

# ABNORMAL MOTION SELECTION IN CROWDS USING BOTTOM-UP SALIENCY

*Matei Mancas, Nicolas Riche, Julien Leroy, Bernard Gosselin*

University of Mons, IT research center/TCTS Lab  
20, Place du Parc, 7000 Mons, Belgium  
matei.mancas@umons.ac.be

## ABSTRACT

This paper deals with the selection of relevant motion from multi-object movement. The proposed method is based on a multi-scale approach using features extracted from optical flow and global rarity quantification to compute bottom-up saliency maps. It shows good results from four objects to dense crowds with increasing performance. The results are convincing on synthetic videos, simple real video movements, a pedestrian database and they seem promising on very complex videos with dense crowds. This algorithm only uses motion features (direction and speed) but can be easily generalized to other dynamic or static features. Video surveillance, social signal processing and, in general, higher level scene understanding can benefit from this method.

*Index Terms*— crowd analysis, social signal processing, saliency, attention, real life, real-time

## 1. FROM SMALL GROUPS TO DENSE CROWDS

Unlike videos containing one or a few objects of interest, when dealing with large or massive groups of moving objects like dense crowds for example, the number of open problems is still important. Individual object tracking is, for example, virtually impossible and it is difficult to acquire databases of specific events.

A first category of papers is related to crowd properties analysis, and a second one to abnormal event detection. Within the first category, a lot of papers estimate crowd density using textures, edges, or global cues [1, 2] or using optical flow [3] to detect stationary crowds. Some people counting in crowds results were also achieved [4].

In the second category, the aim is to detect and if possible to classify abnormal events in crowds. Most of the time, normal behaviors are modelled and deviations from those models are considered abnormal. In [5] authors use HMM and principal component analysis. In [6] an interesting approach uses lagrangian particle dynamics for the detection of flow instabilities and the method seems to be efficient for dense crowds. [7] uses optical flows to detect when abnormal events occur without necessarily pointing the precise region of interest into the frames.

Our approach is a real-time contribution to abnormal event detection and uses the notion of computational attention which quantifies motion saliency. The presented method can be applied from small groups of objects (e.g. 4 objects) to dense touching moving objects like crowds. It is possible to precisely locate the area into the crowd where abnormal or surprising events occur. In section 2 computational attention is defined while section 3 present the proposed motion-based saliency method. Section 4 deals with the validation of the method on synthetic videos, the corresponding real videos and complex groups of moving objects and crowds. Section 5 discusses some issues and concludes.

## 2. COMPUTATIONAL ATTENTION

The aim of computational attention is to automatically predict human attention on multimodal data such as sounds, images, video sequences, etc... The term *attention* refers to the whole attentional process that allows one to focus on some stimuli at the expense of others. Human attention mainly consists of two processes: a bottom-up and a top-down one. Bottom-up attention uses low-level signal features to find the most salient or outstanding objects. Top-down attention uses a priori knowledge about the scene or task-oriented knowledge in order to modify (inhibit or enhance) the bottom-up saliency. While numerous models were provided for attention on still images, videos have been less investigated. Nevertheless, some authors generalized their models to videos ([8, 9] present a more detailed review of saliency algorithms in videos). Most of these methods are mainly applied on more classical mono- or multi-user scenarios and not on dense crowd scenarios. The presented approach is bottom-up and uses the instantaneous spatial context: it compares a given motion behavior to the rest of the motion within the same frame.

## 3. ATTENTION MODEL FOR MOTION SELECTION

This model is an extension of [10] as it can handle more general motion (not only small objects) and it uses motion direction. It is also a generalization to video of [9] which only works on still images. The algorithm presented here has three main steps which are further described.

### 3.1. Motion features extraction

As a first step, features are extracted from the video frames. In order to provide an easy to generalize framework, each frame is divided into square “cells”. In that way some features can be extracted for each cell and compared on the same location and neighborhood basis. In this paper we only extracted the motion vector (speed and motion direction). We choose the Farneback approach [11] as it is quite fast compared to other techniques, it can be easily computed on cells of different sizes which perfectly fits with our approach and the result has good accuracy. Fig. 1, left image displays an example of optical flow extraction showing different speed (arrow length) and directions (arrows color). Ideally, the cells on which the features are extracted should overlap (as in [9]). But this overlap compromises the real-time behavior: that is why we did not use it. The chosen cell size is very small: around 3/5 pixels wide. The features are then discretized into 4 directions (north, south, west, est) and 5 speeds (very slow, slow, mean, fast, very fast).

### 3.2. Spatio-temporal filtering of the features

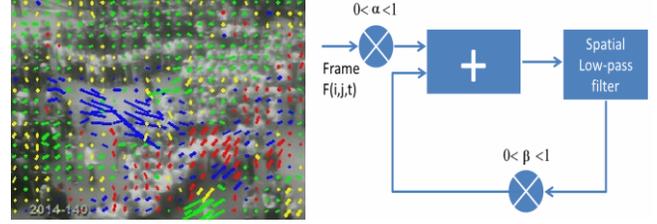
In [9], a multi-scale approach was implemented using low-pass filters with different neighborhoods which roughly summarize the statistics on those neighborhoods (mean). For the video generalization, to keep the algorithm real-time, only two different scales were implemented.

To implement a spatio-temporal low-pass filter which will be applied to each of the discretized feature channels (4 directions and 5 speeds), we separated the space and time dimensions. As it can be seen on Fig. 1, right image, the frames are first spatially low-pass filtered ( $LP_{i,j}$ ). Then, a weighted sum is made on the time dimension by using a loop and a multiplication factor  $\beta$  smaller than 1. This process will tend to provide lower weight to the frames which made the loop several times (the older ones) because of the  $\beta^n$  in Eq. 1 which will be smaller and smaller when the loop iteration  $n$  will be higher. Our approach only takes into account frames from the past (not from the future). This approximation of a 3D convolution (a difference is that this approach provides increasing spatial filtering through iterations) is easy to implement (Fig. 1, right image). The neighborhood of the filtering is obtained by changing the size of the spatial kernel and by modifying the  $\beta$  parameter for the temporal part. If  $\beta$  is closer to 0, the weight applied to the temporal mean will decrease very fast, so the temporal neighborhood will be reduced, while a  $\beta$  closer to 1 will let the temporal dimension be larger. The two filters that we implemented had parameters of 3x3 and 9x9 for the spatial filtering and 0.9 and 0.8 for the temporal filtering.

$$\text{Filtered } 3DVolume = \alpha \times \sum_n \beta^n \times LP_{i,j}(F(i, j, t-n)) \quad (1)$$

### 3.3. From feature detection to feature selection

As in [9], after the filtering of each of the 9 feature channels (4 directions, 5 speeds), the resulting images are separated



**Fig. 1:** Left image: optical flow overlapping a frame, right image: schematization of the 3D low-pass filtering

into 5 bins each, and the self-information ( $I(b_i)$ ) of the pixels for a given bin  $b_i$  is computed as described in Eq. 2. This self-information can be seen as a pixel saliency index.

$$I(b_i) = -\log\left(\frac{H(b_i)}{Card(B)}\right) \quad (2)$$

where  $H(b_i)$  is the value of the histogram  $H$  for the bin  $b_i$  (in other terms how many times the statistics of a video volume resulting from the 3D low-pass filtering can be found within the frame) and  $Card(B)$  the cardinality of the frame (size of the frame in pixels).  $H(b_i)/Card(B)$  is simply the occurrence probability of a pixel of bin  $b_i$ .

The matrices containing the self-information, thus the saliency of the pixels at the two scales (two different 3D filters) are then added. This operation is slightly different from [9] where the  $-\log$  is applied after summing the occurrence probabilities of the pixels at different scales while here we make the sum of the self-information themselves at the different scales, which is equivalent to multiplying the occurrence probabilities. The results are in this case slightly different but the idea remains the same, and this approach is much easier to implement.

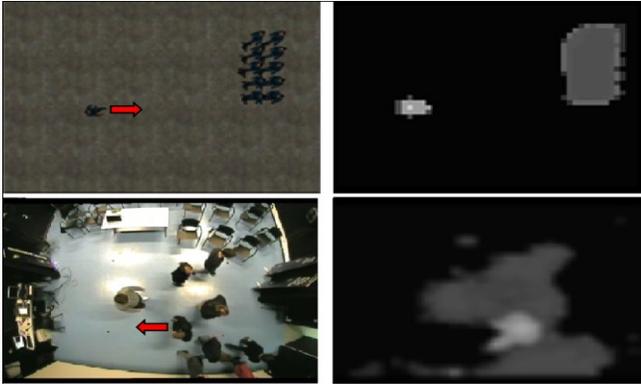
Once a saliency map is computed for each feature channel, a maximum operator is applied to put together the 4 directions into a single saliency map and the 5 speeds into a second saliency map. The two final maps tell us about the rarity of the statistics of a given video volume ( $x,y,t$ ) at two different scales for a given feature. Rare motion is salient.

## 4. MODEL VALIDATION

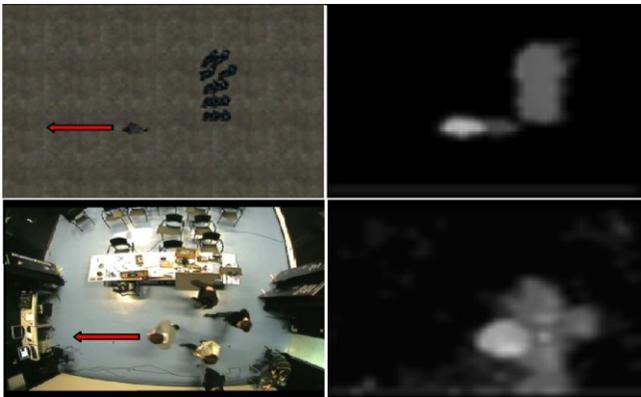
### 4.1. Human motion scene on synthetic and real data

We first performed a validation on synthetic videos made with the graphics engine “Source” and the “Valve Hammer Editor” included in the SDK who comes with the game Half-Life 2 [12]. These tools allow to build and to rehearse various human scenarios in which we have complete control over the video recording. By eliminating the distortion and artifacts, this validation can be used as proof of concept but also enables debugging and testing of the algorithm. The real videos were performed with the same configuration as the virtual scenarios: human displacements which will be tested in various configurations of speed and direction.

In the first scenario, people are at the same speed but one is in the opposite direction.



**Fig. 2:** First column: arrows showing the person which has a different direction compared to the group (virtual scenario on top, real scenario in the bottom). Second column: the corresponding saliency maps of the 4 directions.



**Fig. 3** First column: arrows showing the person which has a different speed compared to the group (virtual scenario on top, real scenario in the bottom). Second column: the corresponding saliency maps of the 5 speeds.

Fig. 2 shows that the model extract correctly the man who is walking differently than the main group (the corresponding are of the saliency map is clearer).

In a second scenario, people have the same direction but one has a higher speed than the main group. Fig. 3 shows that the model extracts correctly the man who is running (higher intensity on the saliency maps). In the last scenario, one person is running with an opposite direction and a higher speed then the crowd cumulating the two motion features.

To asses the performance of our model, we have computed a “detection success” which is the percentage of frames where the system properly detects the correct blob. An error (a blob appearing on other people than the one having the different behavior) is counted only if it persists over five successive frames. The results are excessively high for the virtual scenarios (with several 100% detection as it can be seen in Tab. 1 on the two last columns). To be more selective, for the virtual scenarios an additional second success rate was computed (bold in the two last columns). In that case, the success is counted only if there is no other blob in each frame. Despite this more severe condition, we can

notice that the success rates of the virtual scenarios remain above the real scenarios because of the absence of any artifacts (noise, lens deformation, etc...). Moreover, while the virtual scenarios work even with 3 people, the real scenario begins to work well from 5 people. Finally we remark that more there are people in the scenario, better the method works: the rarity begins to be important.

#	Virtual	Real	People	Success direction	Success speed
1	X		3	92/81.4	
1	X		6	100/98.2	
1	X		11	100/99.6	
1		X	5	95.6	
1		X	9	98.7	
2	X		3		100/92.9
2	X		6		100/97.5
2	X		11		100/98.6
2		X	5		94.1
3	X		3	75/67	100/95
3	X		6	100/94	100/98.4
3	X		11	100/97	100/98.4
3		X	5	97	95

**Tab. 1** First column: scenario number, Second/Third column: virtual or real video, Forth column: the total number of people from which one has a different behavior, Fifth column: the success rate (%) with the method used for both real and virtual videos in normal characters and in bold the more restrictive method, Last column: the success rate (%) with the method used for both real and virtual videos in normal characters and in bold the more restrictive method for the speed feature.

## 4.2. Real Walkways

In this section, we use the more realistic Vasconcelos’s dataset [13] to validate our model. The dataset used here is UCSDped2 and it consists of videos of a crowded pedestrian walkway. In the normal setting, the video contains only pedestrians. Abnormal events are due to the circulation of non-pedestrian entities on the walkways (skate, bikes, cars ...). We tested 9 videos (the ones also containing the ground truth) in the dataset. Initially the ground truth was built to detect the non-pedestrian objects. In our case, also some pedestrians who have rare directions for example are interesting to highlight. That is why their detection was not considered as a false alarm. Fig. 4 shows our result (middle) compared to the ground truth. In that case it is successful as the third small blob in the middle image does not have a 5 frames persistence (noise). As we can see on Tab. 2, the results are very good even in real-life scenarios. Only the videos 3, 7 and 11 are less than 90%. But this is due to the fact that, at some moments in the video, the bike or skate had a speed very close to the one of the pedestrian. As our model only uses motion cues, its less good performance in detecting them is logic.



Fig. 4: Left image: video, Middle image: our model, Right image: the dataset ground truth.

Video #	Success rate	Video #	Success rate
1	93.4 %	7	81.3 %
2	96.6 %	8	94 %
3	79.3 %	9	96.8 %
4	91.3 %	11	75.8 %
5	91.4 %		

Tab. 2: First/third columns: videos number in the dataset. Second/forth columns: rate (for speed or direction).

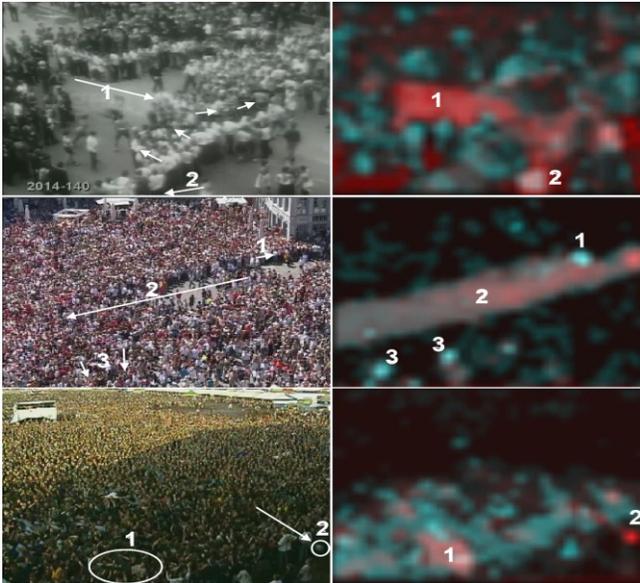


Fig. 5: First column: annotated frames, Second column: color saliency maps. A red dominant means that the speed feature is the most interesting. A cyan dominant means that direction is the important feature. A “white” blob means that both speed and directions may attract attention.

#### 4.2. Crowds abnormalities detection

In this section, we use very complex crowd videos. Surprisingly good results can be found as shown in Fig. 5. On the first row we can see that people running towards the others are detected (1) and the person who is faster and with a different direction (2) is also highlighted. On the second row, the two people walking against the main central flow (1) are well visible. It is also the case with some people having perpendicular directions (3). Finally in the third row one person carried by the crowd (1) is detected as also an object which goes over the crowd thrown by someone with a high speed compared to the other moving objects (2).

## 5. DISCUSSION AND CONCLUSION

We presented a near real-time (~12 fps, small resolution, no optimization) method which provides a bottom-up saliency map. More moving objects there are, better the method works: it is really well adapted for collective behavior analysis. Camera calibration might be very useful to avoid shape apparent deformation in case of a bad camera positioning. Also the addition of static cues would help to detect the bikes even if their speed is the same as the pedestrians. The proposed method also handles (small) camera motion (a lot of motion in the same direction and speed by definition is not rare, so not interesting). It could also be good to stabilize the gaze between saccades (the saliency maps remain quite noisy) and to find an optimum way of fusing different features as speed and direction. One of the main applications is video surveillance, but the analysis could help to discover relations within groups in the crowd, or a set of social features like the interpersonal distance could be used to detect social changes.

## 7. REFERENCES

- [1] A. Marana, S. Velastin, L. Costa, and R. Lotufo. “Estimation of crowd density using image processing,” *Image Processing for Security Applications, IEE Colloquium*, pages 11/1–11/8, 1997.
- [2] R. Ma, L. Li, W. Huang, and Q. Tian. “On pixel count based crowd density estimation for visual surveillance,” *Cybernetics and Intelligent Systems, 2004 IEEE Conference*, vol. 1:170–173, 2004.
- [3] B. Boghossian and S. Velastin. “Motion-based machine vision techniques for the management of large crowds,” *Electronics, Circuits and Systems*, vol. 2:961–964, 1999.
- [4] S.-F. Lin, J.-Y. Chen, and H.-X. Chao. “Estimation of number of people in crowded scenes using perspective transformation,” *Systems, Man and Cybernetics, IEEE Transactions*, 31(6):645–654, 2001.
- [5] E. Andrade, S. Blunsden, and R. Fisher. “Hidden markov models for optical flow analysis in crowds,” *ICPR 2006.*, vol. 1:460–463, 2006.
- [6] S. Ali and M. Shah. “A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis,” *CVPR ’07.*, pages 1–6, 2007.
- [7] N. Ihaddadene, and C. Djeraba. “Real-time crowd motion analysis,” *ICPR 2008*, pages 1–4, 2008.
- [8] M. Mancas, D. Glowinski, G. Volpe, A. Camurri, P. Breteche, J. Demeyer, T. Ravet, P. Coletta. “Real-Time Motion Attention and Expressive Gesture Interfaces,” *Journal On Multimodal User Interfaces (JMUI)*, Springer Berlin/Heidelberg, 2009.
- [9] M. Mancas. “Relative influence of bottom-up and top-down attention,” *Attention in Cognitive Systems, LNCS, Volume 5395/2009*:pp. 212–226, 2009.
- [10] M. Mancas, 2010, “Attention-based Dense Crowds Analysis”, *Proc. of WIAMIS 2010, Desenzano del Garda, Italy*
- [11] G. Farnebäck, “Two-Frame Motion Estimation Based on Polynomial Expansion”, *Lecture Notes in Computer Science*, 2003
- [12] <http://source.valvesoftware.com/>
- [13] <http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm>