Background Initialization in Cluttered Sequences

Andrea Colombari, Andrea Fusiello, and Vittorio Murino
Dipartimento di Informatica - Università degli Studi di Verona
Strada Le Grazie, 15 - 37134 - Verona
colombari@sci.univr.it, {andrea.fusiello, vittorio.murino}@univr.it

Abstract

In this paper we propose a technique to robustly estimate the background in a cluttered sequence, i.e., a sequence where occluding objects persist in the same position for a considerable portion of time. As pixel-level heuristic are not sufficient in this case, we introduce spatial support. First the sequence is subdivided in patches that are clustered along the time-line in order to narrow down the number of background candidates. Then the background is grown incrementally by selecting at each step the best continuation of the current background, according to the principles of visual grouping. The method rests on sound principles in all its stages, and only few, intelligible parameters are needed. Experiments with real sequences illustrate the approach.

1. Introduction

Segmenting moving objects from a static background is a relevant issue in areas such as video surveillance, perceptual interfaces, and content-based video encoding (MPEG4). Foreground objects can be extracted effectively by subtracting the background in the image frames, provided that the background can be estimated. The problem – also called *background initialization* – is defined as follows: Given a video sequence taken with a stationary camera, in which any number of moving occlusors (clutter) can be present, output a single image of the scene *without* clutter, even if such an image have never been captured.

In the most fortuitous cases, clutter has the property to insists on each pixel location for less than 50% of the entire sequence length. In this case background is obtained as the median of each pixel color distribution. Other techniques [16, 4, 19] have been proposed which, like the median, operate at pixel-level, making decisions independently for each pixel. The Adaptive Smoothness Method [10], for example, finds intervals of stable intensity in the sequence. Then, using some heuristics, the longest stable value for each pixel is selected and used as the value that most likely represents the background. However, pixel-level data can be useful in narrowing the number of possible candidate values for the background, but, if clutter is stationary for a long period of time, these techniques fail.

Spatial support must be taken into consideration as an additional heuristics in order to overcome this problem [12, 7, 9]. The Local Image Flow algorithm [2], for instance, considers also information generated by the neighboring locations, namely the local optical flow. Background values hypotheses are generated by locating intervals of relatively constant intensity, which are weighted with local motion information. This technique, however, cannot cope with occlusors that move only in few frames, or equivalently, with the problem of estimating the background from two images only.

Our approach is able to cope with sequences where occlusors persist in the same position for a considerable portion of time. First the sequence is subdivided in patches that are clustered along the time-line in order to narrow down the number of background candidates. Then the background is grown incrementally by selecting at each step the best continuation of the current background. Spatial continuity is enforced through the principles of visual grouping [17].

Related works can be found in the areas of *video inpainting* [18, 13, 8] where the problem is to repair holes in a video sequence with plausible values. Background initialization could be cast as video inpainting if the occlusion masks were known beforehand, which do not makes sense in our case. Moreover we seek to estimate a physically valid view of the background, by choosing pixel values only along the same time-line, whereas this is not usually a constraint in video inpainting.

The closest works to ours are [1] and [14], that deals with background initialization and mosaic completion respectively. They are based on the same scheme: (i) identify an initial region which is sure to be background and then (ii) fill-in the remaining unknown background incrementally by choosing values from the same time-line. At each step, the patch that maximizes a likelihood measure with respect to the surrounding zone, already identified as background, is selected. This entails that the the background should be self-similar (like a building's facade) and that the starting region should be large enough to provide sufficient information. On the contrary, this need not be assumed in our algorithm.

2. Method

Consider a video sequence taken with a stationary camera: Starting from a single pixel in one frame, a temporal line (or *time-line*) piercing all the aligned frames will intersect pixels that correspond to the background and pixels belonging to occlusors. Our method is based on the following hypothesis (as in [2]):

i) the background is stationary;

ii) along each time-line the background is revealed at least once.

The first hypothesis implies that the same background point is imaged always onto the same pixel. The second hypothesis implies that no object can occlude the background for the entire sequence. Please note that this is necessary as we want to use only *observed* values to fill the background at each location.

If hypothesis ii) were stronger, requiring that along each time-line the background is revealed for more than 50% of the entire sequence length, the background could be easily obtained as the median value along the time-line. The technique presented here can deal, in principle, with sequences where background is revealed exactly once.

We model the stabilized video sequence as a 3D array $\mathbf{v}_{x,y,t}$ of pixel values. Each entry contains a color value, which is a triplet (R,G,B). A 3D *patch* $\mathbf{v}_\mathcal{S}$ is a sub-array of the video sequence, defined in terms of the ordered set of its pixel coordinates: $\mathcal{S} = I_x \times I_y \times I_t$, where I_x, I_y, I_t are set of indexes. The set $\mathcal{W} = I_x \times I_y$ is the *spatial footprint* of the patch. A 2D patch $\mathbf{v}_\mathcal{R}$ (or *image patch*) is a 3D patch with a singleton temporal index: $\mathcal{R} = \mathcal{W} \times \{t\}$ or $\mathcal{R} = (\mathcal{W}, t)$.

2.1. Estimating image noise.

The first step is to estimate the noise affecting pixel values in the video sequence. In the following we shall assume that the three color channels are statistically independent, therefore we will consider here one color channel at a time.

Assuming that noise is i.i.d. Gaussian with zero-mean $\mathcal{N}(0, \sigma_m^2)$, the pixel values of the video sequence of length $L-1$ obtained by subtracting each frame from the consecutive one: $\mathbf{n}_{x,y,t} = \mathbf{v}_{x,y,t+1} - \mathbf{v}_{x,y,t}$ are distributed with $\mathcal{N}(0, 2\sigma_m^2)$ plus outliers due to occlusions. The noise standard deviation σ_m is then robustly estimated from $\mathbf{n}_{x,y,t}$. In order to get more statistics, we consider not only the difference between consecutive frames but also frames of distance two and three.

A robust estimator of the spread of a distribution is given by the Median Absolute Difference (MAD):

$$\mathrm{MAD} = \mathrm{med}_i\{|\mathbf{n}_i - \mathrm{med}_i\{\mathbf{n}_i\}|\}. \quad (1)$$

It can be seen [3] that, for symmetric distributions, the MAD coincides with the inter-quartile range, hence, in our case:

$$\mathrm{MAD} = \frac{1}{2}\left(\Phi^{-1}(\tfrac{3}{4}) - \Phi^{-1}(\tfrac{1}{4})\right)\sqrt{2}\sigma_m = \Phi^{-1}(\tfrac{3}{4})\sqrt{2}\sigma_m \approx 0.9539\sigma_m. \quad (2)$$

where $\Phi^{-1}(\alpha)$ is the α-th quantile of the cumulative normal distribution.

2.2. Temporal clustering

The spatial indexes are subdivided into windows $\mathcal{W}_i$ of size $N \times N$, overlapping by half of their size in both dimensions as shown in Fig. 1. Let $\mathbf{v}_\mathcal{S}$, $\mathcal{S} = \mathcal{W}_i \times \{1 \cdots L\}$, be a patch of footprint $\mathcal{W}_i$ which extends in time from the first to the last frame. In order to reduce temporal redundancy, in each 3D patch $\mathcal{S}$ we cluster image patches that depict the same portion of the scene with *single linkage agglomerative clustering* [6]. In agglomerative clustering, starting from all singletons, each sweep combines two clusters into a single cluster. After establishing a distance between objects, a method needs to be chosen to determine which two groups should be linked. The *simple linkage* rule says that the two groups that achieve the smallest inter-group distance between any pair of objects are linked. A *cutoff distance*, i.e., a distance behind which two clusters are not linked, can be set.

In our case, the distance between two image patches $\mathbf{v}_{(\mathcal{W},t_1)}$ and $\mathbf{v}_{(\mathcal{W},t_2)}$ is given by the Sum of Squared Distanced (SSD):

$$\mathrm{SSD}(\mathcal{W}, t_1, t_2) = \frac{1}{2\sigma_m^2} \sum_{x,y \in \mathcal{W}} ||\mathbf{v}_{x,y,t_1} - \mathbf{v}_{x,y,t_2}||^2 \quad (3)$$

The cutoff distance should prevent clustering together image patches that do not depict the same objects. It is obtained from a statistical test, based on the expected distribution of the SSD between two image patches that depict the same *static* portion of the scene. The SSD has a Chi-square distribution with $3N^2$ degrees of freedom, which is evident if we re-write (3) as a *Mahalanobis* distance:

$$\mathrm{SSD}(\mathcal{W}, t_1, t_2) = (\bar{\mathbf{v}}_{\mathcal{W},t_1} - \bar{\mathbf{v}}_{\mathcal{W},t_2})^\top (2\sigma_m^2 \mathbf{I})^{-1} (\bar{\mathbf{v}}_{\mathcal{W},t_1} - \bar{\mathbf{v}}_{\mathcal{W},t_2}) \quad (4)$$

where $\bar{\mathbf{v}}_{\mathcal{W},t}$ is the $3N^2$-dimensional vector obtained by "vectorizing" $\mathbf{v}_{\mathcal{W},t}$ (because $N^2 = |\mathcal{W}|$, and 3 is the number of color channels).

Therefore, given a desired confidence level α, we deem that image patches $\mathbf{v}_{\mathcal{W},t_1}$ and $\mathbf{v}_{\mathcal{W},t_2}$ depict the same static portion of the scene (hence they can be linked in the clustering) if:

$$\mathrm{SSD}(\mathcal{W}, t_1, t_2) < \chi^{-1}_{3N^2}(\alpha) \tag{5}$$

where $\chi^{-1}_n(\alpha)$ is α-th quantile of the cumulative Chi-square distribution with n d.o.f.

Although clusters are made of image patches instead of pixels, the clustering phase implements the same idea as the *intervals of stable intensity* defined in [10], except for clusters need not form a connected temporal interval, and there are no fancy thresholds.

The resulting clusters are 3D patches, with possibly not consecutive temporal indexes. Let $\mathcal{W} \times \mathcal{T}_k$ denote cluster k over spatial footprint $\mathcal{W}$, a representative image patch for that cluster is obtained by averaging pixel values along the time-line:

$$\mathbf{u}_{x,y,k} = \frac{1}{|\mathcal{T}_k|} \sum_{t \in \mathcal{T}_k} \mathbf{v}_{x,y,t} \quad \forall x, y \in \mathcal{W}. \tag{6}$$

As a consequence, the noise affecting the values $\mathbf{u}_{x,y,k}$ is i.i.d. $\mathcal{N}(0, \sigma_k^2)$ with $\sigma_k^2 = \frac{\sigma_m^2}{|\mathcal{T}_k|}$.

In each spatial footprint $\mathcal{W}$ we have now a variable number of cluster representatives $\mathbf{u}_{x,y,k_1} \ldots \mathbf{u}_{x,y,k_\ell}$. The underlying assumption is that (at least) one of them depicts *only* static background: The subsequent stage is devoted to find out which one.

A heuristic that demonstrated helpful to cull the clusters is discarding clusters of size one (i.e., composed by only one frame) provided that this do not eliminate *all* the clusters insisting on a footprint. This is related to the practice of discarding patches with high *motion energy* [14, 2], computed with optical flow or temporal differencing. In our case, as the SSD is related to the motion energy, image patches with high motion energy tends to form clusters of size one.

By introducing this heuristic, we implicitly relax our hypothesis, requiring that along each time-line the background is revealed at least *twice*. However, the method can still cope also with the extreme case of recovering the background from two images only, thanks to the fact that when the clusters insisting on a footprint are *all* singletons, they are kept.

Clustering pixels according to their color along a time-line is analogous to learning a multimodal distribution on a per-pixel basis [16] or finding intervals of stable intensity [2]. In this work, spatial support is introduced by considering rectangular regions instead of single pixels, and the role of clustering is only to reduce the number of background candidates. A decision will be taken based on spatial considerations (Sec. 2.3).

2.3. Background tessellation

The background is constructed following a sequential approach: Starting from seed patches, a tessellation is grown by choosing, at each site, the best continuation of the current background.

The background seeds are the representatives of the largest clusters. Since we assume that no occlusor is stationary in *all* the frames, if the largest clusters have size L (maximal), the seeds are fully reliable. Otherwise, mistakes are possible.

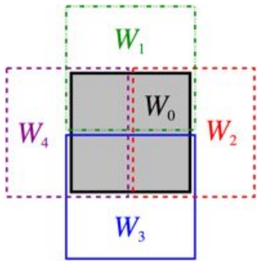

Figure 1. Overlapping footprints.

The growing proceeds as follows. Let $\mathcal{W}_0$ be a spatial footprint where a background patch has already been assigned. We consider in turn each of the four footprints that overlap with it: $\mathcal{W}_i, i = 1, \ldots, 4$ (see Fig. 1) and try to assign a background to each of them (if it was not already assigned) by choosing one of the cluster representatives that insist on $\mathcal{W}_i$. The selected patch has to fulfill two requirements:

i) in the part that overlaps with $\mathcal{W}_0$ it has to depict the same scene points as the background patch, so that it can be stitched seamlessly to it;

ii) in the non-overlapping part it has to represent the "best continuation" of the background.

This procedure is repeated for all the footprints, until all the background has been assigned.

As for the first requirement, the discrepancy of a candidate image patch $\mathbf{u}_{(\mathcal{W}_i,k)}$ with the background patch $\mathbf{u}_{(\mathcal{W}_0,k_0)}$ in the overlapping part is measured with:

$$\mathrm{SSD}(\mathcal{W}_0 \cap \mathcal{W}_i, k_0, k) = \frac{1}{\sigma_{k_0}^2 + \sigma_k^2} \sum_{x,y \in \mathcal{W}_0 \cap \mathcal{W}_i} ||\mathbf{u}_{x,y,k_0} - \mathbf{u}_{x,y,k}||^2. \tag{7}$$

where $M = |\mathcal{W}_0 \cap \mathcal{W}_i|$. By the same token as before (Eq. (5)), $\mathbf{u}_{(\mathcal{W}_i,k)}$ is considered for inclusion in the background with confidence α if

$$\mathrm{SSD}(\mathcal{W}_0 \cap \mathcal{W}_i, k_0, k) < \chi^{-1}_{3M}(\alpha) \tag{8}$$

If $\mathcal{W}_i$ happens to overlap with other footprints than $\mathcal{W}_0$ where the background has already been assigned, the same test is applied, *mutatis mutandi*, to the entire area of overlap.

As for the second requirement, we propose here a method to compare two candidates (if there are more candidates a round robin tournament is used), based on the principles of *visual grouping* [17]. The approach rests on the observation that occlusors generally introduce a discontinuity with the background (as in [5]). When a pure background patch is compared to an image patch containing clutter, their binarized difference defines a partitioning of the pixels into two groups (Fig. 2), i.e. a segmentation. The previous observation implies that the score of this segmentation according to the principles of visual grouping (similarity, proximity, and good continuation) has to be higher in the cluttered patch than in the one containing background. This links the problem of selecting the best continuation of the background to visual grouping.

Figure 2. From left to right: Two cluster representatives that are candidates to fill a background patch and their binarized difference.

Graphs cuts have been proposed in [15] as general computational framework for grouping. The image is represented as a complete weighted undirected graph $G = (V, E)$, by taking each pixel as a node and connecting each pair of pixels by an edge. The weight on that edge reflects the likelihood that the two pixels belong to the same region. Grouping is cast as the problem of partitioning the vertices into disjoint sets, where the similarity among the vertices in a set is high and across different sets is low. The edge weight connecting two nodes i and j is defined as [15]:

$$w_{ij} = e^{-(\mathbf{f}_i - \mathbf{f}_j)^\top (2\mathbf{\Lambda})^{-1} (\mathbf{f}_i - \mathbf{f}_j)} \tag{9}$$

where $\mathbf{f}_i$ is a feature vector containing the spatial position of a pixel i, x_i and y_i, and its RGB color values, R_i, G_i, B_i: $\mathbf{f}_i = [x_i, y_i, R_i, G_i, B_i]$. The diagonal matrix $\mathbf{\Lambda}$ contains normalizing values, which are approximately (the square of) 1/4 of the range of variability of the respective component: $\mathbf{\Lambda}^{1/2} = \mathrm{diag}(N/4, N/4, \sigma_m^R, \sigma_m^G, \sigma_m^B)$.

The graph can be partitioned into two disjoint sets, A and B, $A \cup B = V$, $A \cap B = \emptyset$, by simply removing edges connecting the two parts. This set of edges constitute a *cut*. The cost of the cut, which measures the degree of similarity between the two regions A and B, is the sum of all its edge weights:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{10}$$

The optimal segmentation is the cut with the minimal cost.

Going back to the problem of choosing between two image patches the one that yields the best continuation of the background, let us consider the cut defined by their binarized difference:

$$A = \{(x, y) : (\mathbf{u}_{x,y,k_1} - \mathbf{u}_{x,y,k_2})^\top (\sigma_{k_1}^2 \mathbf{I} + \sigma_{k_2}^2 \mathbf{I})^{-1} \\ (\mathbf{u}_{x,y,k_1} - \mathbf{u}_{x,y,k_2}) < \chi_3^{-1}(\alpha)\} \tag{11}$$

The patch where $cut(A, B)$ is lower is the one containing the occluding pixels (because the cut is along the discontinuity), whereas the same cut in the background patch has a higher cost, because – not being correlated with the structure of the background patch – it is more likely to contain expensive edges.

Our method based on graph-cuts can be seen as a principled way of applying the same continuity criterion as in [5], where a heuristic based on the comparison of the *inner* and *outer* boundaries of the difference region is employed.

2.4. Summary of the method

1. Estimate the image noise σ_m^2 as the sample variance of frames difference, using the MAD (Eq. (2)).

2. Subdivide the spatial domain into overlapping windows $\mathcal{W}$ (footprints).

3. On each footprint $\mathcal{W}$, cluster image patches $\mathbf{v}_{\mathcal{W},t}$ using single linkage agglomerative clustering (see MATLAB), using SSD (Eq. (3)) as the distance and a cutoff based on the Chi-square test (Eq. (5)).

4. For each cluster, compute its representative by averaging (Eq. (6)). Discard clusters of size 1.

5. Select the clusters of maximal length, insert their representatives in the background B.

6. Select a patch in B, select a neighboring footprint W_i which is not represented in B.

7. For each cluster representative $\mathbf{u}_{\mathcal{W}_i,k}$ evaluate the overlap with B (Eq. (7)) and select candidates patches for insertion in B according to Eq. (8).

8. The candidate patches enter into a round robin tournament, where the comparison between two of them is done according to the cost of the cut (Eq. (10)) defined by their binarized difference (Eq. (11)). The higher cost wins. The winner of the tournament is inserted in the background B.

9. Repeat from Step 6 until the background image is complete.

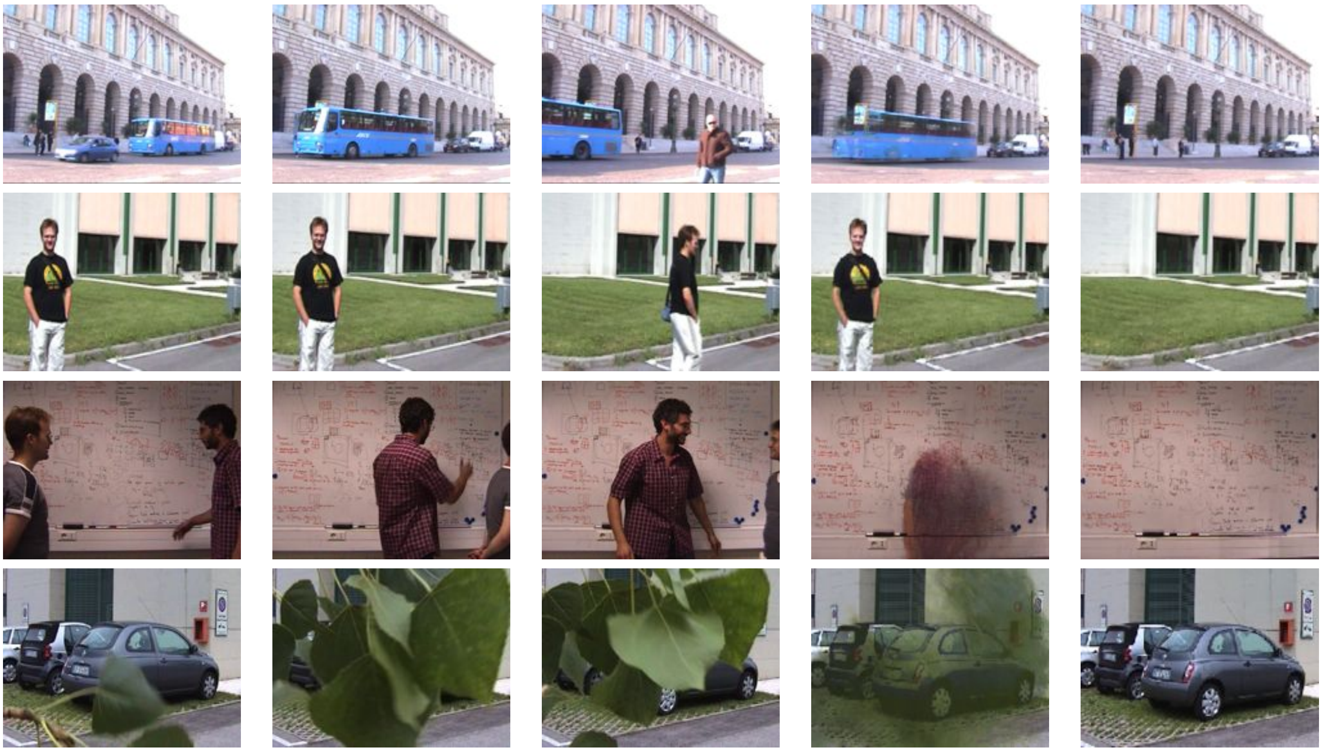

Figure 3. Results with real sequences. From top to bottom: “Granguardia”, “Ca’ Vignal”, “Board”, “Foliage”. From left to right: Three sample frames from the sequence, median, output of our algorithm (background).

As the footprints are overlapping, in the final background image up to four patches might insist on a single pixel (x, y). Let $\mathcal{T}$ be the set of temporal indexes of the frames that contributed to the background value at (x, y), via the cluster representatives. The estimate of the background color $\mathbf{c}_{x,y}$ and its variance $\sigma^2_{x,y}$ are obtained as the sample mean and variance – respectively – of the values $\mathbf{v}_{x,y,\mathcal{T}}$.

3. Experiments

In the experiments we used the following parameter setting. The windows size N must be small enough so as to have the background revealed at least once in every window, but large enough for the overlap test to be reliable. We used $N = 17$ with images 200×260. The confidence level was $\alpha = 0.999999$ in all the tests.

We performed experiments with real sequences (Fig. 3) presenting different challenges. The “Granguardia” is characterized by lots of motionless clutter due to people waiting for the bus, together with a large occlusion due to the bus itself. “Ca’ Vignal” depicts a person standing in the same position for most of the time. “Board” is an indoor sequence where moving objects cast shadows on the background. Finally, “Foliage” presents a serious occlusion due to some foliage shaken in front of the camera, giving rise also to motion blur. For each experiment we report few selected frames of the original sequence, the output of the median (in order to emphasize the toughness of the clutter) and the output of our algorithm.

Our method works with any number of frames, including the minimum, which is two. Fig. 4 shows how the background is recovered without artifacts from a sequence composed by two frames of “Ca’ Vignal”.

One way of verifying the goodness of the resulting background is to use it for foreground recovery. We subtracted each frame to the estimated background and thresholded the result like in Eq. (11). Selected foreground frames are shown in Fig. 5.

4. Conclusions

We illustrated a method for background initialization from cluttered video sequences. The method is robust, as it can cope with serious occlusions caused by moving objects. It is scalable, as it can deal with any number of frames greater or equal than two. It is effective, as it always recovers the background when the assumptions are satisfied. Moreover, our method rests on sound principles in all its stages, and only few, intelligible parameters are needed, namely the confidence level for the tests and the patch size. Future work will aim at estimating it from the data, using a multi-resolution approach.

Our method can be straightforward extended to the case of moving camera, provided that we can compensate for

Figure 4. Background recovery from two frames. From left to right: Original frames, background after step 5, output of the algorithm.

Figure 5. Selected foreground from the test sequences.

camera motion with respect to the background, as in the case of mosaicing [11]. In that case, clutter can be due either to moving objects or to motion parallax.

Acknowledgements

Thanks to Marco Cristani for inspiring discussions.

References

[1] A. Colombari, M. Cristani, V. Murino, and A. Fusiello. Exemplar-based Background Model Initialization. In *Proceedings of the 3rd ACM International Workshop on Video Surveillance & Sensor Networks*, 2005.

[2] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 733–740, 2001.

[3] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1986.

[4] I. Haritaoglu, D. Harwood, and L. Davis. W^4: Who? When? Where? What? a real time system for detecting and tracking people. In *Proceedings of the 3rd International Conference on Face and Gesture Recognition*, 1998.

[5] C. Herley. Automatic occlusion removal from minimum number of images. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 1046–1049, 2005.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[7] O. Javed, K. Shafique, and M. Shah. Hierarchical approach to robust background subtraction using color and gradient information. In *Workshop on Motion and Video Computing*, pages 22–27, 2002.

[8] J. Jia, T. Wu, Y. Tai, and C. Tang. Video repairing: Inference of foreground and background under severe occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[9] L. Li and M. K. H. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112, February 2002.

[10] W. Long and Y. Yang. Stationary background generation: An alternative to the difference of two images. *Pattern Recognition*, 23:1351–1359, 1990.

[11] F. Odone, A. Fusiello, and E. Trucco. Layered representation of a video shot with mosaicing. *Pattern Analysis and Applications*, 5(3):296–305, August 2002.

[12] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring paragios. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1034–1040, 2001.

[13] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 69–72, 2005.

[14] C. Rasmussen and T. Korah. Spatiotemporal inpainting for recovering texture maps of partially occluded building facades. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 125–128, 2005.

[15] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[16] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[17] M. Wertheimer. Laws of Organization in Perceptual Forms. In Ellis Willis D., editor, *A Source Book of Gestalt Psychology*, pages 71–88. Harcourt Brace, New York, 1939.

[18] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 120–127, 2004.

[19] C. Wren, A. Azarbayehani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.