Finding Moving Objects in Video Recordings

Theo Hupkens

Introduction

Today, many military platforms are equipped with electro-optical video systems. Examples are: the Mirador and Sirius onboard of frigates, thermal imager and CCD day vision camera on the Fennek Reconnaissance Vehicles, Day TV and FLIR for the Apache attack helicopters, et cetera. The four new Dutch Ocean Patrol Vessels will be equipped with the multi-spectral, high resolution Gatekeeper system.

These platforms have in common that the video camera is constantly moving. Although human observers are very well able to observe threats or unusual situations on the monitors displaying the video recordings, they become bored and fatigued and less observant when nothing happens for some time. Therefore automatic image processing and pattern recognition systems must be developed, which take over the task of monitoring the video output of the surveillance systems. The development of such systems is not easy because of the difficult situations that are common in military operations: adversaries that try to be as invisible as possible by wearing combat clothing; night-time registrations using infrared cameras or image intensifiers suffering from severe noise; abrupt changes of the camera orientation; a constantly changing sea-background with barely visible swimmers or small boats, and so on. The motion estimation method that is the subject of this study can deal with most of these situations.

Video surveillance systems produce a continuous stream of images. A single image from a video sequence is often called a frame. However, if we want to emphasize the image properties we shall still use the word image. A first step in automatic pattern recognition is segmentation of an image into separate objects. To do so, it is necessary to find out which pixels of an image belong to a certain object. Then this object can be separated from the background and from any other objects. This process is called segmentation. There are several cues that can be used for this purpose, for instance colour differences or texture differences. Segmented regions can be used for further pattern recognition analysis. This paper describes segmentation based on motion. A group of (connected) pixels that move together is assumed to belong to one object. One advantage of using motion is that an object that consists of different parts is detected as one object, whereas if for instance colour differences are used the same object may be detected as several smaller segments which may have to be put together by sophisticated algorithms. Another advantage of the method is that after the motion is estimated, it becomes possible to correct for the changes due to the motion and then average the corrected frames in order to improve the signal to noise ratio.

In this paper, a brief description is given of the original motion estimation method described by [Odobez and Bouthemy, 1995]. This method is very well suited for the estimation of the camera motion. Experimental results obtained with this method from real infrared and colour videos are presented. The results are very accurate, even when noisy videos are used. The same method can be used to estimate the motion of separate

objects as well, by using only regions that move differently than according to the camera motion. This extended method is described in detail and the results when applied to synthetic and real sequences are discussed. After having found the motion parameters of an object, the exact location and the shape of that object as it appears in each video frame is known. Therefore, the extended method can be used for segmentation based on motion. We shall give several examples of this.

Motion parameters estimation for infrared and visual light video sequences

First, we need to define what we mean by "motion". In the simplest motion model, motion is described by two parameters: the velocity in the horizontal and that in the vertical direction. This two-parameter motion model describes a translation in the plane of the image. Several different motions are possible. For instance, an object might move away from the camera. This is almost the same as an object that is becoming smaller in time. A similar effect is obtained when the video camera zooms out. Three parameters are needed for an object that moves away from the camera or for a zooming camera. When an object rotates within the plane of the camera, for instance due the "rolling" of a ship or from an aeroplane that is filming an area while it is making a turn, the object's motion is a combined rotation and translation. Another possible motion would be that the object rotates away from the camera. When using cameras with a frame rate of 25 or more frames per second, this kind of rotation will hardly change the appearance of the object (it will be seen from approximately the some viewpoint), but will look like a slight change of the length to height ratio. All of these motions, and a few more, can be described by just six parameters.

Outline of the method

For the estimation of the motion of any object, an often-used approach is to try to find similar areas in succeeding frames. However, in practice finding similar areas when objects are rotating or changing their appearance is not easy. Therefore, we use a different approach, which is more suitable for the motion model we use. We use local changes of the intensity (that is the *gradient* of the intensity plot) together with intensity changes between successive frames. There exists a known relation between the gradient of the intensity in the neighbourhood of a certain pixel and the change in intensity of that pixel between two frames; this relation also depends on the size of the shift between the two frames. In the simplest model, the shift is proportional to the velocity. Because both the intensity changes and the gradients are known, we are able to calculate the corresponding shift. This is illustrated in Fig. 1. The lower pictures show the intensity of a horizontal line, at two points in times (i.e. for two separate frames of the video recording). Because the motion can just as well take place in the vertical direction, we need the gradient both in the horizontal and vertical direction.

The method starts with a least squares estimate of the parameters of the initial motion of the whole image, based on the relation between the gradients and intensity changes between two succeeding frames. We use high-resolution images only, which tend to be large in terms of numbers of pixels. For these large images, a somewhat different approach is used: in order to avoid excessive computation times, images at several reduced resolutions are used. Decreasing the resolution corresponds to resizing the

image to a smaller size. From every image several resized images are calculated; the first image is resized to 50% (in both directions); the second one to 25% and so on (see Fig. 2). The method starts with a guess of the initial motion of the whole image, at the smallest size that is used.

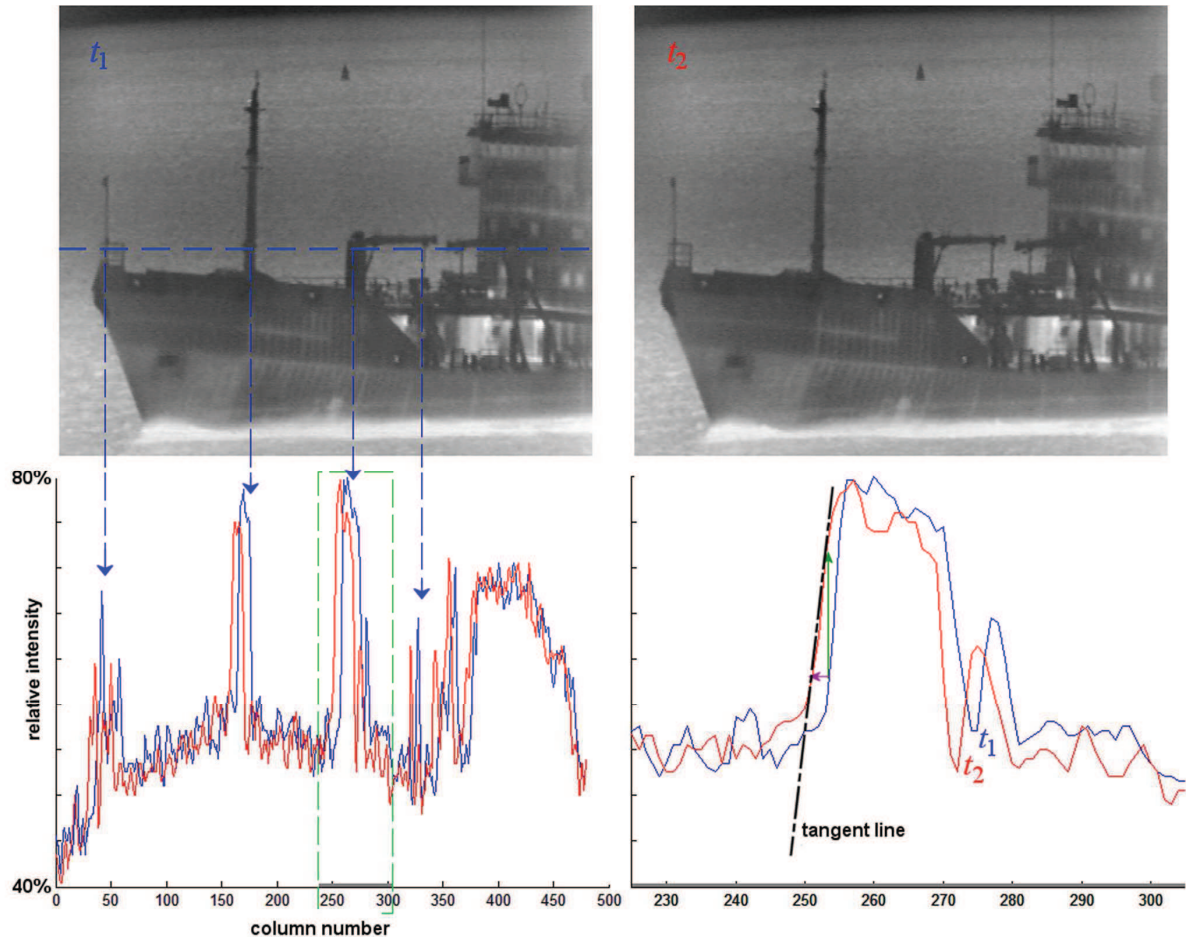


Figure 1. Upper images: two frames at times t_1 and t_2 taken from an infrared video recording. Under left: the intensity at a certain horizontal line for both frames. The vertical elements clearly show up in the intensity plots (blue arrows). Right: a magnified part of the intensity plot, showing the relation between the intensity difference between the two images at some pixel (green vertical arrow), the slope of the curve and the displacement (purple arrow). A larger velocity would result in proportionally larger intensity changes.

The initial estimate will be refined in succeeding steps, using the difference between the motion of every pixel and the estimated overall motion. Pixels that do not fit the current overall motion are called outliers. In the following iterative steps, the outliers will still be used for the determination of the global motion, but will have a smaller weight in the calculations. Due to these weights, after each iteration step the motion model will fit the overall motion better while the outliers will deviate more and more from the motion model. So the weights will increase for non-outliers and decrease for outliers. Fig. 3 illustrates this principle by means of an analogue. After several iterations, the changes in the motion parameters will become less than a preset threshold. At that moment, the weight function will be changed in such a way that pixels that move only *slightly* differently than most other pixels will be considered outliers as well. When after several iterations the motion parameters hardly change anymore, the image with reduced size

will be upsized by a factor of two and the process starts again, using the motion parameters from the last iteration (some parameters have to be adjusted to the new resolution). The process is repeated until a stable solution for the image at its original size is reached. For more details of this method, see [Odobez and Bouthemy, 1995 and Hupkens et al., 2000].

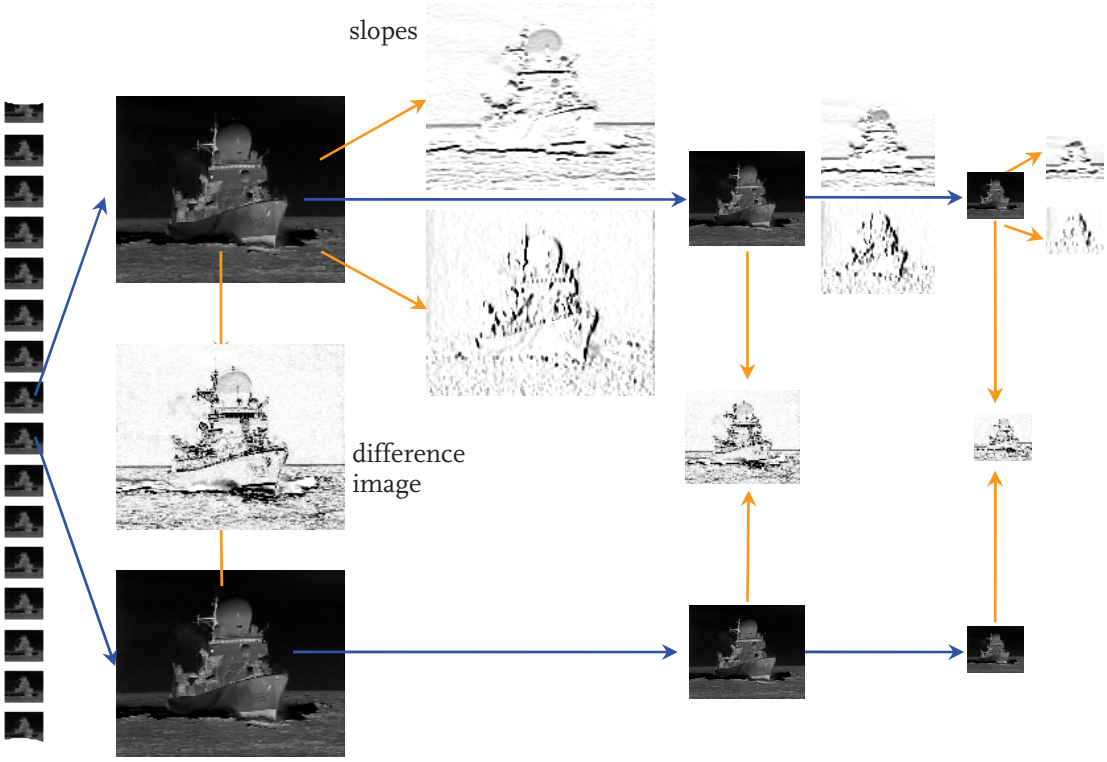


Figure 2. Generation of images to be used with the multi-resolution method. From left to right: from two succeeding frames three images are built: the difference image, an image that contains the horizontal slopes of the first frame and an image that contains the vertical slopes (black = no slope: white is steep slope). The original frames are downscaled by a factor of two and from these images again the difference and slope images are calculated. This process is repeated until the images are small enough for a fast convergence; usually three levels are perfect. The images are analyzed from right to left.

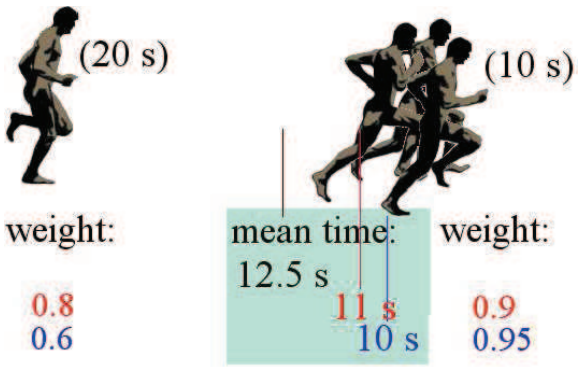


Figure 3. Illustration of the outlier principle: First, the four runners are seen as one group with an estimated 100 m time of 12.5 s. Then the weights are changed and new times are calculated (blue rectangle). After the third iteration, the slowest runner is exposed as an outlier or rather "outrunner". This example also illustrates the segmentation principle: after the third iteration, it has been established that the three fastest runners belong to one group. Then the algorithm will start searching for the next group by running the same algorithm on the outliers. The algorithm will find a second group, which – in this example – is the slowest runner.

New in our approach is that we keep the weights, so after the dominant motion has been found, we can simply go on using the same method on all outliers.

During the video recording, the overall illumination might change. Since it is essential for the method that the illumination does not change, we need to estimate the illumination changes and correct for it. We assume that the illumination changes proportionally to the intensity of each pixel, so we need one extra parameter. During the iterative loops, the intensity of the second frame is gradually changed until it matches that of the first frame. This kind of global intensity change may result from for instance the sun disappearing behind the clouds, although there will always be local differences such as shadows and glittering. Anyway, it appears that a single, global correction works well for all sequences used in this study. It is possible though to include different types of global illumination changes (see for instance [Kim et al., 2005]). It should be noted that the illumination changes usually are very small when two successive frames of a video sequence are used (typically less than 0.2 %). However, in practice it may be necessary to use frames with a much larger time distance, for instance if the expected motions are very slow. If the global illumination changes by even a few percents, inclusion of the illumination change factor is crucial, because otherwise the method might not converge or find wrong motion parameters.

Whenever colour sequences were used, the same method was applied, but with the intensity replaced by a triplet consisting of the primary colour components (red, green and blue). We use only one set of motion parameters (not three sets), which is calculated for the colours together. One might be tempted to think that using three sets of parameters (one set for each primary colour) would improve the segmentation process, because the colour would act as a cue as well. However, any real colour seldom is a pure primary colour (apart from the fact that the red, green and blue colours are additive in contrast to paint colours which are subtractive), so these three colours almost always contain mainly the same information.

Linear motion models, such as described above, have proved to be very useful and robust for motion estimation (see [Fuh and Maragos, 1991] or [Torr and Murray, 1993]), motion segmentation (see [Bouthemy and Rivero, 1987]) and tracking (see [Meyer and Bouthemy, 1992]).

Experimental results: background motion

First, the original method (without the extension for finding multiple objects) was used to see whether background or camera motion could be estimated reliably for real infrared or colour video recordings. All results described in this paper were obtained with identical thresholds and weight function parameters and all were using three resolution levels. The quality of the obtained motion parameters was judged by visual inspection of the *displaced frame differences*; this is a picture of the difference of two successive frames, of which one is corrected for its motion relative to the other frame. Hence, if the difference picture shows regions containing pixels that are non-zero, those regions do have a different motion.

An example involving rotation is shown in Fig. 4. The frames were extracted from a moderate quality AVI movie. Despite the poor quality of some of the intermediate frames, the correct motion is found and the average of 16 frames that are corrected for their relative motion is a sharp image (see Fig. 5). Therefore, the method correctly finds the (irregular) movements of the camera.

Figure 4. Left: first frame of a sequence. Right: sixth frame of the same sequence: blurry images like this one are typical for compressed movies.

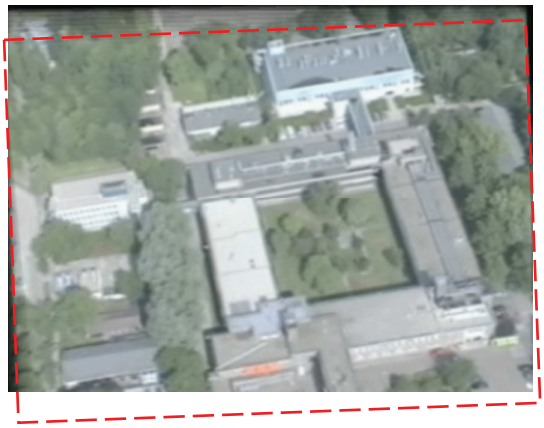


Figure 5. Average of 16 frames. The dashed rectangle shows the borders of the last frame of this sequence after correcting for the observed motion. The borders of all separate frames after correction for the estimated motion are just visible at the upper part of this figure (assuming the printing quality is good enough).

Fig. 6 shows an example of the results of a zoom sequence, taken with a camera on board of a ship. Again, the correct "motion" parameters were found, as can be seen in the averaged image after correction. Fig. 6 also shows a similar example, but now the object is approaching the camera, resulting in an apparent zoom. During the video recording, the vehicle changes its direction a little but also rotates in the plane of the image (compare Figs. 6d and 6e), but these motions are included in the model, so the correct motion is found. In fact, this sequence lasted until the vehicle almost reached the camera. Therefore, in the last frame only the grille of the vehicle was visible. Still the average of all frames was sharp at the position of the grille.

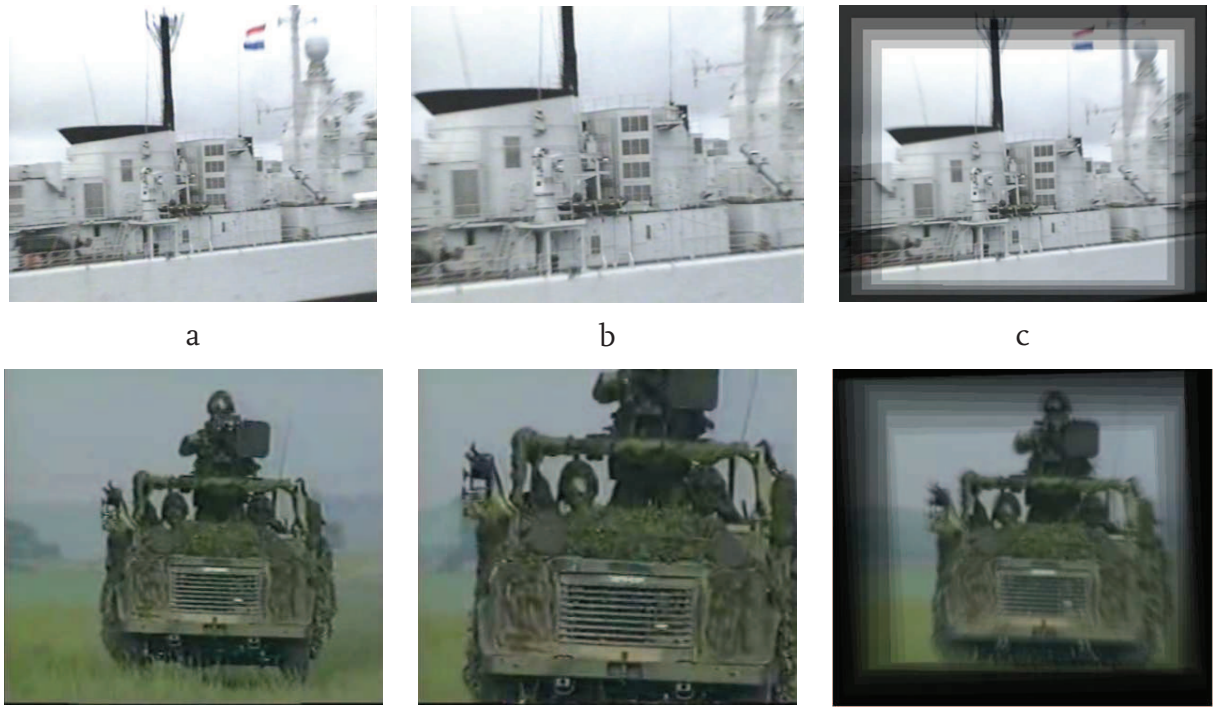


*Figure 6. (*a, b*): first and last frame of the zoom sequence. (*c*): average of five frames, after correcting for the estimated zoom. (*d, e*): first and 11th frame, taken from a long video (source: YouTube). (*f*): averaged frames after correction for the motion of the vehicle.*

The motion detection method was tested further with extremely noisy, realistic images. One result, obtained from an old infrared video recording of the HNLMS Tydeman is shown in Fig. 7. All displaced frame differences for the HNLMS Tydeman contained hardly any structure, so the image moves as a whole and its motion was correctly estimated. Obviously, this "motion" results from camera movements, so in fact it reflects the inverse camera motion. The motion parameters were also estimated by measuring the position of the visible lights, which were mounted on the ship, in successive frames. Both estimates agreed to within experimental error. Even though only six frames were averaged after correction for the motion, the resulting image clearly shows a reduction of the noise (see Fig. 7). From these experiments, it can be concluded that the motion detection method works very well and can be used to correct for camera movements for instance with the purpose of averaging frames.

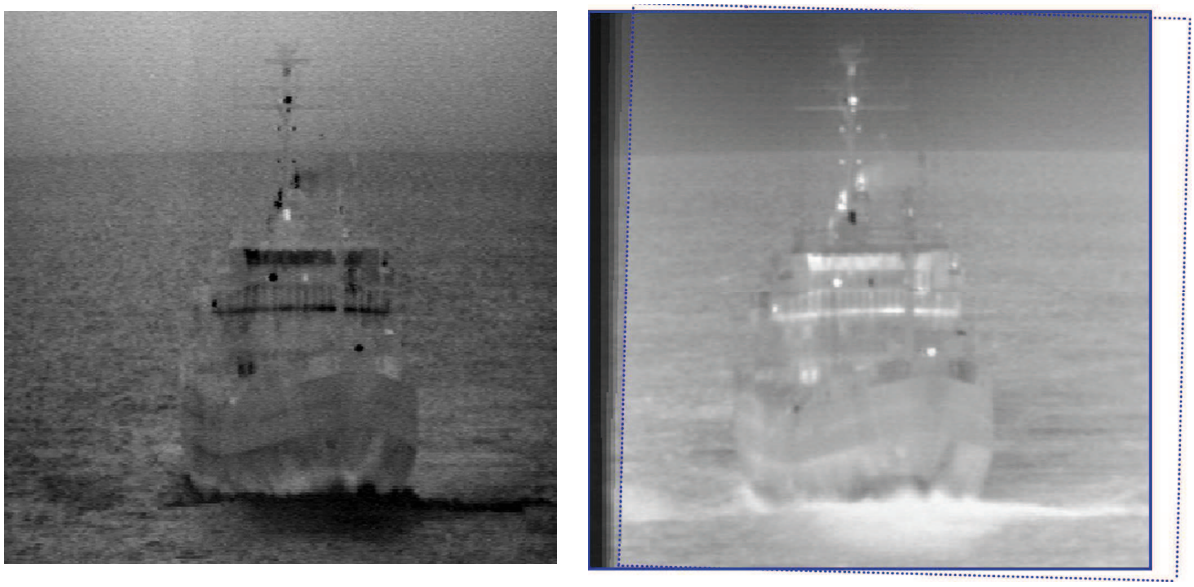


Figure 7. Left: noisy infrared image. The ship rotates by about 0.25 degrees and it 'moves' by 10 to 20 pixels between successive frame due to a moving camera. Right: average of six successive displaced frames. The dotted rectangle indicates the contour of the sixth displaced frame.

Segmentation on the basis of motion

Theoretical description

After the last iteration in the method described above, all the weights of the pixels are known, so it is easy to determine which pixels contribute to the final motion estimate. So in principle we are able to segment the images into areas that move differently. In practice, this is not so straightforward however. Due to noise and other irregularities (such as small sea waves), many isolated pixels belonging to the background will not fit well with the model, and pixels within the foreground will often fit by chance. There are several ways to improve this situation (see for instance [Odobez and Bouthemy, 1994]), one of which will be discussed in this paper.

Since we want the method to find objects autonomously, we need an exact criterion for which pixels 'belong' to the background. Since the weights after the final iteration tend to be either very small (≈ 0) or large (≈ 1), as will be shown later in Fig. 13b, a threshold of 0.5 seems to be a good choice. However, if we simply exclude all points that have a weight below this threshold, many isolated pixels and many pixels that in reality belong to the already found background will still be included. Isolated pixels cannot be used anyway, because the gradient is not defined at these points. There is also a principal problem: pixels can fit several motion models at the same time. This happens for instance in areas with a constant colour, or in areas that have a regular pattern. Pixels that lie in constant areas will be automatically assigned to the background object, whether or not this is correct. Methods to solve this kind of ambiguous situations are beyond the scope of this paper.

In the present study the decision whether or not to exclude a pixel from further calculations, is based on the weights of the 5 × 5 neighbourhood of that pixel. If 12 pixels out of these 25 pixels have a weight above 0.5, the central pixel is excluded from further calculations. Although with this approach good results were obtained for the images used in this study, it can certainly be improved and therefore should be the subject of further studies.

In order to have a flexible system, we implemented the method such that several motion models could be used:

- two parameters for translation only;
- three parameters for translation and rotation over a small angle;
- three parameters for translation and zoom;
- four parameters for translation, rotation and zoom;
- four parameters for translation and asymmetric zoom;
- six parameters for any motion that can be described by a linear motion model.

It is also possible to use a combination of these models: for instance, the two-parameter model can be used if it is likely that the camera movement will result in a pure translation. It can then be followed by the complete six-parameter model if the other objects are likely to have a more complicated motion.

Experimental results on segmentation

Synthetic sequences

The extended method was tested on a number of synthetic sequences, in order to get a feeling for the accuracy of the method. The synthetic images consisted of moving backgrounds with several differently moving objects. A typical example of such a synthetic image set is shown in Fig. 8. Here three objects rotate together about the same axis. The background of the second image is shifted by 1.75 pixels horizontally and 2.75 pixels vertically. Furthermore, the intensity of the background of the second image was deliberately lowered by 6.0%. After the first run of the algorithm, the motion parameters for the background were found to within 0.1%. The estimated global illumination factor was -6.3%. Note that the method can find a translation over a non-integral number of pixels. From the results of this and other synthetic and real sequences (not shown here), it can be concluded that the method can be used to estimate sub-pixel movements as well. The weights at the end of the first run are shown in Fig. 9a. From the weights, areas are calculated that do not fit the motion model (Fig. 9b); these areas can be used directly as an indication of the segments. After the second run of the algorithm, using only these areas, the affine parameters of the rotating rectangles are found. The correct values were found to within 1%. Fig. 10 shows the averages after correcting the second image for the estimated motion.

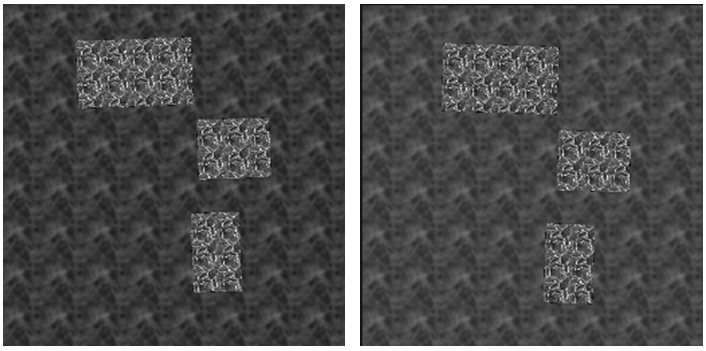
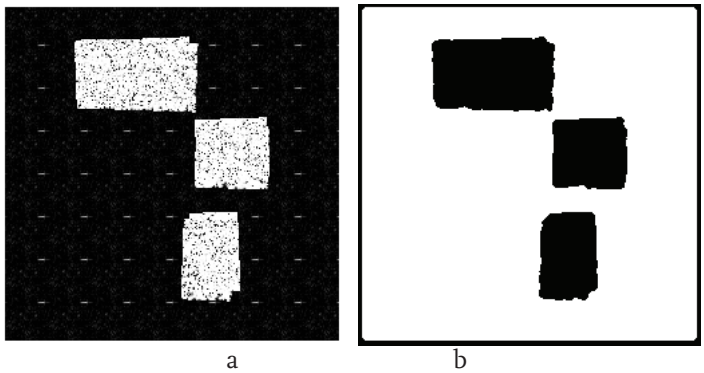
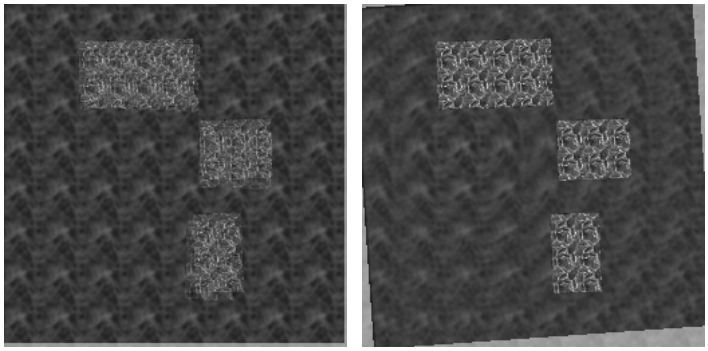


Figure 8. Synthetic frames, the rectangles are rotated clockwise over 4 degrees about a point close to the lower left corner



a b

Figure 9. (a) Weights after first run (black = 0; white = 1). (b) The 5 × 5 neighbourhood requirement ensures that loose pixels are not used in the second run; only the black rectangles are used in the second run.



a b

Figure 10. (a) Average after correcting the second image (Fig. 8b) for the obtained background 'motion', so the background is sharp. (b) Average after correcting the image for the obtained motion of the three rectangles, so the foreground is sharp.

Real sequences

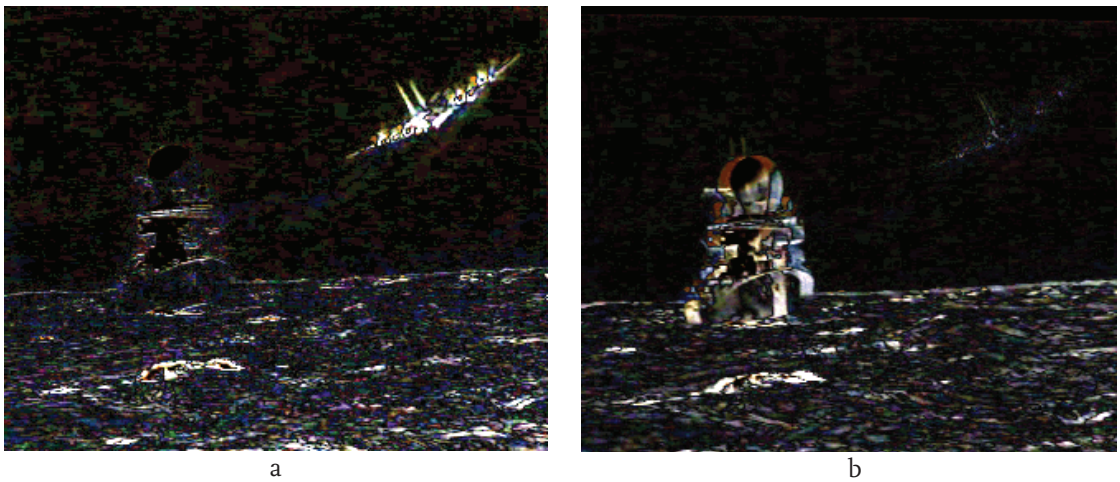
Next, the algorithm was tested on several real sequences. An example is shown in Fig. 11, which contains two objects moving at different speeds. In Fig. 12 the displaced frame differences, corresponding to the example of Fig. 11, are shown and in Fig. 14 the displaced frame averages are shown. Sometimes *averaged* frames are preferred to show the results, because if an averaged frame is not perfect, this is noticed immediately by the human eye. On the other hand, the difference between two frames may be very small even if an imperfect correction is applied, so the displaced frame difference not always gives a clear view of the quality of the motion estimate. Averaged frames have the added advantage that they can be calculated for any desired number of frames. The results for

the sequence of Fig. 11 as shown here were obtained with a four-parameter (translation, rotation and zoom) model.

With this example, inaccurate results were obtained with the six-parameter model. In this case both the airplane and the white wave that is visible at the foreground move with respect to the background. Since the airplane and the wave move in opposite directions this is approximately the same as a rotation of both objects about a point somewhere between the two objects. In such cases, very many pixels are required to find a correct solution, even if there is not much noise. With a limited number of (possibly noisy) pixels, many more ambiguous solutions are possible. If the motion parameters must be used directly for some application, this may cause a problem, but if we only need the result of the motion (for instance for calculating averaged images, as illustrated before), this is less important. However, if one knows beforehand that certain motions are not very likely, it is better to use a motion model with as few parameters as possible. If for instance 25 or 50 images per second are recorded, it is very unlikely to have a large rotation component.

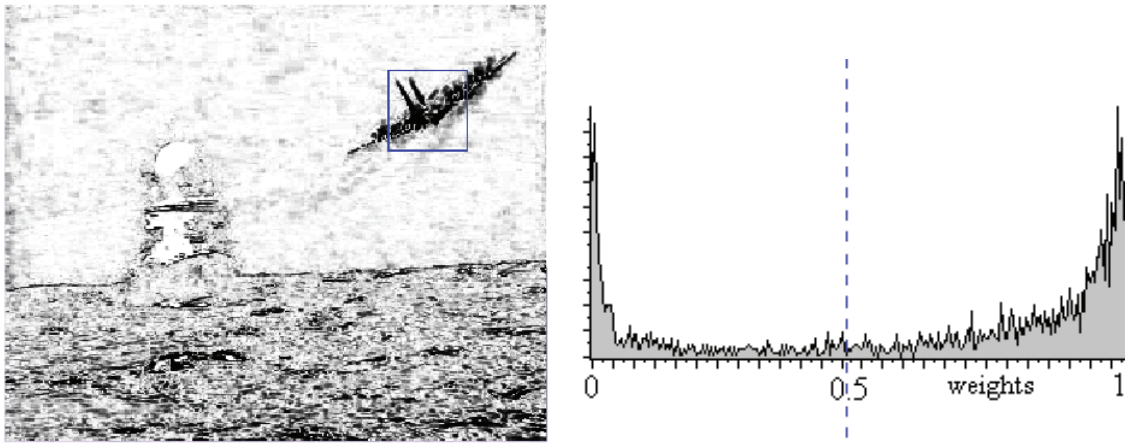
a b

Figure 11. (a) One frame taken from a sequence; the aeroplane is approaching the frigate (b) Black: pixels that are used for further calculations, based on the weights after the first run (black = 1; white = 0).



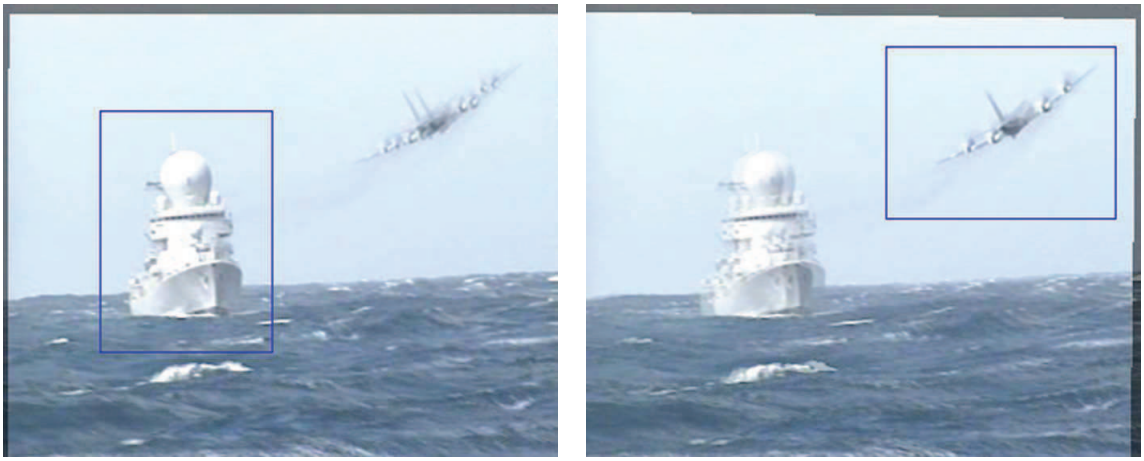
a b

Figure 12. Displaced frame differences (black = no difference; all intensities are multiplied by 4 to make the differences more clearly visible. (a) After the first run, the estimated motion is that of the largest object, so the aeroplane is visible. (b) After the second run, the estimated motion is that of the aeroplane.



a b

Figure 13. (a) Weights after the first run (black = 0, white = 1). (b) Histogram of the weights in a small rectangular area around the aircraft.



a b

Figure 14. Displaced averages of two frames after first (a) and second (b) motion estimate. The rectangles indicate the objects of which the motion parameters were found, so these objects should be sharp.

With the restricted motion model we obtained very good results for this particular sequence. We agree with [Wu and Kittler, 1990] that it is far from clear whether more complicated motion models will yield better results in practice. With many of our test sequences good results were obtained if *only* translations were taken into account. In such cases, models using four parameters almost always gave somewhat better results than the six-parameter model. One example of such a situation is shown in Fig. 15a. The camera tries to follow the two men. It is very likely that the camera motion is a simple translation. Indeed, when a two parameter motion model is used, the algorithm converges rapidly and it finds the correct background motion. In this particular case a much slower six-parameter motion model yields about the same results, because there are no ambiguous situations and because a clear structure is present in the background. During the second run of the method, the motion of the two men should be found. Since the men are moving freely, any apparent motion can be possible, so now a six-parameter affine model is used. We did not compensate for intensity changes this time.

Of course a more advanced model is needed to account for all details of the movements involved. The results are so good that in the movie that is made of the frames corrected for the motion of the foreground object, the two men are quite sharp and you can even see the hand of one of the men moving. We cannot show this in print, so instead the average of all frames, after compensation for the estimated motion of the two men is shown in Fig. 15d.

a b

c d

Figure 15.
(a) *One of the original frames.*
(b) *All pixels that move with equal speed (black).*
(c) *Average of six frames after compensation for the motion of the background. The background is very sharp. At the underside of the image you can see how much the individual frames had to be shifted to obtain the compensation.*
(d) *Average of six frames after compensation for the motion of the foreground "object".*

Men in camouflage uniforms caught in the act

The motion estimation method is particularly suited to find objects that do have a clearly visible structure but are still hardly visible because of the resemblance to the structure of its surroundings (for instance men in camouflage uniforms against a background that resembles the camouflage pattern). Figs. 16 and 17 show two examples. From two succeeding frames the location of the moving objects are already found. For every new frame, the contours of the moving objects will become more certain (see Fig. 16d). In both cases, the two men can be found by other techniques, not based on motion, because the structures of their uniforms differ considerably from the background. However, when the background changes due to camera movements, many standard methods to find moving objects, which are often based on differences between succeeding images, will fail [Nascimento and Marques, 2006; Boult et al. 2001]. Using synthetic images, we have been able to show that our method works very well, even when the objects have exactly the same structure as the background. It is then impossible to see the objects, unless they move. If they move, humans are able to see the exact shape of the moving objects immediately. The method described in this paper will also find the exact shape after analyzing just a few frames.

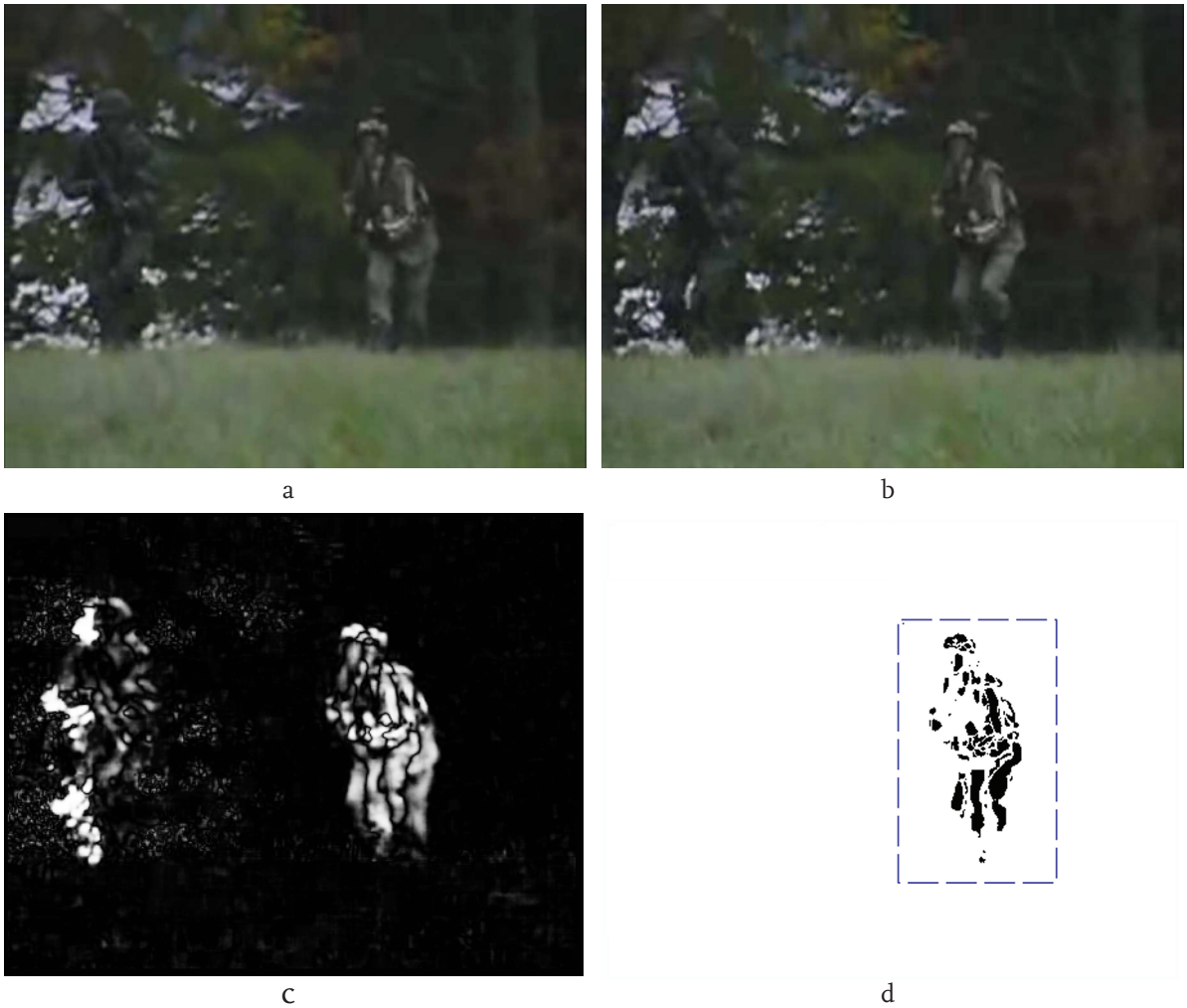


Figure 16 a, b*: two succeeding frames. Source: YouTube.* c*: weights after first run (black: does* **not** *fit the background motion).* d*: the combined result of three succeeding frames; the contours of the right man are already clearly visible; the left man is less visible (not shown in this image) because he hardly moves.*

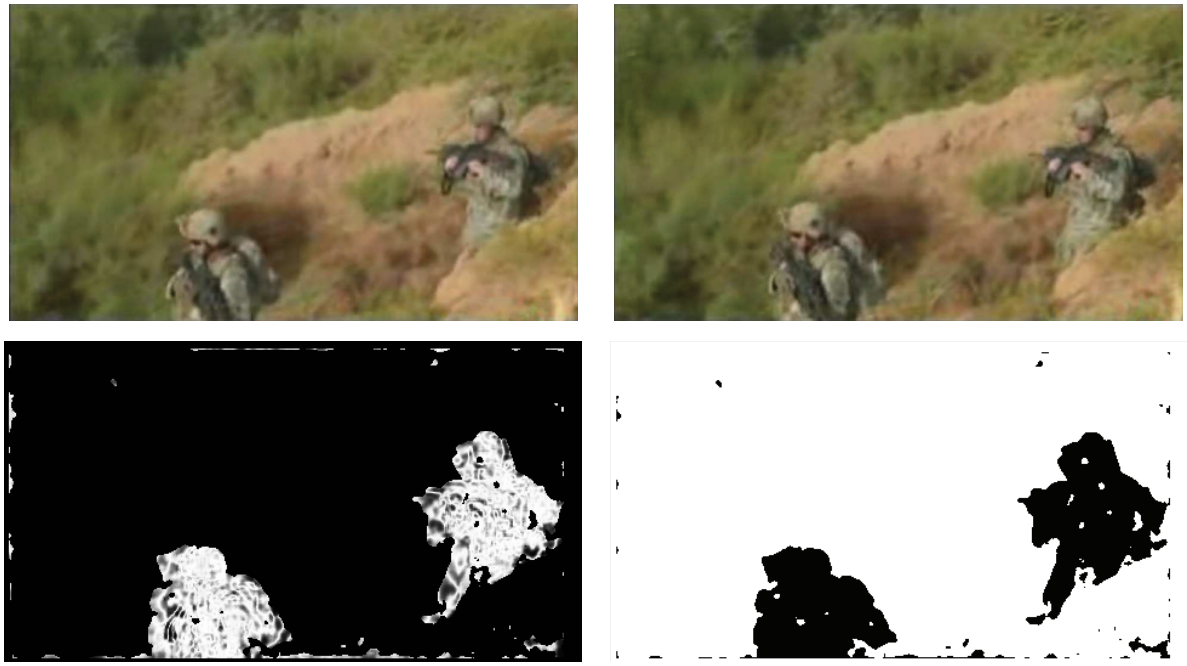


Figure 17. Above: two succeeding frames. Source: YouTube. Below, left: weights after first run (black: does not fit the background motion). Below, right: black indicates the presence of objects that move with respect to the background.

Real time calculations

It is evident that for most military applications all processing must take place in real time or near real time. Although the simulations were not optimized for speed, on a 3 GHz personal computer the current implementation needs only a few seconds for the calculations of images of about 512 × 512 pixels. Therefore, it may be expected that with optimizations and special purpose hardware near real time applications will be possible.

Conclusions

The motion estimation method as described by Odobez and Bouthemy gives good results even for noisy sequences. The method yields correct parameters for camera movements. We have shown that the method can be applied repeatedly, while excluding pixels that were assigned to a motion region in the previous estimation step. The weights after each final motion estimate can then be used to find the contours of the regions where the object is located (segmentation). Experiments with synthetic as well as with real infrared sequences show very good results. The extended method is able to find objects moving at different speeds. A possible application is finding moving, camouflaged objects. The method is able to find the motion parameters of the separate objects, so it becomes possible to make averages where one of the objects is sharp, even if the object is moving, to suppress noise. However, one may have to restrict the model when ambiguous motions are possible. It is suggested that in future work the method is extended by adding a mechanism to consider only contiguous areas, in order to avoid artefacts due to ambiguous situations. In principle, this method can be repeated until all objects that move differently are found, but in practice the method will work only with rather large, well-structured objects.

Acknowledgements

The infrared sequences were kindly supplied by Dr. P.B.W. Schwering of the Electro-Optics Group of TNO Defence, Safety and Security in The Hague, The Netherlands. The visible light video sequences (Figs. 4, 6a, 6b, 11 and 15) are courtesy of the "Audiovisuele Dienst Defensie" (Dutch Defense-AudioVisual service).

References

Boult, T.E., Micheals, R.J., Gao, X and Eckmann, M. (2001) Into the Woods: Visual Surveillance of Non-Cooperative and Camouflaged Targets in Complex Outdoor Settings, *Proceedings of the IEEE*, pp 1382-1402.

Bouthemy P. and Rivero J.S. (1987) A hierarchical likelihood approach for region segmentation according to motion-based criteria, *Proceedings of the 1st Int. Conf. In Comp. Vision,* pp 463-467.

Fuh C.S. and Maragos, P. (1991) Affine models for image matching and motion detection, *Tech. Report CICS-P-280*, Center for Intelligent Control Systems.

Hupkens Th.M., Vos M. de, Patras I. and Hendriks E.A. (2000) Segmentation Based on Successive Robust Motion Estimates, *Proceedings of the sixth annual conference of the Advanced School for Computing and Imaging*, pp 237-244.

Kim Y-H, Martinez, A.M. and Kak A.C. (2005) Robust motion estimation under varying illumination, *Image and Vision Computing* 23 (4), pp 365-375.

Nascimento, J.C. and Marques, J.S. (2006) Performance Evaluation of Object Detection Algorithms for Video Surveillance. *IEEE Transactions on Multimedia 8(4)*: 761-774.

Meyer F. and Bouthemy P. (1992) Region-based tracking in an image sequence, *Proceedings of the 2nd Europ. Conf. In Comp. Vision*, pp 476-484.

Odobez J.M. and Bouthemy P. (1994) Detection of multiple moving objects using multiscale MRF with camera motion compensation. *Proceedings of the 1st IEEE ICIP,* vol. 2, pp 257-261.

Odobez J.M. and Bouthemy P. (1995) Robust Multiresolution Estimation of Parametric Motion Models, *J. Visual Comm. Image Representat.*, **6**(4), pp 348-365.

Torr P.H.S. and Murray D.W. (1993) Statistical detection of independent movement from a moving camera, *Image and Vision Computing* 11(4), pp 180-187.

Wu S.F. and Kittler J. (1990) A differential method for simultaneous estimation of rotation, change of scale and translation, *Signal Processing: Image Communication* 2, pp 69-80.