Patch-based Background Initialization in Heavily Cluttered Video

Andrea Colombari and Andrea Fusiello *Member, IEEE,*

Abstract

In this paper we propose a patch-based technique for robust background initialization that exploits both spatial and temporal consistency of the static background. The proposed technique is able to cope with heavy clutter, i.e, foreground objects that stand still 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, a tessellation is grown incrementally by selecting at each step the best continuation of the current background. The method rests on sound principles in all its stages, and only few, intelligible parameters are needed. Experimental results show that the proposed algorithm is effective and compares favorably with existing techniques.

Index Terms

Background initialization, Bootstrapping, Background modelling, Motion segmentation, Content-based representation

I. Introduction

Segmenting moving objects from a static background is a relevant issue in areas such as video surveillance [1], [2], perceptual interfaces [3], and content-based video encoding (MPEG4) [4]. Foreground objects can be extracted effectively by subtracting the background in the image frames, provided that an updated model of the background is available at any time. This is achieved by *initialization* (also called *bootstrapping*) of the background followed by its *maintenance*. Therefore it is assumed that a short segment at the beginning of the sequence is reserved for initialization.

A. Colombari is with eVS embedded Vision Systems S.r.l., Strada Le Grazie 15, I-37134 Verona, Italy. e-mail: andrea.colombari@evsys.net tel.: +39.045.802.7027 fax: +39.045.802.7027. A. Fusiello is with the Dipartimento di Informatica, Università degli Studi di Verona, Strada Le Grazie 15, I-37134 Verona, Italy. e-mail: andrea.fusiello@univr.it. tel.: +39 045 8027088. fax +39 045 8027068

It is clear that initialization is crucial to allow a proper maintenance. Nevertheless, there has been a large amount of work addressing the issues of background model representation and maintenance [2], [1], [3], [5], [6], [7], [8], [9], [10] but not as much focusing on model initialization [11], [12], [13]. The main reason is that often the assumption is made that initialization can be achieved by exploiting some clean frames at the beginning of the sequence. Obviously this assumption is hardly met in real scenarios, because of continuous clutter presence.

In this paper we address the background initialization problem, which is defined as follows: Given a video sequence taken with a stationary camera, in which any number of moving occluders (clutter) can be present, output a single image of the scene *without* clutter, even if such an image has never been captured.

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 both pixels that correspond to the background and pixels belonging to foreground. The goal is to reconstruct a background image by picking the correct color (or gray-level) from each time-line.

Some assumptions are customary, that make this task feasible:

i) the background has to be globally stationary, i.e. only small local motion may occur (e.g. waving trees);
ii) in each pixel, or small region, the background is revealed for at least a short interval of the sequence;
iii) the processing occurs after the end of the sequence.

The second hypothesis implies that no object can occlude the background for the entire sequence. This is necessary for we want to use only *observed* values to fill the background at each location, as opposed to video inpainting [14], [15], [16], where *plausible* values are used for filling holes.

If hypothesis ii) were stronger, requiring that clutter has to rest on each pixel location for less than 50% of the entire sequence length, the background could be easily obtained as the median of each pixel color distribution [17].

Other techniques [1], [3], [6] have been proposed which, like the median, operate at pixel-level, making decisions independently for each pixel. The Adaptive Smoothness Method [5], for example, finds intervals of stable intensity in the time-line. Then the longest stable value for each pixel is selected and used as the value that most likely represents the background.

Better performances can be obtained if the spatial support of a pixel is taken into account. The Local Image Flow algorithm [11], for instance, considers also information generated by the neighboring locations of a pixel. Background values hypotheses are generated by locating intervals of relatively constant intensity, which are weighted with local motion information gathered from the optical flow.

The algorithm that is closer in spirit to ours is [13]. The main difference is that [13] relies on temporal stability, whereas we do not. It applies a rough segmentation of the input sequence into foreground and background blocks. Segmentation aims to find stable intervals of time for each block, i.e. intervals with low motion energy. This is done block by block using an iterative process that minimizes a cost function proportional to the motion energy. The stable interval of a block is used to initialize the stable interval of the adjacent blocks, thereby exploiting the spatio-temporal continuity of the sequence. F

All the techniques based on temporal stability, however, are doomed to fail in presence of very persistent clutter, which eventually becomes part of the background. This is also referred to as the *waking person* or *sleeping person* problem [2].

The algorithms proposed in [12] and [18] solve the sleeping person problem, borrowing some ideas from video inpainting. They are based on the same scheme: (i) identifying an initial background region and then (ii) filling-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 background should be self-similar (like a building facade) and that the starting region should be large enough to provide sufficient structure information.

In summary, our algorithm proceeds as follows. First the sequence is subdivided in patches that are clustered along the temporal line in order to narrow down the number of background candidates. Then the background is grown incrementally by selecting at each step the patch that provides the *best continuation*, according to the same continuity measure implemented by the spatial graph-cuts [19] segmentation.

With respect to the state of the art, our approach i) is more effective in dealing with the sleeping person problem than techniques based on the notion of temporal stability, such as [11]; ii) it needs not to assume a self-similar background like [12], [18]; iii) it makes only mild assumptions, unlike [13], for example, which requires that the top left block of the sequence be clear of clutter. A preliminary version of this work appeared in [20].

The paper is organized as follows. In Sec. II our continuity-based method is explained step by step and summarized in Sec. II-D. Experimental results are shown in Sec. III: Our algorithm is directly compared with the one proposed in [13] and with those evaluated in [2] for the specific case of the bootstrapping sequence. Finally, conclusions are drawn in Sec. IV.

II. METHOD

We model the 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 *spatio-temporal 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 indices. The window $\mathcal{W} = I_x \times I_y$ is the *spatial footprint* of the patch. An *image patch* $\mathbf{v}_{\mathcal{R}}$ is a spatio-temporal patch with a singleton temporal index: $\mathcal{R} = \mathcal{W} \times \{t\}$ or $\mathcal{R} = (\mathcal{W}, t)$.

Our method for background initialization is based on the following hypothesis:

i) the background is constant;

ii) in each spatio-temporal patch (of a given footprint size) the background is revealed at least once;

iii) foreground objects introduce a color discontinuity with the background.

The first hypothesis implies that the same background point is imaged always onto the same pixel with the same intensity (if visible), and it implicitly defines the background model: an image with single valued pixels, as opposed to models that represent disjoint set of values at each pixel, as in [1], [2].

Small camera motion can be modelled by a global projective transformation compensated as in [21], while little intensity changes can be taken into account as noise.

The second hypothesis differs from the corresponding one stated in the introduction: We require visibility at patch level instead of pixel level – which is stronger, but we relax it along the time dimension. Indeed, the technique presented here can deal, in principle, with sequences where the background is revealed exactly once, differently from [13], [5], [11].

The third hypothesis (also used in [22]) excludes camouflage and makes possible to grow the background *by continuity*. In other words, it expresses a bias toward a continuous background.

A. Estimating camera 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 (R,G,B) are statistically independent, therefore we will consider here one color channel at a time. Albeit questionable, this is a simplifying assumption that many authors made, including [1].

Assuming that noise is i.i.d. Gaussian with zero-mean $\mathcal{N}(0, \sigma_{\mathrm{n}}^2)$, differences of pixel values along the time-line $\mathbf{n}_{x,y,t} = \mathbf{v}_{x,y,t} - \mathbf{v}_{x,y,t+1}$ are distributed as $\mathcal{N}(0, 2\sigma_{\mathrm{n}}^2)$ plus outliers due to moving foreground objects. The noise standard deviation σ_{n} is then estimated robustly from $\mathbf{n}_{x,y,t}$. In order to get more statistics, we consider not only the differences between consecutive pixel values but also between frames of distance two and three.

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

$$\mathrm{MAD} = \mathrm{med}_i\{|\mathbf{n}_i - \mathrm{med}_i\{\mathbf{n}_i\}|\}. \tag{1}$$

It is proven [23] that, for symmetric distributions, the MAD coincides with the inter-quartile range, hence, in our case:

$$\mathrm{MAD} = \tfrac{1}{2}(\Phi^{-1}(\tfrac{3}{4}) - \Phi^{-1}(\tfrac{1}{4}))\sqrt{2}\sigma_{\mathrm{n}} = \Phi^{-1}(\tfrac{3}{4})\sqrt{2}\sigma_{\mathrm{n}} \tag{2}$$

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

B. Temporal clustering

The spatial indices 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.

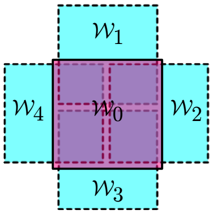


Fig. 1. Each footprint $\mathcal{W}_o$ has four overlapping neighbors: $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3$ and $\mathcal{W}_4$.

In order to reduce temporal redundancy, in each spatio-temporal patch, we cluster image patches that depict the same static portion of the scene with *single linkage agglomerative clustering* [24]. Starting from all singletons, each sweep of the algorithm combines two clusters into a single cluster according to the *simple linkage* rule which says that the two clusters 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, is set.

Let $\mathbf{v}_{\mathcal{S}}$, $\mathcal{S} = \mathcal{W}_i \times \{1 \cdots L\}$, be a spatio-temporal patch with footprint $\mathcal{W}_i$ which extends in time from the first to the last frame. 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 Distances (SSD):

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

The cutoff distance should prevent clustering together image patches that do not have the same content. 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:

$$\mathrm{SSD}(\mathcal{W}, t_1, t_2) = (\bar{\mathbf{v}}_{\mathcal{W},t_1} - \bar{\mathbf{v}}_{\mathcal{W},t_2})^\top \frac{\mathbf{I}}{2\sigma_{\mathrm{n}}^2} (\bar{\mathbf{v}}_{\mathcal{W},t_1} - \bar{\mathbf{v}}_{\mathcal{W},t_2}) \tag{4}$$

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 [5], except for the fact that clusters do not need to form a connected temporal interval, and tricky thresholds are avoided.

The resulting clusters are spatio-temporal patches, with possibly not consecutive temporal indices. 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 $\sigma_k^2 = \sigma_{\mathrm{n}}^2 / |\mathcal{T}_k|$.

In each spatial footprint $\mathcal{W}$ we now have a variable number of cluster representatives $\mathbf{u}_{\mathcal{W},k_1} \dots \mathbf{u}_{\mathcal{W},k_\ell}$ (see Fig. 2). The assumption ii) that we made implies that (at least) one of them depicts *only* static background: The subsequent stage is devoted to find out which one.

Motion energy heuristic: A heuristic that demonstrated helpful to cull the clusters consists of rejecting those of size one (i.e., composed by one frame) provided that this do not eliminate *all* the clusters with a given footprint. This corresponds to the practice of discarding patches with high *motion energy* [18], [13], [11], computed with optical flow or temporal differencing. Indeed, image patches containing fast moving objects tend to form size-one clusters. In order to obtain the correct result, however, we ought to strengthen the hypothesis: in each spatio-temporal patch the background must be revealed at least *twice*.

C. Background tessellation

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

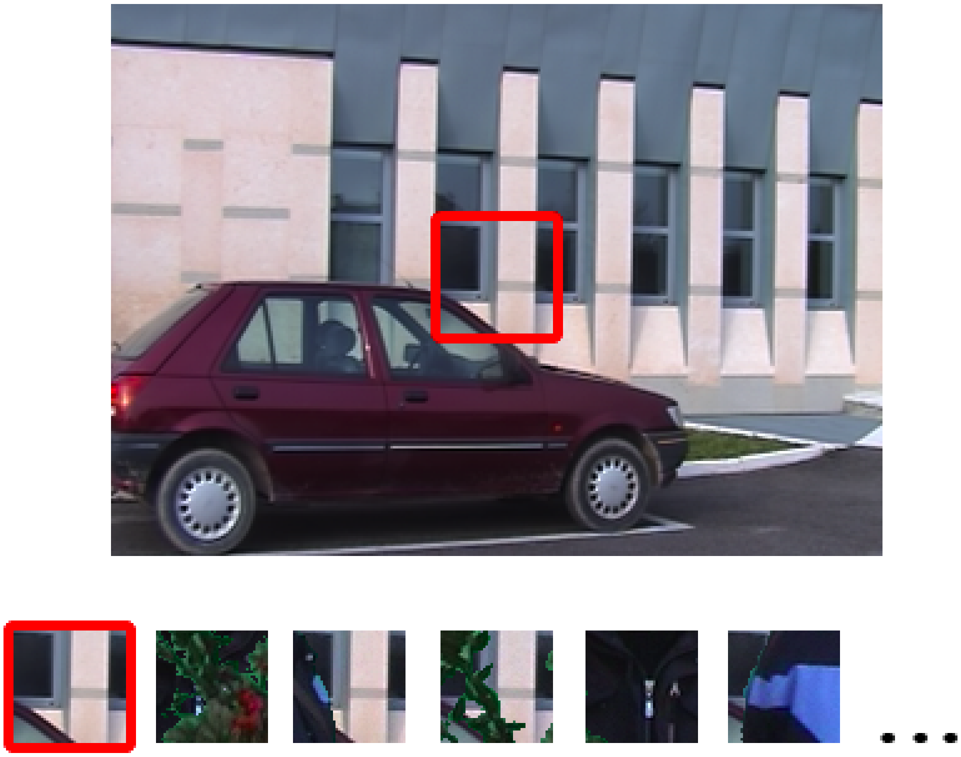


Fig. 2. Some cluster representatives over the same footprint: The first one depicts the background, whereas the others contain clutter. The “people&foliage” sequence is depicted in Fig. 6

The background seeds are the patches that represent the largest clusters. Since we assume that no foreground object is stationary in *all* the frames, if the largest clusters have size L (maximal), they are fully reliable. Otherwise, mistakes are possible as they could represent a persistent foreground object.

The growing proceeds as follows. Let $\mathcal{W}$ be a spatial footprint where a background patch has not been assigned yet, but it has been assigned to at least one of its neighbors, $\mathcal{W}_0$. This means that $\mathcal{W}$ overlaps with some background. The algorithm assigns the background patch to $\mathcal{W}$ by choosing one from the cluster representatives with footprint $\mathcal{W}$.

The selected patch has to fulfill two requirements:

1) *Seamlessness*. In the part that overlaps with $\mathcal{W}_0$ it has to depict the same content as the background patch, so that it can be stitched seamlessly to it;
2) *Best continuation*. In the non-overlapping part it has to represent the “best continuation” of the current background, meaning that, among several candidates, the patch that introduces the least discontinuity is chosen.

This procedure is repeated for all the footprints, until all the background has been assigned (Fig. 3). Details on how 1) and 2) are implemented are given in the next two subsections.

Time-intersection heuristic: Another requirement, drawn from [13], that might be considered before 1) and 2) as a culling heuristic consists in requiring that the intersection of the temporal indices of adjacent clusters that are selected to represent the background is not empty. This is related to a general implicit assumption of spatio-temporal continuity of the foreground motion.

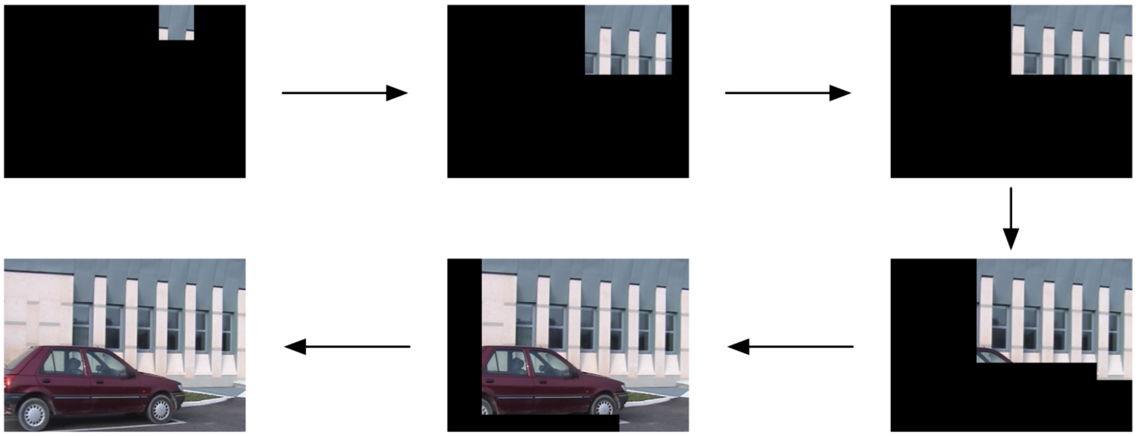


Fig. 3. Snapshots of the background as the tessellation proceeds.

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

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

By the same token as before (in (5)), $\mathbf{u}_{(\mathcal{W},k)}$ is considered for inclusion in the background with confidence α if

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

where $M = |\mathcal{W}_0 \cap \mathcal{W}|$.

If $\mathcal{W}$ 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.

2) Best continuation: 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* [25]. The approach rests on the assumption iii) that foreground objects introduce a discontinuity with the background. When a pure background patch is compared to an image patch containing foreground, their binarized difference defines a partitioning of the pixels into two groups (Fig. 4), 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) must be higher in the patch containing foreground than in the one containing background. This links the problem of selecting the best continuation of the background to the visual grouping theory.

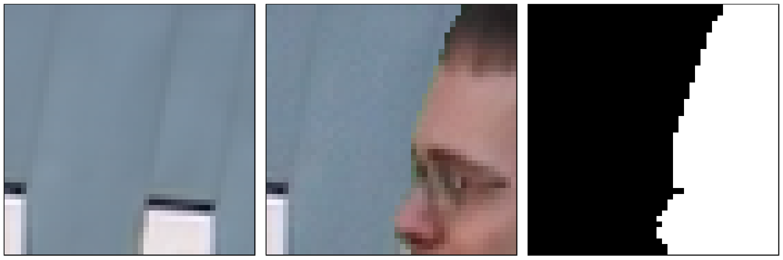


Fig. 4. From left to right: two cluster representatives (background candidates) and their binarized difference.

Graphs cuts have been proposed in [19] as a 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 the two nodes i and j is defined as [19]:

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

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: $\mathbf{\Lambda}^{1/2} = \mathrm{diag}(N/4, N/4, \sigma_\mathrm{n}, \sigma_\mathrm{n}, \sigma_\mathrm{n})$.

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 constitutes 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:

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

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, consider the cut (A, B) defined by their binarized difference:

$$A = \{(x, y) : \tfrac{1}{\sigma^2_{k_1}+\sigma^2_{k_2}} \, ||\mathbf{u}_{x,y,k_1} - \mathbf{u}_{x,y,k_2}||^2 < \chi_3^{-1}(\alpha)\}. \tag{11}$$

Let us assume that one of the two patches contains only background, while the other contains also some foreground clutter; the cost $cut(A, B)$ is more likely to be lower in the second, because it runs along the discontinuity, whereas in the background patch the same cut is likely to contain more expensive edges.

Our method, inspired by graph-cuts[1], can be seen as a principled way of applying the same continuity criterion as in [22], where a heuristic based on the comparison of the *inner* and *outer* boundaries of the difference region is employed.

D. Summary of the algorithm

1) Estimate the camera noise σ^2_{n} as the sample variance of frames difference, using the MAD as in (2).
2) Subdivide the spatial domain into overlapping windows $\mathcal{W}$ or footprints.
3) On each footprint $\mathcal{W}$, cluster image patches $\mathbf{v}_{\mathcal{W},t}$ with single linkage agglomerative clustering using SSD (3) as the distance and a cutoff based on the Chi-square test (5).
4) Compute cluster representative by averaging with (6).
5) Select the clusters of maximal length, insert their representatives in the background B.
6) Select a footprint $\mathcal{W}$ which is only partially filled in B.
7) For each cluster representative $\mathbf{u}_{\mathcal{W},k}$ evaluate the discrepancy with B using (7) and select candidates patches for insertion in B according to (8).
8) The candidate patches enter a round robin tournament, where the comparison between any two of them is done according to cost of the cut (10) defined by their binarized difference (11). The higher cost wins. The winner of the tournament in inserted in B.
9) Repeat from Step 6 until the background image is complete.

As the footprints are overlapping, up to four pixels from the original sequence might rest on a single pixel (x, y) in the final background image. Let $\mathcal{T}$ be the set of temporal indices of the frames that contributed to the background value at (x, y), via the cluster representatives. The output of the algorithm

[1] Please note that we do not actually perform segmentation. We simply rely on the graph-cuts paradigm to compute the cost of a given segmentation.

is an estimate of the background $\mathbf{b}_{x,y}$ and its variance $\sigma^2_{b,x,y}$ obtained as the sample mean and variance – respectively – of the values $\mathbf{v}_{x,y,\mathcal{T}}$. A variance image is shown in Fig. 5.

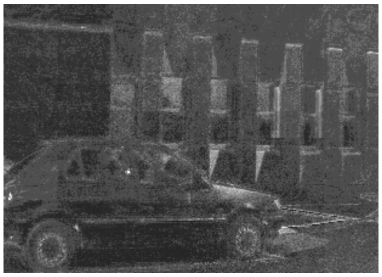


Fig. 5. Gray level visualization of the per-pixel variance of the "people&foliage" background (values are normalized in [0,255]).

The culling heuristics described previously, namely *motion energy* and *time intersection*, do not affect significantly the quality of the result, rather they reduce the computing time, as they cut the number of cluster candidates. Occasionally, however, they might discard the true background cluster, thereby preventing the algorithm to produce the correct result. For this reason we prefer to consider them as optional features that can be turned on if one is willing to trade off speed for quality of the result.

The complexity of the algorithm is dominated by the round robin tournament, in the background tessellation step. The worst case is when all clusters have dimension one and all of them enters the tournament, thus requiring $\frac{L\times(L-1)}{2} = O(L^2)$ trials for each footprint, where L is the number of images. Computing the cost of the cut costs $O(N^4)$ in the worst case, where N is the footprint size. Hence, the algorithm complexity is $O(M \times L^2 \times N^4)$, where M is the number of footprints. Since $M \approx 4\frac{Q}{N^2}$, where Q is the total number of pixels, the complexity can be rewritten as $O(Q \times L^2 \times N^2)$.

Our unoptimized MATLAB implementation takes around 6 hours on a 2GHz Pentium IV with 2Gb of RAM to process a 300 frames long sequence with a resolution of 240×320 and a patch size of 39x39.

III. Experimental Results

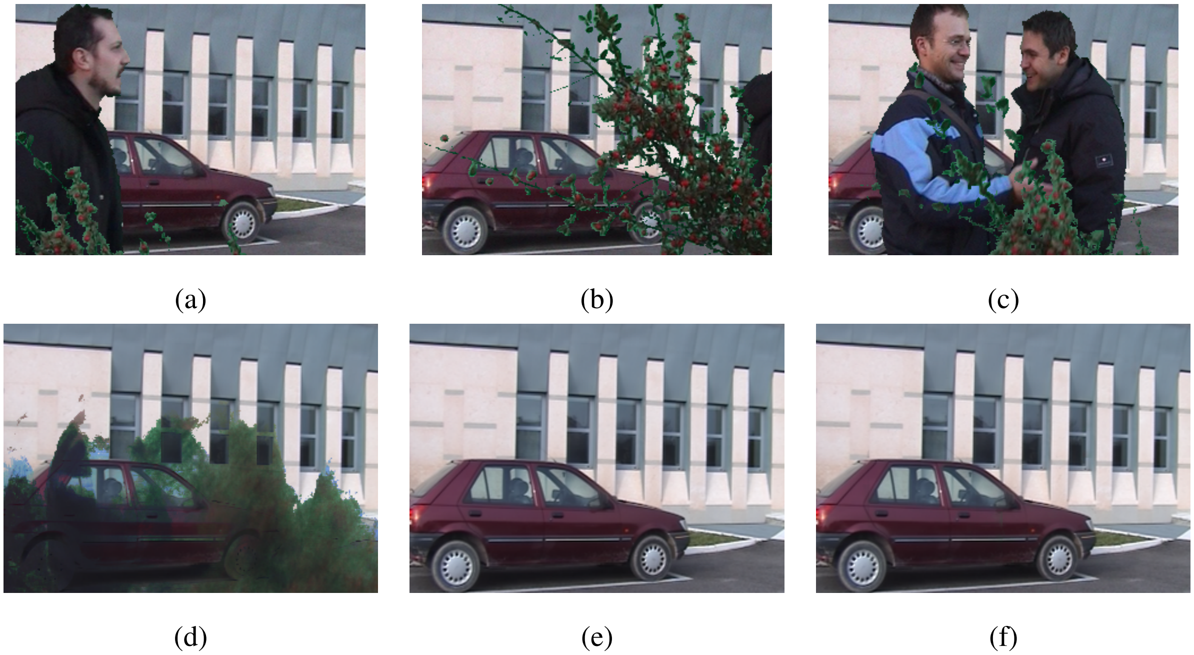
In the following we will refer to the method presented in this paper as PBI (Patch-based Background Initialization).

The experiments are organized into three parts. In the first part we used the sequence "people&foliage" that we constructed on purpose with heavy clutter in order to challenge the PBI algorithm. The ground truth for the sequence is known, hence a quantitative analysis was performed in terms of closeness to the true background (with PSNR). The sequence is available on the web[2] for comparison.

In the second part we compared PBI with the technique proposed in [13] (henceforth referred to as RBE), using the same data. Comparison is only visual for one sequence and quantitative (PSNR) for the other.

Finally, in the third part, we tested PBI on the "bootstrapping" sequence, that was used in [2] to compare several background modelling algorithms. Unfortunately the ground truth is provided only for the segmentation, and not for the background itself. As a consequence, we were able to compare performances of PBI only through the results of segmentation. It turns out that PBI algorithm perform better than the ones evaluated in [2].

The only critical parameter in the algorithm is the window size, which must be small enough to be clear of clutter at least once in the sequence, but large enough for the overlap test to be reliable. We used $N = 39$ with images 240×320 (scaled proportionally with other image sizes). The confidence level was $\alpha = 0.99999$ in all the tests.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Results on the "people&foliage" sequence. (a), (b), (c) are some sample frames, (d) is the median, (e) is the true background, and (f) is the background produced by PBI.

The "people&foliage" sequence (240×320) was obtained starting with a clean background sequence

[2] http://profs.sci.univr.it/∼fusiello/demo/bkg/

TABLE I

COMPARISON AGAINST THE GROUND TRUTH FOR "PEOPLE&FOLIAGE".

Algorithm	*PSNR*	*# wrong pixels*
average	14.12	47860
median	13.64	24015
PBI	44.89	5

onto which we pasted, with chroma-keying, two other sequences: One made by waving leaves, and another depicting some people (sample frames are shown in Fig. 6). The clutter (leaves and people) is present in every frame, and it is fairly persistent, as testified by the median background, also reported in Fig. 6 as a baseline case. The numerical comparison with the ground truth, reported in Tab. I, demonstrates that PBI recovers the correct background with negligible errors.

The number of wrong pixels in the background image was computed as follows. First the true background $\mathbf{g}$ and the camera noise σ_g were obtained by computing average and standard deviation (resp.) of the the clean background sequence. Then, using again the Mahalanobis distance, two pixels $\mathbf{g}_{x,y}$ and $\mathbf{b}_{x,y}$ belonging to the true background and to the computed background respectively are deemed to come from the same Gaussian distribution (i.e, the pixel is correctly classified as background) with confidence α if:

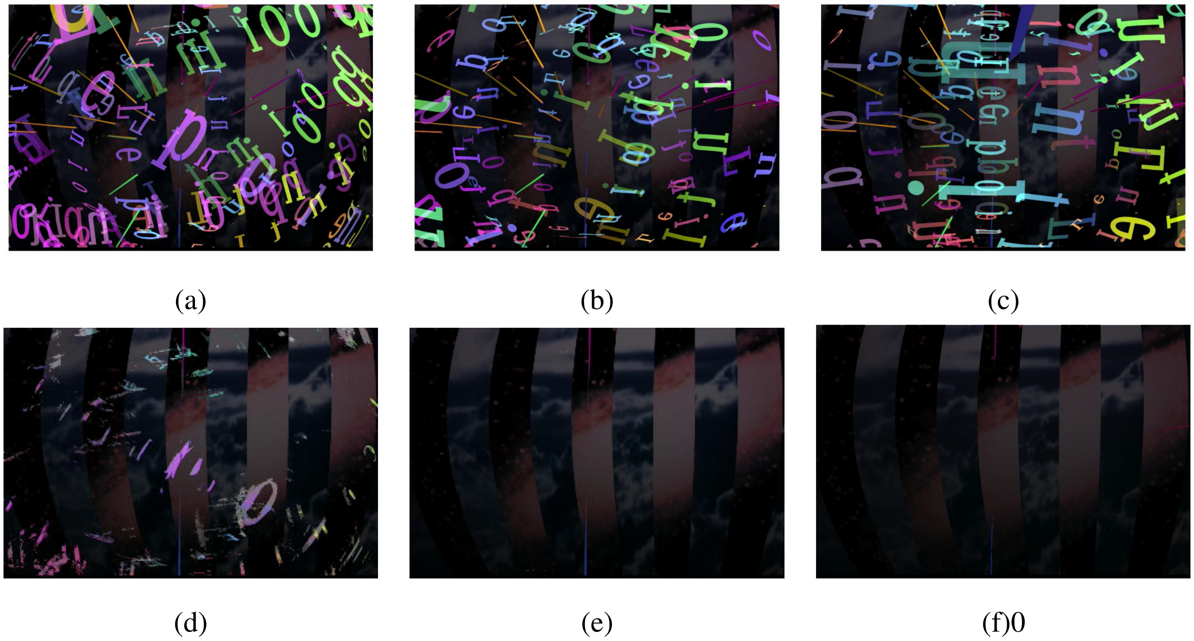
$$\frac{1}{\sigma^2_{b,x,y} + \sigma^2_{g,x,y}} ||\mathbf{b}_{x,y} - \mathbf{g}_{x,y}||^2 < \chi_3^{-1}(\alpha). \tag{12}$$

In this case the background pixel is correct, otherwise it is wrong.

Fig. 7 shows the results of RBE and PBI on the "VQEG-17" sequence (from Video Quality Experts Group Test Sequences[3]). As a baseline, the median background is also shown. By visual comparison (for the ground truth is not available) both algorithms produce a reasonably clean background and both perform better than the median, which is corrupted by small artifacts.

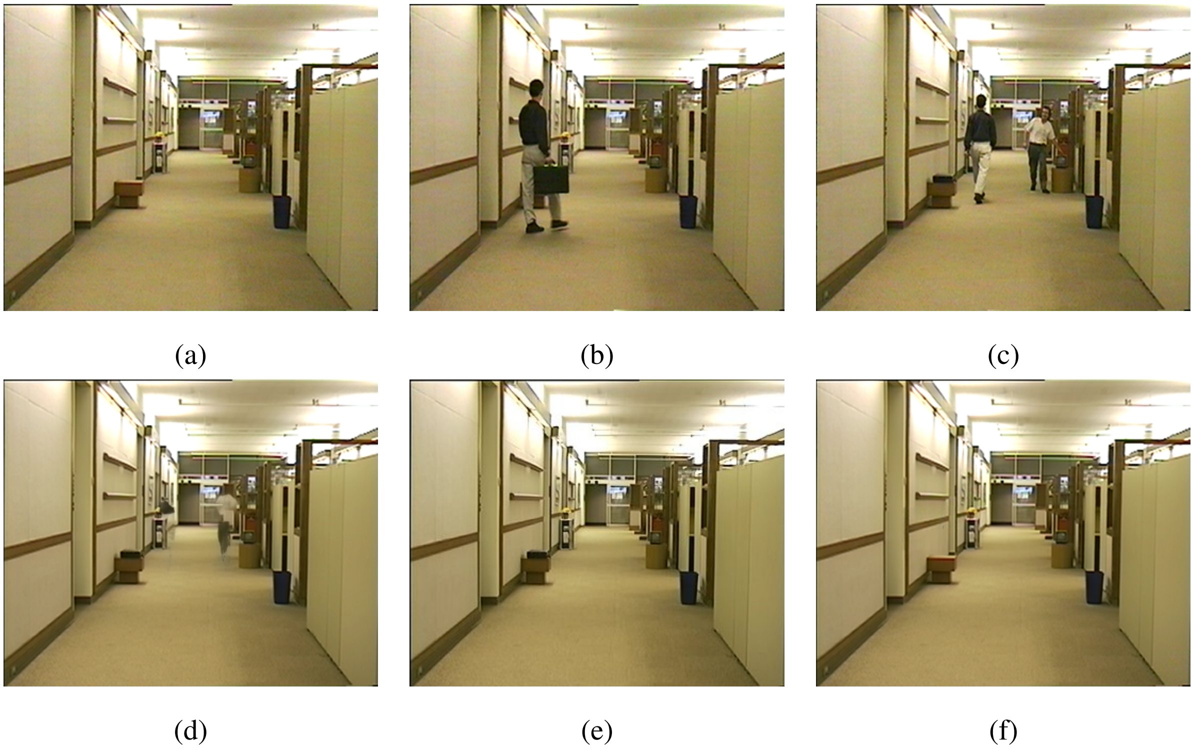
A quantitative comparison with RBE can be obtained on the "hall&monitor" sequence (Fig. 8), for which a ground truth background was computed as the average of the first four clean frames as written in [13]. Table II shows that PBI produces a background closer to the ground truth. As the reader can notice, the main difference between PBI results and RBE is the black suitcase that is not present at the beginning, it is left behind by a person and then remains in the same position until the end. Whether the

[3] http://www.vqeg.org/


(a) (b) (c)

(d) (e) (f)0

Fig. 7. Results on the “VQEG-17” sequence. (a), (b), (c) are some sample frames, (d) is the median, (e) is the output of RBE, and (f) is the output of PBI.


(a) (b) (c)

(d) (e) (f)

Fig. 8. Results on the “hall&monitor” sequence. (a), (b), (c) are some sample frames, (d) is the median, (e) is the output of RBE, and (f) is the output of PBI.

TABLE II

COMPARISON AGAINST THE GROUND TRUTH FOR "HALL&MONITOR".

Algorithm	*PSNR*
average	29.86
median	30.89
RBE	35.15
PBI	41.88

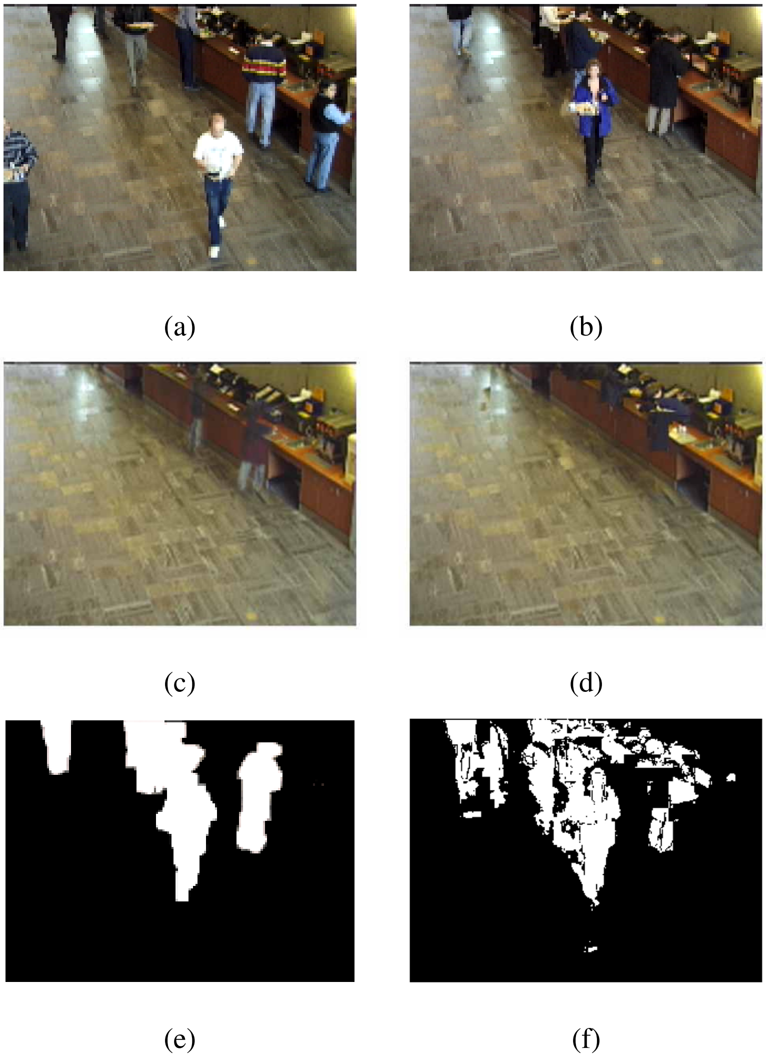
background should include the suitcase or not is clearly disputable. PBI solves the dilemma by favoring the most continuous background, hence the solution without suitcase is preferred. Albeit arbitrary, this behavior is consistent, i.e., it does not depend on how long the suitcase has been still.

This remark opens the question of defining *the* background (or at least what PBI consider to be background). Any definition based on temporal stability incurs in the "sleeping person" problem, and we avoided it on purpose. On the other hand, spatial continuity cannot be used alone: a white sheet waving in front of the camera would otherwise obtain a white background. A non-operarational characterization of the background produced by PBI is an unsettled issue.

In order to overcome the "sleeping person" problem, temporal persistence cannot be considered, so "background is the most persistent object" is not a valid answer. Also "background is the most continuous image that can be composed using all the available patches" is not a good definition, because it discards the notion of object stationariety (a white sheet that is seen in turn in all the patches would produce a white background). We must admit that we do not have a general definition of background that can cope with the suitcase example. In this paper we are assuming that the background is implicitly defined by the algorithm itself, by its hypothesis and behaviour.

Fig. 9 shows results obtained on the "bootstrapping" sequence, that was used in [2] specifically to evaluate (in terms of foreground segmentation) the initialization phase of background modeling algorithms. It consists of 3054 frames, the first 200 were used for initialization and testing occurs at the 299^{th} frame, for which a ground truth segmentation is provided (manually).

As we did before, foreground/background segmentation is cast as a statistical test using the Mahalanobis distance: A pixel $\mathbf{v}_{x,y,t}$ of the video sequence is deemed to belong to the background with confidence α



(a) (b)

(c) (d)

(e) (f)

Fig. 9. The top row shows the first (a) and the 299$^{\rm th}$ frame (b) of the “bootstrapping” sequence: (c) is he median, (d) is the output of PBI. The ground truth segmentation for frame #299 is shown in (e) and (f) is the result of PBI & Mahalanobis.

if:

$$\frac{1}{\sigma_{\rm n}^2+\sigma_{b,x,y}^2}\,||\mathbf{v}_{x,y,t}-\mathbf{b}_{x,y}||^2 < \chi_3^{-1}(\alpha)\,; \tag{13}$$

otherwise it is assigned to the foreground. We will refer to this segmentation algorithm as “PBI & Mahalanobis”.

This is a challenging sequence, and the recovered background by PBI & Mahalanobis has some errors in the area of the condiment bar, because the corresponding patches were permanently occluded. Consequently, the segmentation is far from perfect; nevertheless it turns out that it is better than the one obtained with other state-of-the-art algorithms.

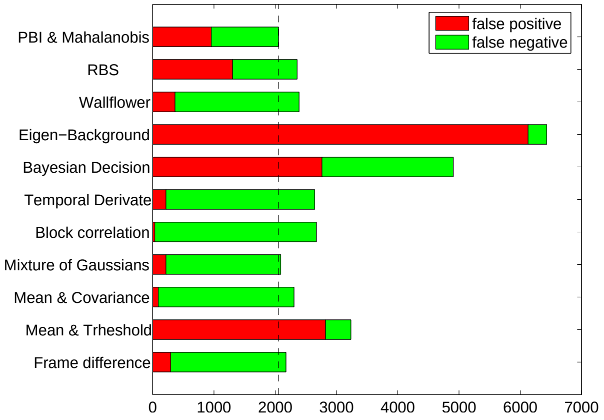


Fig. 10. Comparison of several background initialization algorithms on the "bootstrapping" sequence (see [2] for the description of the algorithms)

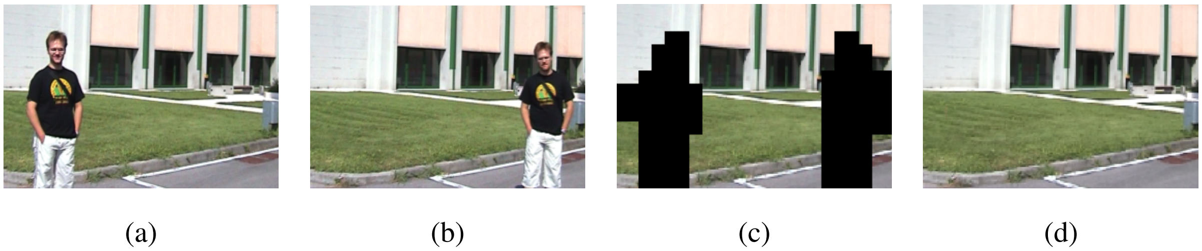


Fig. 11. Background recovery from two frames. (a) and (b) are the original pictures, (c) depicts the clusters of length two, and (d) is the output of PBI.

Fig. 10 reports the number of false positives and false negatives for the algorithms considered in [2] with the addition of RBS [10] and PBI & Mahalanobis (this paper). As the reader can notice, our method achieves the lowest total error (sum of false positives and false negatives).

Finally, the example shown in Fig. 11 demonstrate that our algorithm is not limited to videos, but can work with as few as two pictures, where time statistics are not available.

IV. Conclusions

We illustrated an incremental, patch-based method for background initialization in a video sequence, founded on the idea of continuation. 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. The latter can be tuned manually or automatically by a multiresolution approach. Experimental results show that our algorithm compares favorably with the state-of-the-art.

Acknowledgments: We would like to thank Andrea Prati for providing the code of the RBS algorithm [10] and for pointing us to some interesting papers. Dirk Farin kindly provided data and results of [13]. Stefano Fiore performed some of the experiments.

REFERENCES

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

[2] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," pp. 255–261, 1999.

[3] 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.

[4] P.Nunes, P.Correia, and F.Pereira, "Coding video objects with the emerging mpeg-4 standard," in *I Conferência Nacional de Telecomunicações*, April 1997.

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

[6] 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*, vol. 19, pp. 780–785, 1997.

[7] N. Paragios and V. Ramesh, "A mrf-based approach for real-time subway monitoring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 1034–1040.

[8] 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*, 2002, pp. 22–27.

[9] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 105–112, February 2002.

[10] S. Calderara, R. Melli, A. Prati, and R. Cucchiara, "Reliable background suppression for complex scenes," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 2006, pp. 211–214.

[11] 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*, 2001, pp. 733–740.

[12] 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.

[13] D. Farin, P. H. N. de With, and W. Effelsberg, "Robust background estimation for complex video sequences," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, 2003, pp. 145–148.

[14] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. 120–127.

[15] 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*, vol. 2, 2005, pp. 69–72.

[16] 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.

[17] B. Gloyer, H. Aghajan, K. Siu, and T. Kailath, “Video-based freeway monitoring system using recursive vehicle tracking,” in *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science & Technology – Image and Video Processing*, vol. 2421, 1995, pp. 173–180.

[18] 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*, vol. 3, 2005, pp. 125–128.

[19] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[20] A. Colombari, A. Fusiello, and V. Murino, “Background initialization in cluttered sequences,” in *5th Workshop on Perceptual Organization in Computer Vision*, 2006, in conjunction with CVPR 2006.

[21] ——, “Video objects segmentation by robust background modeling,” in *International Conference on Image Analysis and Processing (ICIAP 2007)*. Modena, Italy: IEEE Computer Society, 10-14 September 2007, pp. 155–164.

[22] C. Herley, “Automatic occlusion removal from minimum number of images,” in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, 2005, pp. 1046–1049.

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

[24] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[25] M. Wertheimer, “Laws of Organization in Perceptual Forms,” in *A Source Book of Gestalt Psychology*, Ellis Willis D., Ed., Harcourt Brace, New York, 1939, pp. 71–88.