A robust method for camera motion estimation in movies based on optical flow

Nhat-Tan Nguyen* and Denis Laurendeau

The Computer Vision and Systems Laboratory,
Department of Electrical and Computer Engineering,
1065, Avenue de la Medecine
Laval University,
Quebec (QC) G1V 0A6, Canada
E-mail: ntnguyen@gel.ulaval.ca
E-mail: laurend@gel.ulaval.ca
*Corresponding author

Alexandra Branzan-Albu

The Computer Vision and Systems Laboratory,
Department of Electrical and Computer Engineering,
Laval University,
University of Victoria,
Victoria (BC) V8W3P6, Canada
E-mail: aalbu@ece.uvic.ca

Abstract: Camera motion estimation plays an important role in digital video analysis algorithms such as video indexing and retrieval or automatic movie analysis. Several algorithms have been proposed to solve this problem in MPEG videos. This paper presents an optical flow-based approach for the camera motion estimation in all kinds of digital video formats, especially in movies. It compares the motion vector fields (MVFs) with six predefined templates to determine the type of motion. The MVFs are generated by using high-accuracy optical flow computation. The advantage of the method lies in its robustness to noisy environments such as false motion vectors and object motions. Comprehensive experiments with video clips extracted from well-known feature movies demonstrate the performance of the proposed approach.

Keywords: camera motion detection; optical flow; MVF; motion vector field; shot detection.

Reference to this paper should be made as follows: Nguyen, N-T., Laurendeau, D. and Branzan-Albu, A. (2010) 'A robust method for camera motion estimation in movies based on optical flow', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 9, Nos. 3/4, pp.228–238.

Biographical notes: Nhat-Tan Nguyen is a PhD Student in the Department of Electrical and Computer Engineering, Laval University, Quebec, Canada. He received is ME in Electrical Engineering in 2006 from Lava University. His research interests are is the areas of medical image analysis, video processing and human motion recognition.

Denis Laurendeau is a Full Professor in the Department of Electrical and Computer Engineering, Laval University, Quebec, Canada as well as Chief Director of the Computer Vision and Systems Laboratory. He received a PhD from Laval University in 1986. His area of interest include artificial 2D/3D vision applied to fixed and mobile robotics, telerobotics and artificial vision applied to biomedical engineering.

Alexandra Branzan-Albu is an Associated Professor in the Department of Electrical and Computer Engineering in the University of Victoria, Victoria, BC, Canada. She received her BE and PhD, both in Electronics, from Polytechnical University of Bucarest, Romania in 1992 and 2000, respectively. Her current areas of research interests include computer vision, pattern recognition, image and video processing, motion analysis and tracking, medical imaging, virtual reality and simulation.

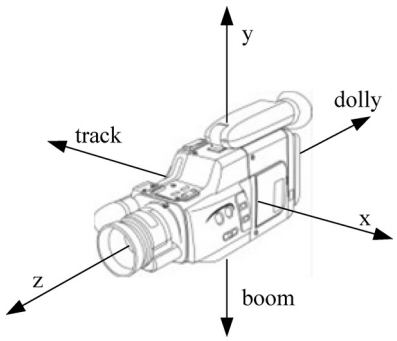
1 Introduction

In this paper, we develop tools based on low-level information to analyse camera motion as part of the E-Inclusion project. The goal of the E-Inclusion project is to create powerful audio–video tools that will allow multimedia content producers to improve the richness of the multimedia experience for the blind, the deaf, the hard of hearing and the hard of seeing, by automating key aspects of the multimedia production and post-production processes.

The estimation of camera motion is important for several video analysis systems. Camera motion is often used as an expressive element in film production. Given the types of camera motion, a video sequence can be decomposed into temporal segments of video with smoothly changing content called 'shots'.

There are different types of camera motion: rotation around one of the three axes and translation along the x- and y-axis. The axes of a camera are presented in Figure 1. In the context of our approach, the translation along the x-axis (y-axis) and the rotation around the y-axis (x-axis) are considered as one type of motion. Besides, translation along the z-axis can be considered as equivalent to zoom in and zoom out. The proposed approach is able to distinguish six types of camera motion on motion vector fields (MVFs).

Figure 1 The axes of a camera



Some recent techniques use motion data available from compressed video files such as motion vectors and discrete cosine transformation (DCT) coefficients in the MPEG files instead of performing optical flow computation (Ewerth et al., 2004; Gillespie and Nguyen, 2004; Kim et al., 2000; Lee and Hayes, 2002; Tiburzi and Bescos, 2007). This direction seems very computationally effective but it has its own issue. The MPEG motion vectors often do not represent the true motion of a frame sufficiently due to outlier vectors. Thus, we choose to perform the accurate optical flow calculation in order to generate the MVFs. Then we define templates so as to represent camera motion and estimate template parameters from the computed MVFs.

A comprehensive test set has been created consisting of four video sequences extracted from "The Fabulous Destiny of Amélie Poulain" (Le Fabuleux destin d'Amélie Poulain) including many kinds of camera motion. The main challenge in using sequences extracted from motionpictures is that we do not have any control or knowledge of camera motion and/or calibration.

The remainder of this paper is organised as follows. In Section 2, related work is discussed. Our approach to estimate camera motion in movie sequences is presented in Section 3. Section 4 illustrates how the algorithm can be used to determine the type of camera motions in real movie videos. Section 5 concludes this paper and outlines areas for future work.

2 Related work

The related work can be categorised into two different groups according to the camera motion model used. The first group analyses motion field vectors then estimates model parameters. These parameters are afterward evaluated to reveal the camera motion involved (Ewerth et al., 2004; Tiburzi and Bescos, 2007). The second group directly implies the observed optical flow pattern by using the angular distribution or the power of optical flow vectors (Patel and Sethi, 1997; Xiong and Lee, 1998).

For example, Tiburzi and Bescos (2007) describe the camera motion by the set of six parameters of the affine model. Camera motion patterns can be identified via thresholding of a set of classification functions in which each is sensitive to the presence of pan, tilt, zoom or roll patterns, respectively. This method is proposed for working in real time on MPEG sequences. Kim et al. (2000) assume a two-dimensional affine camera model and state that they can detect six types of motion: zoom, rotation, pan, tilt, object motion and stationary. Thus, results for translation along the x- (y-axis) and rotation around the y- (x-axis) are merged. Ewerth et al. (2004) present an approach which can distinguish between camera rotation and translation. Their approach is based on an appropriate 3D camera model that includes rotation, translation and zoom in and out. They extract the motion vectors directly from the compressed MPEG stream and then apply an outlier removal algorithm to obtain a reliable MVF. Gillespie and Nguyen (2004) have chosen the four-parameter affine model to represent camera motion. This paper shows the improvement in the estimation of the camera motion model parameters, but it does not show how the algorithm can classify between the types of camera motion, such as pan and zoom.

Xiong and Lee (1998) divide image-frames into subregions and then analyse the projected x and y components of optical flow separately in the different subregions of the images. Different camera motions (such as zoom, rotation, pan, tilt, object motion and stationary) are recognised by comparing the computed mean values and standard deviations with the prior known patterns. Naito et al. (2006) propose a camera motion detection method using

a background image generated by video mosaicking based on the correlation between feature points on a frame pair. To detect the camera motion, the position of frames on the background image is converted to feature parameters so as the coordinate of the centre of each quadrangle camera frame and the distance between the centre and vertex of the quadrangle frame, respectively.

3 Camera motion estimation

In this section, we present our algorithms for the estimation of camera motion (pan, tilt and zoom). The method is based on analysing the optical flow information. It consists of following steps which are explained in detail below:

1 optical flow computation
2 MVF processing
3 motion templates
4 template matching.

3.1 Optical flow computation

Our implementation follows Brox's algorithm to compute optical flow because it provides the lowest error rate among the noted algorithms (Brox et al., 2004). Details of implementation, especially the numerical approach, have been adopted from Brox's work. Extension to different colour channels, and the concept of a local smoothing function, have been borrowed from the work of Sand and Teller (2006). The algorithm is briefly introduced below.

It is known that a basic assumption in computing the optical flow is that *intensity is conserved before and after the motion*, that is, $\mathrm{d}I(x, y, t)/\mathrm{d}t = 0$. Moreover, the gradient of the image intensity is also assumed not to vary due to the displacement. The gradient constraint equation is, therefore, derived as:

$$\nabla I(x, y, t) = \nabla I_t(x + u, y + v, t + 1) \tag{1}$$

where $I_t(x + u, y + v, t + 1)$ denotes the partial time derivative of $I(x, y, t)$ and $\nabla I(x, y, t) = (I_x(x, y, t), I_y(x, y, t))^{\mathrm{T}}$ denotes the spatial gradient. Brox has defined a further assumption: the smoothness of the flow field. As the optimal displacement field will have discontinuities at the boundaries of objects in the scene, it is worthwhile to generalise the smoothness assumption by demanding a piecewise smooth flow field. Let $\mathbf{x} = (x, y, t)^{\mathrm{T}}$ and $\mathbf{w} = (u, v, 1)^{\mathrm{T}}$, the computation of optical flow is a combination of intensity and smoothness terms of the images involved.

$$E_D(u, v) = \int_{\Omega} \Psi\left(\left|I_w\right|^2 + \gamma\left|\nabla I_w\right|^2\right) \mathbf{dx} \tag{2}$$

$$E_S(u, v) = \int_{\Omega} \Psi\left(\left|\nabla_3 u\right|^2 + \left|\nabla_3 v\right|^2\right) \mathbf{dx} \tag{3}$$

with

$$I_w = I(\mathbf{x} + \mathbf{w}) - I(\mathbf{x})$$

$$\nabla I_w = \nabla I(\mathbf{x} + \mathbf{w}) - \nabla I(\mathbf{x}))$$

The total energy is the weighted sum between the data term and the smoothness term

$$E(u, v) = E_D(u, v) + \alpha E_S(u, v) \tag{4}$$

with some regularisation parameter $\alpha > 0$.

The algorithm also uses a coarse-to-fine strategy called the 'warping' technique. With this multiscale approach, optical flow estimation gives significantly smaller angular errors and excellent robustness to noise.

3.2 MVF processing

We use a basic yet efficient description of the camera motion at each frame via phase histograms of the motion vectors. This assumes that the intensity of the camera motion is constant in each segment, which is usually the case.

The classification of camera motion using phase histograms often separate a frame into several subregions (Lee and Hayes, 2002; Tiburzi and Bescos, 2007). Working on the subregions of images individually can alleviate the effectiveness of objects located in some local regions of the frame. Therefore, a division into four basic regions is quite appropriate for our goal. The horizontal and vertical optical flows then form the MVF. Motion vectors within a particular region, or a frame often shows a tendency to stand in correlation during camera operation. Thus, let us define the average magnitude and direction of the motion vectors within a frame as follows:

$$M[n, t] = \frac{1}{N_n}\sqrt{\left(\sum_{k=1}^{N_n} v_{x,k}[n, t]\right)^2 + \left(\sum_{k=1}^{N_n} v_{y,k}[n, t]\right)^2} \tag{5}$$

$$\big(V_x[n, t], V_y[n, t]\big) = \left[\frac{1}{N_n}\sum_{k=1}^{N_n} v_{x,k}[n, t], \frac{1}{N_n}\sum_{k=1}^{N_n} v_{y,k}[n, t]\right] \tag{6}$$

where $M[n, t]$ is the magnitude of the nth subregion in frame t, $v_{x,k}[n, t]$ and $v_{y,k}[n, t]$ are the horizontal and vertical motion vectors for the kth pixel of the nth subregions, respectively, N_n is the total number of pixels in the nth subregion of the selected frame, $V_x[n, t]$ and $V_y[n, t]$ are the components of a motion vector representing the motions occurring in the nth subregion. Then $M[n, t]$ is used to decide whether or not a given frame has an adequate percentage of motion associated with basic camera operations and $V_x[n, t]$ and $V_y[n, t]$ are used as parameters to assign a camera operation into one of six different types.

3.3 Motion templates

An image is divided into four basic subregions as shown in Figure 2, w and h are the width and height of a frame. The size of the subregion is chosen as a fraction ρ of the frame height and width. Optical flow is investigated in different subregions. We build templates based on these subregions that are used to classify six basic camera motions. Each subregion in the template is represented by a vector $(V_x[n, i], V_y[n, i])$. These vectors have a unique length, but, depending on the types of motion, have a different orientation. Figure 3 shows the templates with their associated vectors. For further use, each template has been labelled by a number.

Figure 2 Basic four-subregion division of a frame (see online version for colours)

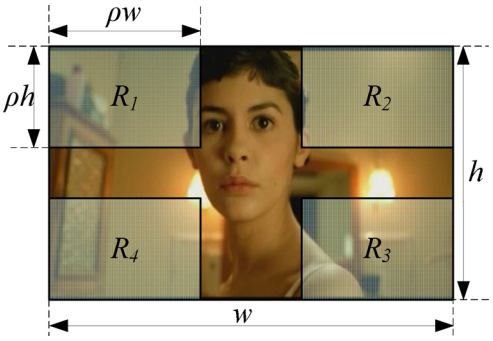
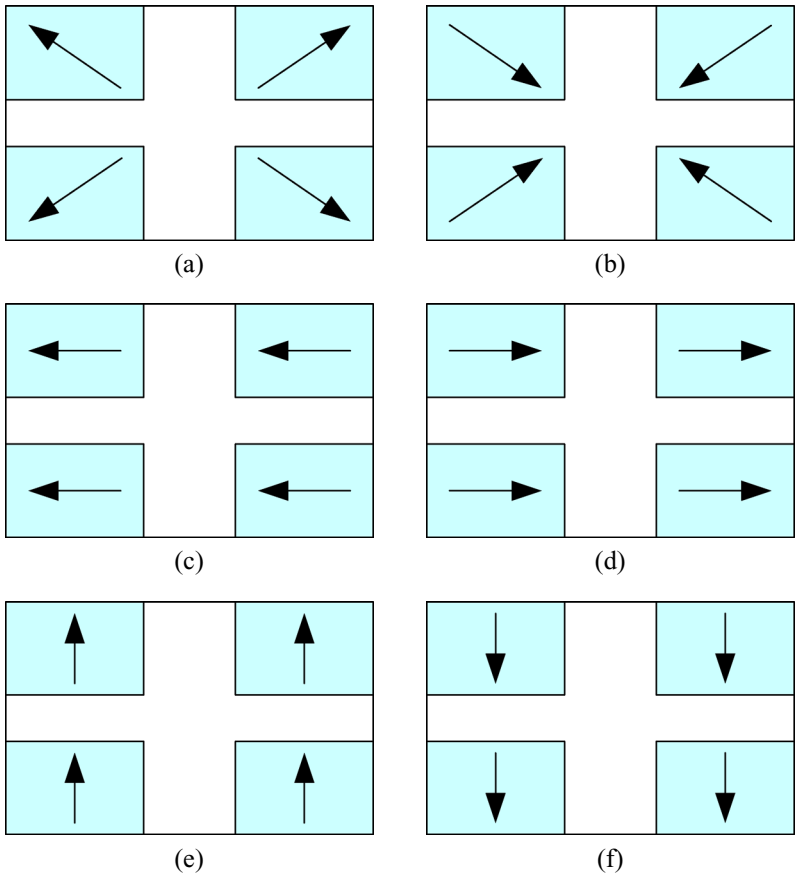


Figure 3 Basic camera motion templates: (a) zooming in – 1; (b) zooming out – 2; (c) panning left – 3; (d) panning right – 4; (e) tilting up – 5 and (f) tilting down –6 (see online version for colours)



3.4 Template matching

We suggest that most of the magnitude of the background subregions is greater than some threshold during camera movement. To determine if there is camera motion (Lee and Hayes, 2002), a magnitude function, D_{bg}, is defined by

$$D_{\text{bg}} = \frac{1}{N_{\text{bg}}} \sum_{n \in B} S[n, t] \tag{7}$$

with

$$S[n, t] = \begin{cases} 1, & \text{if } M[n, t] \geq \tau_{\text{mag}} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where N_{bg} is the total number of the background subregions, B is the set of background subregions and τ_{mag} is the magnitude threshold. If D_{bg} is greater than or equal to a threshold, τ_D, it signifies that there is a camera movement within the given frame. Afterwards, the template motion matching is processed. So given a motion vector $(V_x[n, t], V_y[n, t])$ of the nth subregion and a template vector $(V_x[n, i], V_y[n, i])$ of the same nth subregion in the ith template, then the angle between these vectors can be determined by:

$$\alpha_n^i = \arccos\left(\frac{(V_x[n, t] \times V_x[n, i]) + (V_y[n, t] \times V_y[n, i])}{\sqrt{V_x[n, t]^2 + V_y[n, t]^2} \times \sqrt{V_x[n, i]^2 + V_y[n, i]^2}} \right) \tag{9}$$

With the purpose of identifying the most appropriate template for camera motions of a given frame, a measure is defined by

$$A = \arg\min_i \left[\sum_{n=1}^{N_{\text{bg}}} \alpha_n^i \right], \quad 1 \leq i \leq N_{\text{tp}} \tag{10}$$

where N_{tp} is the number of templates. Therefore, the value found in A is associated with a type of camera motion that is predefined in the templates, then the frame is marked as having the given camera motion.

4 Experimental results

To test the proposed camera motion estimation algorithm on a realistic video, we have applied the algorithm to four short movie sequences. These sequences, listed in Table 1, are extracted from the movie 'The Fabulous Destiny of Amélie Poulain' (Le Fabuleux destin d'Amélie Poulain). The sequences consist of various scenes, camera motions with different speeds and view angles. These sequences are available at http://vision.gel.ulaval.ca/~ntnguyen/download.html.

Figure 4 shows some typical images from the sequences. In this study, parameters $[\rho, B, \tau_{\text{mag}}, \tau_D]$ are set experimentally to [0.3, 4, 0.08, 0.75]. We found that different sequences may need slightly different parameters to obtain the best results, but we used a single set of parameters to obtain reasonable results for all of the videos.

Table 1 List of test sequences

			Camera motion types		
Video sequence	*Total frames*	*Video shots*	*Zoom*	*Pan*	*Tilt*
Videosample01.avi	712	8	3	2	1
Videosample02.avi	632	11	2	0	1
Videosample03.avi	348	4	2	0	1
Videosample04.avi	537	4	2	1	2

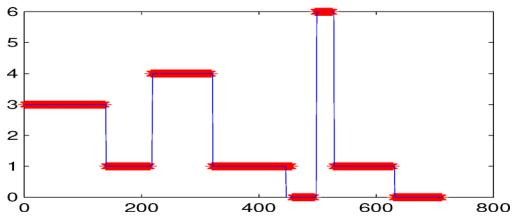
Figure 4 Sample images from the sequences (see online version for colours)

Table 2 shows the experimental results from applying the algorithm to the four test sequences. The sequences have been entirely analysed by a human observer for providing ground truth reference on camera motion. In first tested sequence (Videosample01.avi), our algorithm produced 12 missed and 12 false detections over 712 frames. These errors are often found on the frames within the transition between two different camera motions. We find similar situations in the second and the third movie sequences. In the second movie sequence (Videosample02.avi), there are a total of 632 frames, the algorithm produced 612 correct, 17 false , and 3 missed detections. The third movie sequence (Videosample03.avi) has 348 frames with only 7 false and 7 missed detections. In the fourth sequence (Videosample04.avi), the algorithm was tested on 537 frames, with 26 missed and 26 false detected frames. Several camera motion frames are mis-classified as static camera because in the process of shooting movies, for example, the cameraman usually commences a zooming-shot with a light zoom, then increasing it and finishing with a light zoom before stopping the shot (frames 450 to 480 in Videosample04.avi are detected as camera-static frames). Other error annotations are also caused by most of the objects in the video which move horizontally, a situation that gets even worse when these objects dominate the main portion of the screen.

Table 2 Results of detecting camera motion from the four test sequences

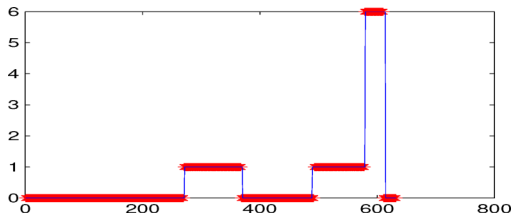
Sequence	*Motion type*	*Detected/actual*	*Missed*	*False*	*Precision(%)*
Videosample01.avi	Static	122/133	11	0	96.77
	Zooming in	317/307	0	10	
	Zooming out	0/0	0	0	
	Panning left	138/139	1	0	
	Panning right	104/103	0	1	
	Tilting up	0/0	0	0	
	Tilting down	31/30	0	1	
Videosample02.avi	Static	416/408	0	12	98.26
	Zooming in	184/189	0	5	
	Zooming out	0/0	0	0	
	Panning left	0/0	0	0	
	Panning right	0/0	0	0	
	Tilting up	0/0	0	0	
	Tilting down	32/35	3	0	
Videosample03.avi	Static	49/52	3	0	97.99
	Zooming in	115/119	4	0	
	Zooming out	95/93	0	2	
	Panning left	0/0	0	0	
	Panning right	0/0	0	0	
	Tilting up	89/84	0	5	
	Tilting down	0/0	0	0	
Videosample04.avi	Static	23/0	0	23	94.04
	Zooming in	176/199	23	0	
	Zooming out	59/57	0	2	
	Panning left	142/141	0	1	
	Panning right	0/0	0	0	
	Tilting up	0/0	0	0	
	Tilting down	137/140	3	0	

Figure 5 Results from the algorithm vs. manually annotated ground truth of the test sequences: (a) Videosample01.avi; (b) Videosample02.avi; (c) Videosample03.avi and (d) Videosample04.avi (see online version for colours)

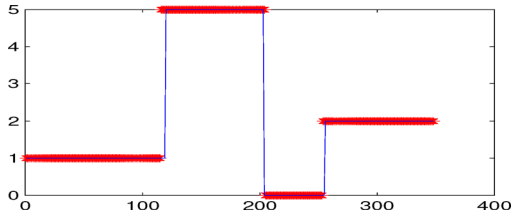


(a)

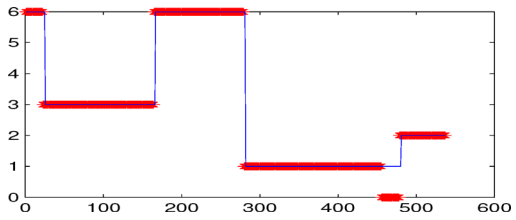
Figure 5 Results from the algorithm vs. manually annotated ground truth of the test sequences: (a) Videosample01.avi; (b) Videosample02.avi; (c) Videosample03.avi and (d) Videosample04.avi (see online version for colours) (continued)



(b)



(c)



(d)

Note: The blue line represents ground truth and the red dots indicate the output of the algorithm. The value from 1 to 6 represent the motion types as indicated by the template labels in Figure 3, label 0 represents the static camera.

Figure 5 presents the results of the algorithm performed on the sequences and their ground truth. The blue is the ground truth, the red dots represent the output of the algorithm. The four subfigures reveal that the algorithm's outputs and the ground truth are fitting closely.

5 Conclusion

This paper illustrates a simple and robust approach to characterise camera motion in the context of movie indexing, which is an important step for motion analysis of objects and video retrieval. While other camera motion estimation methods use affine models' parameters or perform an iterative optimisation algorithm, our method directly analyses optical flow vectors without any transformation. Using only the magnitude and the angle

of the vectors on several specified subregions of image, camera motion can be classified into six types. The test sequences are extracted from a real movie consisting of complex camera/object motions and noise. The experimental results provide a good illustration of the robustness of the method.

However, because the proposed approach is based on optical flow, it is limited by the computationally expensive optical flow estimation. This has been one of the key problems in computer vision for years. In the last two decades, the quality of optical flow estimation methods has increased dramatically as well as the performance of the computer. That is the motive for us to implement the optical flow-based approach.

Camera motion information, that is, dominant motion, can be exploited in further video analysis. The idea is using this information to make a motion compensation then extract the moving objects from the background.

Acknowledgement

This work is financially supported by the Department of Canadian Heritage through Canadian Culture Online.

References

Brox, T., Bruhn, A., Papenberg, N. and Weickert, J. (2004) *High Accuracy Optical Flow Estimation Based on a Theory for Warping*. Vol. 3024, pp.25–36.

Ewerth, R., Schwalb, M., Tessmann, P. and Freisleben, B. (2004) 'Estimation of arbitrary camera motion in MPEG videos', *Proceedings of the 17th International Conference on Pattern Recognition. ICPR 2004*, Vol. 1, pp.512–515.

Gillespie, W.J. and Nguyen, D.T. (2004) 'Robust estimation of camera motion in MPEG domain', *Proceedings of the IEEE Region 10 Conference (TENCON)*. Vol. 1, pp.395–398.

Kim, J-G., Chang, H.S., Kim, J. and Kim, H-M. (2000) 'Efficient camera motion characterization for MPEG video indexing', *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2000)*. Vol. 2, pp.1171–1174.

Lee, S. and Hayes, M.H. (2002) 'Real-time camera motion classification for content-based indexing and retrieval using templates', *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*. Vol. 4, pp.3664–3667.

Naito, M., Matsumoto, K., Hoashi, K. and Sugaya, F. (2006) 'Camera motion detection using video mosaicing', *Proceedings of the IEEE International Conference on Multimedia and Expo*. pp.1741–1744.

Patel, N.V. and Sethi, I.K. (1997) 'Video shot detection and characterization for video databases', *Storage and Retrieval for Image and Video Databases (SPIE)*. pp.218–225.

Sand, P. and Teller, S. (2006) 'Particle video: long-range motion estimation using point trajectories', *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2, IEEE Computer Society pp.2195–2202.

Tiburzi, F. and Bescos, J. (2007) 'Camera motion analysis in on-line MPEG sequences', *Proceedings of the Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '07)*. pp.42–46.

Xiong, W. and Lee, J.C-M. (1998) 'Efficient scene change detection and camera motion annotation for video classification', *Computer Vision and Image Understanding*, Vol. 71, pp.166–181.