Remove Noise in Video with 3D Topological Maps

Donatello Conte, Guillaume Damiand

Remove noise in video with 3D topological maps*

Donatello Conte[1] and Guillaume Damiand[2]

[1] Université François-Rabelais de Tours, LI EA 6300, F-37200 France
[2] Université de Lyon, CNRS, LIRIS, UMR5205, F-69622 France

Abstract. In this paper we present a new method for foreground masks denoising in videos. Our main idea is to consider videos as 3D images and to deal with regions in these images. Denoising is thus simply achieved by merging foreground regions corresponding to noise with background regions. In this framework, the main question is the definition of a criterion allowing to decide if a region corresponds to noise or not. Thanks to our complete cellular description of 3D images, we can propose an advanced criterion based on Betti numbers, a topological invariant. Our results show the interest of our approach which gives better results than previous methods.

Keywords: Video denoising; 3D Topological Maps; Betti numbers.

1 Introduction

Several video analysis applications, like video surveillance or traffic monitoring, require as a preliminary sub-task the identification within the scene of the moving objects (foreground) as opposed to the static parts of the scene (background).

Many algorithms have been proposed in the literature most based on the background subtraction technique [17, 3, 11]. These algorithms are quite efficient but no one is the best for all situations (see [5] for a comparison of the most widely used background subtraction algorithms).

Some authors give up looking for an algorithm that directly provides the ideal foreground mask, and apply, instead, some post-processing in order to reduce or eliminate noise pixels, that is pixels erroneously detected as foreground. For example in [15] the authors show a method to remove shadows, or in [6] the authors propose some heuristics for removing some errors in the foreground mask. Even if these approaches are efficient, they are based on some assumptions that are not always true, being too dependent from the specific video characteristics.

* Paper published in Proceedings of 18th Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, LNCS 8621, pp. 213-222, August 2014. Thanks to Springer Berlin Heidelberg. The original publication is available at `http://dx.doi.org/10.1007/978-3-662-44415-3_22`

Our paper fall in the last category: we propose an approach to reduce noise on foreground masks. But we present a general method that can be used on any video, in contrast to the more video-dependent approaches in the literature.

The basic idea of the method is that noise cannot be detected and removed analyzing a single frame of the video (as the other approaches do), but noise is easier to detect if more successive frames are examined: in fact real objects present, over the sequence, a regularity that noise seems not to have. Therefore the approach is based on a 3D structural representation of the foreground for a certain number of frames, and noise removal is done through structural operations on that data structure.

The remainder of the paper is organized as follows: in Sect. 2 the 3D structural representation of the scene is presented and explained, then in Sect. 3 the noise removal algorithm is given; the validation of the method, together with a comparison with other approaches, is made by a robust quantitative experimentation in Sect. 4; finally conclusions and perspectives are drawn in Sect. 5.

2 Definitions and Representation

We recall here the standard notions around 2D and 3D digital images, before introducing the notions of cellular subdivision and Betti numbers.

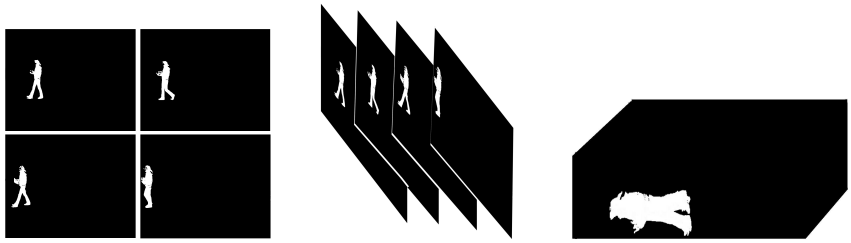
2.1 Digital 2D and 3D Images and Video

A *pixel* (resp. *voxel*) is an element of the discrete space $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) denoted by its coordinates (x, y) (resp. (x, y, z)). A 2D (resp. 3D) *image* is a set of pixels (resp. voxels) and a mapping between these pixels (resp. voxels) and a set of colors or gray levels. Each pixel (resp. voxel) e is associated with its *color* or *gray level* $c(e)$. Furthermore, each pixel (resp. voxel) e is associated with a *label* $l(e)$ from a finite set of labels L. These labels can be obtained from the image by a segmentation algorithm.

In this work, a temporal sequence of 2D images is considered as a 3D image. Each image of the sequence is associated with a time t. Thus each pixel is now considered as a *temporal pixel*, described by three coordinates (x, y, t), (x, y) being the spatial coordinates and t being the temporal coordinate. Thus, a temporal sequence of 2D images can be seen as a 3D image, where each voxel is in fact a temporal pixel.

We use the classical notion of α-*adjacency*. Two voxels (x_1, y_1, z_1) and (x_2, y_2, z_2) are 6-adjacent if $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1$; they are 26-adjacent if $max(|x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2|) = 1$ and they are 18-adjacent if they are 26-adjacent and if $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1$ or 2. Adjacency relations are extended to set of voxels: two sets of voxels S_1 and S_2 are α-adjacent if there is $v_1 \in S_1$ and $v_2 \in S_2$ such that v_1 and v_2 are α-adjacent.

Given an α-adjacency, an α-path between two voxels v_1, v_2 is a sequence of voxels starting from v_1 and finishing from v_2, such that two voxels of the



(a) (b) (c)

Fig. 1. The 3D representation of a video. (a) The image sequence. (b) The construction of the 3D image. (c) The final representation.

sequence are α-adjacent. A set of voxels S is α-connected if there is an α-path between any pair of voxels in S having all its elements in S.

A region in 3D is a maximal set of 6-connected voxels having same label. In addition to all the regions present in the labeled image, another region is considered, called R_0, which contains all the voxels that do not belong to the image. R_0 is the complementary of the image.

2.2 Cellular Subdivision of Video

A video seen as a 3D image is thus decomposed in 3D regions which form a partition of the image (i.e. any voxel belongs to exactly one region and the union of all the regions is equal to the entire image). Figure 1 shows an example of the 3D representation of a video. The partition is decomposed in the following cells: 0-cells are vertices, 1-cells are edges, 2-cells are faces and 3-cells are volumes.

Volumes describe the boundaries of 3D regions. Each volume is bounded by a surface, i.e. a set of adjacent faces, each face corresponding to a maximal contact area between two adjacent regions. Faces are bounded by edges, each edge corresponding to a maximal contact between two adjacent faces; and edges are bounded by vertices. Incidence relations are defined between the cells: two cells are incident if they have different dimensions and if one belongs to the boundary of the second one.

This cellular subdivision is a generalization of a region adjacency graph (RAG [16]) which is a graph having a vertex for each region, and an edge between each pair of adjacent regions. Vertices of the graph correspond to regions, and edges correspond to faces which describe the adjacency relations. This RAG was extended in a multi-graph, called multi-RAG, in order to represent multi-adjacency relations between regions (when two regions are adjacent several times). However our cellular structure is much more rich than RAG and multi-RAG since it describes also the multi-adjacency relations but the relations are ordered (given a region we can iterate through the adjacent regions in an ordered way which is not directly possible with graphs); furthermore in our structure all the cells are represented (RAG instead describes only 3-cells and 2-cells).

The cellular subdivision is represented thanks to 3D topological maps [7], an efficient 3D model based on combinatorial maps [13, 8] which represent the subdivisions in cells plus all the incidence and adjacency relations between the cells. In this work, 3D topological maps are used as an external tools and thus we do not go into detailed definitions, see above references for more details.

Having a cellular subdivision makes it possible to use some classical tool of algebraic topology, since our subdivision is an abstract cellular complex [2, 12]. In this work we use Betti numbers, a well-known topological invariant [14], in order to characterize the topology of regions. These numbers are related to homology groups, but we give here their intuitive presentation. In 3D, there are three non null Betti numbers: $\mathfrak{b}_0$ is the number of connected components, $\mathfrak{b}_1$ is the number of tunnels and $\mathfrak{b}_2$ is the number of voids. For a region R, $\mathfrak{b}_0(R) = 1$ because by definition a region is connected, $\mathfrak{b}_1(R)$ is the number of *tunnels* of R (a tunnel corresponds to a path of voxels in R that cannot be contracted into a point) and $\mathfrak{b}_2(R)$ is the number of *voids* of R (a void is a set of connected voxels that do not belong to R but that are fully surrounded by voxels in R). An incremental method to compute Betti numbers and to update them during region merging is given in [10].

3 Our Method

The main principle of our method is detailed in Algo. 1. First a given video of foreground masks is cut in consecutive slices of 2D images. Then each slice is consider as a 3D image (as explained in the previous section) and a 3D topological map is built to describe the corresponding cellular subdivision. This gives a set of regions, each one being labeled 0 or 1 depending if it corresponds to a set of background voxels (0) or of foreground ones (1).

Algorithm 1: Reduce noise on foreground masks

Input: A video of foreground masks V;
A boolean function $criterion(r_1, r_2)$.
Result: V is modified by merging all the adjacent pairs of regions satisfying *criterion*.

$T \leftarrow$ build the 3D topological map describing V;
foreach *region* $R_1 \in T$ *labeled* 1, $R_1 \neq R_0$ **do**
  foreach *region* R_2 *adjacent to* R_1, $R_2 \neq R_0$ **do**
    if $criterion(R_1, R_2)$ **then**
      $R \leftarrow$ merge(R_1, R_2);
      update region R;
return *the partition described by* T;

The 3D topological map corresponding to a given slice is built by using the algorithm given in [7] (wich is the extension in 3D of similar algorithm in 2D [9]).

During this construction, a cube is created for each voxel, and 6-adjacent voxels having the same label are merged. Doing the merging during the construction allows to process large video by avoiding to build the full model composed of all the cubes describing all voxels.

Then, each pair of adjacent regions (R_1, R_2) are considered so that R_1 is labeled 1. Indeed in order to reduce the noise, it is enough to merge some white regions with the background, thus there is no need to process black regions. If the pair (R_1, R_2) satisfies a given criterion, the two regions are merged. Merging two regions is done using the algorithm given in [10] which mainly consists in removing the faces separating the two regions, and possibly updating the edges and the vertices if needed.

Additional information associated with region R (which is the result of the merging of R_1 and R_2) must be updated. In this work, each region stores its number of voxels and its label. The number of voxels of R is the sum of the number of voxels of R_1 and the number of voxels of R_2. The label of R is always fixed to 0. Indeed, R_1 is labeled 1, thus by definition of regions, R_2 is labeled 0 (two adjacent regions can not have the same label). Since our goal is to reduce the noise, region R, considered as noise, and which is the merging of one background region and one foreground region, must stay in the background.

At the end of the algorithm, all the pairs of regions were considered and the new video is returned: this is the partition described by the modified 3D topological map.

The complexity of Algo. 1 is linear in number of adjacencies between regions times the complexity of the criterion. The number of adjacencies between regions is equivalent to the number of edges in the multi-RAG. Indeed thanks to the cellular decomposition we can iterate through all these adjacencies which are explicited by the faces, and the regions around each face are directly retrieved thanks to the incidence relations.

Now the main question is the definition of a criterion. Indeed, this is the main tool used during the reduce noise algorithm and only a correct definition will give good results. A first simple criterion, given in Eq. 1, consists in testing if the size of the white region is smaller than a threshold τ given by the user. The idea of this criterion comes from the fact that noise in image produces often small regions comparing to real objects. It is also important to highlight that for real objects there is always an overlapping between their appearances in consecutive frames, even at a low frame rate. Video used in experiments are at 10 fps and the overlapping between masks for real objects is always held.

$$size(R_1) < \tau \quad (1)$$
(R_1 being the region labeled 1)

The main interest of this criterion is to be very simple and computed in constant time since each region stores its size and the sizes are updated incrementally during the region merging. Note that this solution can be implemented

using a multi-RAG data-structure instead of 3D topological maps. Indeed additional information described by topological maps are here not used.

One problem of this first criterion is that some white regions representing noise can have a size larger than the threshold and thus are not removed. By studying such regions, we have observed that they are often very porous because noise is non regular and noisy adjacent pixels have often different labels. For this reason, these regions have many voids and tunnels contrary to regions describing objects which have generally a small number of voids and tunnels. This observation tends to show that the threshold associated with regions having many voids and tunnels must be increased in order to have an higher chance to be removed. For that, we propose in Eq. 2 a second criterion which mixes the size of the white region and its Betti numbers. This second criterion has two parameters: τ the threshold for the size of small regions, and φ, a percentage which is multiplied by the sum of the Betti numbers of R_1.

$$size(R_1) < \tau * (1 + \varphi * (\mathfrak{b}_1(R_1) + \mathfrak{b}_2(R_1))) \quad (2)$$
(R_1 being the region labeled 1)

This second criterion illustrates the interest of having an advanced description of regions (more precise than a RAG) allowing to compute and to mix several characteristics on regions. The complexity of this algorithm is equal to the complexity of the Betti number computation, i.e. linear in number of vertices, edges and faces describing region R_1. These numbers can be bounded by the number of voxels of R_1 times a constant number (8 for vertices, 12 for edges and 6 for faces).

4 Experiments

4.1 Dataset and Algorithms

We use the PETS 2010 Dataset [1]. This dataset is a standard database widely used for the performance evaluation of tracking and surveillance algorithms.

In order to evaluate the performances of the proposed denoising algorithm, we start from foreground detection masks on PETS video sequences, resulting from the application of a basic background subtraction (BS) algorithm. We expressly used the basic BS algorithm without any improvement and parameter optimization, because we want to show that the proposed algorithm can clean detection masks without any pre-processing prior. This allows to be not dependent on the specific video sequence and it avoids the optimization parameters phase that is tedious and not always possible.

Starting from the same detection masks, we compare our algorithm with:

A1 a denoising algorithm that uses only morphological operations (erosion and dilatation);

A2 the algorithm proposed in [6] that adds, to the basic subtraction algorithm, several post-processing improvements;

A3 the algorithm [6] with the addition of the grouping phase proposed by the same authors in [4].

As shown in [4], these algorithms are effective in reducing noise regardless of the method for foreground detection. Other approaches are not considered because of their high computational complexity. Note that the algorithms A2 and A3 require several parameters to set. A3 also requires a camera calibration phase (for details see [4]) for each video (taken with different camera settings). Therefore, in this experimentation we preliminary optimized these parameters, which are therefore specific for each sequence.

Our new method has two parameters: τ the threshold for the size, and nb which is the number of frames grouped in a same 3D slice. The method based on Betti numbers has an additional parameter: φ the percentage of Betti numbers added to the size.

4.2 Performance Index

We use an evaluation scheme inspired by the method presented in [18]; it takes into account one-to-one as well as many-to-one and one-to-many matches.

The goal of a detection evaluation scheme, on a frame, is to take a list of ground truth boxes $G = G_1, \ldots, G_n$ and a list of detected boxes $D = D_1, \ldots, D_m$ and to measure the quality of the match between the two lists. From the two lists D and G two overlap matrices σ and τ are created. The rows $i = 1 \ldots |G|$ of the matrices correspond to the ground truth boxes and the columns $j = 1 \ldots |D|$ correspond to the detected boxes.

The values are calculated as follows:

$$\sigma_{ij} = \frac{area(G_i \cap D_j)}{area(G_i)} \qquad \tau_{ij} = \frac{area(G_i \cap D_j)}{area(D_j)} \tag{3}$$

The matrices can be analyzed for determining the correspondences between the two lists:

One-to-One Matches: G_i matches against D_j if row i of both matrices contains only one non-zero element at column j and column j of both matrices contains only one non-zero element at row i. The overlap area needs to have a certain size compared to the rectangle in order to be considered successful ($\sigma_{ij} \geq e_1$ and $\tau_{ij} \geq e_2$).

One-to-Many Matches with One Ground Truth Box: G_i matches against several detected boxes if row i of the matrices contains several non-zero elements. The additional constraint of $\sum_j \sigma_{ij} \geq e_3$ ensures that the single ground truth rectangle is sufficiently detected.

One-to-Many Matches with One Detected Box: D_j matches against several ground truth boxes if column j of the matrices contains several non-zero elements. Also here we add the constraint of $\sum_i \tau_{ij} \geq e_4$.

Parameters $e_1, \ldots, e_4$ measure how much detected boxes against ground truth have to overlap. For most applications a value of 0.8 (80% of overlapping) is good; therefore we set $e_1 = \cdots = e_4 = 0.8$.

Based on this matching strategy, the recall and precision measures are defined as follows:

$$\text{recall} = \frac{\sum_i Match_G(G_i)}{|G|} \qquad \text{precision} = \frac{\sum_j Match_D(D_j)}{|D|} \tag{4}$$

where $Match_G(G_i)$ is defined as follows:

$$Match_G(G_i) = \begin{cases} 1 & \text{if } G_i \text{ matches against a single detected box} \\ 0 & \text{if } G_i \text{ does not match against any detected box} \\ 0.8 & \text{if } G_i \text{ matches against several detected boxes} \end{cases} \tag{5}$$

and $Match_D(D_j)$ accordingly.

The indexes Recall and Precision for a video sequence are the average values of recall and precision over all the frames of the sequence.

4.3 Results

Results of our experiments are given in Table 1 for the precision and recall measures, and in Table 2 for the F-score values (the harmonic mean of precision and recall). In all the arrays, dark grey cells are the best scores for each video, and light grey cells the second best scores. In these experiments, nb is always fix to 15. $tXXX$ is the value obtained by the method with the size criterion with $\tau = XXX$, and $tXXX\text{-}pYYY$ is the value obtained by the method with the size and Betti numbers criterion with $\tau = XXX$ and $\varphi = YYY$.

	v1		**v3**		**v4**		**v5**		**v6**		**v7**		**v8**	
	Rec	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*
A1	**0.55**	0.09	**0.38**	0.27	**0.54**	0.24	**0.55**	0.16	0.44	0.45	**0.57**	0.12	**0.68**	0.14
A2	0.44	0.29	0.20	0.39	0.46	0.36	0.44	0.32	0.41	**0.55**	0.47	0.31	0.60	**0.38**
A3	0.49	0.22	0.31	0.42	0.52	0.33	0.50	0.20	0.43	0.47	0.52	0.19	0.65	0.24
t2000	**0.54**	0.20	0.36	0.45	**0.55**	0.37	**0.54**	0.26	**0.46**	0.50	**0.55**	0.23	**0.66**	0.25
t3000	**0.54**	0.20	0.35	**0.46**	**0.55**	**0.39**	0.53	0.28	**0.46**	0.50	**0.55**	0.22	**0.66**	0.25
t4000	0.53	0.21	0.33	**0.46**	**0.54**	**0.40**	0.53	0.28	**0.46**	0.51	0.54	0.23	**0.66**	0.26
t2000-p.05	**0.54**	0.23	0.34	**0.46**	**0.54**	0.38	0.53	0.33	**0.46**	0.53	0.54	0.33	**0.66**	0.29
t2000-p.1	0.48	0.28	0.22	**0.47**	0.31	**0.39**	0.51	**0.36**	0.44	**0.54**	0.46	**0.37**	0.56	0.32
t2000-p.15	0.39	0.36	0.08	0.34	0.09	0.37	0.45	**0.35**	0.36	0.52	0.37	**0.36**	0.47	**0.33**
t3000-p.05	0.50	0.26	0.25	0.45	0.53	0.38	0.53	**0.36**	**0.45**	**0.54**	0.50	0.34	0.62	0.31
t3000-p.1	0.37	0.37	0.07	0.28	0.08	0.32	0.45	**0.35**	0.35	0.52	0.36	**0.37**	0.46	**0.33**
t3000-p.15	0.23	0.37	0.03	0.14	0.03	0.17	0.32	0.30	0.29	0.46	0.27	0.32	0.34	0.29
t4000-p.05	0.47	0.29	0.17	0.44	0.26	0.36	0.49	**0.36**	0.44	**0.54**	0.46	**0.37**	0.55	**0.33**
t4000-p.1	0.26	**0.38**	0.03	0.14	0.03	0.17	0.36	0.32	0.32	0.48	0.29	0.34	0.39	0.30
t4000-p.15	0.14	**0.40**	0.01	0.07	0.01	0.06	0.21	0.22	0.19	0.33	0.22	0.29	0.25	0.28

Table 1. The values of the indexes precision and recall for the considered algorithms.

These results show that our new method is competitive comparing with the three previous algorithms. Generally, merging more regions (either by increasing τ or by increasing φ) decreases the recall while increases the precision until a certain point. Thus better results are obtained by finding the good thresholds giving the best compromise for precision and recall.

These results show a second important conclusion: the method using Betti numbers can greatly improve the results. This is for example the case for video $v7$ with $\tau = 2000$, where the precision is improved from 0.23 without Betti to 0.37 with Betti using $\varphi = .1$.

These results are confirmed by the F-score values given in Table 2 which allow to find the best compromise between precision and recall. For all videos, the best scores are often obtained by the method using Betti numbers with $\tau = 2000$ (best score for 3 videos, and second best score for the 4 other videos).

	v1	**v3**	**v4**	**v5**	**v6**	**v7**	**v8**
	Fsc	*Fsc*	*Fsc*	*Fsc*	*Fsc*	*Fsc*	*Fsc*
A1	0.15	0.31	0.33	0.25	0.44	0.20	0.23
A2	0.35	0.26	0.40	0.37	0.47	**0.37**	**0.46**
A3	0.30	0.36	0.40	0.28	0.45	0.28	0.35
t2000	0.29	**0.40**	**0.45**	0.35	**0.48**	0.32	0.36
t3000	0.30	**0.40**	**0.45**	0.37	**0.48**	0.32	0.37
t4000	0.30	**0.39**	**0.46**	0.37	**0.48**	0.32	0.37
t2000-p.05	0.33	**0.39**	**0.45**	0.41	**0.49**	**0.41**	0.40
t2000-p.1	0.35	0.30	0.34	**0.42**	**0.49**	**0.41**	**0.41**
t2000-p.15	**0.37**	0.13	0.14	0.39	0.43	**0.37**	0.38
t3000-p.05	0.34	0.32	0.44	**0.43**	**0.49**	**0.41**	**0.41**
t3000-p.1	**0.37**	0.12	0.13	0.39	0.42	0.36	0.38
t3000-p.15	0.28	0.05	0.05	0.31	0.36	0.30	0.31
t4000-p.05	**0.36**	0.25	0.30	0.41	**0.49**	**0.41**	**0.41**
t4000-p.1	0.31	0.05	0.05	0.34	0.38	0.31	0.34
t4000-p.15	0.20	0.02	0.01	0.21	0.24	0.25	0.26

Table 2. The values of the F-score for the considered algorithms.

5 Conclusion

In this paper, we presented a new method of noise reduction on foreground video masks. Thanks to a 3D cellular description of the video, our method is defined in an high abstraction level considering regions and adjacency relations between these regions. This simplifies the denoising algorithm which consists mainly to merge foreground regions with the background. A second main advantage is the possibility of defining high level criteria on the regions. In this paper we use a simple criterion using the size of regions, and a more advanced criterion using

Betti numbers (that gives better results). This second criterion can be defined thanks to the full cellular representation, while this is not directly possible using simpler data-structures such as region adjacency graph.

As future work, we first want to work on the automatic computation of the parameters of our method. A second perspective is to define other criteria. Thanks to our representation, many possibilities could be studied mixing geometrical criteria and topological ones. A last perspective is to use similar techniques (considering the 3D cellular description of a video) in other fields of video processing such that objects or activities recognition.

Acknowledgement: This work has been partially supported by the French National Agency (ANR), project SOLSTICE ANR-13-BS02-01.

References

1. Pets2001 dataset. http://www.cvg.rdg.ac.uk/pets2001/.
2. P.S. Aleksandrov. *Elementary concepts of topology*. Dover Publications Inc., New York, 1961.
3. O. Barnich and M. V. Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):17091724, 2011.
4. D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. An algorithm for recovering camouflage errors on moving people. In *Int. Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR)*, pages 365–374, 2010.
5. D. Conte, P. Foggia, G. Percannella, F. Tufano, and Mario Vento. An experimental evaluation of foreground detection algorithms in real scenes. *EURASIP Journal on Advances in Signal Processing*, 2010(373941):1–11, 2010.
6. D. Conte, P. Foggia, M. Petretta, F. Tufano, and M. Vento. Evaluation and improvements of a real-time background subtraction method. In *Int. Conf. on Image Analysis and Recognition (ICIAR)*, volume LNCS 3656, pages 1234–1241, 2005.
7. G. Damiand. Topological model for 3d image representation: Definition and incremental extraction algorithm. *CVIU*, 109(3):260–289, March 2008.
8. G. Damiand. Combinatorial maps. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.9 edition, 2010.
9. G. Damiand, Y. Bertrand, and C. Fiorio. Topological model for two-dimensional image representation: Definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2):111–154, February 2004.
10. A. Dupas and G. Damiand. Region merging with topological control. *Discrete Applied Mathematics*, 157(16):3435–3446, August 2009.
11. M. Haque and M. Murshed. Perception-inspired background subtraction. *IEEE Trans. on Circuits and Systems for Video Technology*, 23(12):2127–2140, 2013.
12. V.A. Kovalevsky. Finite topology as applied to image analysis. *CVGIP*, 46:141–161, 1989.
13. P. Lienhardt. N-Dimensional generalized combinatorial maps and cellular quasi-manifolds. *Int. Journal of Computational Geometry and Applications*, 4(3):275–324, 1994.
14. J.R. Munkres. *Elements of Algebraic Topology*. Perseus Books, 1984.
15. A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara. Detecting moving shadows: algorithms and evaluation. *IEEE Trans. on PAMI*, 25(7):918–923, 2003.

16. A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26(1):24–33, 1974.
17. C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on PAMI*, 22-8:747–757, 2000.
18. C. Wolf and J.-M. Jolion. Model based text detection in images and videos: a learning approach. Technical report, LIRIS INSA de Lyon, 2004.