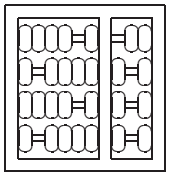


INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Robust Estimation of Camera Motion using Optical Flow Models

Jurandy Almeida *Rodrigo Minetto*
Tiago A. Almeida *Ricardo da S. Torres*
Neucimar J. Leite

Technical Report - IC-09-24 - Relatório Técnico

July - 2009 - Julho

Robust Estimation of Camera Motion using Optical Flow Models

Jurandy Almeida[1] Rodrigo Minetto[1] Tiago A. Almeida[2]
Ricardo da S. Torres[1] Neucimar J. Leite[1]

[1] Institute of Computing
University of Campinas
13083-970 – Campinas – SP – Brazil
{jurandy.almeida,rodrigo.minetto,rtorres,neucimar}@ic.unicamp.br

[2]School of Electrical and Computer Engineering
University of Campinas
13083-970 – Campinas – SP – Brazil
tiago@dt.fee.unicamp.br

July 6, 2009

Abstract

The estimation of camera motion is one of the most important aspects for video processing, analysis, indexing, and retrieval. Most of existing techniques to estimate camera motion are based on optical flow methods in the uncompressed domain. However, to decode and to analyze a video sequence is extremely time-consuming. Since video data are usually available in MPEG-compressed form, it is desirable to directly process video material without decoding. In this paper, we present a novel approach for estimating camera motion in MPEG video sequences. Our technique relies on linear combinations of optical flow models. The proposed method first creates prototypes of optical flow, and then performs a linear decomposition on the MPEG motion vectors, which is used to estimate the camera parameters. Experiments on synthesized and real-world video clips show that our technique is more effective than the state-of-the-art approaches for estimating camera motion in MPEG video sequences.

1 Introduction

Advances in data compression, data storage, and data transmission have facilitated the way videos are created, stored, and distributed. The increase in the amount of video data has enabled the creation of large digital video libraries. This has spurred great interest for systems that are able to efficiently manage video material [1–3].

Making efficient use of video information requires that the data be stored in an organized way. For this, it must be associated with appropriate features in order to allow any future

retrieval. An important feature in video sequences is the temporal intensity change between successive video frames: apparent motion. The apparent motion is generally attributed to the motion caused by object movement or introduced by camera operation. The estimation of camera motion is one of the most important aspects to characterize the content of video sequences [4].

Most of existing techniques to estimate camera motion are based on analysis of the optical flow between consecutive video frames [5–10]. However, the estimation of the optical flow, which is usually based on gradient or block matching methods, is computationally expensive [11].

Since video data are usually available in MPEG-compressed form, it is desirable to directly process the compressed video without decoding. A few methods that directly manipulate MPEG compressed video to extract camera motion have been proposed [4, 11, 12]. These approaches use MPEG motion vectors as an alternative to optical flow which allows to save high computational load from two perspectives: full decoding the video stream and optical flow computation [4].

In this paper, we present a novel approach for estimating camera motion in MPEG video sequences. It consists of three main steps. First, we extract the raw motion vectors from MPEG stream by partial decoding. Next, we create prototypes of optical flow, and then perform a linear decomposition on the MPEG motion vectors, which is used to estimate the camera parameters. Finally, we apply a robust estimation technique to reduce the influence of outliers.

In order to validate our approach, we use a synthetic test set and real-world video sequences including all kinds of camera motion and many of their possible combinations. Further, we have conducted several experiments to show that our technique is more effective than the state-of-the-art approaches for estimating camera motion in MPEG video sequences.

The remainder of the paper is organized as follows. In Section 2, we review three existing approaches used as reference in our experiments. Section 3 presents our approach for the estimation of camera motion. The experimental settings and results are discussed in Section 4. Finally, Section 5 presents conclusions and directions for future work.

2 Related Work

In this section we review three approaches used as reference in our experiments. These methods were implemented and their effectiveness are compared in Section 4.

Kim et al. [4] have used a two-dimensional affine model to detect six types of motion: panning, tilting, zooming, rolling, object motion, and stationary. Beforehand, motion vector outliers are filtered out by a simple smoothing filter. The camera parameters for the model are estimated by using a least squares fit to the remaining data.

Smolic et al. [12] have used a simplified two-dimensional affine model to distinguish between panning, tilting, zooming, and rolling. They use the M-estimator approach [13] to deal with data corrupted by outliers. It is basically a weighted least square technique, which reduces the effect of outliers by using an influence function.

Gillespie et al. [11] have extended such approach in order to improve its effectiveness by using a robust Least Median-of-Squares (LMedS) [13] to estimate the camera parameters and minimize the influence of outliers.

3 Our Approach

The previous approaches simply find the best-fit affine model to estimate camera motion by using the least squares method. However, the affine parameters are not directly related to the physically meaningful camera operations.

In this sense, we propose a novel approach for the estimation of camera motion based on optical flow models. The proposed method generates the camera model using linear combinations of prototypes of optical flow produced by each camera operation. It consists of three main steps: (1) feature extraction; (2) motion model fitting; and (3) robust estimation of the camera parameters.

3.1 Feature Extraction

MPEG videos are composed by three main types of pictures: intra-coded (I-frames), predicted (P-frames), and bidirectionally predicted (B-frames). These pictures are organized into sequences of groups of pictures (GOPs) in MPEG video streams.

A GOP must start with an I-frame and can be followed by any number of I and P-frames, which are usually known as anchor frames. Between each pair of consecutive anchor frames can appear several B-frames. Figure 1 shows a typical GOP structure.

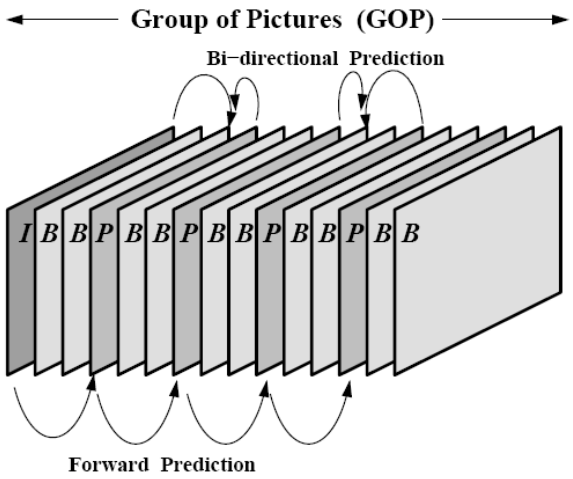


Figure 1: A typical group of pictures (GOP) in MPEG video sequences.

An I-frame does not refer to any other video frame. On the other hand, the encoding of a P-frame is based on a previous anchor frame, while the encoding of a B-frame can be based on two anchor frames, a previous as well as a subsequent anchor frame.

Each video frame is divided into a sequence of non-overlapping macroblocks. For each macroblock, a motion vector which points to a similar block in an anchor frame is estimated. Motion estimation algorithms try to find the best block match in terms of compression efficiency. This can lead to motion vectors that do not represent the camera motion at all [14].

The motion vectors are extracted directly from the compressed MPEG stream. Only the motion vectors from P-frames are processed in our approach. They were chosen in the following considerations. First, usually each third until fifth frame in a MPEG video is a P-frame, and thus, the temporal resolution is suficient for most applications. Further, both the prediction direction and the temporal distance of motion vectors are not unique in B-frames, resulting in additional computational complexity.

3.2 Motion Model Fitting

A camera projects a 3D world point into a 2D image point. The motion of the camera may be limited to a single motion such as rotation, translation, or zoom, or some combination of these three motions. Such camera motion can be well categorized by few parameters.

If we consider that the visual field of the camera is small, we can establish ideal optical flow models, which are noise-free, by using a numerical expression for the relationship of the MPEG motion vectors, and creating prototypes of optical flow models.

Hence, we can approximate the optical flow by a weighted combination of optical flow models:

$$f = P \cdot p + T \cdot t + Z \cdot z + R \cdot r, \tag{1}$$

where p, t, z, and r are the prototypes generated by panning, tilting, zooming, and rolling, respectively.

The parameter-estimation problem is now to obtain an estimate for the parameters P, T, Z, and R, based on a set of measured motion vectors $\{\hat{f}_i\}$. Since the measurements are not exact, we can not assume that they will all fit perfectly to the model. Hence, the best solution is to compute a least-squares fit to the data. For this, we define the model error as the sum of squared norm of the discrepancy vectors between the measured motion vectors $\hat{f}_i$ and the motion vectors obtained from the model:

$$E = \sum_i \|(P \cdot p_i + T \cdot t_i + Z \cdot z_i + R \cdot r_i) - \hat{f}_i\|^2, \tag{2}$$

where P, T, Z, and R represent the motion induced by the camera operations of panning (or tracking), tilting (or booming), zooming (or dollying), and rolling, respectively.

To minimize the model error E, we can take its derivatives with respect to the motion

parameters

$$\frac{\partial E}{\partial P} = \sum_i 2\ p_i^T\ (P \cdot p_i + T \cdot t_i + Z \cdot z_i + R \cdot r_i - \hat{f}_i),$$
$$\frac{\partial E}{\partial T} = \sum_i 2\ t_i^T\ (P \cdot p_i + T \cdot t_i + Z \cdot z_i + R \cdot r_i - \hat{f}_i),$$
$$\frac{\partial E}{\partial Z} = \sum_i 2\ z_i^T\ (P \cdot p_i + T \cdot t_i + Z \cdot z_i + R \cdot r_i - \hat{f}_i),$$
$$\frac{\partial E}{\partial R} = \sum_i 2\ r_i^T\ (P \cdot p_i + T \cdot t_i + Z \cdot z_i + R \cdot r_i - \hat{f}_i),$$

and set them to zero, giving

$$\sum_i (P\ p_i^T p_i + T\ p_i^T t_i + Z\ p_i^T z_i + R\ p_i^T r_i - p_i^T \hat{f}_i) = 0,$$
$$\sum_i (P\ t_i^T p_i + T\ t_i^T t_i + Z\ t_i^T z_i + R\ t_i^T r_i - t_i^T \hat{f}_i) = 0,$$
$$\sum_i (P\ z_i^T p_i + T\ z_i^T t_i + Z\ z_i^T z_i + R\ z_i^T r_i - z_i^T \hat{f}_i) = 0,$$
$$\sum_i (P\ r_i^T p_i + T\ r_i^T t_i + Z\ r_i^T z_i + R\ r_i^T r_i - r_i^T \hat{f}_i) = 0,$$

which can be written in matrix form as

$$\begin{bmatrix} \sum_i \langle p_i, p_i \rangle & \sum_i \langle p_i, t_i \rangle & \sum_i \langle p_i, z_i \rangle & \sum_i \langle p_i, r_i \rangle \\ \sum_i \langle t_i, p_i \rangle & \sum_i \langle t_i, t_i \rangle & \sum_i \langle t_i, z_i \rangle & \sum_i \langle t_i, r_i \rangle \\ \sum_i \langle z_i, p_i \rangle & \sum_i \langle z_i, t_i \rangle & \sum_i \langle z_i, z_i \rangle & \sum_i \langle z_i, r_i \rangle \\ \sum_i \langle r_i, p_i \rangle & \sum_i \langle r_i, t_i \rangle & \sum_i \langle r_i, z_i \rangle & \sum_i \langle r_i, r_i \rangle \end{bmatrix} \begin{pmatrix} P \\ T \\ Z \\ R \end{pmatrix} = \begin{pmatrix} \sum_i \langle p_i, \hat{f}_i \rangle \\ \sum_i \langle t_i, \hat{f}_i \rangle \\ \sum_i \langle z_i, \hat{f}_i \rangle \\ \sum_i \langle r_i, \hat{f}_i \rangle \end{pmatrix}, \tag{3}$$

where

$$\langle u, v \rangle = u^T v$$

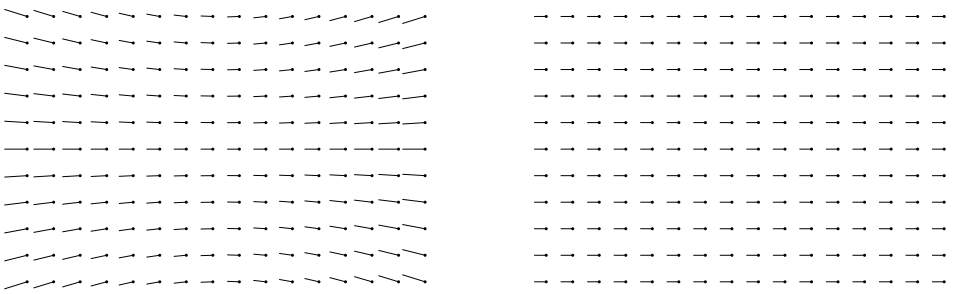
is the inner product between the vectors u and v.

Here, we define the optical flow model for panning (p), tilting (t), zooming (z), and rolling (r), respectively, as:

$$p(x,y) = \begin{pmatrix} -1 \\ 0 \end{pmatrix},\ t(x,y) = \begin{pmatrix} 0 \\ -1 \end{pmatrix},\ z(x,y) = \begin{pmatrix} -x \\ -y \end{pmatrix},\ r(x,y) = \begin{pmatrix} y \\ -x \end{pmatrix},$$

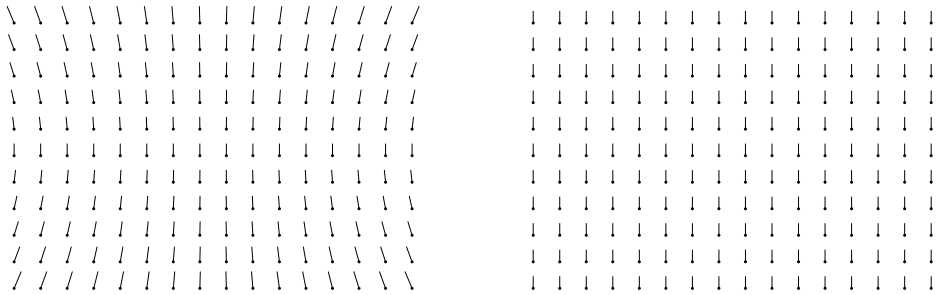
where (x, y) is the sample point whose coordinate system has origin at the center of the image.

Figures 2, 3, 4, and 5 represent the prototypes which consist of optical flow models generated by panning, tilting, zooming, and rolling, respectively. These optical flow models express the amount and direction of the camera motion parameters, respectively.



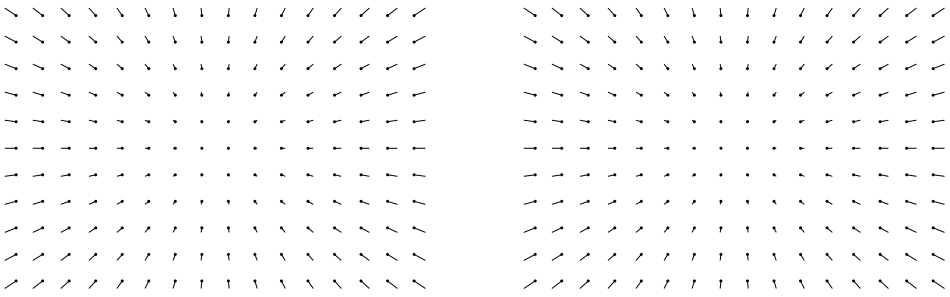
(a) Optical flow. (b) Prototype.

Figure 2: The optical flow and the prototype generated by panning.



(a) Optical flow. (b) Prototype.

Figure 3: The optical flow and the prototype generated by tilting.

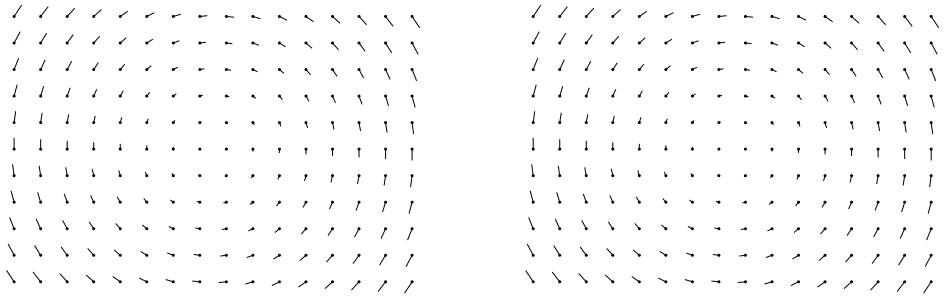


(a) Optical flow. (b) Prototype.

Figure 4: The optical flow and the prototype generated by zooming.

3.3 Robust Estimation of the Camera Parameters

The direct least-squares approach for parameter estimation works well for a small number of outliers that do not deviate too much from the correct motion. However, the result is significantly distorted when the number of outliers is larger, or the motion is very different



(a) Optical flow. (b) Prototype.

Figure 5: The optical flow and the prototype generated by rolling.

from the correct camera motion. Especially if the video sequence shows independent object motions, a least-squares fit to the complete data would try to include all visible object motions into a single motion model.

To reduce the influence of outliers, we apply a well-known robust estimation technique called RANSAC (RANdom SAmple Consensus) [15]. The idea is to repeatedly guess a set of model parameters using small subsets of data that are drawn randomly from the input. The hope is to draw a subset with samples that are part of the same motion model. After each subset draw, the motion parameters for this subset are determined and the amount of input data that is consistent with these parameters is counted. The set of model parameters with the largest support of input data is considered the most dominant motion model visible in the image.

4 Experiments and Results

In order to evaluate the performance of the proposed method for estimating camera motion in MPEG video sequences, experiments were carried out on both synthetic and real-world video clips.

4.1 Results with noise-free synthetic data

First, we evaluate our approach on synthetic video sequences with known ground-truth data. For this, we create a synthetic test set with four MPEG-4 video clips[1] (640 × 480 pixels of resolution) based on well textured POV-Ray scenes of a realistic office model (Figure 6), including all kinds of camera motion and many of their possible combinations. The main advantage is that the camera motion parameters can be fully controlled which allows us to verify the estimation quality in a reliable way.

The first step for creating the synthetic videos is to define the camera's position and orientation in relation to the scene. The world-to-camera mapping is a rigid transformation

[1] All video clips and ground-truth data of our synthetic test set are available at `http://www.liv.ic.unicamp.br/~minetto/videos/`.

(a) (b) (c)

Figure 6: The POV-Ray scenes of a realistic office model used in our synthetic test set.

which takes scene coordinates $p_{\mathrm{w}} = (x_{\mathrm{w}}, y_{\mathrm{w}}, z_{\mathrm{w}})$ of a point to its camera coordinates $p_{\mathrm{c}} = (x_{\mathrm{c}}, y_{\mathrm{c}}, z_{\mathrm{c}})$. This mapping is given by [16]

$$p_{\mathrm{c}} = Rp_{\mathrm{w}} + T, \tag{4}$$

where R is a 3×3 rotation matrix that defines the camera's orientation, and T defines the camera's position.

The rotation matrix R is formed by a composition of three special orthogonal matrices (known as *rotation matrices*)

$$R_x = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix},$$

$$R_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) \\ 0 & -\sin(\beta) & \cos(\beta) \end{bmatrix},$$

$$R_z = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where α, β, γ are the angles of the rotations.

We consider the motion of a continuously moving camera as a trajectory where the matrices R and T change according to the time t, in homogeneous representation,

$$\begin{bmatrix} p_{\mathrm{c}} \\ 1 \end{bmatrix} = \begin{bmatrix} R(t) & T(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{\mathrm{w}} \\ 1 \end{bmatrix}. \tag{5}$$

Thus, to perform camera motions such as tilting (gradual changes in R_x), panning (gradual changes in R_y), rolling (gradual changes in R_z), and zooming (gradual changes in focal distance f), we define a function $F(t)$ which returns the parameters (α, β, γ, and f) used to move the camera at the time t. We use a smooth and cyclical function

$$F(t) = \mathcal{M} * \frac{1 - \cos(2\pi t/\mathcal{T})(0.5 - t/\mathcal{T})}{0.263}, \tag{6}$$

where $\mathcal{M}$ is the maximum motion factor and $\mathcal{T}$ is the duration of camera motion in units of time. We create all video clips using the maximum motion factor $\mathcal{M}$ equals to 3^o for tilting (α), 8^o for panning (β), 90^o for rolling (γ), and 1.5 for zooming (f).

Figure 7 shows the main characteristics of each resulting video sequence (M_i). The terms P, T, R, and Z stand for the motion induced by the camera operations of panning, tilting, zooming, and rolling, respectively. The videos M_3 and M_4 have combinations of two or three types of camera motions. In order to represent a more realistic scenario, we modify the videos M_2 and M_4 to have occlusions due to object motion.

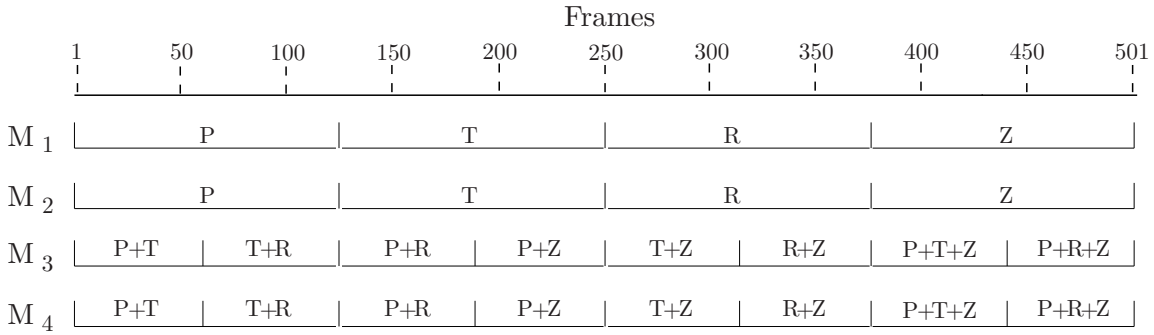


Figure 7: The main characteristics of each video sequence (M_i) in our synthetic test set.

We assess the effectiveness of the proposed method using the well-known Zero-mean Normalized Cross Correlation (ZNCC) metric [17], defined by

$$\mathrm{ZNCC}(\mathcal{F},\mathcal{G}) = \frac{\sum_t (\mathcal{F}(t)-\bar{\mathcal{F}})(\mathcal{G}(t)-\bar{\mathcal{G}})}{\sqrt{\sum_t (\mathcal{F}(t)-\bar{\mathcal{F}})^2 \sum_t (\mathcal{F}(t)-\bar{\mathcal{G}})^2}} \tag{7}$$

where $\mathcal{F}(t)$ and $\mathcal{G}(t)$ are the estimate and the real camera parameters, respectively, at the time t. It returns a real value between -1 and $+1$. A value equals to $+1$ indicates a perfect estimation; and -1, an inverse estimation.

Tables 1, 2, 3, and 4 compare our approach with the techniques presented in Section 2. Clearly, the use of optical flow models for estimating camera motion in MPEG video sequences is more effective than the affine model-based approaches.

Table 1: Effectiveness achieved by all approaches in video clip M_1.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.981419**	**0.996312**	**0.999905**	**0.964372**
Gillespie et al.	0.970621	0.987444	0.999830	0.958607
Smolic et al.	0.950911	0.994171	0.999199	0.949852
Kim et al.	0.649087	0.912365	0.994067	0.858090

Despite MPEG motion vectors improve the runtime performance, they often do not model real motion adequately [14]. Note that the effectiveness achieved by all methods is

Table 2: Effectiveness achieved by all approaches in video clip M_2.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.981029**	**0.995961**	**0.999913**	**0.965994**
Gillespie et al.	0.972189	0.988062	0.999853	0.959516
Smolic et al.	0.936479	0.991438	0.999038	0.949367
Kim et al.	0.633559	0.821266	0.986408	0.865052

Table 3: Effectiveness achieved by all approaches in video clip M_3.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.587136**	**0.950760**	**0.999624**	**0.956845**
Gillespie et al.	0.575178	0.931957	0.999521	0.954215
Smolic et al.	0.559669	0.940782	0.999037	0.951701
Kim et al.	0.501764	0.942563	0.997240	0.942588

Table 4: Effectiveness achieved by all approaches in video clip M_4.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.592071**	**0.949922**	**0.999659**	**0.956440**
Gillespie et al.	0.577467	0.932568	0.999545	0.954286
Smolic et al.	0.557849	0.940886	0.998920	0.951640
Kim et al.	0.498081	0.941956	0.997334	0.944102

reasonably reduced for tilting operations in presence of several types of camera motions at the same time.

4.2 Results with real-world video sequences

We also evaluate our technique over four real-world video sequences[2]. These video clips were shot with a hand-held consumer-grade DVR (Canon Optura 40) with variable zoom. They were recorded in MPEG format at 320×240 resolution, 14.98 frames per second.

Table 5 summarizes the main characteristics of each resulting real-world video sequence (R_i). All videos clips were affected by natural noise. The videos R_3 and R_4 have occlusions due to object motion.

In these experiments, we analyze the effectiveness of motion vector-based techniques in relation to the well-known optical flow-based estimator presented in [9]. Each video clip (R_i) takes less than 1 second to process the whole sequence using our approach on a Intel Core 2 Quad Q6600 (four cores running at 2.4 GHz), 2GB memory DDR3. It is important

[2]All real-world video sequences are available at `http://www.liv.ic.unicamp.br/~minetto/videos/`.

Table 5: The main characteristics of each real-world video sequence (R_i).

Video	Frames	Camera Operations
R_1	338	P,T,R,Z
R_2	270	P,T,R,Z
R_3	301	P,T,R,Z
R_4	244	P,T,R,Z

to realize that the optical flow-based method requires a magnitude of almost one second per frame.

Tables 6, 7, 8, and 9 compare our approach with the techniques presented in Section 2. In fact, the use of optical flow models for estimating camera motion in MPEG video sequences outperforms the affine model-based approaches.

Table 6: Effectiveness achieved by all approaches in video clip R_1.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.986287**	**0.986294**	**0.987545**	**0.982227**
Gillespie et al.	0.982345	0.978892	0.980464	0.964398
Smolic et al.	0.984085	0.976381	0.977135	0.966526
Kim et al.	0.982998	0.884470	0.795713	0.944286

Table 7: Effectiveness achieved by all approaches in video clip R_2.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.914379**	**0.954113**	**0.929268**	**0.684219**
Gillespie et al.	0.863166	0.931218	0.899512	0.357249
Smolic et al.	0.874244	0.952316	0.919447	0.611227
Kim et al.	0.899520	0.901673	0.846316	0.670006

Table 8: Effectiveness achieved by all approaches in video clip R_3.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.964425**	**0.960878**	**0.957735**	**0.454204**
Gillespie et al.	0.949270	0.931442	0.927145	0.379836
Smolic et al.	0.957662	0.953751	0.956303	0.444741
Kim et al.	0.954097	0.912121	0.924798	0.368722

Table 9: Effectiveness achieved by all approaches in video clip R_4.

Method	*Tilting*	*Panning*	*Rolling*	*Zooming*
Our Approach	**0.976519**	**0.958020**	**0.927920**	**0.577974**
Gillespie et al.	0.948314	0.902511	0.851247	0.308588
Smolic et al.	0.969314	0.956417	0.903442	0.523507
Kim et al.	0.969613	0.938639	0.839906	0.474439

Note that the optical flow models identify the camera operations better than the affine parameters. For instance, considering zooming operations in the video R_4, our method is more than 10% ($\approx$ 5 percentual points) better than the best affine model-based one.

5 Conclusions

In this paper, we have presented a novel approach for the estimation of camera motion in MPEG video sequences. Our technique relies on linear combinations of optical flow models. Such models identify the camera operations better than the affine parameters.

We have validated our technique using synthesized and real-world video clips including all kinds of camera motion and many of their possible combinations. Our experiments have showed that the use of optical flow models for estimating camera motion in MPEG video sequences is more effective than the affine model-based approaches.

Future work includes an extension of the proposed method to distinguish between translational (tracking, booming, and dollying) and rotational (panning, tilting, and rolling) camera operations. In addition, we want to investigate the effects of integrating the proposed method into a complete MPEG system for camera motion-based search-and-retrieval of video sequences.

Acknowledgment

The authors thank the financial support of Microsoft EScience Project, CAPES/COFECUB Project (Grant 592/08), and Brazilian agencies FAPESP (Grants 07/54201-6 and 08/50837-6), CNPq (Grant 142466/2006-9), and CAPES (Grant 01P-05866/2007).

References

[1] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatio-temporal queries," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 8, no. 5, pp. 602–615, 1998.

[2] A. Hampapur, A. Gupta, B. Horowitz, C.-F. Shu, C. Fuller, J. R. Bach, M. Gorkani, and R. Jain, "Virage video engine," in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1997, pp. 188–198.

[3] D. B. Ponceleon, S. Srinivasan, A. Amir, D. Petkovic, and D. Diklic, "Key to effective video retrieval: Effective cataloging and browsing," in *ACM Multimedia*, 1998, pp. 99–107.

[4] J.-G. Kim, H. S. Chang, J. Kim, and H.-M. Kim, "Efficient camera motion characterization for mpeg video indexing," in *ICME*, 2000, pp. 1171–1174.

[5] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 497–501, 2000.

[6] S.-C. Park, H.-S. Lee, and S.-W. Lee, "Qualitative estimation of camera motion parameters from the linear composition of optical flow," *Pattern Recognition*, vol. 37, no. 4, pp. 767–779, 2004.

[7] B. Qi, M. Ghazal, and A. Amer, "Robust global motion estimation oriented to video object segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 958–967, 2008.

[8] P. Sand and S. J. Teller, "Particle video: Long-range motion estimation using point trajectories," *IJCV*, vol. 80, no. 1, pp. 72–91, 2008.

[9] M. V. Srinivasan, S. Venkatesh, and R. Hosie, "Qualitative estimation of camera motion parameters from video sequences," *Pattern Recognition*, vol. 30, no. 4, pp. 593–606, 1997.

[10] T. Zhang and C. Tomasi, "Fast, robust, and consistent camera motion estimation," in *CVPR*, 1999, pp. 1164–1170.

[11] W. J. Gillespie and D. T. Nguyen, "Robust estimation of camera motion in MPEG domain," in *TENCON*, 2004, pp. 395–398.

[12] A. Smolic, M. Hoeynck, and J.-R. Ohm, "Low-complexity global motion estimation from p-frame motion vectors for mpeg-7 applications," in *ICIP*, 2000, pp. 271–274.

[13] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. John Wiley and Sons, Inc., 1987.

[14] R. Ewerth, M. Schwalb, P. Tessmann, and B. Freisleben, "Estimation of arbitrary camera motion in MPEG videos," in *ICPR*, 2004, pp. 512–515.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, Inc., 2003.

[17] J. Martin and J. L. Crowley, "Experimental comparison of correlation techniques," in *Int. Conf. on Intelligent Autonomous Systems*, 1995.

Erratum added in November 25, 2009.

Page	Line	For	Read
5	7	rolling (r), respectively, as:	rolling (r), respectively, as [10]:
13	18	[10] T. Zhang and C. Tomasi, "Fast, robust, and consistent camera motion estimation," in	[10] R. Minetto, N. J. Leite, and J. Stolfi, "Reliable detection of camera motion based on
13	19	CVPR, 1999, pp. 1164–1170.	weighted optical flow fitting," in VISAPP, 2007, pp. 435–440.