Real Time Illumination Invariant Background Subtraction Using Local Kernel Histograms

Philippe Noriega and Olivier Bernier
France Telecom Research & Development
2, av. Pierre Marzin, 22300 Lannion, France
{philippe.noriega, olivier.bernier}@francetelecom.com

Abstract

Constant background hypothesis for background subtraction algorithms is often not applicable in real environments because of shadows, reflections, or small moving objects in the background: flickering screens in indoor scenes, or waving vegetation in outdoor ones. In both indoor and outdoor scenes, the use of color cues for background segmentation is limited by illumination variations when lights are switched or weather changes. This problem can be partially allievated using robust color coordinates or background update algorithms but an important part of the color information is lost by the former solution and the latter is often too specialized to cope with most of real environment constraints. This paper presents an approach using local kernel histograms and contour-based features. Local kernel histograms have the conventional histograms advantages avoiding their inherent drawbacks. Contour based features are more robust than color features regarding scene illumination variations. The proposed algorithm performances are emphasized in the experimental results using test scenes involving strong illumination variations and non static backgrounds.

1   Introduction

Background subtraction provides important cues for numerous applications in computer vision, for example surveillance tracking [6] or human pose estimation [9]. However, background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images on screens lead to background changes. In a same way, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes.

Background variations problem can be solved by using background update algorithms [10]. However, complex background models often prevent real time processing. Another solution consists in finding robust features. For example, robust color features alleviates the problem of illumination variations [3] but they lead to a trade-off: more robust color coordinates means a less discriminative power.

Alternatively, histograms are a convenient tool in computer vision because they are easy to compute and invariant to rotation or translation. Moreover, they bring robustness

when small objects move in the background because they are robust to partial occlusions. However, histograms have some inherent drawbacks: they lack any spatial information, deal with large amounts of data and are sensitive to quantization errors.

Thus, building an algorithm able to cope with real environments is a challenging task, especially if real time is desired. As tracking applications often need real time solutions, algorithms must solve most common background problems while remaining efficient in term of computation costs. The goal of the algorithm presented in this paper is to reach a balance between robustness and computation cost. The next section presents the related works, section 3 describes the color-based local kernel histograms, section 4 explains their adaptation to contour-based features, experimental results are presented in section 5 and section 6 concludes this paper.

2   Related works

Some papers focuses on background modelization including a module to update the background. Elgammal et al. [2] use a non parametric model with a Gaussian-like camera noise model where each background pixel is compared to a recent sample of intensity values for this pixel using a Gaussian kernel function. Samples are progressively replaced in a fifo manner. Background subtraction is computed from the intersection of a short-term and a long-term model results. False detections are handled by a second detection stage that tests if a model of a positive pixel match with background models of the neighborhood pixels. This method does not cope with false detections produced by camera noise. [9] solves this problem using the level set method to skeletonize a silhouette and then, inflates the skeleton (a stick like human model) to provide a false positive free silhouette. Even with fast marching methods, this process is very time consuming. A kernel function is also used to model the background in [8]. Results are enhanced including a non parametric model of the foreground in the computation of the likelihood. A Markov network takes into account the dependency between neighborhood pixels and graph-cuts are used to segment foreground from background. Wallflower [10] uses three modules that proceed at different image scales. At pixel level, background pixel values are predicted using a Wiener filter. The region level is used to detect foreground moving regions and provide some error corrections. In the case of rapid and strong illumination changes, the frame level is used to match stored background models with the new illumination conditions to update the background. Thus, if no model matches, the system fails.

In real scenes, illumination conditions can change instantaneously when lights are switched on or off increasing the difficulty in differencing foreground from background. In this case, the use of an illuminant invariant color representation is an alternative solution to complex background models. One of the simplest invariant in a RGB image is provided by dividing each channel value by the sum of the three channels. A more sophisticated model is proposed by [4]: illumination invariant color coordinates are computed from RGB channels according to the following formulas: $l_1 = (R-G)^2/D$, $l_2 = (R-B)^2/D$ and $l_3 = (G-B)^2/D$ with $D = (R-G)^2 + (R-B)^2 + (G-B)^2$.

Other robust parameters are extracted after normalizing each mean-subtracted color channel RGB, computing the covariance matrix and using as indexing numbers the three angles formed by the inverse cosine of the covariances [3]. The main drawback of these methods is that illumination invariant color features are less informative about the image

content than original coordinates.

Another way of gaining robustness to illumination changes consists in using more features in addition to the color feature. Since the gradient value is less sensitive to light changes, gradient-based features may be a relevant hint: [1] uses histograms of oriented gradients (HOG) to classify images that contains pedestrians. [5] is an example of exploiting multiple features with a Bayesian framework that incorporates spectral, spatial, and temporal features in the background modeling. To model a static background pixel, color and Sobel gradients are used. For dynamic background pixels like flickering screen or moving tree branches, the color co-occurrence that records a pixel color through two consecutive frames is used.

Histograms are a convenient tool if their inherent drawbacks are avoided. Local kernel histograms [7] can retrieve spatial information using small histograms size for real-time processing and including smoothing features to cope with small movements and camera noise. However, coupling local kernel histograms with color feature does not solve the problem of illumination variations. The main idea of this article consists in adapting the local kernel histograms to contour-based features to gain robustness in this case.

3 Color local kernel histograms

Because of camera noise or little moving objects such as leaves in trees, pixel-wise background subtraction algorithms often leads to false detections. Local kernel histograms naturaly supress most of false positives in case of outdoor windy scenes and camera noise. They are fast and provide a pixel-wise smoothed silhouette. This section presents briefly the color local kernel histograms for background subtraction, for further details, see [7].

Local kernel histograms associated with color features consists in segmenting the image into overlapped local square areas. One histogram is computed for each area to provide accurate spatial information. To significantly reduce 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 initialization frame. A double Gaussian kernel, one in the image space and one in the color space bring robustness against noise.

Considering a squared local area of size $n \times n$ pixels S, $G_S(\mu_s, \sigma_s, S)$ and $G_C(\mu_c^j, \sigma_c, S)$ are the both Gaussian kernels, respectively on the image space (with the mean value corresponding to the area center) and the color space (the mean is the color associated with the histogram bin h_j). The value for an histogram bin h_j representing one of the n_c selected colors is:

$$h_j \atop {1 \leq j \leq n_c} = \sum_{k=1}^{n^2} G_S(\mu_s, \sigma_s, S_k) G_C(\mu_c^j, \sigma_c, S_k) \,. \tag{1}$$

An unselected color bin is added for each histogram, his value is the sum of equation (1) for all the unselected colors. The Bhattacharryya distance is used to compare the histograms computed from the initial and the actual frame to provide the probability P_l that the local area l belongs to the background. For a pixel S in the image, given the N_a
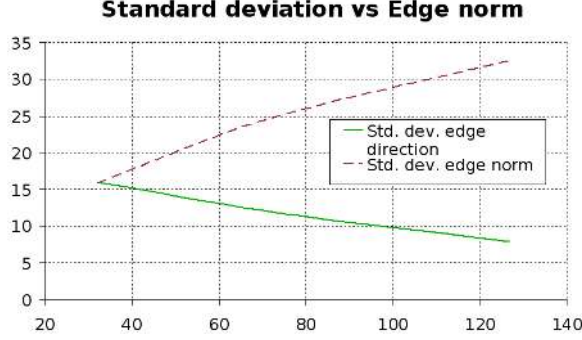
Figure 1: Standard deviation of the computed edge magnitude and orientation versus edge norm.

local areas that this pixel belongs to, the pixel-wise probability P_s is:

$$P_s = \frac{1}{\sum_{l=1}^{N_a} G_S^l(\mu_s^l, \sigma_s, S)} \sum_{l=1}^{N_a} G_S^l(\mu_s^l, \sigma_s, S) P_l \; . \tag{2}$$

4 Local kernel histograms of oriented gradient: contour-based features for local kernel histograms

When they are associated with color features, local kernel histograms cannot handle illumination changes. This section present an implementation of local kernel histograms with contour based features which are robust to illumination changes. Thus, polar histograms are computed from an image area distributing gradient vectors in their corresponding bins according to their orientation.

4.1 Histogram computation

The Shen-Castan algorithm provide for each pixel in the image, the norm ($\|\overrightarrow{E_S}\|$) and the orientation ($dir(\overrightarrow{E_S})$) of the gradient vector $\overrightarrow{E_S}$. The norm corresponds to the contour strength and the gradient direction is orthogonal to the orientation contour. The standard deviation of the gradient magnitude and orientation were measured during a short sequence of a test card representing lines with different contrasts. The results represented on figure 1 show how the gradient norm can be modeled by a Gaussian kernel $G_n(\mu_n, \sigma_n)$ with standard deviation proportional to the edge magnitude ($\sigma_n = a_n.\|\overrightarrow{E_S}\| + b_n$) and with mean centered on the norm ($\mu_n = \|\overrightarrow{E_S}\|$). In the same way, a gradient direction value $dir(\overrightarrow{E_S})$ for a pixel S is modeled by a Gaussian kernel in orientation $G_o(\mu_o, \sigma_o)$ centered on the computed orientation: ($\mu_o(\overrightarrow{E_S}) = dir(\overrightarrow{E_S})$) and a linear model for standard deviation versus the edge magnitude: ($\sigma_o(\overrightarrow{E_S}) = -a_o.\|\overrightarrow{E_S}\| + b_o$).

For a given pixel S and his computed gradient $\overrightarrow{E_S}$, each histogram bin h_o^S representing
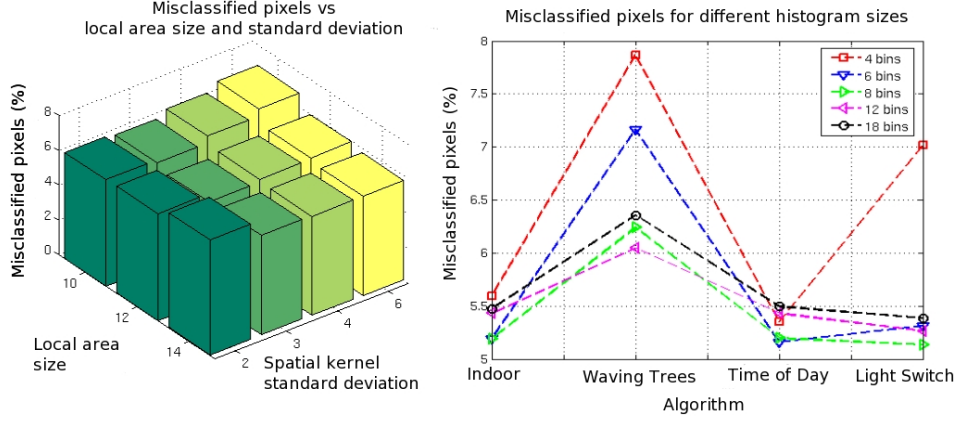
Figure 2: Left: influence of local area size and standard deviation of the Gaussian space kernel on the percentage of misclassified pixel in images. Right: percentage of misclassified pixels for different scenes and different kernel histogram of gradient size. The best performances are reached for 8 or 12 bins. As smaller histograms are computed faster, 8 is chosen.

the edge direction o is filled according the Gaussian kernel $G_o\big(\mu_o(\overrightarrow{E_S}), \sigma_o(\overrightarrow{E_S})\big)$:

$$h_o^S = G_o\big(o, \mu_o(\overrightarrow{E_S}), \sigma_o(\overrightarrow{E_S})\big) \times \|\overrightarrow{E_S}\| \tag{3}$$

To retrieve spatial information, image is segmented into overlapped squared local areas. A polar histogram is computed with all the pixel $S(x, y)$ in a local area A according to (3) and a spatial Gaussian kernel $G_s(\mu_s, \sigma_s)$ with the mean value corresponding to the area center:

$$h_o^A = \sum_{S \in A} G_o\big(o, \mu_o(\overrightarrow{E_S}), \sigma_o(\overrightarrow{E_S})\big) G_s(S, \mu_s, \sigma_s) \times \|\overrightarrow{E_S}\| \tag{4}$$

4.2   Histogram comparison

Unlike [7], histograms are not normalized and the Bhattacharyya distance cannot be used to provide a similarity measure between them. Instead, the normal cumulative distribution function of the gradient norm Gaussian kernel $F\big(G_n(\mu_n, \sigma_n)\big)$ is used to compare histogram bins. Thus, a similarity probability can be computed exploiting the results of $F(.)$ for the two compared histogram bin values.

Given H^A the histogram computed from the local area A and h_o^A, the bin representing the direction o and considering the Gaussian model adopted for the gradient norm, then the probability $P\big(h_o^A = B_o\big)$ and $P\big(h_o^A < B_o\big)$ are given by:

$$P\big(h_o^A = B_o\big) = G_n\big(B_o, h_o^A, \sigma_n(h_o^A)\big), \tag{5}$$

$$P\big(h_o^A < B_o\big) = F\Big(B_o, G_n\big(h_o^A, \sigma_n(h_o^A)\big)\Big). \tag{6}$$

The similitude between the reference and the current image histogram bins $h_o^{A_{ref}}$ and $h_o^{A_{cur}}$ is computed for the average mean coordinate point μ because this setting gives com-
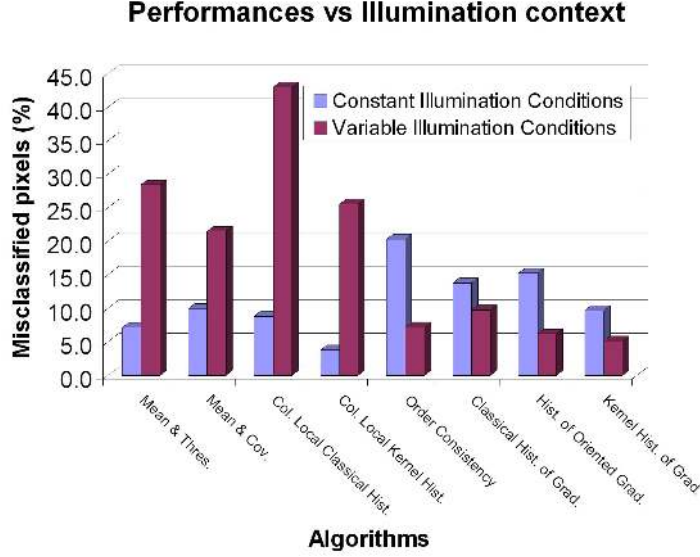
Figure 3: Performances in constant and variable illumination scenes.

plementary probabilities for equations (8) and (9) and allows a Bhattacharyya distance calculation (10):

$$\mu = \frac{h_o^{A_{ref}} + h_o^{A_{cur}}}{2} \tag{7}$$

$p_o^{A_{ref}}$ and $p_o^{A_{cur}}$ are defined as:

$$p_o^{A_{ref}} = P\left(h_o^{A_{ref}} < \mu\right) = F\left(\mu, G_n\left(h_o^{A_{ref}}, \sigma_n(h_o^{A_{ref}})\right)\right), \tag{8}$$

$$p_o^{A_{cur}} = P\left(h_o^{A_{cur}} < \mu\right) = F\left(\mu, G_n\left(h_o^{A_{cur}}, \sigma_n(h_o^{A_{cur}})\right)\right). \tag{9}$$

According to these formulas, if histogram bins are very similar $h_o^{A_{cur}} \simeq h_o^{A_{cur}}$ then $p_o^{A_{ref}} \simeq p_o^{A_{cur}} \simeq 0.5$. In the opposite case, one of the probability $p_o^{A_{ref}}$ or $p_o^{A_{cur}}$ is near to 1 and the other is near to 0. If $\overline{p}_o^{A_{cur}}$ and $\overline{p}_o^{A_{ref}}$ are the complement of $p_o^{A_{cur}}$ and $p_o^{A_{ref}}$ then the Bhattacharyya distance give the probability of histogram bin similitude p_o^A:

$$p_o^A = \sqrt{p_o^{A_{ref}} p_o^{A_{cur}}} + \sqrt{\overline{p}_o^{A_{ref}} \overline{p}_o^{A_{cur}}} \tag{10}$$

The Similitude between the whole histograms H_{ref}^A and H_{cur}^A can be seen as the probability P^A that the local area A belongs to the background in the reference image: $P^A = \prod_o p_o^A$.

5 Experimental results

Kernel histograms of gradients are compared with seven other algorithms. Color-based algorithms use chrominance channels UV from YUV color space, contour-based algorithms use only the luminance channel Y. The eight algorithms are:
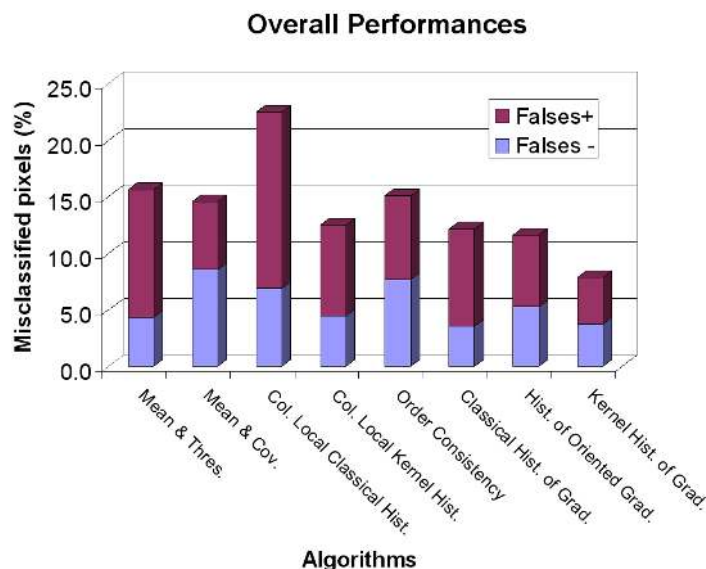
Overall Performances

Figure 4: Algorithms overall performances.

Mean & Threshold: Pixel-wise color 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 of UV color channels are computed from the recent samples values for pixels. Foreground pixels are determined using a threshold.

Color Local Classical 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. This is similar to [6].

Color Local Kernel Histograms: the image is partitioned into overlapped squared local areas with a kernel histogram computed using one Gaussian kernel in image space and one other in the color space. This is similar to [7].

Order Consistency: as pixel value ordering is preserved in local neighborhoods, the order consistency is tested between reference and actual image on the four cardinal directions from the tested pixel. Computed probability of order consistency is thresholded for each pixel to provide the output image. This is similar to [11].

Classical Histograms of Gradients: if a pixel gradient norm is greater than a threshold, it is classified on the histogram bin corresponding to its orientation. Other pixels are dropped to minimize the camera noise influence. Histograms are normalized and compared with the Bhattacharyya distance. Thresholding this last result provide the binary output image.

Histograms of Oriented Gradients (HOG): L2 Normalized local histograms of image gradient orientation are computed from a dense grid for the reference and actual images. The Bhattacharyya distance between them provide a probability map which is thresholded to classify foreground pixels from the background. This is the algorithm of [1].

Local Kernel Histograms of Oriented Gradients: The method explained in this paper, HOG are computed from Gaussian weighted overlapped local areas. Two other Gaussian

Mean Thres.	Mean Cov.	Color Hist.	Color Ker. Hist.	Order Consist.	Cl. Hist. of Grad.	HOG	Kern. Hist. of Grad.
23	23	17	16	7	18	20	16

Table 1: Frames per second for the tested algorithms.

kernels are added in the computation of the gradients to take into account the uncertainty due to camera noise.

Both indoor and outdoor test sequences are used (see figure 5). Some of them ("foreground covers monitor pattern", "waving trees" and "time of day") were used by [10] in their results. They are available from the web[1]. The five sequences show classical difficulties for background subtraction:

Soft 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.

Time of Day: The sequence shows a darkened room that gradually becomes brighter over a period of several minutes. A person walks in and sits on a couch.

Light Switch: During this test sequence, the room starts with soft lights, after a few minutes a person walks in and turns on the light.

Mean and Threshold and Mean and Covariance algorithms are trained with the first twenty images of each scene. Other algorithms are trained with a single image. In the scene Time of Day, the training phase begins at the 800 th frame.

The local area size $n \times n$ and the standard deviation σ_S which are chosen for the spatial Gaussian kernel are tuned by trying various settings and counting, for each one, the percentage of error made for the both scenes: "shadows and reflections" and "waving trees". Accounting on these experimental results (see figure 2), it appears that the best setting is $n = 12$ and $\sigma_S = 3$. Kernel histograms of gradients size is tuned during experiment to find the best setting. Figure (2) shows that 8 bins give the best performances.

Qualitative results are shown on figure 5 and quantitative ones are represented on figure 3 and 4. If local kernel histograms of gradients provide the overall best results, color local kernel histograms outperform them in constant illumination scenes but, in varying illumination conditions, they are unable to provide a useful silhouette. Considering only the scenes with illumination variations, good results are provided by the histograms of oriented gradient (HOG). This algorithm is dedicated to feed a SVM classifier that detect the presence of humans in scenes [1]. Order consistency gives a very noisy result, principally on the waving trees scene where histogram-based algorithms prove their smoothing capacity. In this case, pixel-wise algorithms performs poorly except the mean and covariance but this one gives some false negatives.

---

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

6 Conclusion

Local kernel of oriented gradient provide the overall best results proving that this algorithm is the best adapted to most purposes and circumstances tested on this paper. He is able to provide good results when small objects moves in the background as well when strong illuminations changes occur. In all the cases, Gaussian kernels improve the quantization error rate to reduce both false positives and negatives. Thus, time consuming post processing is not needed allowing effective real time background subtraction (see table 1).

References

[1] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, June 2005.

[2] Ahmed M. Elgammal, David Harwood, and Larry S. Davis. Non-parametric model for background subtraction. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 751–767, London, UK, 2000. Springer-Verlag.

[3] Graham D. Finlayson, Subho S. Chatterjee, and Brian V. Funt. Color angular indexing. In *ECCV (2)*, pages 16–27, 1996.

[4] T. Gevers and A. Smeulders. A comparative study of several color models for color image invariant retreival. In *Proc. 1st Int. Workshop on Image Databases & Multimedia Search, Amsterdam, Netherlands.*, pages 17–24, 1996.

[5] Liyuan Li, Weimin Huang, Irene Y. H. Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. In *IEEE Transactions on Image Processing*, volume 13, pages 1459–1472, 2004.

[6] Michael Mason and Zoran Duric. Using histograms to detect and track objects in color video. In *AIPR*, pages 154–162, 2001.

[7] Philippe Noriega, Benedicte Bascle, and Olivier Bernier. Local kernel color histograms for background subtraction. In INSTICC Press, editor, *VISAPP*, volume 1, pages 213–219, 2006.

[8] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, 27(11):1778–1792, November 2005.

[9] Cristian Sminchisescu and Alexandru Telea. Human pose estimation from silhouettes. a consistent approach using distance level sets. In *WSCG International Conference on Computer Graphics,Visualization and Computer Vision*, 2002.

[10] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV*, pages 255–261, 1999.

[11] Binglong Xie, Visvanathan Ramesh, and Terrance E. Boult. Sudden illumination change detection using order consistency. *Image Vision Comput.*, 22(2):117–125, 2004.
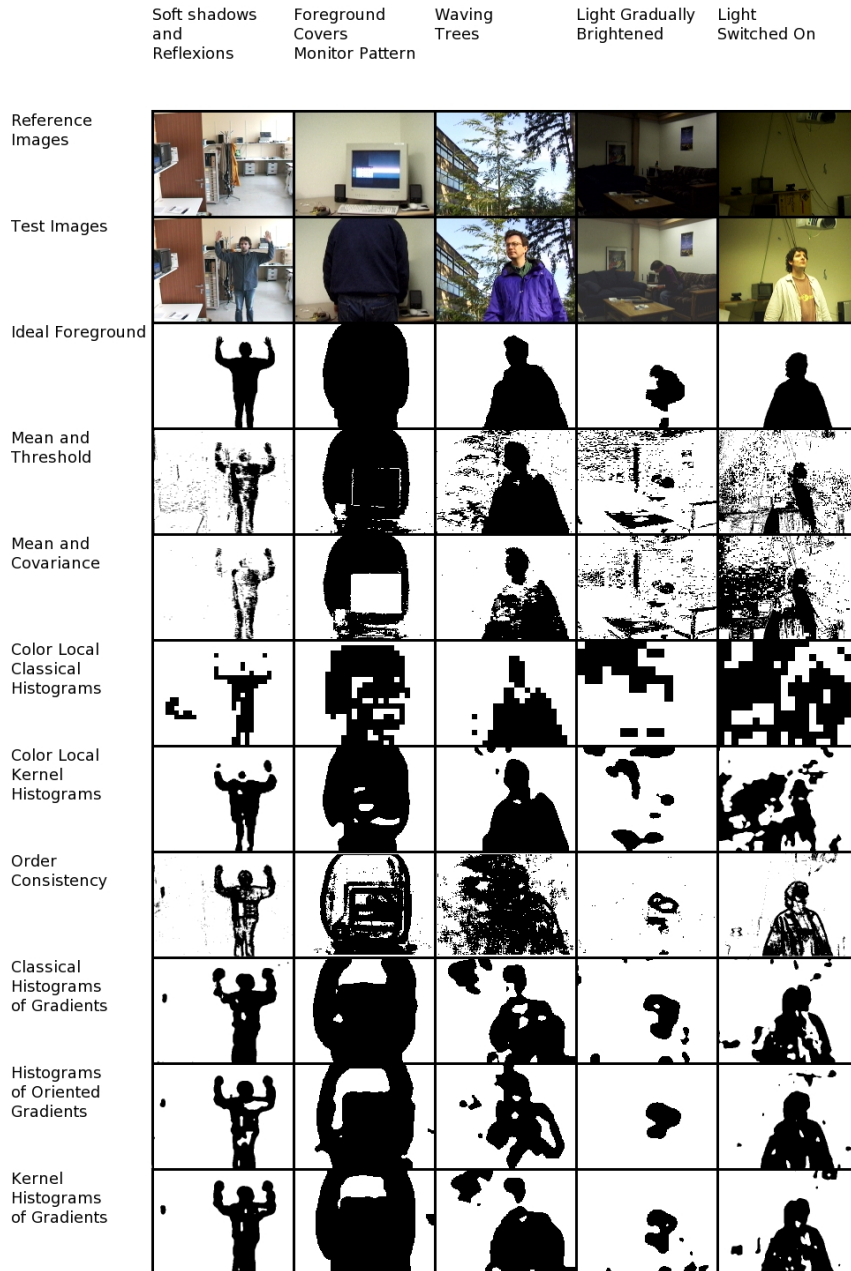
Figure 5: Comparison of background subtraction algorithms for different kind of scenes. The top row shows reference images used for initialization, the second row corresponds to test images, third row represents hand segmented ground truth, each other row shows the results for one algorithm and each column represents a conventional problem in background subtraction.