of data by quantizing the feature space before histogram computation. Considering color histograms, quantization consists in putting close colors in the same histogram bin. Quantization can be performed in different color spaces. (M. Mason, 2001) applies a color depth reduction formula to transform the 24-bit RGB color space to 12-bit, (Crandall and Luo, 2004) work in CIE LAB color space and reduce it to 267 standard colors in a first stage before keeping only 10 basic colors. The CIE LAB space has the advantage of being perceptually uniform i.e. the Euclidean distance between two colors corresponds to the human perception difference. The calculation of the distance between two histograms is another way to reach real time computation. In the case of quadratic histogram distance (J. L. Hafner, 1995), the weight matrix that contains the coefficients denoting the similarity between histogram bins can be diagonalized offline.

Filling several histogram bins with a unique pixel is a good method to reduce influence of noise and of quantization errors in histogram computation (Han and Ma, 2002). Quadratic distance (J. L. Hafner, 1995) yields the same advantage but use only the Euclidean distance in histogram similarity computation.

3 COLOR LOCAL KERNEL HISTOGRAMS

In the proposed technique, image is segmented into overlapped **local** squares with a **histogram** for each one to provide accurate spatial information. To reduce significantly the amount of data without loosing important information because of coarse quantization, the color space is quantized according to the most representative colors extracted from the scene. A double Gaussian **kernel**, one in the image space and one in the color space bring robustness against noise. Technical implementation is described further below.

3.1 Image partitioning

Histograms must be computed from a group of pixels. For maximum spatial accuracy, the image is partitioned in $n \times n$ square like regions that are overlapped with the same gap g for both image axis coordinates. So, excluding the image edges, a pixel belongs to $N_a = (n/g)^2$ regions.

On one hand, n must be large enough to smooth both camera vibrations and waving objects in the scene. On the other hand, too large regions prevent accurate objects of interest detection. In experimental results, n is fixed at 12 pixels with a gap $g = 3$. More overlapping requires excessive computing resources.

3.2 Color quantization

Quantization allows saving computer resources by reducing the histogram sizes. Because camera noise prevents distinguishing between all the 256×256 colors in the UV space, this last is reduced to 40×40 colors. Then, a good option is quantizing taking into account the most representative colors in the scene. In this way, n_c colors are selected from the image reference to be associated to n_c histogram bins. An undefined color bin is added for other unselected colors. Thus, all pixels not corresponding to one of the selected colors is associated with the undefined color bin.

This approach brings a great improvement in term of computation time. To represent more than ninety percent of reduced colors in a cluttered scene, the color histogram size is set to only 15 color bins. Moreover, this size is smaller than those reached by good quantization: 64 with fuzzy histograms in CBIR application (Han and Ma, 2002) or 1600 bins in CIELAB (Crandall and Luo, 2004), and much smaller than those usually reached: 4096 for the tracking algorithm presented in (M. Mason, 2001) or 9796 for color cooccurrence histograms (Chang and Krumm, 1999).

3.3 Kernels

Instead of associating one pixel with a unique region and a unique histogram bin, Gaussian kernels are introduced in both image and color space to bring more flexible fuzzy associations between image and histograms. Gaussian kernels are also chosen because of its smoothing properties and are easily computed. For computation efficiency, the kernels are pre-computed and stored in lookup tables.

3.3.1 Spatial Gaussian kernel

Pixels $S_k(x_k, y_k)$ in a local area l are weighted in terms of distance from the area center. Thus, to compute the local histogram H_l, the pixel contributions are weighted according to a bi-dimensional spatial Gaussian kernel $G_l^S(\mu_l^S, \sigma_s)$ with mean $\mu_l^S(x_l, y_l)$ on the area center and standard deviation σ_s (see Figure 1). K^S is a normalization coefficient:

$$\begin{aligned} d_x &= x_k - x_l \ , \\ d_y &= y_k - y_l \ , \\ G_l^S(S_k) &= \frac{K^S}{2\pi\sigma_s} exp\left(-\frac{d_x^2 + d_y^2}{2\sigma_s^2}\right) \ . \end{aligned} \tag{1}$$

The ratio weight between the border and center area of regions must be low enough to provide good smoothing properties. Thus, the standard deviation σ_s is chosen to be about a quarter of the local area size. This setting brings 95 percent of the Gaussian kernel inside the area and gives a ratio weight of about 0.135. K^S normalizes the kernel on the area: $\sum_{k=1}^{n^2} G_l^S(S_k) = 1$.

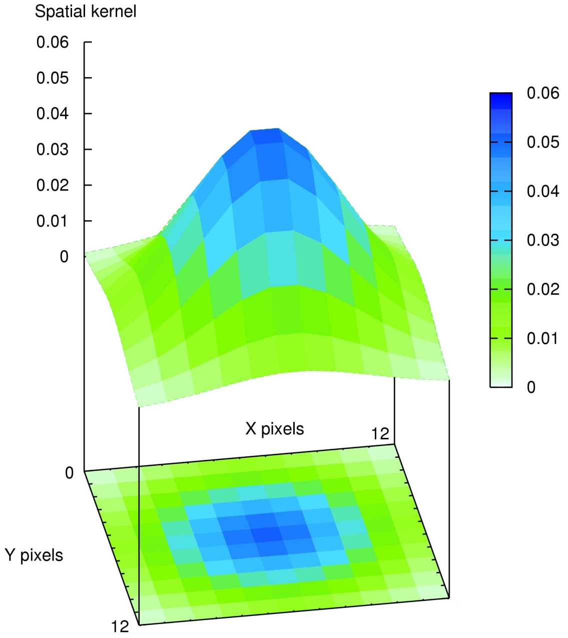

Figure 1: Spatial Gaussian kernel on a local area of 12×12 pixels. The standard deviation is low enough ($\sigma_s = 3$) to provide good smoothing properties

3.3.2 Color Gaussian kernel

Two different colors falling in two separate histogram bins are considered dissimilar even if they are very close. This is a significant classical histogram drawback.

Using a color Gaussian kernel alleviates this problem and takes into account colors similarity. Instead of falling into a unique histogram bin, a pixel is shared between several bins according to a Gaussian weight G^C. In YUV color space and given h_j, a bin representing the color (U_j, V_j) in the chrominance histogram, the contribution of the pixel $S_k(x_k, y_k, Y_k, U_k, V_k)$ to the h_j bin is:

$$\begin{aligned} d_U &= U_k - U_j \ , \\ d_V &= V_k - V_j \ , \\ G_j^C(U_k, V_k) &= \frac{K_j^C}{2\pi\sigma_c} exp\left(-\frac{d_U^2 + d_V^2}{2\sigma_c^2}\right) \ . \end{aligned} \tag{2}$$

K_j^C is a normalization coefficient determined for the color (U_j, V_j) among the n_r colors in the reduced space: $\sum_{i=1}^{n_r} G_j^C(U_i, V_i) = 1$. Standard deviation σ_c is estimated by taking into account the camera noise.

3.4 Local kernel histograms computation

As explained above, local kernel histograms are computed from image overlapped regions taking into account the two Gaussian kernels: the former in image space and the second in color space.

In a local area l, the value of a h_j histogram bin corresponding to a selected color ($1 \leq j \leq n_c$) is:

$$h_j = \sum_{k=1}^{n^2} G_l^S(S_k) G_j^C(S_k) \ . \tag{3}$$

For the undefined color bin, all occurs as if histogram contains all the n_r colors in the reduced space ($1 \leq j \leq n_r$). Then, the value of the undefined color bin is the sum of unselected color bins:

$$h_{j+1} = \sum_{(j=n_c+1)}^{n_r} h_j \ . \tag{4}$$

Of course, for fast histogram computation, contributions of each colors in the reduced color space are pre-computed in lookups tables. The normalized histogram H_l contains n_c colors bins plus the undefined color bin. It is normalized due to the normalization constants K^C and K^S.

4 APPLICATION TO BACKGROUND SUBTRACTION

In background subtraction, histograms are often used to extract spatial or temporal features of background. Those can being color or contours orientation for spatial features or pixel value versus frame number in the case of temporal features. For example (A. Elgammal and Davis, 2000) compute their background model using histograms that describe temporal statistics for pixels values. In their region scale process, (K. Toyama and Meyers, 1999) use histograms to compare moving regions between frames. This section describes how to apply local kernel histograms in color background subtraction to obtain a pixel scale probability map.

4.1 Local area probability

As histograms are normalized, the Bhattacharyya distance between them provides a result between 0 and 1 which can be assimilated as a probability. Given histograms $H_l^{t_0}$ and H_l^t computed from the same area l respectively in reference and current image, the probability P_l that l belongs to the background is computed applying Bhattacharyya distance to the histogram bins h_j.

$$P_l = \sum_{j=1}^{n_c+1} \sqrt{h_j^{t_0} h_j^t} \; . \qquad (5)$$

4.2 Pixel probability

Area histogram similarity computation provide an identical probability for all the pixels in the area. Thus, the resulting probability map is heavily aliased. If it is suitable for tracking (M. Mason, 2001), background subtraction needs generally more spatial accuracy. Overlapping between areas reduce aliasing but there is a trade off between computation time and gap size between areas. To provide a pixel scale map while preserving computation ressources, the probability is computed with the probabilities resulting from the N_a areas that a pixel belongs to. Taking account of the spatial kernel G^S, the probability P_s for the pixel $S(x_k, y_k, Y_k, U_k, V_k)$ is:

$$P_s = \frac{1}{\sum_{l=1}^{N_a} G_l^S} \sum_{l=1}^{N_a} G_l^S P_l \; . \qquad (6)$$

5 EXPERIMENTAL RESULTS

The local kernel histograms are compared with three other algorithms in the field of background subtraction. Each algorithm use chrominance channels UV from YUV color space:

Mean & Threshold: Pixel-wise mean values are computed during a training phase, and pixels within a fixed threshold of the mean are considered background.

Mean & Covariance: The mean and covariance are computed from the recent samples values for pixels. Foreground pixels are determined using a threshold. This is similar to the background algorithm used in (A. Elgammal and Davis, 2000).

Histograms: Frames are segmented into 50% large overlapped square zones of 20 pixels. A conventional color histogram is computed from each zone for both reference and current image. Similarity is computed with histogram intersection and a threshold determines foreground pixels: see (M. Mason, 2001).

Local Kernel Histograms: The method explained in this paper, probability map is thresholded to extract silhouettes.

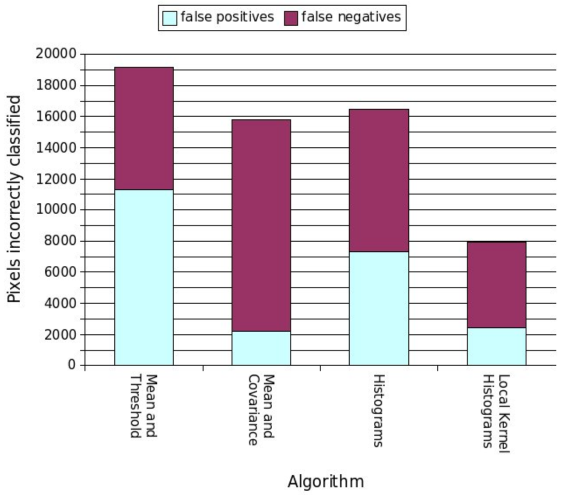

Figure 2: Algorithms overall performance.

Both indoor and outdoor test sequences are used (see Figure 3). The third (foreground covers monitor pattern) and the fourth (waving trees) sequences were used by (K. Toyama and Meyers, 1999). They are available from the web[1] in color with a 160×120 pixels resolution. The first indoor scene was grabbed

[1] http://research.microsoft.com/users/jckrumm/WallFlower/TestImages.htm

with a color CCD camera using 384×288 pixels resolution and the last outdoor scene with a webcam and a 320×240 pixels resolution. Image quality is relatively poor. The five sequences show classical difficulties for background subtraction:

Camera Vibrations: Camera is not strongly fixed and vibrations cause small image motion.

Shadows and Reflections: A person stays between the window and the door. Shadow and reflections slightly modify the background on the left side of the picture.

Foreground Covers Monitor Pattern: A monitor lies on a desk with rolling interference bars. A person walks into the scene and occludes the monitor.

Waving trees: A person walks in front a swaying tree.

Gust of Wind: A person walks in front of swaying flowers. Suddenly, a gust of wind occurs. The flowers move with more intensity.

The test images are shown in Figure 3. Tests are performed on a single frame from each sequence and consist in segmenting a human subject from the background. Mean & Threshold and Mean & Covariance algorithms are both initialized during the first 200 frames before the test. Histogram based algorithms are only trained with the first image of the test sequence.

Because histograms naturally have the capacity to smooth noise, camera vibrations and swaying flowers do not affect histogram based algorithms. If the Mean and Covariance algorithm succeeds on the waving trees scene, it needs a certain time to adapt its background model causing false detections when an unexpected event occurs e.g. a gust of wind. On the other hand, conventional histograms fail when shadows and reflections appear in the scene. In this case, UV colors channels are slightly modified yielding pixels jumps between histograms bins and obviously, conventional histograms bin by bin comparison measures fails. It is a classical histogram drawback. However, small color changes do not affect local kernel histograms because the color kernel reduces quantization errors. Conventional histograms (M. Mason, 2001) result in strongly aliased foreground detection. Thus, because of their poor spatial accuracy, histograms are generally not suitable for silhouette pose or gesture analysis. However, local kernel histograms provide spatial accurate probability maps (cf. § 4) for silhouette extraction.

The results of the tests are shown in Figure 2 and table 1. As in (K. Toyama and Meyers, 1999), performances are evaluated in term of number of foreground pixels marked as background (false negatives) and background pixels marked as foreground (false positives). Ground truth is provided by hand segmentation. It is obvious that the few test sequences produced in this paper are not sufficient to correctly evaluate the difference between the algorithms. However, results underline the capacity of local kernel histograms to naturally smooth noise from camera, soften shadows or reflections and waving background objects.

In terms of computation load, the local kernel histograms modelizes a local area including n^2 pixels with a histogram comprising n_c+1 bins. In our experiments, 144 pixel in a local area are modelized with only 16 bins. Moreover, Gaussian kernels are precomputed and stored in lookup tables, yielding a fast histogram computation. Thus, even with strong overlapping between local areas, computation times are close to those required by the Mean & Threshold algorithm.

6 CONCLUSION

As shown in experimental results, the local kernel histogram based algorithm is a robust and efficient method to extract color information from images. Even in noisy environment with camera vibrations or swaying vegetation, they provide useful and accurate probability map for background subtraction. This method is easily generalizable to other features e.g. contours, and can be useful in many fields of computer vision e.g. content based image retrieval (CBIR) or tracking. This paper has demonstrated that local kernel histograms combine conventional histograms advantages and avoid their inherent drawbacks to provide robust, fast and accurate spatial information. Illumination robust background subtraction using contour features and local kernel histograms will be addressed in future works.

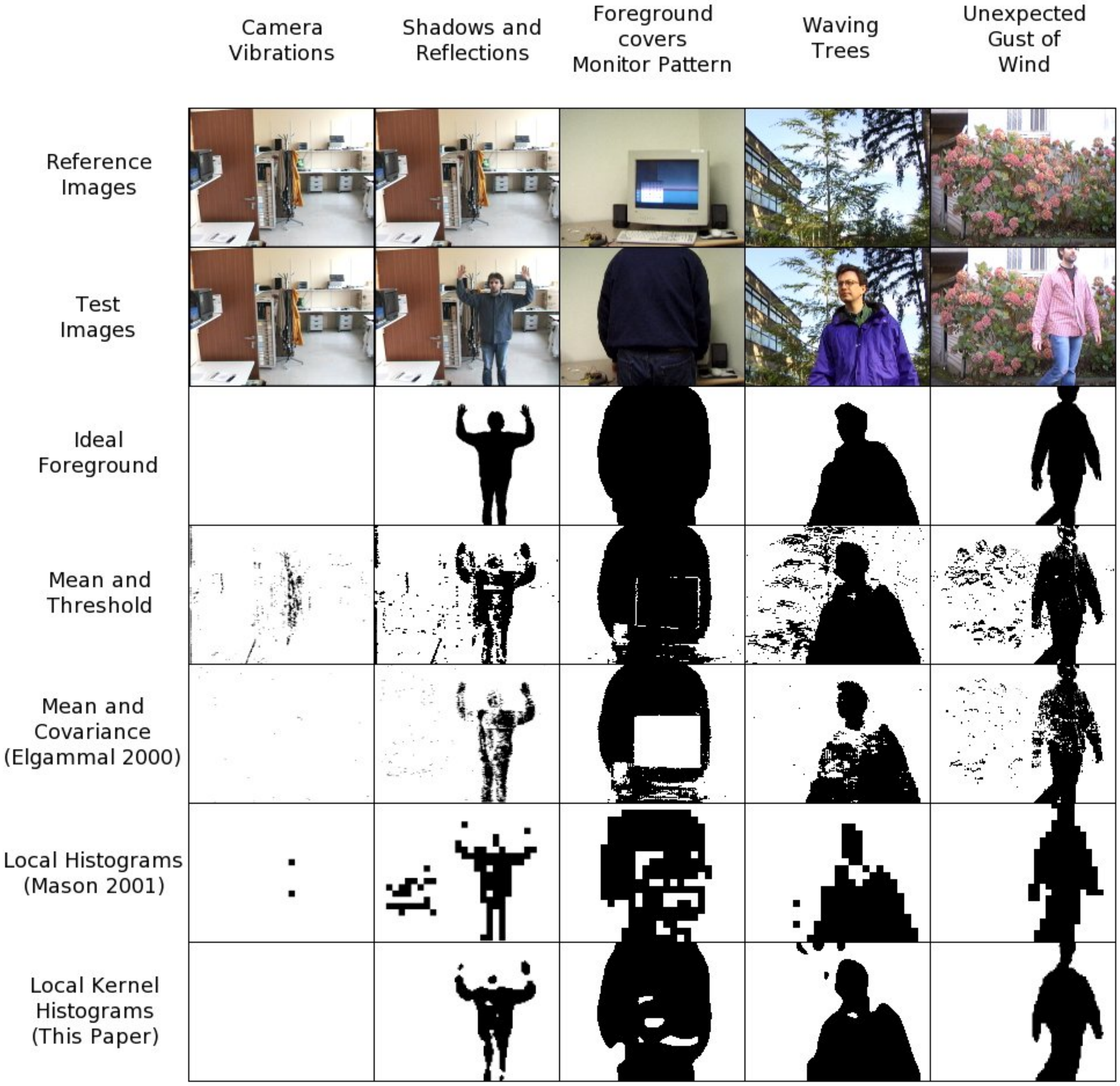

Figure 3: Comparison of color background subtraction algorithms with color local kernel histograms. The top row shows reference images used to initialize background subtraction. The second row corresponds to original images extracted from indoor and outdoor scenes. Third row represents hand segmented ground truth. Each other row shows the result for one algorithm and each column represents a conventional problem.

REFERENCES

A. Elgammal, D. H. and Davis, L. S. (2000). Non-parametric model for background subtraction. In *European Conference on Computer Vision, volume II, pages 751-767*. Springer-Verlag.

B. Han, C. Yang, R. D. and Davis, L. (2005). Bayesian filtering and integral image for visual tracking. In *Special session on Real-Time Object Tracking: Algorithms and Evaluation in Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*.

Chang, P. and Krumm, J. (1999). Object recognition with color cooccurrence histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

Crandall, D. and Luo, J. (2004). Robust color object detection using spatial-color joint probability functions. In *IEEE Computer Society Conference on Computer Vi-*

Table 1: Performance of algorithms on various images test.

		Problem Type and his Associated Frame Test					
Algorithm	Error Type	Camera Vibrations (frame 235)	Indoor (frame 235)	Foreground Covers Monitor (frame 251)	Waving Trees (frame 247)	Unexpected Gust of Wind (frame 246)	Total Errors
Mean and Threshold	false neg.	0	6351	457	104	971	
	false pos.	2787	1848	195	1905	4554	19172
Mean and Covariance	false neg.	0	6788	3273	977	2589	
	false pos.	49	603	89	116	1333	15817
Histograms	false neg.	0	4525	2455	931	1207	
	false pos.	338	4507	32	96	2359	16450
Local Kernel Histograms	false neg.	0	3247	664	195	1390	
	false pos.	0	692	146	495	1126	7955

sion and Pattern Recognition (CVPR'04) - Volume 1, pp 379-385. IEEE Computer Society.

Gargi, U. and Kasturi, R. (1999). Image database querying using a multiscale localized color representation. In *IEEE Workshop on ContentBased Access of Image and Video Libraries*. IEEE Computer Society.

H. Yamamoto, H. Iwasa, N. Y. H. T. (1999). Content-based similarity retrieval of images based on spatial color distribution. In *Int. Conf. on Image Analysis and Processing (ICIAP), pp. 951-956*. Springer.

Han, J. and Ma, K. K. (2002). Fuzzy color histogram and its use in color image retrieval. In *IEEE Transactions on Image Processing, vol. 11, no. 8, pp. 944-952*. IEEE Computer Society.

Huang, J., Kumar, S., Mitra, M., Zhu, W., and Zabih, R. (1997). Image indexing using color correlograms. In *Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec., pages 762-768*. IEEE Computer Society.

J. L. Hafner, H. S. Sawhney, W. E. M. F. W. N. (1995). Efficient color histogram indexing for quadratic form distance functions. In *IEEE Transactions. Pattern Anal. Mach. Intell. 17(7): 729-736*. IEEE Computer Society.

K. Toyama, J. Krumm, B. B. and Meyers, B. (1999). Wallflower: principles and practice of background maintenance. In *ICCV, pages 255-261*. IEEE Computer Society.

M. Mason, Z. D. (2001). Using histograms to detect and track objects in color video. In *30th AIPR Workshop. pp. 154-159*. IEEE Computer Society.

Pass, G. and Zabih, R. (1996). Histogram refinement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision*. IEEE Computer Society.