Wallflower: Principles and Practice of Background Maintenance

Kentaro Toyama, John Krumm, Barry Brumitt, Brian Meyers
Microsoft Research
Redmond, WA 98052
{kentoy|jckrumm|barry|brianme}@microsoft.com

Abstract

Background maintenance *is a frequent element of video surveillance systems. We develop* Wallflower, *a three-component system for background maintenance: the pixel-level component performs Wiener filtering to make probabilistic predictions of the expected background; the region-level component fills in homogeneous regions of foreground objects; and the frame-level component detects sudden, global changes in the image and swaps in better approximations of the background.*

We compare our system with 8 other background subtraction algorithms. Wallflower is shown to outperform previous algorithms by handling a greater set of the difficult situations that can occur.

Finally, we analyze the experimental results and propose normative principles for background maintenance.

1. Introduction

Video surveillance systems seek to automatically identify people, objects, or events of interest in different kinds of environments. Typically, these systems consist of stationary cameras directed at offices, parking lots, and so on, together with computer systems that process the images and notify human operators or other processing elements of salient events.

A common element of such surveillance systems is a module that performs background subtraction for differentiating background pixels, which should be ignored, from foreground pixels, which should be processed for identification or tracking. The difficult part of background subtraction is not the differencing itself, but the maintenance of a background model – some representation of the background and its associated statistics. We call this modeling process *background maintenance*. An ideal background maintenance system would be able to avoid the following problems:

Moved objects: A background object can be moved. These objects should not be considered part of the foreground forever after.
Time of day: Gradual illumination changes alter the appearance of the background.
Light switch: Sudden changes in illumination and other scene parameters alter the appearance of the background.
Waving trees: Backgrounds can vacillate, requiring models which can represent disjoint sets of pixel values.
Camouflage: A foreground object's pixel characteristics may be subsumed by the modeled background.
Bootstrapping: A training period absent of foreground objects is not available in some environments.
Foreground aperture: When a homogeneously colored object moves, change in the interior pixels cannot be detected. Thus, the entire object may not appear as foreground.
Sleeping person: A foreground object that becomes motionless cannot be distinguished from a background object that moves and then becomes motionless.
Waking person: When an object initially in the background moves, both it and the newly revealed parts of the background appear to change.
Shadows: Foreground objects often cast shadows which appear different from the modeled background.

No perfect system exists. In this paper, we hope to further understanding of background maintenance through a threefold contribution: In the next section, we describe *Wallflower*, a background maintenance algorithm that attempts to address many of the problems enumerated. Then in Section 3, we present a comparison of Wallflower with several well-known algorithms on a common dataset of image sequences that contain the above problems. Finally, we analyze the results of our experiments and propose several principles that should be taken into account when considering background maintenance.

2. The Wallflower Algorithm

Wallflower was intended to solve as many of the canonical problems as possible.

To handle problems that occur at various spatial scales, it processes images at the pixel, region, and frame levels. At the *pixel level*, Wallflower maintains models of the background for each pixel. Pixel-level processing makes the preliminary classifications of foreground versus background and also handles adaptation to changing backgrounds. The pixel-level avoids many of the common problems immediately: moved objects, time of day, waving trees, camouflage, and bootstrapping. All pixel-level processing happens at each pixel independently and ignores information observed at other pixels. The *region-level* considers inter-pixel relationships that might help to refine the raw classification of the pixel level; doing so avoids the foreground aperture problem. Finally, the *frame level* addresses the light switch problem: it watches for sudden changes in large parts of the image and swaps in alternate background models that explain as much of the new background as possible.

0-7695-0164-8/99 $10.00 (c) 1999 IEEE

2.1. Pixel Level

Wallflower's core consists of per-pixel models of the background. The pixel-level algorithm makes probabilistic predictions about what background pixel values are expected in the next live image using a one-step Wiener prediction filter.

The Wiener filter is a linear predictor based on a recent history of values. Any pixel that deviates significantly from its predicted value is declared foreground. For a given pixel, the linear prediction of its next value in time is

$$\underset{\sim}{s}_t = -\sum_{k=1}^{p} a_k \, s_{t-k} \ ,$$

where $\underset{\sim}{s}_t$ is the predicted value of the pixel at frame t, the s_{t-k} is a past value of the pixel, and the a_k are the prediction coefficients. The filter uses p past values to make its prediction. The expected squared prediction error, $E[e_t^2]$, is

$$E[e_t^2] = E[s_t^2] + \sum_{k=1}^{p} a_k \, E[s_t \, s_{t-k}]$$

The a_k are computed from the sample covariance values of the s_n. Details of the computations can be found in textbooks and a tutorial article [6]. In our implementation, we use the past 50 values of s_n to compute p=30 prediction coefficients. (All parameters in our system were chosen by hand and tuned on a varied set of image sequences.) If the actual value of the next pixel differs by more than $4.0 * \sqrt{E[e_t^2]}$ from its predicted value, the pixel is considered to be foreground.

The linear predictor works well for periodically changing pixels, and for random changes it produces a larger value of $E[e_t^2]$.

The main advantage of a predictive technique is that it reduces our uncertainty in a pixel's value by accounting for how it varies with time. For instance, one of the test sequences shows a CRT and the inevitable interference "bars" rolling up the screen. While the expected deviation from the mean pixel value on the CRT is 14.2 gray levels, the expected deviation from the predicted value is only 6.7. This allows a tighter threshold for background subtraction and a greater chance of avoiding the camouflage problem.

One potential problem occurs when a foreground object corrupts the history values. To solve this, we keep a history of *predicted* values for each pixel as well as the history of actual values. For each new pixel, we compute two predictions based on the actual history and the predicted history. If either prediction is within the tolerance, the pixel is considered background.

To handle adaptation, the prediction coefficients are recomputed for every new frame. We retain the new coefficients for a pixel if the corresponding expected prediction error is less than 1.1 times the previous error. This is especially helpful for bootstrapping, where the predictor locks onto to the most stationary history it has seen. The error is allowed to increase slightly (10%), to handle moved objects that might now be in a noisier range of values for our particular camera.

2.2. Region Level

We would like to segment whole objects, rather than isolated pixels, as foreground. When an object is homogeneously colored, the problem is an instance of the classic vision problem where a moving homogeneous region exhibits no perceivable motion. We consider inter-pixel relations to propel these regions into the foreground.

Moving homogeneous regions are necessarily bracketed by pixels which occur on the frontier edges of object movement, and which exhibit identical properties as those inside the region (see Figure 1a, where a homogeneously-valued disk translates to the right. Regions in black register movement). So, we cast new foreground regions that have been discovered by the pixel level as seed regions that are grown by backprojecting the pixel values that occur in the seeds. Our implementation runs as follows:

As each new pair of raw and foreground-marked images, I_t and F_t, arrives,

1. Compute image differences (Figures 1a and b):

$$J_t(\mathbf{x}) = \begin{cases} 1, & \text{if } |I_t(\mathbf{x}) - I_{t-1}(\mathbf{x})| > k_{\text{motion}}, \\ 0, & \text{otherwise.} \end{cases}$$

2. Compute the subset of pixels which occur at the intersection of adjacent pairs of differenced images [1] and the previous foreground image (Figure 1c):

$$K_t(\mathbf{x}) = J_t(\mathbf{x}) \wedge J_{t-1}(\mathbf{x}) \wedge F_{t-1}(\mathbf{x}).$$

3. Find 4-connected regions, $\mathbf{R}_i$, in K_t, discarding regions consisting of less than $k_{\min}$ pixels [2].
4. Compute H_i, the normalized histogram of each $\mathbf{R}_i$, as projected onto the image I_{t-1} (s is a pixel value):

$$H_i(s) = \frac{\left| \{ \mathbf{x} : \mathbf{x} \in \mathbf{R}_i \text{ and } I_{t-1}(\mathbf{x}) = s \} \right|}{\left| \mathbf{R}_i \right|}$$

5. Backproject histograms in I_t: For each $\mathbf{R}_i$, compute $F_{t-1} \wedge \mathbf{R}_i$, and from each point in the intersection, grow L_i, the 4-connected regions in the image,

$$L_i(\mathbf{x}) = \begin{cases} 1, & \text{if } H_i(I_t(\mathbf{x})) > \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

where we use $k_{\text{motion}} = 16$, $k_{\min} = 8$, $\varepsilon = 0.1$.

Steps 1 and 2 determine those regions that can be reliably considered foreground, and not just part of a newly revealed background. Steps 3 and 4 compute the histogram of foreground regions found. Finally, Step 5 backprojects the histogram locally (via 4-connectivity) so that some pixels considered background by the pixel-level are thrust into the foreground.

We considered using region-level processing to avoid the sleeping person and waking person problems. For example, we could collect a histogram of pixels surrounding a foreground object and backproject them onto neighboring pixels to suppress them from appearing as foreground. Informal experiments showed, however, that the harm from this suppression outweighed the marginal benefit, so we left it out of the final implementation.

(a) A homogeneous disk moves to the right. Change is visible in the black regions only (J_{t-1} in text).

(b) The same thing happens one frame later (J_t).

(c) Only the intersection ($J_{t-1} \wedge J_t$) is certain to be foreground in the middle image.

Figure 1. Finding the right foreground.

2.3. Frame Level

The frame level background maintenance mechanism utilizes multiple background models (which are simply collections of pixel-level models for each pixel in the image) to enable rapid adaptation to previously observed situations, as exemplified by the light switch problem. To pixel-level models, this change appears to be no more than the sudden appearance of foreground objects. Image changes in this situation, however, affect a majority of pixels in the scene simultaneously. Therefore, it makes sense to use a frame-wide model.

Our approach to frame-level background maintenance is to automatically retain a representative set of scene background models and then to provide an automatic mechanism for switching between them. The algorithm chooses representative background models in a training phase with a *k*-means clustering algorithm [2] applied to a set of pixel-level background models. We compute the best model by computing the number of foreground pixels that result as each model attempts to explain the input image. We then choose the model that produces the fewest number of foreground pixels. For our tests, we anticipated only two states (lights on and lights off), so we set $k = 2$.

We decide when to change models by monitoring the fraction of foreground pixels in the image. If this fraction exceeds 0.7, we trigger the consideration of a new model. We select the model that minimizes the fraction of foreground pixels in the current image.

Lastly, triggering of a frame-wide model substitution suppresses application of the region-level algorithm because during these events, too much of the image becomes included in K_t (from Step 2 of that algorithm) for region processing to be useful.

3. Experiments

We compare ten different background maintenance algorithms in situations that typify the first seven canonical problems mentioned in the introduction. (The other three problems were not considered for reasons to be explained in Section 4.) Algorithms that make an attempt to adapt to changing backgrounds are marked with an [A]:

Adjacent Frame Difference: Each image is subtracted from the previous image in the sequence. Absolute differences greater than a threshold are marked as foreground.

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

Mean & Covariance [A]: The mean and covariance of pixel values are updated continuously [5, 10]. Foreground pixels are determined using a threshold on the Mahalanobis distance. This is similar to the background algorithm used in [10].

Mixture of Gaussians [A]: A pixel-wise mixture of 3 Gaussians models the background. Each Gaussian is weighted according to the frequency with which it explains ($\pm 2\sigma$) the observed background. The most heavily weighted Gaussians that together explain over 50% of past data are considered background [3].

Normalized Block Correlation: Images are split into blocks. The pixels in each block are represented as their respective medians over the training images. Each block is represented as its median template and the standard deviation of the block-wise normalized correlation from the median over the training images. For each incoming block, normalized correlation values that deviate too much from the expected deviations cause the block to be considered foreground ([7] – a yet more sophisticated algorithm that is mentioned in this paper was not fully implemented).

Temporal Derivative: In the training phase, the minimum and maximum values of each pixel are saved along with the maximum inter-frame change in intensity at each pixel. Any pixel that deviates from its minimum or maximum by more than the maximum inter-frame change is considered foreground [4]. We additionally enforced a minimum inter-frame difference of 10 pixels after the regular training phase.

Bayesian Decision [A]: Pixel value probability densities, represented as normalized histograms, are accumulated over time, and backgrounds are determined by a straightforward maximum *a posteriori* criterion [8].

Eigenbackground: Images of motionless backgrounds are collected. Principle component analysis is used to determine means and variances over the entire sequence (whole images as vectors). Incoming images are projected onto the PCA subspace. Differences between the projection and the current image greater than a threshold are considered foreground [9].

Linear Prediction [A]: This is the algorithm described in Section 2.1.

Wallflower [A]: This is a combination of the pixel, region and frame level algorithms as described in Section 2.

All of the test sequences were taken with a 3-CCD camera recording to digital tape. We stored the sequences at a size of 160x120 pixels, sampled at 4Hz. For algorithms that adapt, adaptation rates were set so that absorption into

Algorithm	Error Type	Problem Type: moved object	time of day	light switch	waving trees	camou-flage	bootstrap	foreground aperture	Total Errors
Frame difference	false neg.	0	1165	2479	3509	9900	1881	3884	
	false pos.	0	193	86	3280	170	294	470	22957
Mean + threshold	false neg.	0	873	1116	17	194	415	2210	
	false pos.	0	1720	15116	3268	1638	2821	608	27178
Mean + covariance	false neg.	0	949	1857	3110	4101	2215	3464	
	false pos.	0	535	15123	357	2040	92	1290	30379
Mixture of Gaussians	false neg.	0	1008	1633	1323	398	1874	2442	
	false pos.	0	20	14169	341	3098	217	530	24081
Block correlation	false neg.	0	1030	883	3323	6103	2638	1172	
	false pos.	1200	135	2919	448	567	35	1230	19281
Temporal derivative	false neg.	0	1151	752	2483	1965	2428	2049	
	false pos.	1563	11842	15331	259	3266	217	2861	41257
Bayesian decision	false neg.	0	1018	2380	629	1538	2143	2511	
	false pos.	0	562	13439	334	2130	2764	1974	26937
Eigen-background	false neg.	0	879	962	1027	350	304	2441	
	false pos.	1065	16	362	2057	1548	6129	537	14699
Linear prediction	false neg.	0	961	1585	931	1119	2025	2419	
	false pos.	0	25	13576	933	2439	365	649	23959
Wallflower	false neg.	0	961	947	877	229	2025	320	
	false pos.	0	25	375	1999	2706	365	649	10509

Figure 1. Performance of algorithms on test sequences.

the background would be complete after 50 frames from the moment that a background ceases to change. Some algorithms were designed for intensity images only – for

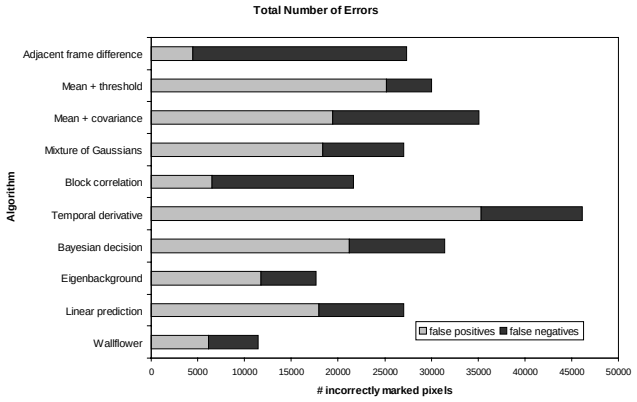

Figure 3. Overall performance.

these, we made the simplest generalization to RGB values by concatenating all three images and using the L_1 norm for detecting differences. On every output image, we performed speckle removal to eliminate islands of 4-connected foreground pixels consisting of less than 8 pixels. For all other parameters, we experimented with different settings of the adjustable parameters of each algorithm until the results seemed optimal over the entire dataset. We did not change parameters between sequences.

We took one sequence of images to represent each of the first seven problems listed in the introduction. Each test sequence begins with at least 200 background frames for training the algorithms, except for the bootstrap sequence (the first 200 frames of the bootstrap sequence were nevertheless used for training algorithms that require a training phase). Because tracking of people is a common application of background maintenance, the goal in these tests is to mark people, and only people, as foreground objects. We have deliberately excluded objects such as cars, which might be considered foreground in some applications, from the sequences. The sequences are:

Moved Object: A person walks into a conference room, makes a telephone call, and leaves with the phone and a chair in a different position. We evaluate the background image 50 frames after the person leaves the scene, giving those algorithms that adapt time to do so.

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: First, there is a training period in which the camera sees a room with the lights both on and off. Then, during the test sequence, the room starts with the lights off. After a few minutes a person walks in, turns on the light, and moves a chair. We stop the algorithms shortly after the light comes on, leaving the various pixel-level adaptation mechanisms inadequate time to adapt. In addition to the person, the recently moved chair is considered foreground.

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

Camouflage: A monitor sits on a desk with rolling interference bars. A person walks into the scene and occludes the monitor.

Bootstrapping: The sequence consists of several minutes of an overhead view of a cafeteria. There is constant motion, and every frame contains people.

Foreground Aperture: A person is asleep at his desk, viewed from the back. He wakes up and slowly begins to move. His shirt is uniformly colored.

Partially in consideration of space constraints, we did not consider three problems mentioned in the introduction. We left out the sleeping person and waking person problems because none of the algorithms handle them (the

following section discusses this point further); therefore, we did not expect them to contribute any interesting comparative information. Next, although shadows classified as foreground are a problem on images based on brightness, we felt the problem was not fundamental to background maintenance itself, but rather to the pixel type. Shadows do not cause as significant a problem for range images, for example.

The results of the tests are shown in Figure 4. We evaluated each result numerically in terms of the number of false negatives (the number of foreground pixels that were missed) and false positives (the number of background pixels that were marked as foreground). "Ground truth" foreground pixels were marked by hand. Tallies appear in Figure 2 and are charted in Figure 3, but they should be taken with a grain of salt: The absolute values of the numbers in Figure 2 are meaningless on their own, and the final count in Figure 3 is not intended to be a definitive ranking of the algorithms. Such a ranking is necessarily task-, sequence-, and application dependent.

4. Analysis and Principles

Background maintenance seems simple at first. But it turns out to be a problem rich with hard cases and subtle tradeoffs. This section attempts to crystallize the issues by proposing a set of principles to which background maintenance modules should adhere.

Principle 1: No Semantics

On one hand, only Wallflower succeeds on the foreground aperture test, where movement of the interior of subject's shirt is undetectable. The region-filling algorithm fills most of the area that the other algorithms miss.

On the other hand, Wallflower outputs false positives in the waving trees sequence, where part of the sky, considered background by the pixel-level, becomes foreground after region-level processing. The region-level algorithm is therefore an unsound heuristic, the use of which is not justified in general because it is an attempt to extract object semantics from low-level vision:

> Semantic differentiation of objects should not be handled by the background maintenance module.

Background subtraction is never an end in itself. Larger systems seeking a high-level understanding of image sequences use it as a component. Put another way, a background maintenance module handles the default model for everything in a scene that is not modeled explicitly by other processing modules. Thus, the module performing background maintenance should not attempt to extract the semantics of foreground objects on its own.

This restriction implies that there are some problems that are inappropriate for background maintenance systems to attempt to solve. In particular, attempts to solve the sleeping person, waking person, and foreground aperture problems are bound to be unsound, given the low-level nature of background maintenance.

A corollary to this principle is that while background maintenance might be useful in determining gross traffic statistics of objects such as people and cars, which can alternately be moving or motionless, attempts to use it alone as a preprocessing step for continuous, *accurate* tracking are bound to fail – some foreground objects will either be incorrectly adapted into the background or moved background objects will remain forever in the foreground.

We note that in some systems, the higher-level module provides feedback to background maintenance about what pixels should not be adapted into the background. In one person-tracking system, a high-level module tracks a single user and prevents pixels representing the user from becoming background [10]. This is different from attempting to segment whole people, as such, in the background module itself; the background module merely accepts additional information from above but does not generate it on its own.

Principle 2: Proper Initial Segmentation

As long as there is to be a background maintenance module, we must differentiate the task of *finding* foreground objects from the task of *understanding* whether or not they are of interest to the system.

> Background subtraction should segment objects of interest when they first appear (or reappear) in a scene.

Instead of semantic understanding, a more realistic goal for background subtraction is to pick out objects of interest that can then be recognized, tracked, or ignored by higher-level processing. In environments where adaptation is necessary (see Principle 4), maintaining foreground objects as foreground is not a reasonable task for background modelers, since such accurate maintenance requires semantic understanding of foreground. We can evaluate background maintenance by how closely it comes to finding all foreground pixels (as defined by the end task) when a foreground object *first* appears in the scene, while simultaneously ignoring all others. Object recognition and tracking can be computationally expensive tasks – good background subtraction eliminates the need to perform these tasks for each frame, on every subregion.

Principle 3: Stationarity Criteria

The principle of proper initial segmentation depends on the notion of object salience. How, in general, should one determine what is of interest in video sequences? Objects are salient when they deviate from some invariant property of the background. The key question, then, is how this invariance is modeled and what it means to deviate from it.

Backgrounds, for instance, are not necessarily defined by absence of motion. Consider a fluttering leaf on a tree. As the leaf moves on and off a pixel, that pixel's value will change radically. No unimodal distribution of pixel values can adequately capture such a background, because these models implicitly assume that the background, apart from some minimal amount of noise, is static. Indeed, all

unimodal models failed to capture the complexity in the background required to handle the waving trees and camouflage experiments.

Those background systems which can model multimodal distributions of pixels, or which attempt to make predictions about the expected background fared well, however. These algorithms succeed because they are founded on a stronger notion of when a pixel deviates from the background.

> An appropriate pixel-level stationarity criterion should be defined. Pixels that satisfy this criterion are declared background and ignored.

We define *stationarity* as that quality of the background that a particular model assumes to be approximately constant. (We note that this is different, and more ambiguous, than the formal definition of stationarity from random signal processing.) In order to understand the strengths and weaknesses of a particular algorithm in a specified environment, it is crucial that this stationarity criterion be made explicit. Carelessness in defining the stationarity criterion can lead to the waving trees and camouflage problems.

Principle 4: Adaptation

Backgrounds often change, even with liberal definitions of stationarity. For instance, all of the adaptive background maintenance systems are able to handle the moved object sequence by absorbing the chair into the background when it regains stationarity. Ergo,

> The background model must adapt to both sudden and gradual changes in the background.

This is an obvious requirement, but it makes sense only under Principles 1 and 2.

The line between gradual and non-gradual changes should be chosen to maximize the distinction between events that cause them. The surprisingly good performance of frame differencing suggests that fast adaptation works quite well if the foreground consists of moving people. Ultimately, however, the choice of this line is arbitrary. Some parts of a scene may remain in the foreground unnecessarily long if adaptation is slow, but other parts will disappear too rapidly into the background if adaptation is fast. Neither approach is inherently better than the other – a point that emphasizes the inadequacy of background maintenance for all but the initialization of tracking. This point is also consistent with Principle 2, which does not distinguish between algorithms that adapt quickly or slowly, as long as new foreground objects will still appear in the foreground.

Principle 5: Multiple Spatial Levels

Sudden light changes were best handled by normalized block correlation, eigenbackground, and Wallflower. On the other hand, neither eigenbackgrounds nor block correlation deals with the moved object problem or the bootstrapping problem, because they lack adaptive pixel-level models. So, our final principle is the following:

> Background models should take into account changes at differing spatial scales.

Most background maintenance algorithms maintain either pixel-wise models or whole-frame models, but not both. Pixel-level models are necessary to solve many of the most common background maintenance problems, while the light switch and time-of-day problems suggests that frame-wide models are useful, as well. A good background maintenance system is likely to explicitly model changes that happen at different spatial scales. Much of Wallflower's success is attributable to its separate models for pixels and frames.

5. Conclusion

Background maintenance, though frequently used for video surveillance applications, is often implemented *ad hoc* with little thought given to the formulation of realistic, yet useful goals. We presented Wallflower, a system that attempts to solve many of the common problems with background maintenance. Comparison of Wallflower with other algorithms establishes a case for five principles that we proposed based on analysis of the experiments.

References

1. Cutler, R. and M. Turk. "View-Based Interpretation of Real-Time Optical Flow for Gesture Recognition." in *Int'l Conf. on Automatic Face and Gesture Recog.* 1998. Nara, Japan.
2. Duda, R.O. and P.E. Hart, *Pattern Classification and Scene Analysis*. 1973: Wiley.
3. Grimson, W.E.L., *et al.* "Using Adaptive Tracking to Classify and Monitor Activities in a Site." in *IEEE Conference on Computer Vision and Pattern Recognition*. 1998. Santa Barbara, CA: IEEE Computer Society.
4. Haritaoglu, I., D. Harwood, and L.S. Davis. "W4s: A Real-Time System for Detecting and Tracking People in 2 1/2 D." in *5th European Conference on Computer Vision*. 1998. Freiburg, Germany: Springer.
5. Koller, D., J.W. Weber, and J. Malik. "Robust Multiple Car Tracking with Occlusion Reasoning." in *European Conf. on Computer Vision*. 1994.
6. Makhoul, J., "Linear Prediction: A Tutorial Review." in *Proceedings of the IEEE*, 1975. **63**(4): p. 561-580.
7. Matsuyama, T., T. Ohya, and H. Habe, *Background Subtraction for Non-Stationary Scenes*, . 1999, Department of Electronics and Communications, Graduate School of Engineering, Kyoto University: Sakyo, Kyoto, Japan.
8. Nakai, H. "Non-Parameterized Bayes Decision Method for Moving Object Detection." in *Asian Conf. Computer Vision*. 1995. Singapore.
9. Oliver, N., B. Rosario, and A. Pentland. "A Bayesian Computer Vision System for Modeling Human Interactions." in *Int'l Conf. on Vision Systems*. 1999. Gran Canaria, Spain: Springer.
10. Wren, C.R., *et al.*, "Pfinder: Real-Time Tracking of the Human Body." in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. **19**(7): p. 780-785.

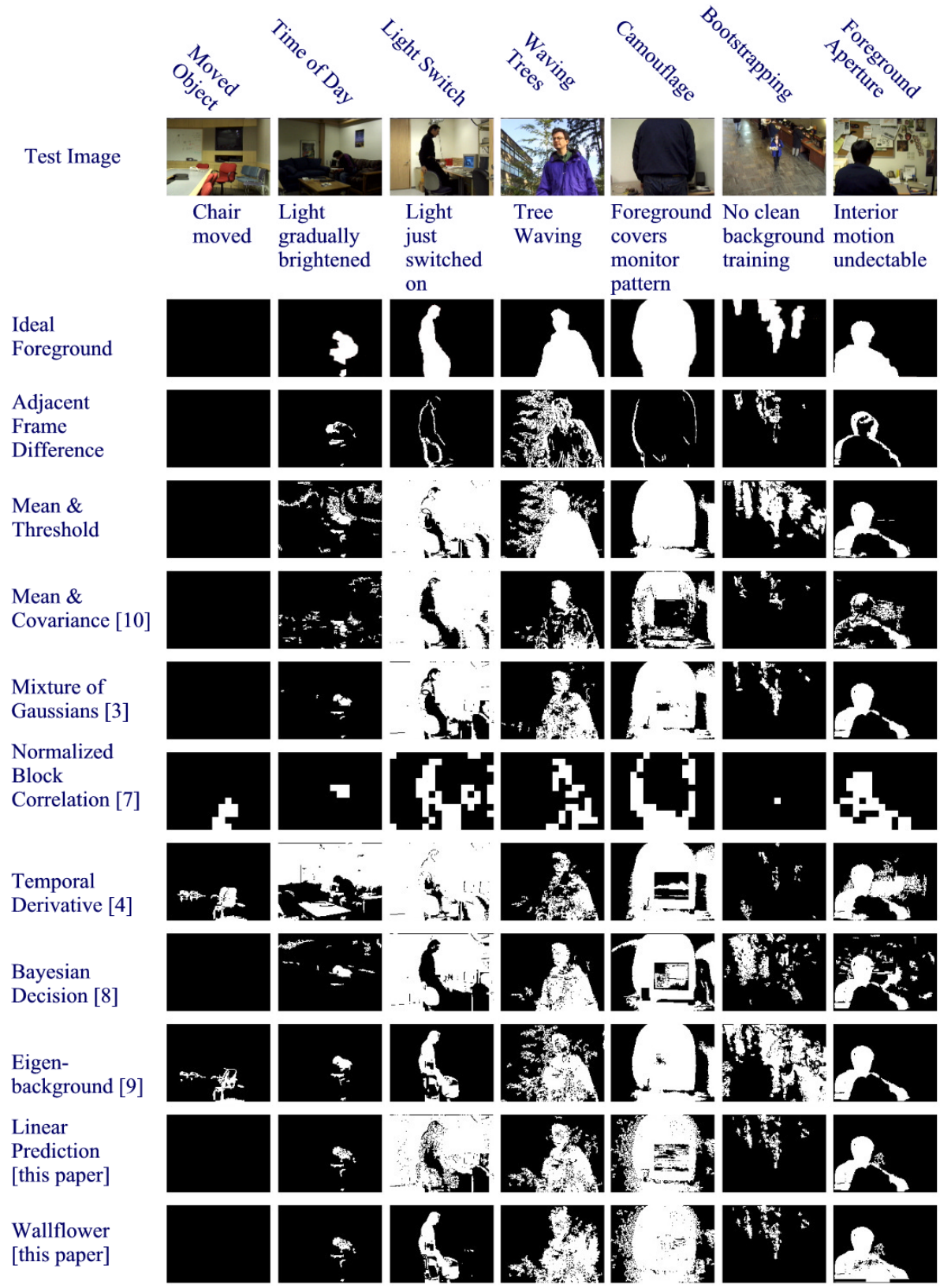

Figure 4. Tests of ten background maintenance algorithms on the seven canonical background problems. Each row shows the results of one algorithm, and each column represents one canonical problem. The top row shows the image in the sequence at which the processing was stopped. The second row has hand-segmented images of the foreground used as "ground truth" for our quantitative comparison.