3D Saliency for Abnormal Motion Selection: the Role of the Depth Map

Nicolas Riche[1], Matei Mancas[1], Bernard Gosselin[1], Thierry Dutoit[1]

[1] University of Mons (UMONS), Faculty of Engineering (FPMs)
20, Place du Parc, 7000 Mons, Belgium
{Matei.Mancas, Nicolas.Riche, Bernard.Gosselin, Thierry.Dutoit}@umons.ac.be

Abstract. This paper deals with the selection of relevant motion within a scene. The proposed method is based on 3D features extraction and their rarity quantification to compute bottom-up saliency maps. We show that the use of 3D motion features namely the motion direction and velocity is able to achieve much better results than the same algorithm using only 2D information. This is especially true in close scenes with small groups of people or moving objects and frontal view. The proposed algorithm uses motion features but it can be easily generalized to other dynamic or static features. It is implemented on a platform for real-time signal analysis called Max/Msp/Jitter. Social signal processing, video games, gesture processing and, in general, higher level scene understanding can benefit from this method.

Keywords: Saliency, Attention, 3D features, Kinect, Depth map, Gestures

1 Computational Attention

Computational attention intends to provide algorithms which predict human attention. Attention refers to the process that allows one to focus on some stimuli at the expense of others and it is divided into two complementary influences. Bottom-up attention uses signal characteristics to find the salient objects. Top-down attention uses a priori knowledge to modify the bottom-up saliency. The relative importance of bottom-up and top-down attention depends on the situations [1].

In this paper we focus on bottom-up attention which uses the instantaneous spatial context: it compares a given motion behavior to the rest of the motion within the same frame. Some of the authors providing static attention approaches generalized their models to the time dimension: Dhavale and Itti [2], Tsotsos et al. [3], Parkhurst and Niebur [4], Itti and Baldi [5], Le Meur [6] or Bruce [7]. Motion has a predominant place and the multi-scale temporal contrast of its features is mainly used to highlight important movements. Boiman and Irani [8] provided a model which is able to compare the current movements with others from the video history or a database. Nevertheless, at our best knowledge, none of the motion-based attention models takes into account video depth from a 3D camera while very few use depth for static images [9]. In the next section, the importance of the depth motion extraction is shown in section 2. In section 3, we describe a near real-time motion-based attention model which highlights rare, surprising thus, abnormal motion. In section 4, we show the improvement brought to our model by the use of the depth information, especially in close scenes with frontal view. Finally we discuss and conclude in section 5.

2 Why 3D Features for Attention?

The 2D motion features extraction from videos can identify the relevant motion within the *(X, Y)* plane. However, they show their limits when movement occurs on the *Z* (depth) axis. We can see an example in Figure 1 where the relevant motion is poorly captured with 2D motion features as the main movement is along the *Z* axis.

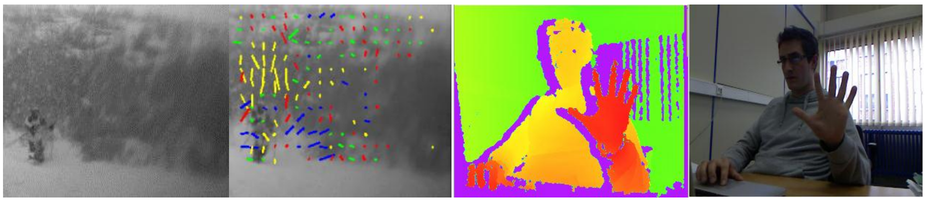

Fig. 1: From left to right: a frame with a skier coming towards the camera (depth – Z axis velocity); 2D motion features (optical flow for X and Y velocity); Depth map (red: close, green: far); RGB corresponding frame

The left image shows the initial video frame while the second image from the left shows the extracted optical flow. The *(X,Y)* motion is properly captured: the snow falling vertically (*Y* axis) above the skier is detected (yellow vertical lines) and the snow moved by the skier on his right on the *X* axis (blue horizontal lines). But the motion of the skier himself is not well described: the image shows several lines of different colors (*X,Y* directions) on the skier while in reality he is coming towards the camera (*Z* axis). This example shows that detection of the motion on the *Z* axis would assign the skier with his real displacement. Obviously, a better feature extraction will also enhance the attention model performance.

The availability of low-cost 3D sensors with active infra-red illumination (as the Microsoft "Kinect" [10]) is an opportunity to easily extract scene depth (Z) information along with classical videos providing *(X,Y)* information. As shown in Figure 1 (third and fourth image from the left), these cameras provide us with RGB video (forth image) and the corresponding depth map (third image). The color map of the third image shows pixels close to the camera in red and pixels far from the camera in green. The pixels in violet are pixels where the information is not available (too close to the camera, too far from it, infra-red shadows, etc.).

The third image from Figure 1 shows that the depth map is homogenous and its quality is well behind the one of classical stereo cameras. This fact is very interesting for the extraction of the movement along the *Z* axis. The implementation for both 3D feature extraction and bottom-up attention computation was carried out on Max MSP [11] using the Jitter and FTM [12] libraries. Max is a platform for real-time signal processing which allows both fast prototyping by using visual programming with libraries supported by a large community and flexibility by the possibility to build additional blocks if needed. Jitter is a library added to Max which provides the possibility to work with matrices, and thus with images and video. FTM is a shared library for Max providing a small and simple real-time object system and a set of optimized services to be used within Max externals. Its capability to handle matrices makes it complementary to Jitter.

3 Attention Model for Motion Selection

The proposed algorithm has three main steps (Figure 2). First, motion features are extracted from the video. Static features could also be extracted, but here only motion-related features were used. A second step is a spatio-temporal filtering of the features at several scales to provide multi-scale statistics. Finally, a third step uses those statistics to quantify at several scales the features' rarity within the video frame.

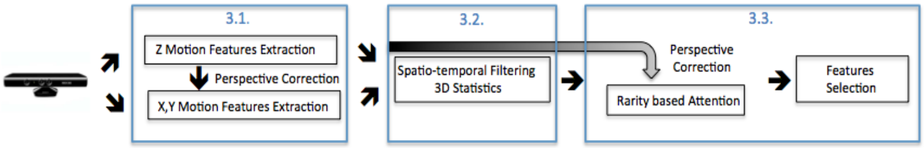

Fig. 2 Block diagram of algorithm used to detect salient motion events using the depth map

3.1 Motion features extraction

3.1.1. Part 1: X and Y extraction

We first work in the plane *(X,Y)*. On the video from the RGB camera, we apply an optical flow algorithm. Optical flow is a measure of the velocity of each pixel between two consecutive frames. (Figure 1, second image). We choose the Farneback approach [13] as it is quite fast and pick Δx and Δy.

3.1.2. Part 2: Z extraction

We make the difference between two consecutive frames of the depth map to get Δz. Some noise is present on the depth map (violet pixels, Figure 1, third image). The noise of the depth map is eliminated by saturating the shadows and a separation between motion and noise can be achieved by thresholding (Figure 3, first row).

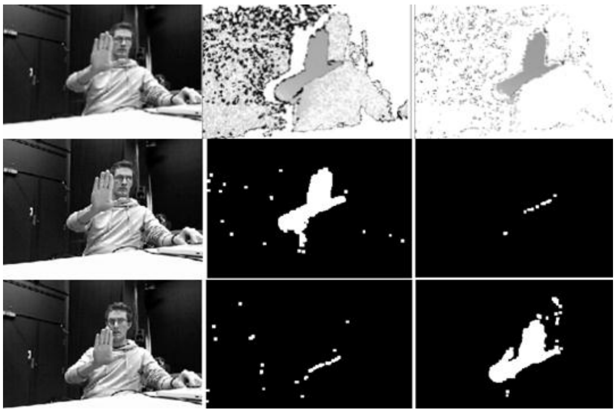

Fig. 3 First row, from left to right: video frame (the hand moves on the *Z* axis); frame differencing of the noisy depth map; frame differencing on the denoised depth map. Second and third row, First column: video frames with the hand going towards the camera (up) and in the opposite direction (down), middle column: feature map of the direction towards the camera, left column: feature map of the direction opposite to the camera

After noise elimination, the Z axis speed is given by the absolute value of Δz (Figure 3, first row, left image) while the direction is given by the sign of Δz (Figure 3, rows 2 and 3 middle and right).

3.1.3. Part 3: Depth-based perspective correction

Figure 4 (first and second images) shows the perspective problem. The perspective view of a camera will provide wrong apparent sizes of moving objects (people far from the camera seem smaller than people close to the camera) and also wrong apparent speeds of the moving objects (an object moving close to the camera will seem to have a much higher speed than an object moving far from the camera). This perspective view will have negative effects on the speed computation and on the attention model (on the X and Y axes), especially within close camera scenes.

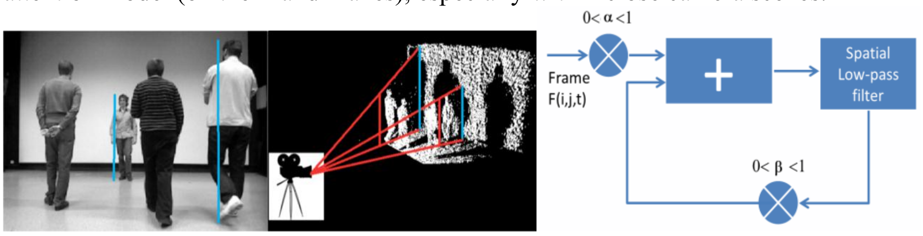

Fig. 4 From left to right: View from the camera: the apparent size of people is different function of their distance with the camera (vertical blue lines); Reconstructed image from the Kinect: people have similar sizes (red vertical lines) but the shadows (apparent sizes) are different (blue vertical lines); Schematization of the 3D low-pass filtering

To remove this effect, we need to compute the distance ("*dist*") of each pixel relative to the camera. This distance will let us know the real objects speed and sizes (Thales theorem). Thanks to the depth map of the Kinect, this depth distance can be directly used to compute the speed and to correct the objects sizes. This is also crucial in attention computation as pixels' rarity depends on objects size. The corrected 3D speed is obtained with the Eq. (1) where Δx and Δy are computed using the optical flow on the RGB video, Δz the frame differencing on the depth map and *dist* comes from the depth map. The features are then discretized into 6 directions (north, south, west, east, front, back) and 5 speeds (very slow, slow, mean, fast, very fast).

$$Speed_{3D} = \sqrt{dist \times \Delta x^2 + dist \times \Delta y^2 + \Delta z^2} \qquad \textbf{(1)}$$

3.2 Spatio-temporal filtering of the features

We use low-pass spatio-temporal filters to roughly summarize the statistics of the feature maps (6 directions and 5 speeds). Several scales (both in space and time) of those filters should be applied to the feature maps, but, to keep the algorithm real-time, only two scales were taken into account. To implement a spatio-temporal low-pass filter which will be applied to each of the discretized feature channels (6 directions and 5 speeds), we separated the space and time dimensions. As it can be seen on Figure 4, third image, the frames (F) are first spatially low-pass filtered ($LP_{i,j}$ in Eq. 2). Then, a weighted sum is made on the time dimension by using a feedback and a multiplication factor β<1. This process will tend to provide lower weight to the frames which made the feedback several times (the older ones) because of the β^n in Eq. 2 which will be smaller and smaller when the feedback iteration n will be higher. Our approach only takes into account frames from the past (not from the future).

The neighborhood of the filtering is obtained by changing m (diameter-1 of the spatial kernel $A(h,k)$ (Eq. 3)) and by modifying the β parameter for the temporal part (Eq. 2). If β is closer to 0, the weight applied to the temporal mean will decrease very fast, so the temporal neighborhood will be reduced, while a β closer to 1 will let the temporal dimension be larger. The two filters that we implemented had parameters of m =2 and 8 for the spatial filtering and $\beta = 0.4$ and 0.3 for the temporal filtering.

$$\hat{F}(i,j,t) = \alpha \times \sum_n \beta^n \times LP_{i,j}(F(i,j,t-n))^n \tag{2}$$

where $LP_{i,j}$ is a classical Gaussian spatial low-pass filtering:

$$LP_{i,j}(F(i,j,t-n)) = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h,k) \times F(i-h, j-k, t-n) \tag{3}$$

3.3 From feature detection to feature selection

After the filtering of each of the 11 feature maps (6 directions, 5 speeds), the resulting images are separated into 3 bins each. The occurrence probability $P_s(b_i)$ of each bin and for a given scale s is computed as described in Eq. 4:

$$P_s(b_i) = \frac{H(dist \times b_i)}{\sum dist \times \|B\|} \tag{4}$$

where $H(dist \times b_i)$ is the value of the histogram H for the bin b_i (how many times the statistics of a video volume resulting from the 3D low-pass filtering can be found within the frame). The pixels belonging to the bin b_i are previously multiplied by the distance that separates them from the camera: this operation provides a higher weight to pixels which are far from the camera and which belong to objects which have an apparent size smaller than their real size. In that way, the effect of the perspective is cancelled. $||B||$ is the cardinality of the frame (size of the frame in pixels) and $\sum dist$ the sum of distances of the all the pixels. $P_s(b_i)$ is the occurrence probability of the pixels of the bin b_i where the perspective has been cancelled.

Finally, the self-information $I(b_i)$ for the pixels of each bin is computed after taking into account $P_s(bi)$ at the different scales s at which it was computed. This self-information represents the bottom-up attention or saliency for all the pixels of bin b_i (Eq. 5). In order to keep real-time processing, only two scales were used here, so s=2.

$$I(b_i) = -\log\left(\frac{\sum_s P_s(b_i)}{s}\right) \tag{5}$$

Once a saliency map is computed for each of the 6 direction feature maps, they are merged into a *(X,Y,Z)* direction saliency map using the maximum operator but with a coefficient of 2 for the *Z* axis and 1 for the *X* and *Y* axis (Eq. 6). For the speed saliency maps, the speed on the Z axis is incorporated into the 5 saliency maps already existent in 2D (very slow, slow, mean, fast, very fast). Those maps are merged using the same approach as for the direction maps (Eq. 6). The coefficient of 2 is empirical and it is due to the fact that the motion on *(X,Y)* on one hand and the motion on *Z* on the other hand are not extracted using the same approach (optical flow for *(X,Y)* and frame differencing for *Z*).

$$S = Max(2 \times S_Z + S_X + S_Y) \tag{6}$$

The final *(X,Y,Z)* map tells us about the rarity of the statistics of a given video hyper-volume *(X,Y,Z,t)* at two different scales for a given feature. *Rare motion is salient.*

4 Model Validation

4.1 When using 3D features?

We represented the speed and direction saliency maps by using a RGB final saliency map. A red dominant means that the speed feature is the most interesting. A cyan dominant means that direction is the most important. A white blob (which is a mix of red and cyan) means that both speed and directions may attract attention. Here we used only 2D motion features on complex real scenes (Figure 5). In the first two images from the first row there is a close scene with a frontal view. The other scenes contain wider and wider views with mostly top views. Surprisingly good results can be found on those wide scenes as shown in Figure 5.

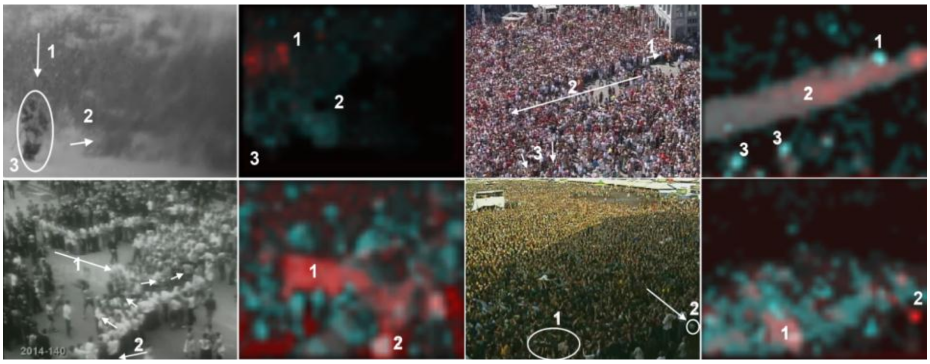

Fig. 5: First and third column: annotated frames, Second and fourth column: color saliency maps. A red dominant means that the speed feature is the most interesting. A cyan dominant means that direction is the important feature.

On the second row, first and second images, 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 first row, third and fourth images, 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 second row, third and fourth images one person carried by the crowd (1) and a thrown object (2) are also well detected with a higher speed compared with the other moving objects.

Nevertheless, the results are very poor for the first row, first and second image in Figure 5. While the rapidly falling snow (*Y* axis motion) is well detected (1) and the snow pushed by the skier (*X* axis motion) on his right (2) is also detected, the skier himself (3) is not detected at all! The skier is the only moving object on the *Z* axis, thus it is very salient, but as only 2D features are extracted, he is not well detected.

This scene comparison in 2D shows that the more the scene is wide and the camera has a top view, the less important the *Z* axis motion is. Indeed, a top-view will map most of the motion on the *(X,Y)* plane and very small people doing gestures on *Z* (like jumping for example) are almost not detectable in those configurations.

An interesting conclusion is that, while in videosurveillance-like situations (wide field of view, almost top-view) one does not need a precise knowledge about the Z axis, for ambient intelligence and robot-like situations (smaller field of view, frontal view), the knowledge of the Z axis is crucial. This is convenient, as the Kinect sensor horopter is between 25 cm and 6 meters.

4.2 Scenarios used for validation

Four people take part to three scenarios. Each one of the scenarios is designed to validate the model along a specific axis (*X*, *Y* and *Z)*. For a first run, for each axis, the purpose is to validate the direction (one of the four people will always be in the opposite direction of the three others) and, during a second run, the goal is to validate the speed (a person will walk faster than the three others). To quantify the model results we define a success rate which is the ratio between the number of frames where the maximum of saliency is located onto the person with a different behavior (in terms of speed or direction) than the others and the total number of valid frames. The valid frames are the frames where the four people are in motion (as only motion features are taken into account). Each of the 6 video runs last around 1 or 2 minutes.

4.3 Validation of the perspective correction

To show the contribution of the perspective correction effect, we processed the scenario along the *X* axis (Table 1, left-side) and the one along the *Z* axis (Table 1, right-side) with or without perspective correction.

Table 1: Influence of perspective correction on the *X* axis scenario (left-side) in terms of success rate using the *(X, Y)* saliency maps and on the *Z* axis scenario using the *Z* saliency map.

	X-axis Scenario		Z-axis Scenario	
	Sal. Map XY no correct.	Sal. Map XY with correct.	Sal. Map Z no correct.	Sal. Map Z with correct.
Direction	64.6 %	80 %	80.5 %	93 %
Speed	67.1 %	81.3 %	69 %	77 %

In the *Y* axis scenario, there is no perspective effect as all the participants are at the same distance from the camera. Table 1 shows significant improvement for success rate if the perspective correction is applied. Thereafter, we will always use the perspective correction in the following experiments.

4.4 Scenarios used for validation

As stated in section 4.2, in a first run of the three scenarios, the goal was to validate the attention-based motion direction selection. In each of the three scenarios, people move at very similar speed but one of the four moves in the opposite direction with respect to the three others. Some results are shown in Figure 6. The white blobs are pointing towards the image areas with a salience higher than 96% of the maximum of the saliency map. Figure 6 shows that the model correctly extracts the man who is walking differently with respect to the main group. Table 2 (left-side) provides the quantitative details of the test on the different sequences:

Table 2: Success rate percentage for salient direction (left-side) and speed (right-side) detection on the three axis using the *(X,Y), (X,Y,Z) and Z* saliency maps.

	Direction			Speed		
	Sal. Map XY	Sal. Map XYZ	Sal. Map Z	Sal. Map XY	Sal. Map XYZ	Sal. Map Z
X-axis	80 %	80 %	47 %	81.3 %	77 %	42 %
Y-axis	94 %	90.5 %	51 %	86.2 %	84.4 %	33.3 %
Z-axis	54.3 %	83.3 %	93 %	44 %	71 %	77 %

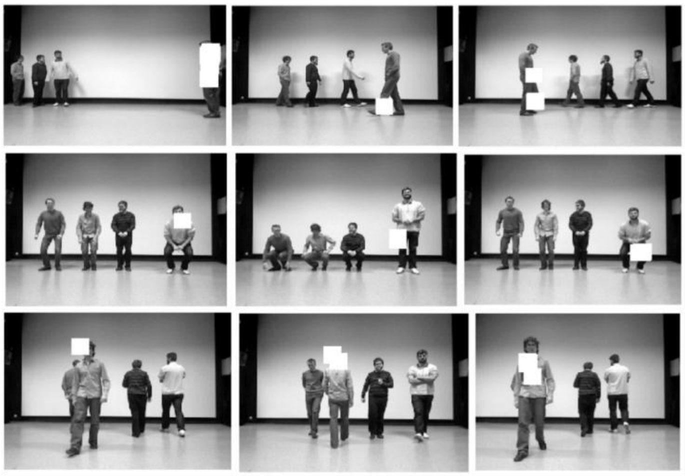

Fig 6: Direction scenarios along 3 axes. The white blobs locate the person which has different direction on the X axis (first row), on the Y axis (second row) and on the Z axis (third row).

Table 2 (left-side) provides the success rates for the three axes in selecting the salient person (the one having different behavior compared to the others). The figures are given for the 2D saliency map *(X,Y)*, the 3D saliency map *(X,Y,Z)* and the saliency map of the *Z* axis alone. A first remark is that the *(X,Y)* saliency map performs very poorly on the *Z*-axis (54.3%). A second remark is that the *Z* saliency map performs very well on the *Z* axis (93%) while it performs very poorly on the *X* (47%) and *Y* (51%) axes. Finally, a third remark is about the fusion system (Eq. 6). While the fusion of the *Z* axis saliency with the *(X,Y)* axes saliency seems to work well on the *X*-axis scenario (both have 80%), for the *Y*-axis scenario the *(X,Y)* saliency map has a 94% success rate while the *(X,Y,Z)* saliency map has only a 90.5% success rate. This shows that the use of the information from the *Z* axis on a scenario concerning mainly the *Y* axis slightly decreased the system performance. Concerning the *Z*-axis scenario, the conclusion is the same as for the *Y* axis: the *Z* saliency map provides very good results (93%) while the addition of *X* and *Y* information in a *(X,Y,Z)* saliency map decreases the result to 83.3%. This third remark shows an issue in the empirical fusion strategy proposed in Eq. 6: the *(X,Y)* saliency map works better on the X and Y scenarios than the *(X,Y,Z)* saliency map, while the *Z* saliency map alone works better on the *Z* scenario than the *(X,Y,Z)* saliency map.

4.5 Motion speed validation

To validate the speed, we use the same principle as for the direction. In each scenario, one person has a higher speed than the main group. Figure 7 shows that the model extracts correctly the man who is faster than the others on all the axes. Table 2 (right-side) provides the success rates for the three axes in selecting the salient person (the one having different speed compared to the three others). The figures of this table, even if the overall performances of the speed are slightly lower the ones of the direction, lead to the same remarks than for the previous section: the key role of the *Z* saliency map for the *Z* axis scenario is confirmed and also the fusion issue which slightly decrease the overall system performances.

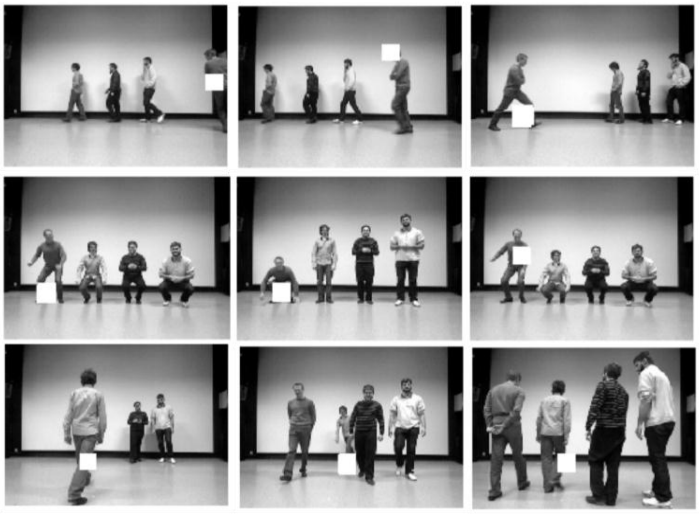

Fig 7: Speed scenarios along 3 axes. The white blobs locate the person which has different speed on the X axis (first row), on the Y axis (second row) and on the Z axis (third row).

5 Discussion and Conclusion

We presented a novel near real-time (20 fps for small-sized videos not optimized) bottom-up saliency model. This model uses motion-based 2D or 3D features, but it can be easily extended to other motion features or static features. The use of the depth information has proven, especially in close scenes with frontal view, its crucial importance. The quantitative results on a real scenario show substantial success rate increase in selecting the abnormal motion when the depth information is used along to the classical 2D features. Moreover, the proposed algorithm can handle small motion of the camera without important performance decrease. The fusion issue which leads to a slight performance decrease compared to the best results of the separate *(X,Y)* and *Z* maps can be solved by using a common method for feature extraction for all the exes as a 3D optical flow. The use of depth features opens perspectives for small groups and gesture analysis in frontal views.

Acknowledgements

This work is funded by Numediart (www.numediart.org), Walloon region, Belgium

References

1. Mancas, M.: Relative influence of bottom-up and top-down attention. Attention in Cognitive Systems, LNCS, Volume 5395/2009:pp. 212–226, Springer (2009)
2. Dhavale, N., Itti, L.: Saliency-based multifoveated MPEG compression. In Signal Processing and Its Applications. Proceedings. pp. 229-232, (2003)
3. Tsotsos, J. K., Liu, Y., Martinez-Trujillo, J.C., Pomplun, M., Simine, E., Zhou, K.: Attenting to visual motion. J. of Computer Vision and Image Understandig, (2005)
4. Parkhurst, D.J., Niebur, E.: Texture contrast attracts overt visual attention in natural scenes. European Journal of Neuroscience, 19(3):783–789, (2004)
5. Itti, L., Baldi, P.: Bayesian Surprise Attracts Human Attention. Advances in Neural information Processing Systems, 18:547, (2006)
6. Le Meur, O., Le Callet, P., Barba, D., Thoreau, D.: A Coherent Computational Approach to Model Bottom-Up Visual Attention. PAMI, pages 802–817, (2006)
7. Bruce, N.D.B., Tsotsos, J.K.: Saliency, attention, and visual search: an information theoretic approach. Journal of Vision, 9(3):5, (2009)
8. Boiman, O., Irani, M.: Detecting Irregularities in Images and in Video. International Journal of Computer Vision, 74(1):17–31, (2007)
9. Ouerhani, N., Huegli, H.: Computing visual attention from scene depth. In Proc. of Int’l Conf. on Pattern Recognition. Vol. 1. (2000)
10. Microsoft Kinect sensor, http://www.xbox.com/kinect
11. Max MSP, http://cycling74.com
12. FTM library, http://ftm.ircam.fr/index.php/Main_Page
13. Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. Scandinavian Conference on Image Analysis, LNCS 2749. pp. 363-370, (2003)