



# Unsupervised dense crowd detection by multiscale texture analysis

Fagette Antoine, Nicolas Courty, Daniel Racoceanu, Jean-Yves Dufour

► **To cite this version:**

Fagette Antoine, Nicolas Courty, Daniel Racoceanu, Jean-Yves Dufour. Unsupervised dense crowd detection by multiscale texture analysis. Pattern Recognition Letters, Elsevier, 2013, pp.1-27. <hal-00904210>

**HAL Id: hal-00904210**

**<https://hal.archives-ouvertes.fr/hal-00904210>**

Submitted on 14 Nov 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Unsupervised Dense Crowd Detection by Multiscale Texture Analysis

Antoine Fagette<sup>a,d</sup>, Nicolas Courty<sup>b</sup>, Daniel Racocceanu<sup>c,d</sup>, Jean-Yves  
Dufour<sup>e</sup>

<sup>a</sup>*Thales Solutions Asia Pte Ltd*  
28 Changi North Rise - Singapore 498755, Singapore  
antoine.fagette@thalesgroup.com

Tel: +65 8200 8957 - Fax: +65 6478 9920

<sup>b</sup>*IRISA - Université de Bretagne Sud*  
Campus de Tohannic - 56000 Vannes, France

<sup>c</sup>*University Pierre and Marie Curie*  
4 Place Jussieu, 75005 Paris, France

<sup>d</sup>*CNRS, IPAL UMI 2955*

1 Fusionopolis Way, #10-18 Connexis (South Tower) - Singapore 138632, Singapore

<sup>e</sup>*Thales Services, Campus Polytechnique*  
1 Avenue Augustin Fresnel - 91767 Palaiseau Cedex, France

---

## Abstract

This study introduces a totally unsupervised method for the detection and location of dense crowds in images without context-awareness. With the perspective of setting up fully autonomous video-surveillance systems, automatic detection and location of crowds is a crucial step that is going to point which areas of the image have to be analyzed. After retrieving multiscale texture-related feature vectors from the image, a binary classification is conducted to determine which parts of the image belong to the crowd and which to the background. The algorithm presented can be operated on images without any prior knowledge of any kind and is totally unsupervised.

*Keywords:* dense crowd, segmentation, feature extraction, texture analysis, quadtree, diffusion maps, multiscale

---

## 1. Introduction

Crowd monitoring has become a major concern of the beginning of the 21<sup>st</sup> century. With the increasing number of CCTV networks in public areas, the enhancement of the computing power of modern computers and the progress made these past decades in computer sciences and computer vision in particular, the possibility to entrust an automatic system with the security and the monitoring of events involving large crowds is within reach.

This paper is dealing with the problem of detection and location of a dense crowd in the image. It focuses on a method that does not need any training set nor any prior knowledge of any kind on the context from where the picture or the video has been taken. Our method is based on the assumption that a crowd is visually identified by a type of texture characterized by great variations of the color vectors as well as of the orientations of the borders. The features that we extract from the image are representative of these variations. We are also taking into account the multiscale aspect of a crowd by appending several feature vectors computed with several sizes of spatial neighborhood, thus forming a multiscale feature vector. Unlike the previous studies on this same topic, detailed in Section 2, our work, described in Section 3, aims at providing a method that is totally unsupervised and independent of the shooting conditions. It is based on the appearance of the crowd and not exclusively on its motion. We are therefore able to locate dynamic as well as static crowds on images taken from cameras and poses we know nothing of. We have run tests on static images of both synthetic and real scenes within which the ground truth is known. The results of this experimentation,

25 detailed in Section 4, prove that our method is successful at detecting dense  
26 crowds.

## 27 **2. State of the art**

28 When it comes to crowd detection, several methods have been developed,  
29 each valid for its own context, mainly depending on the density of the crowd  
30 to be detected and its distance to the camera. The goal of this Section is to  
31 give a brief overview of the different methods that are used for the detection  
32 and the location of crowds in video-surveillance.

33 Through different surveys on crowd analysis in general ([1] and [2]) and  
34 on pedestrian detection and human visual analysis in particular ([3] and  
35 [4]), crowd detection can be generalized into three main ways: to detect the  
36 pedestrians themselves, to proceed with background subtraction methods  
37 and/or to assume that in the observed scene every moving object is part of  
38 a crowd.

39 The process of recognizing each pedestrian of a crowd to detect assumes  
40 that, in the image, it is possible to segment each pedestrian from the back-  
41 ground or from a group of pedestrians. It requires from the camera to be  
42 close enough to have a number of pixels per pedestrian high enough to run  
43 the algorithm but has the advantage to work theoretically well with still  
44 images as well as with mobile cameras. Nevertheless, this method reaches  
45 its limits when the crowd is too dense and the number of occlusions too  
46 important for the algorithm to match its human model with the objects it  
47 detects. It may also fail when the relative motion between the camera and  
48 the pedestrians is too chaotic to enable a good capture of the phenomenon.

49 The pedestrian recognition is used by Leibe *et al.* in [5] where they combine  
50 local and global cues via a probabilistic top-down segmentation to identify  
51 the human beings. Wu and Nevatia in [6] use edgelet features to segment  
52 pedestrians even partially occluded, and so do Lin *et al.* in [7] by generating  
53 a body part template to match as well as possible the detected shapes among  
54 the crowd. Dalal and Triggs, in [8], prove the efficiency of the Histograms  
55 of Oriented Gradients to detect a pedestrian in the image. Finally, Tu *et*  
56 *al.* in [9] detect heads and shoulders as a first guess on the positions of the  
57 pedestrians and then associate every squared sub-part of the image to the  
58 most probable pedestrian or to the background.

59 Another way of proceeding is to use a background subtraction algorithm.  
60 This method goes with the assumption that each object that is not part of  
61 the background is going to be a pedestrian or that an algorithm is able,  
62 afterwards, to classify the detected objects as pedestrian or non-pedestrian  
63 (in the latter, the techniques are quite close to those described in the previous  
64 paragraph). This technique is not able to deal with video streams taken  
65 from a mobile camera. However, it is very efficient to monitor places such as  
66 pedestrian zones, stadiums or fairs where the environment is well controlled  
67 and only pedestrians are expected. Dong in [10] manages to detect human  
68 beings even with some occlusions by matching the shapes detected from the  
69 background subtraction with models and combined models. Wang and Yung  
70 in [11] match 3D human models with silhouette obtained via background  
71 subtraction and helps to find the best position for his models by locating  
72 the heads of the pedestrians through a head detector.

73 The third method that is commonly used for crowd detection assumes

74 that a crowd is never static and that it evolves in an environment that is, it-  
75 self, non-dynamic. Therefore, by using for example an optical flow algorithm,  
76 one can detect the areas of the image where something is moving and deduct  
77 the position of the crowd. This process finds its limits when the camera  
78 itself is moving (beyond possible correction) or in the case when the crowd is  
79 standing still (*e.g.* sit-ins, commemorations, etc.). Boghossian and Velastin  
80 in [12] use an optical flow algorithm to get the motion, introduce continu-  
81 ity properties to remove the noise and detect slow movements by running  
82 this optical flow algorithm between two frames separated by several others.  
83 Reisman *et al.* in [13] use an optical flow algorithm as well and detect the  
84 movements that can only be made by a human crowd with specific classi-  
85 fiers, thus eliminating the vehicles. Rabaud and Belongie in [14] use both  
86 the motion and the fact that a crowd is composed of objects that are similar  
87 in shape to locate the crowd and its pedestrians. Finally, Ali and Shah in  
88 [15] use a set of particles combined with an optical flow algorithm to detect  
89 the flow and to go further by identifying its instabilities.

90 Recently, some work has been done to detect and locate a crowd in the  
91 image using texture analysis. Indeed, a dense crowd has a very particular  
92 aspect, made of a patchwork of colors, that lead researchers to consider this  
93 feature to segment a crowd from the background. This is precisely the idea  
94 that is exploited by Manfredi *et al.* in [16] to detect and locate groups of  
95 pedestrians in open spaces, using classification. In [17], Rodriguez *et al.*  
96 combine a head detector, belonging to the first technique described above,  
97 to the results given by density estimation to robustify their crowd detection  
98 and pedestrian location.

99 The method described in this paper belongs to this last category of meth-  
 100 ods. Following Manfredi *et al.*, we use a texture-based approach. However,  
 101 our priority is given to the unsupervised learning to separate the crowd from  
 102 the background.

### 103 3. Overview of the method

104 Our method aims at detecting large dense crowds in which it is impossible  
 105 to segment each individual and without any training dataset. It is based on  
 106 a texture analysis technique. First, from each pixel of the image, features  
 107 relevant to the crowd texture are extracted. These features are stored in  
 108 a vector of features attached to the described pixel. Then the pixels are  
 109 classified either as belonging to the crowd or to the background. This binary  
 110 classification is performed using a diffusion map with the extracted features as  
 111 data. This last operation raises a problem of time and volume of computation  
 112 that leads us to consider reducing the amount of data to be treated.

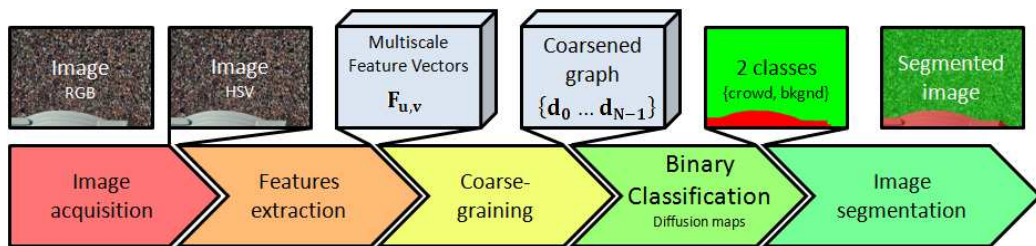


Figure 1: Overview of the method

#### 113 3.1. Features extraction

114 We base our work on the assumptions that the color vectors describing a  
 115 crowd show great spatial variations and that the borders of objects (namely

116 the pedestrians) are orientated in all possible directions. These assumptions  
 117 are to be opposed to the one assuming that the background tends to be  
 118 more uniform on wider areas. Retrieving features representing the level of  
 119 variation of the color vector and of the orientation of the borders at each  
 120 pixel of the image gives us a good level of information on the presence or not  
 121 of the crowd on this pixel.

### 122 3.1.1. Window of observation dimensioning

123 The study deals with images taken by non-calibrated cameras. There-  
 124 fore, the size of the sample from which the features are extracted cannot be  
 125 determined uniquely. This problem is avoided by considering that each size  
 126 of this window of observation gives a different value of the same feature.

Therefore, if we suppose that we extract  $n$  features using  $m$  sizes of win-  
 dow of observation  $\{r_1, \dots, r_m\}$ , then, at every pixel  $(u, v)$  of the image  $I$ ,  
 the value of the  $i^{th}$  feature for the  $j^{th}$  size of window of observation  $r_j$  is  $f_{u,v}^{i,r_j}$   
 and we obtain the multiscale feature vector  $\mathbf{F}_{\mathbf{u},\mathbf{v}}$ :

$$\forall (u, v) \in I, \mathbf{F}_{\mathbf{u},\mathbf{v}} = \begin{pmatrix} f_{u,v}^{1,r_1} \\ \vdots \\ f_{u,v}^{1,r_m} \\ \vdots \\ f_{u,v}^{n,r_1} \\ \vdots \\ f_{u,v}^{n,r_m} \end{pmatrix} \quad (1)$$

### 127 3.1.2. Features definition

128 Three types of features are implemented for this study: the Laplacian  
 129 of Gaussian (LoG), the entropy and the Histogram of Oriented Gradients



130 (HOG). As we want to use the information carried by the color itself, we  
 131 choose to work within the HSV colorspace. We use the hue component  $I_h$  in  
 132 radians to compute the LoG and the entropy and weight each of these two  
 133 features with the saturation component  $I_s$ . The HOG is computed with the  
 134 value component  $I_v$ .

Therefore, for each pixel  $(u, v)$  of the image  $I$  and for each size of window of observation  $r_j$ , the computation of these three features gives respectively  $f_{u,v}^{1,r_j}$ ,  $f_{u,v}^{2,r_j}$  and  $f_{u,v}^{3,r_j}$ . Because of the angular nature of the terms of  $I_h$ , we use their complex values  $\tilde{I}_h$  and the smallest angular difference  $\Delta_{\theta_1}^{\theta_2}$  for the computation of the LoG:

$$\forall (u, v) \in I, \tilde{I}_h(u, v) = \exp(i \cdot I_h(u, v)) \quad (2)$$

$$\Delta_{\theta_1}^{\theta_2} = (\theta_2 - \theta_1 + \pi) \bmod(2\pi) - \pi \quad (3)$$

In the following,  $\otimes$  denotes a term-by-term multiplication,  $*$  a convolution:  
 $\forall (u, v) \in I,$

$$f_{u,v}^{1,r_j} = (G_{\sigma_j} * LoG_I)(u, v) \quad (4)$$

$$f_{u,v}^{2,r_j} = \left( - \sum_{k=0}^b \frac{G_{\sigma_j} * B_k \otimes \log_2(G_{\sigma_j} * B_k)}{\log_2(N)} \right) \otimes (G_{\sigma_j} * I_s)^\beta(u, v) \quad (5)$$

$$f_{u,v}^{3,r_j} = \|\mathbf{f}_{\mathbf{u},\mathbf{v}}^{3,r_j}\| \quad (6)$$

with  $LoG_I$  the customized LoG:

$$LoG_I(u, v) = \sum_{U=u-r_j}^{u+r_j} \sum_{V=v-r_j}^{v+r_j} \Delta_{arg((G_{\sigma_1} * \tilde{I}_h)(U,V))}^{arg((G_{\sigma_1} * \tilde{I}_h)(u,v))} \cdot (I_s(u, v) \cdot I_s(U, V))^\alpha \quad (7)$$

and  $G_{\sigma_j}$  and  $G_{\sigma_1}$ , the normalized gaussian filters defined respectively by  $\sigma_j = \frac{r_j}{3}$  and  $\sigma_1 = \frac{1}{3}$ .

$B_k$  is the binary image corresponding to the  $k^{th}$  bin of the histogram of  $b$  bins used to compute the entropy:

$$\forall (u, v) \in I_h, B_k(u, v) = \begin{cases} 1 & \text{if } \frac{2k\pi}{b} \leq I_h(u, v) < \frac{2(k+1)\pi}{b}, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

135 and  $\mathbf{f}_{\mathbf{u}, \mathbf{v}}^{\mathbf{3}, \mathbf{r}, \mathbf{j}}$  is the result of the convolution of each bin of the HOG performed  
 136 on  $I_v$  by the gaussian filter  $G_{\sigma_j}$  at pixel  $(u, v)$ . It is a vector of dimension  $d$  if  
 137 the orientation is binned into  $d$  bins. Usually, and it is our case,  $d = 8$ . For  
 138 more details on the HOG, the reader may refer to [8] by Dalal and Triggs.

139 With the experience, we choose:  $\alpha = \beta = 0.25$ .

140 In the end, we obtain, for each pixel  $(u, v)$  of the image, a multiscale  
 141 feature vector of dimension  $3 \cdot m$ . This multiscale feature vector is then  
 142 normalized in order to ensure the coherence of the data.

### 143 3.2. Clustering using a diffusion map and classification

144 Once the multiscale feature vectors have been computed, pixels have to  
 145 be labeled as the crowd or the background. As we are focusing on an unsu-  
 146 pervised method we are looking for a clustering algorithm that could separate  
 147 the pixels according to the value of their attached multiscale feature vectors.  
 148 However, as opposed to traditional methods, such as K-means, that are often  
 149 considering only the distance between two data to determine whether they  
 150 belong or not to the same cluster, we want to use both the lengths and the  
 151 density of the different paths between these data, as suggested by Farbman  
 152 *et al.* in [19]. This can be achieved using a spectral clustering.

153 It is also reasonable to assume that the different densities observed in  
 154 the scene with different illumination conditions will lead to different feature

155 vectors that will nevertheless belong to the same manifold. The diffusion  
156 maps, as initially introduced by Coifman and Lafon in [20], are an interesting  
157 tool that preserves the similarity between those samples while providing a  
158 low-dimensional embedding which encodes the structural information of the  
159 manifold. Beside the spectral clustering aspect, the diffusion maps algorithm  
160 is also embedding a parameter, the diffusion parameter, hereafter noted  $t$ ,  
161 that can be seen as a scaling parameter. Scrolling this parameter from one  
162 value to another can strengthen or weaken the relationship existing between  
163 two data points. The diffusion maps algorithm is therefore used to divide  
164 the multiscale feature vectors in two clusters.

165 A good introduction to the diffusion maps can be read in the paper of de  
166 la Porte *et al.* [21]. The idea of using this approach for clustering is described  
167 by Nadler *et al.* in [22]. We base our work with the diffusion maps on these  
168 papers, using a gaussian kernel to map the multiscale feature vectors in the  
169 diffusion space.

170 Then, for both clusters, the mean vector of all the attached multiscale  
171 feature vectors is computed. The one with the highest norm gets the crowd  
172 label, the other the background one.

173 The diffusion maps technique is a powerful tool yet subject to some lim-  
174 itations regarding the amount of data to be processed. Using the algorithm  
175 directly on the multiscale feature vectors associated to each pixel of an im-  
176 age with a 4CIF resolution implies clustering 405504 elements. The diffusion  
177 matrix holds therefore more than 160 billion values and the complexity is  
178 skyrocketing. The amount of data to be processed has to be limited in order  
179 to reduce the time and volume of computation.

180 This problem has been addressed in various ways. Among them, an  
 181 approach described by Fowlkes *et al.* in [23] and used by Farbman *et al.* in  
 182 [19] is based on the Nyström method. It approximates the eigenvalues and  
 183 eigenvectors of the diffusion matrix using a smaller sample of the data, taken  
 184 randomly. Afterwards, it computes the missing points using the Nyström  
 185 extension. Another method, developed by Lafon and Lee in [24] is, with  
 186 the same idea of sub-sampling, to regroup data that are similar into clusters  
 187 and to build the diffusion map with these clusters and no longer with the  
 188 data themselves. The coarse-grained version of the original diffusion map is  
 189 supposed to have the same spectral properties provided that the choice of  
 190 the clusters has been made correctly.

191 Our approach is different in the sense that it tries to coarsen the graph  
 192 while considering the spatial relationship between the elements of the graph.  
 193 It is based on the computation of a quadtree.

### 194 3.3. Quadtree computation

195 The difficulty with quadtrees lies into finding the criterion that will indi-  
 196 cate if a region of rank  $k$  contains data that are homogeneous enough or else  
 197 if it needs to be split into four sub-regions of rank  $k + 1$ .

In our case, the data used for each sub-region is the mean vector of all  
 the multiscale feature vectors attached to the pixels contained in this sub-  
 region. We note  $M_{0_i}$  the  $i^{th}$  region of rank  $k$ ,  $M_{1_i}$ ,  $M_{2_i}$ ,  $M_{3_i}$  and  $M_{4_i}$  its four  
 sub-regions of rank  $k + 1$  and  $\mathbf{m}_{0_i}$ ,  $\mathbf{m}_{1_i}$ ,  $\mathbf{m}_{2_i}$ ,  $\mathbf{m}_{3_i}$  and  $\mathbf{m}_{4_i}$  their respective  
 data. The level of homogeneity  $H$  is then evaluated using  $\mathbf{V}_{0_i}$ , the variance  
 vector of the four data  $\mathbf{m}_{1_i}$ ,  $\mathbf{m}_{2_i}$ ,  $\mathbf{m}_{3_i}$  and  $\mathbf{m}_{4_i}$  of the sub-regions, and  $\mathbf{V}_{\mathbf{I}}$

the variance vector of all the multiscale feature vectors of the image.

$$\mathbf{V}_{\mathbf{0}_i} = Var(\{\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \mathbf{m}_{3_i}, \mathbf{m}_{4_i}\}) \quad (9)$$

$$\mathbf{V}_{\mathbf{I}} = Var(\{\mathbf{F}_{\mathbf{u},\mathbf{v}}\}_{(u,v) \in I}) \quad (10)$$

$$H = \begin{cases} True & \text{if } \forall l \in \{0, \dots, 3 \cdot m - 1\}, \mathbf{V}_{\mathbf{0}_i}[l] < \alpha \cdot \mathbf{V}_{\mathbf{I}}[l] \\ False & \text{otherwise.} \end{cases} \quad (11)$$

With  $\alpha$  a parameter set by the user. With the experience, we choose  $\alpha$  between 0% and 20%. If  $H$  is false, the region  $M_{0_i}$  is considered as not homogeneous, it is split into the four sub-regions  $M_{1_i}$ ,  $M_{2_i}$ ,  $M_{3_i}$  and  $M_{4_i}$  which level of homogeneity is going to be tested at the next iteration of  $k$ . If  $H$  is true, the region  $M_{0_i}$  is considered as homogeneous, it will not be further split and becomes a leaf of our quadtree. If  $M_{0_i}$  is the  $j^{th}$  leaf of the quadtree we note:

$$L_j = M_{0_i} \quad (12)$$

$$\mathbf{d}_j = \mathbf{m}_{0_i} \quad (13)$$

198 In the end, the quadtree is composed of  $N$  leaves  $\{L_0, \dots, L_{N-1}\}$  with  
 199 their respective data  $\{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ . These are the data used to compute  
 200 the diffusion map and perform the binary classification described in 3.2.

## 201 4. Validation

202 For the validation, two datasets of images<sup>1</sup> are used. The first one is com-  
 203 posed of images that have been retrieved from Google Images and manually

---

<sup>1</sup>These datasets are available for download at <http://www.ipal.cnrs.fr/download>

204 annotated. These images show large dense crowds in urban environments,  
205 presenting challenging backgrounds with textures ranging from near-regular  
206 to near-stochastic. Their resolution is diverse and they are all JPEG-encoded.  
207 The second dataset was generated with the synthetic crowd generator Ago-  
208 raset built by Allain *et al.* and described in [26]. This second dataset is  
209 automatically annotated by the crowd generator. The annotation for both  
210 datasets is the following: green represents the crowd, red the background.  
211 The same colors are used to display our results.

212 In Section 3 three parameters of the algorithm have been declared:

- 213 •  $m$ , the number of sizes of window of observation and the value of these  
214 sizes (see Subsection 3.1)
- 215 •  $\alpha$ , the quadtree parameter for the homogeneity (see Subsection 3.3)
- 216 •  $t$ , the diffusion parameter of the diffusion maps algorithm (see Subsec-  
217 tion 3.2)

218 In this Section, we give the conclusions of a series of tests that study the  
219 influence of each of these parameters on the performances of the algorithm.

220 Then, taking advantage of the capability of the crowd generator to pro-  
221 duce different textures, we challenged our algorithm into finding a crowd on  
222 different backgrounds. The results are illustrated and commented in this  
223 Section.

224 We compare also the efficiency of our algorithm to the efficiency of the  
225 traditional K-means.

226 Finally, we provide some of the results produced by our method as well  
227 as an evaluation of its performances on two different sets of images issued

228 from our datasets.

229 To evaluate the performance of our algorithm we use the F-score indicator  
230 and assign Positive to the crowd class and Negative to the background class.

#### 231 4.1. Influence of $m$

232 The parameter  $m$  is embedding two characteristics: the number of win-  
233 dows of observation  $m$  itself and the sizes of these windows of observation  
234  $\{r_1, \dots, r_m\}$ . We have therefore conducted two tests here. For these two  
235 tests, we fixed  $\alpha = 10\%$  and  $t = 1$ .

236 First, we chose five intervals of values ( $r_1 = 1$  and  $r_m = 5, 10, 20, 50$   
237 or 100) with a unit step between each values,  $r_{i+1} - r_i = 1$ . The results  
238 confirmed the intuition that the range covered by the number of windows  
239 of observation has a positive impact on the results given by the algorithm.  
240 The wider the range, the better the results. However, in order to have a  
241 reasonable size for our multiscale feature vectors, we limit the range covered  
242 from  $r_1 = 1$  to  $r_m = 50$ .

243 In the second test, we fixed  $r_1 = 1$  and  $r_m = 50$  but we took five different  
244 values for the step: 1, 2, 3, 5 and 10. The results showed that the value  
245 of the step has no significant influence on the performance, provided that it  
246 allows a good sampling of the range  $r_1$  to  $r_m$ .

247 From the results of these two tests, we choose to keep  $m = 5$  so that  
248  $r_1 = 1$  and  $r_{i+1} - r_i = 10$ .

#### 249 4.2. Influence of $\alpha$

250 The parameter  $\alpha$  determines the level of subdivision of our quadtree,  
251 therefore the number of leaves and so the complexity of the diffusion maps

252 algorithm that is run afterwards. For this set of tests,  $\alpha$  took successively  
253 the following values: 0%, 5%, 10%, 15% and 20%. We fixed  $t = 1$  and  $m = 5$   
254 so that  $r_1 = 1$  and  $r_{i+1} - r_i = 10$ . As expected, the results showed that the  
255 smaller  $\alpha$  is, the more precise the segmentation is but often at a level that is  
256 not wished. For scenes with a low complexity,  $\alpha$  has little influence however,  
257 when the complexity grows, it is preferable to have  $\alpha$  not too small in order  
258 to avoid an over-segmentation.

259 Moreover, the parameter  $\alpha$  is a bargain parameter. We are trading preci-  
260 sion in order to enhance the speed of the diffusion maps part of the algorithm.  
261 A balance has to be found and from the tests we have conducted, we choose  
262 to keep  $\alpha$  between 5% and 10%. It is hard to quantify the time that is saved  
263 as a function of  $\alpha$  because the level of subdivision of the tree depends on  
264 the complexity of the image itself. However, we have set the smallest size a  
265 leaf can take to 5 pixels. Below, the information of homogeneity does not  
266 make sense any more. Therefore, should the quadtree go to its maximum  
267 allowed subdivision, for a 4CIF image, it would have 4096 leaves, *i.e.* 4096  
268 data to be treated by the diffusion maps algorithm to be compared to the  
269 405504 pixels that would have to be clustered if the coarse-graining part  
270 was skipped. It represents almost a hundred times less data, therefore the  
271 number of operation is divided by  $10^6$ .

### 272 4.3. Influence of $t$

273 The diffusion parameter  $t$  rules the proximity between the data in the dif-  
274 fusion space and has an influence on the final K-means clustering performed  
275 in the diffusion space. For this batch of tests, we have fixed  $\alpha = 10\%$  and  
276  $m = 5$  so that  $r_1 = 1$  and  $r_{i+1} - r_i = 10$  and  $t$  took the following values: 1,



277 2, 3, 5, 10, 20 and 100.

278 The results showed that the parameter  $t$  gives optimum results for  $t = 1$ .  
279 For  $t$  greater than 1, the diffusion map algorithm tends to bring the data too  
280 close from each other in the diffusion space, which leads to a bad separation  
281 of the data. That is the reason why we choose to keep  $t = 1$ .

#### 282 4.4. *Influence of the background*

283 In order to test the robustness of our detection method under different  
284 background conditions, we used the synthetic ground truth of Figure 2a to  
285 test various background conditions. We chose a progression in the complexity  
286 of the texture ranging from quasi-flat to quasi-noise, by gradually increasing  
287 the level of noise while downgrading the geometric structure of the texture.  
288 The first image Figure 2b corresponds to a flat color background and, as  
289 such, is the simplest. In the second image Figure 2c, the scene is illuminated  
290 with a global illumination model based on photon maps producing shading  
291 effects on the ground. In Figure 2d, to the same illumination model is added  
292 a virtual sun which casts shadows over the ground. Next, volumetric textures  
293 are used with various levels of geometric structures. First a marble texture is  
294 used on Figure 2e, then a Brownian noise on Figure 2f, and finally a Markov  
295 Random Field over the 3 color components on Figure 2g. This last image is  
296 assumed to be the most challenging for our algorithm.

297 As expected, the performances are quite good on the first three images.  
298 The algorithm achieves the detection of the crowd. However, it does not  
299 perform a complete detection of the isolated pedestrians, thus downgrading  
300 the F-score. The result displayed on Figure 2d shows that the algorithm  
301 is sensitive to the shadows but classifies those belonging to the pedestrians

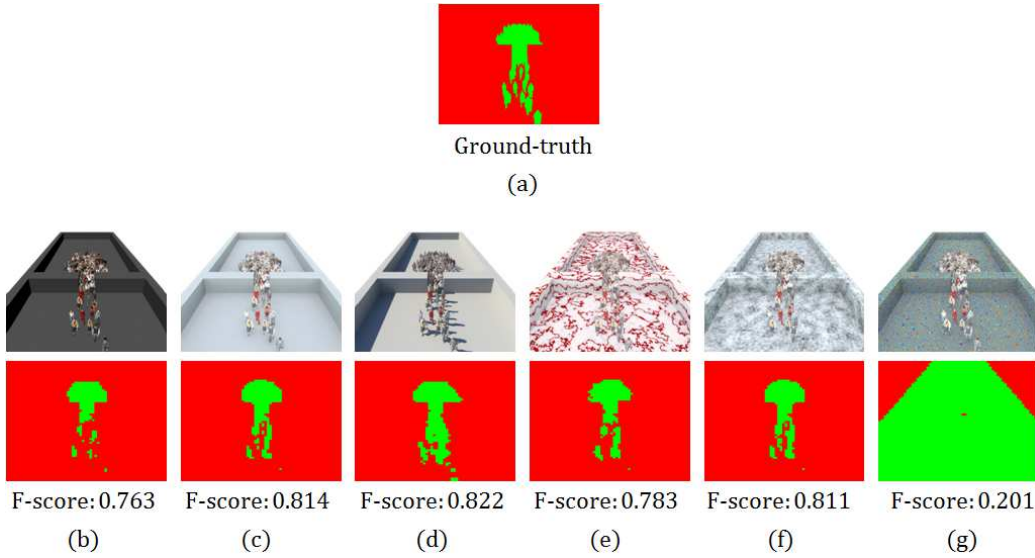


Figure 2: Comparison of the results obtained on a same image of crowd but with various textures for the background.

302 into the crowd and those belonging to the walls as part of the background.  
 303 On the marble and Brownian noise backgrounds, the algorithm proves to  
 304 be very efficient with results comparable to those obtained on the simpler  
 305 backgrounds.

306 Finally, our algorithm is out-challenged by the Markov Random Field  
 307 background. Due to the type of features used by the algorithm, this last  
 308 result was expected.

309 *4.5. Comparison with the traditional K-means*

310 In this Subsection we compare the performances of the K-means algorithm  
 311 with the performances of our algorithm. We used the K-means algorithm to  
 312 separate the  $m$ -dimension space containing the multiscale feature vectors into

313 two clusters. For this set of tests we use four images, two taken from Google  
 314 Images and two synthetic.

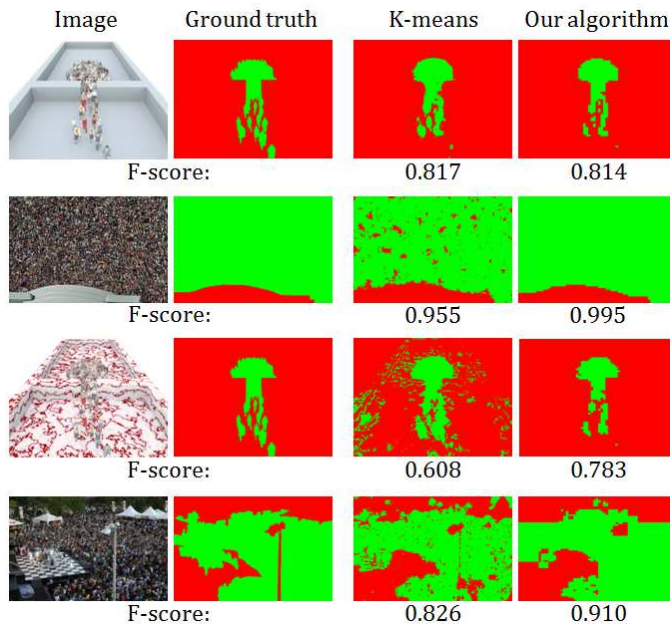


Figure 3: Comparison of the results obtained via a K-Means algorithm with those given by our method.

315 The results, displayed on Figure 3, show that the performances of the  
 316 K-means algorithm are comparable with those of our algorithm for scenes of  
 317 a lower complexity. However, as the complexity grows, the performances of  
 318 the K-means algorithm decrease significantly whereas those of our algorithm  
 319 remain higher.

320 Moreover, the K-means algorithm considers all the pixels independently  
 321 from each others. This leads to the non-regularity of the two classes (espe-  
 322 cially for the crowd class). Our algorithm avoids this problem which gives  
 323 results closer to the human perception.

324 *4.6. More results and performances*

325 This Subsection is providing more results and performances of our al-  
326 gorithm on two sets of images. The first set is composed of ten images  
327 synthesized by Agoraset, the second has ten images from Google Images.

328 In the results that we are providing on Figure 4, one can see that our  
329 algorithm comes with good performances and detects efficiently the crowds  
330 on the various images that have been used. The F-score computed for the  
331 images from the synthetic dataset indicates that our results are less accurate  
332 on those images than on the ones from the Google Images dataset. This  
333 can be explained by two factors. First, the ground truth for the synthetic  
334 dataset is computed by the simulator itself which segments very precisely  
335 each pedestrian. On the dataset taken from the Internet, the ground truth  
336 has been annotated manually and is therefore subject to the simplifications  
337 a human-being tends to make naturally. Since our algorithm works with a  
338 quadtree, it mimics this behavior. The second reason that explains the lower  
339 performances on the synthetic dataset is that this dataset contains images  
340 with isolated pedestrians. Even though the algorithm achieves to detect  
341 these pedestrians, it fails most of the time to detect them entirely causing  
342 the F-score to drop down.

343 We would like also to emphasize the problem of the subjectivity inherent  
344 to the definition of a crowd. It is indeed debatable until what extent a  
345 group of human being can be considered as a crowd or as part of it. On an  
346 image, are the inter-individual spaces part of the crowds, or should they be  
347 considered as part of the background? Furthermore, are the persons sitting  
348 at the terrace of a cafe part of the crowd gathered on the street right in

Synthetic dataset				Internet dataset			
Image	Ground truth	Result	F-score	Image	Ground truth	Result	F-score
			0.956				0.995
			0.947				0.968
			0.940				0.958
			0.876				0.957
			0.875				0.952
			0.793				0.943
			0.776				0.941
			0.774				0.910
			0.737				0.904
			0.712				0.821

Figure 4: More results and performances: on the left the synthetic dataset, on the right the dataset constituted with images taken from Google Images.

349 front of them? The result shown on Figure 5a indicates that our algorithm  
350 considers that both the inter-individual spaces as well as the people sitting  
351 at the terrace of a cafe are part of the crowd.

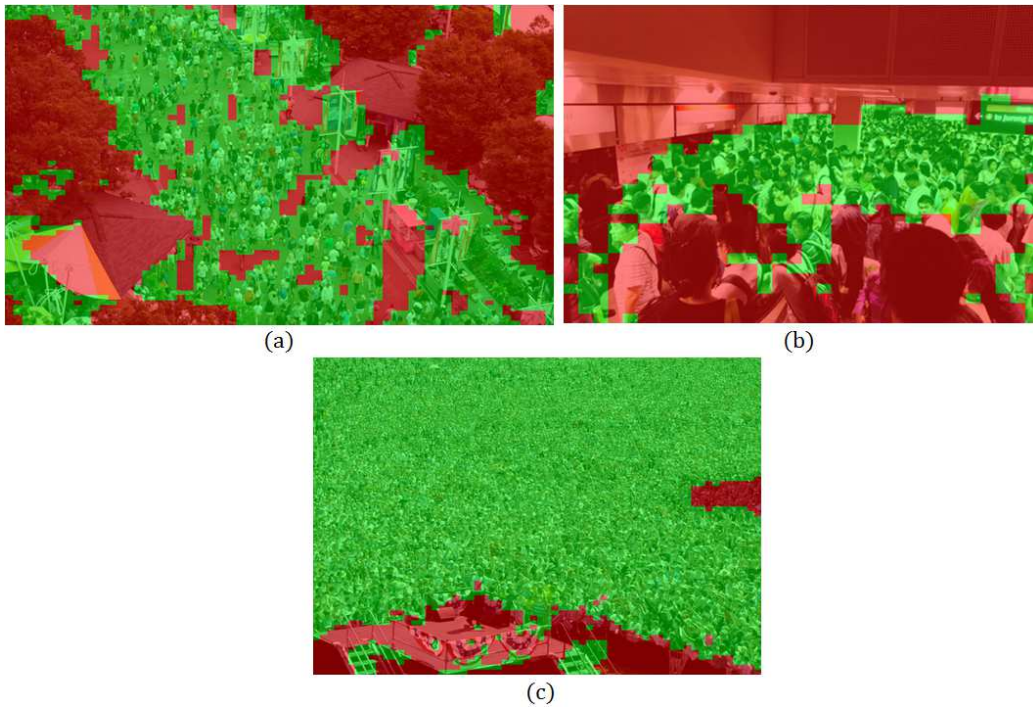


Figure 5: Example of the limitations encountered by our algorithm.

352 Moreover, as noted in 4.4, the background has an influence on the per-  
353 formances of the algorithm. It has been shown that when the background  
354 has a texture too chaotic, the algorithm fails to segment the crowd properly  
355 and tends to allocate elements of the background to the crowd class, as for  
356 example colorful flags or signboards. Conversely, elements of the crowd can  
357 be confused and sorted into the background class if their texture is similar to  
358 a background texture. These two cases are illustrated on Figure 5a and 5c.

359 The entanglement of beams combined with the steps at the bottom of Figure  
360 5c or the furniture present on the right side of Figure 5a show a structural  
361 information too close to the crowd for our algorithm. The shadow on the  
362 right part of Figure 5c tricks it too. Experience shows that with a higher  $\alpha$   
363 this last problem disappears.

364 Finally, another limitation of our algorithm is displayed on Figure 5b.  
365 The image was shot in a train station with a low-positioned camera resulting  
366 in an almost horizontal field of view. As a result, people in the foreground  
367 appear much bigger than those in the background. Our algorithm detects  
368 successfully the crowd except for the heads closer to the camera which are  
369 classified as part of the background. This is due to the fact that they have  
370 the same texture as background objects.

## 371 5. Conclusion

372 In this paper, we have combined three kinds of features, extracted at  
373 different scales of observation, in order to build a high dimensional multiscale  
374 feature vector for each pixel of the image. To separate these multiscale feature  
375 vectors into two classes, we have used the diffusion distance instead of the  
376 traditional Euclidean distance because we wanted to consider the length and  
377 the density of the path between our data. Finally, to optimize the time  
378 and volume of computation, we have explored a new technique of coarse-  
379 graining using a quadtree. With the combination of these different blocks,  
380 we are providing a new and fully unsupervised crowd detection and location  
381 algorithm. To conclude our paper, we would like to point out some interesting  
382 directions of research for any future work on this method.



383 First, it is reasonable to think that the three types of features that we are  
384 using so far are not the only ones that are relevant for the targeted purpose.  
385 Therefore, our multiscale feature vector could be enriched with some new  
386 features. We have focused in this paper on static images, however, as it has  
387 been reminded in Section 2, motion is also a very important feature of a  
388 crowd. We are therefore thinking of including some dynamic features to our  
389 multiscale feature vectors.

390 Second, the way to compute the multiscale feature vector can most cer-  
391 tainly be improved in two ways: each type of feature can be explored within  
392 its own range of size of windows of observation. Some features are better  
393 used locally, some others are more relevant when computed at a larger scale.  
394 Moreover, the weight attributed to each type of feature is a point that de-  
395 serves to be studied furthermore.

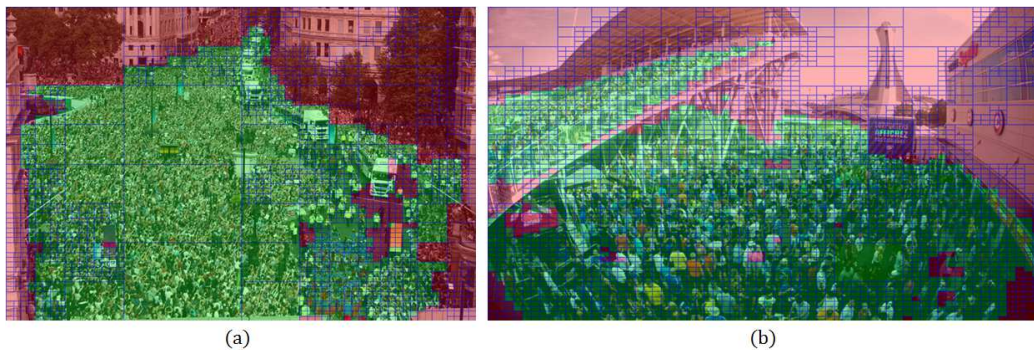


Figure 6: Quadtree and segmentation superimposed: one can see how the algorithm approximates the crowd area.

396 Finally, we believe that if a human-being is able to detect and locate a  
397 crowd in an image in spite of its resemblance to other natural phenomenon,



398 it is because of his capacity to extract from this image some higher-level  
399 information. Raising the problem at a semantic level would provide us with  
400 a context that could help us quantify the probability of dealing with a crowd  
401 or not.

## 402 **References**

- 403 [1] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, L.-Q. Xu, Crowd  
404 analysis: a survey, *Machine Vision and Applications* 19 (2008) 345–357.
- 405 [2] J. C. Silveira Jacques Jr., S. Raupp Musse, C. Rosito Jung, Crowd anal-  
406 ysis using computer vision techniques, *IEEE Signal Processing Magazine*  
407 27 (2010) 66–77.
- 408 [3] D. Gerónimo, A. López, A. D. Sappa, Computer vision approaches to  
409 pedestrian detection: Visible spectrum survey, *Pattern Recognition and*  
410 *Image Analysis* 4477 (2007) 547 – 554.
- 411 [4] D. M. Gavrilu, The visual analysis of human movement: a survey, *Com-*  
412 *puter Vision and Image Understanding* 73 (1999) 82 – 98.
- 413 [5] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded  
414 scenes, in: *Proc. of the 2005 IEEE Computer Society Conference on*  
415 *Computer Vision and Pattern Recognition, CVPR 2005, Vol. 1, 2005,*  
416 pp. 878 – 885.
- 417 [6] B. Wu, R. Nevatia, Detection of multiple, partially occluded humans  
418 in a single image by bayesian combination of edgelet part detectors, in:

- 419 Proc. of the 10th IEEE International Conference on Computer Vision,  
420 ICCV 2005, Vol. 1, 2005, pp. 90 – 97.
- 421 [7] Z. Lin, L. S. Davis, D. Doermann, D. Dementhon, Hierarchical part-  
422 template matching for human detection and segmentation, in: Proc. of  
423 the 11th IEEE International Conference on Computer Vision, 2007, pp.  
424 1 – 8.
- 425 [8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detec-  
426 tion, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005.  
427 IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886 –  
428 893.
- 429 [9] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, T. Yu,  
430 Unified crowd segmentation, in: Proc. of the 10th European Conference  
431 on Computer Vision, ECCV 2008, 2008, pp. 691 – 704.
- 432 [10] L. Dong, Fast crowd segmentation using shape indexing, in: Proc. of the  
433 11th International Conference on Computer Vision, ICCV 2007, 2007,  
434 pp. 1 – 8.
- 435 [11] L. Wang, N. H. C. Yung, Crowd counting and segmentation in visual  
436 surveillance, in: Proc. of the 16th IEEE International Conference on  
437 Image Processing, ICIP 2009, 2009, pp. 2573 – 2576.
- 438 [12] B. A. Boghossian, S. A. Velastin, Motion-based machine vision tech-  
439 niques for the management of large crowds, in: Proc. of the 6th IEEE  
440 International Conference on Electronics, Circuits and Systems, 1999,  
441 Vol. 2, 1999, pp. 961–964.

- 442 [13] P. Reisman, O. Mano, S. Avidan, A. Shashua, Crowd detection in video  
443 sequences, in: Proc. of the IEEE Intelligent Vehicles Symposium, 2004,  
444 pp. 66 – 71.
- 445 [14] V. Rabaud, S. Belongie, Counting crowded moving objects, in: Proc.  
446 of the IEEE Computer Society Conference on Computer Vision and  
447 Pattern Recognition, Vol. 1, 2006, pp. 705 – 711.
- 448 [15] S. Ali, M. Shah, A lagrangian particle dynamics approach for crowd flow  
449 segmentation and stability analysis, in: IEEE International Conference  
450 on Computer Vision and Pattern Recognition, 2007.
- 451 [16] M. Manfredi, R. Vezzani, S. Calderara, R. Cucchiara, Detection of  
452 crowds gathered in open spaces by texture classification, in: Proc. of the  
453 1st International Workshop on Pattern Recognition and Crowd Analy-  
454 sis, PRCA 2012, 2012, pp. 32 – 35.
- 455 [17] M. Rodriguez, I. Laptev, J. Sivic, J.-Y. Audibert, Density-aware person  
456 detection and tracking in crowds, in: Proc. of the International Confer-  
457 ence on Computer Vision, ICCV 2011, 2011.
- 458 [18] N. I. Fisher, Statistical analysis of circular data, Cambridge University  
459 Press, 1995.
- 460 [19] Z. Farbman, R. Fattal, D. Lischinski, Diffusion maps for edge-aware im-  
461 age editing, in: ACM Transactions on Graphics (TOG), Vol. 29, ACM,  
462 2010, p. 145.
- 463 [20] R. R. Coifman, S. Lafon, Diffusion maps, Applied and Computational  
464 Harmonic Analysis 21 (1) (2006) 5 – 30.

- 465 [21] J. De la Porte, B. Herbst, W. Hereman, S. van der Walt, An introduction  
466 to diffusion maps, in: Proceedings International, 2008.
- 467 [22] B. Nadler, S. Lafon, R. Coifman, I. G. Kevrekidis, Diffusion maps - a  
468 probabilistic interpretation for spectral embedding and clustering algo-  
469 rithms, Principal manifolds for data visualization and dimension reduc-  
470 tion 10 (2008) 238 – 260.
- 471 [23] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using  
472 the nystrom method, Pattern Analysis and Machine Intelligence, IEEE  
473 Transactions on 26 (2) (2004) 214–225.
- 474 [24] S. Lafon, A. B. Lee, Diffusion maps and coarse-graining: A unified  
475 framework for dimensionality reduction, graph partitioning, and data  
476 set parameterization, Pattern Analysis and Machine Intelligence, IEEE  
477 Transactions on 28 (9) (2006) 1393–1403.
- 478 [25] Wikipedia, Quadtree, <http://en.wikipedia.org/wiki/Quadtree>.
- 479 [26] P. Allain, N. Courty, T. Corpetti, AGORASET: a dataset for crowd  
480 video analysis, in: 1st ICPR International Workshop on Pattern Recog-  
481 nition and Crowd Analysis, Tsukuba, Japan, 2012.