A Robust and Hierarchical Approach for Camera Motion Classification

Yuliang Geng, De Xu, Songhe Feng, and Jiazheng Yuan

Institute of Computer Science and Technology,
Beijing Jiaotong University, Beijing, 100044, China
gengyuliang@hotmail.com

Abstract. Camera motion classification is an important issue in content-based video retrieval. In this paper, a robust and hierarchical camera motion classification approach is proposed. As the Support Vector Machine (SVM) has a very good learning capacity with limited sample set and does not require any heuristic parameter, the SVM is first employed to classify camera motions into translation and non-translation motions in preliminary classification. In this step, four features are extracted as input of the SVM. Then, zoom and rotation motions are further classified by analyzing the motion vectors' distribution. And the directions of translation motions are also identified. The experimental results show that the proposed approach achieves a good performance.

1 Introduction

Camera motion classification is an important issue in content-based video retrieval. Taking no account of scene depth variation, there are four basic camera motion categories, namely, still, zoom (includes zoom in, zoom out), rotation and translation (includes panning right, tilting down, panning left and tilting up). Extracting camera motion will help understand higher-level semantic content, especially in some specific domains, such as sports video, movie video and surveillance video. Usually, zoom-in motion will give the details about the characters or objects do, or imply an important event may happen. Zoom-out motion gives a distant framing, which shows the spatial relations among the important figures, objects, and setting in a scene. Translation motion often indicates the dominant motion direction of a scene or gives an overview of mise en scene. So camera motion classification is essential to video structure analysis and higher semantic information extraction.

There are a number of methods proposed to detect camera motion in recent literatures [1,2,3,4]. Most of prior work is focused on parameter model estimation, such as affine model, perspective model, etc [1,2,3]. But high computational complexity and noise sensibility are still the main problems of parameter model estimation approach, especially in massive video analysis and retrieval. In [2] Huang *et al.* utilize feature points selection to improve performance of parameter estimation. In [3], Kumar *et al.* utilize parameter-decomposition estimation to reduce computational complexity. In fact, it is not necessary for video parsing

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 340–348, 2006.

and understanding to extract accurate motion parameters. Qualitative camera motion classification helps improve computational performance and reduce noise influence. Zhu *et al.* [4] propose a qualitative method, which employs motion vectors mutual relationship to implement camera motion classification, and obtains a satisfying results.

In this paper, we propose an effective and efficient camera motion classification approach. First, cinematic rules are utilized to filter abnormal noise and foreground motion noise in preprocessing step. Then, the SVM is employed to classify camera motions into translation and non-translation motions in preliminary classification of camera motion. Finally, we refine the camera motion categories. In this step, the zoom and rotation motions are further distinguished, and the translation direction is also identified by analyzing the motion vectors' distribution. Experimental results validate the effectiveness of our proposed approach. The block diagram of our approach is shown in Fig. 1.

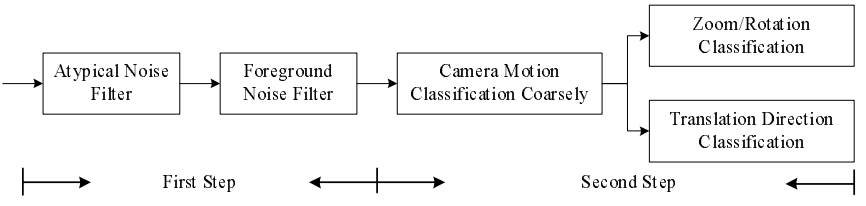


Fig. 1. Block diagram of the proposed approach

The organization of this paper is as follows. In Section 2, we represent the preprocessing step of camera motion classification, which is used to reduce abnormal noise and foreground motion noise. A robust and hierarchical approach for camera motion classification is proposed in Section 3. Section 4 and 5 give the experimental results and draw the conclusions.

2 The Preprocessing of Motion Vector Field (MVF)

Before estimating camera motion categories, we need filter motion noises in MVF because they might result in an error motion estimation. There are two different motion noises. One is abnormal motion noise that is generated from motion estimation by block-matching algorithm or optical flow equation. The other is foreground noise that is generated from the foreground object motion.

First, we filter the abnormal motion noise. In data analysis, Interquartile Range is an effective way to detect noise data in a given data set [5]. That is, any data that is less than $LQ - 1.5IQR$ or greater than $UQ + 1.5IQR$ is regarded as noise data, where LQ is the lower quartile, UQ is the upper quartile, and IQR is the interquartile range which is defined as $UQ - LQ$.

For a given MVF, we suppose that the magnitudes of motion vectors satisfy Gaussian distribution. We remove the abnormal motion vectors by computing the Interquartile Range, and denote the valid motion vector set as $\mathbf{V}_1$.

Then we further reduce the foreground noise. In mise en scene, filmmaker often places the foreground object on the center region of screen, also called attention region, to attract viewers' attention [6]. The attention region is determined by Golden Section spatial composition rule, which suggests dividing the screen into 3×3 regions in $3:5:3$ proportion in both directions. We denote the center region, that is attention region, as $\mathbf{C}$, and the surrounding region as $\mathbf{B}$. As the foreground objects and background have conspicuously different motion vectors, we compute the motion saliency map based on the valid motion vectors $\mathbf{V}_1$ as

$$S(i,j) = |E(i,j) - (\omega_1 \bar{E}_{\mathrm{B}} + \omega_2 \bar{E}_{\mathrm{C}})| \tag{1}$$

where $E(i,j)$ is the motion energy of block (i,j). ω_1, ω_2 are the preassigned weight values, and $\omega_1 \geq \omega_2$, $\omega_1 + \omega_2 = 1$. As the discussed above, the surrounding region plays more important role in camera motion classification, so we assign a greater value to ω_1 than ω_2. $\bar{E}_{\mathrm{B}}$, $\bar{E}_{\mathrm{C}}$ are the average motion energies of region B and C respectively.

Thus, we get the foreground motion region approximately by binarizing the motion saliency map. The binarization threshold is estimated in an adaptive method. We filter the foreground motion and achieve valid motion vector set, which is denoted as $\mathbf{V}_2$. The camera motion classification is based on $\mathbf{V}_2$.

3 Hierarchical Camera Motion Classification

The hierarchical approach for camera motion classification is composed of two steps as Section 3.1 and 3.2 depicted. Before camera motion classification, the still camera motion is detected. We regard the camera motion as still category if the average motion energy of the valid motion vectors is less than a given threshold TH_{still}. TH_{still} is an empirical value, and is set as 2.

3.1 Camera Motion Preliminary Classification Based on SVM

As the translation motions have similar motion vector fields, which are different from the ones for zoom and rotation motions. Namely, the MVF for translation is composed of parallel motion vectors with uniform magnitudes; and the MVF for zoom is composed of radial vectors whose magnitudes are proportional to their distance from the center of focus (COF). The motion vectors for zoom-in/zoom-out point inward to/outward from the COF. The vertical MVF for rotation has the same characteristic as the MVF for zoom.

As the discussed above, we first classify camera motions into two categories: translation and non-translation motion (includes rotation and zoom). Here, we extract four features to characterize the camera motions as follows.

1) Motion Direction Feature. The motion vector direction is classified into 12 categories: $(-15^\circ + 30^\circ i,\ 15^\circ + 30^\circ i)$, $i = 0, 1, \ldots, 11$. Let $H^{\mathrm{A}}(i)$ represent the percentage of motion vectors at the ith direction. Then the motion direction consistency is computed as

$$F^{\mathrm{AngEn}} = -\sum_{i=0}^{11} (H^{\mathrm{A}}(i) \log H^{\mathrm{A}}(i)) \tag{2}$$

2) Motion Direction Relationship. To characterize motion direction relationship, we first compute included angles among the valid motion vectors. Then the included angles are classified into 8 categories: $(22.5°i, 22.5°(i+1))$, $i = 0, 1, \ldots, 7$. Let $H^{\mathrm{I}}(i)$ represent the percentage of included angles at the ith direction. The motion direction relationship is characterized by the mean and entropy of angular histogram $H^{\mathrm{I}}(i)$.

$$F^{\mathrm{InAngMean}} = \sum_{i=0}^{7} (15 \times i \times H^{\mathrm{I}}(i))/(N_2(N_2-1)/2) \tag{3}$$

$$F^{\mathrm{InAngEn}} = -\sum_{i=0}^{7} (H^{\mathrm{I}}(i) \log H^{\mathrm{I}}(i)) \tag{4}$$

where N_2 is the size of the valid motion vector set $\mathbf{V}_2$.

3) Motion Energy Feature. We compute the motion energy histogram of valid motion vectors with 10 equally spaced bins, and denote it as $H^{\mathrm{M}}(i)$. Then the motion energy distribution is characterized as

$$F^{\mathrm{MagEn}} = -\sum_{i=0}^{9} (H^{\mathrm{M}}(i) \log H^{\mathrm{M}}(i)) \tag{5}$$

The four feature values can be taken as a feature vector, and each component is normalized by the Gauss normalization formula. Thus, we denote the feature vector as $\mathbf{F} = [\bar{F}^{\mathrm{AngEn}}, \bar{F}^{\mathrm{InAngMean}}, \bar{F}^{\mathrm{InAngEn}}, \bar{F}^{\mathrm{MagEn}}]$.

As the SVM has a very good learning capacity with limited sample set, and does not require any heuristic parameter [7], we select the SVM as the classifier in camera motion preliminary classification. We set $\mathbf{F}$ as the input vector of the SVM. There are three common kernel functions: Radial Basis Function $K(x, y) = \exp(-||\mathbf{x} - \mathbf{y}||^2/2\sigma^2)$, Polynomial Function $K(x,y) = (\mathbf{x} \cdot \mathbf{y} + b)^d$, and Sigmoid Kernel function $K(x,y) = \tanh[b(\mathbf{x} \cdot \mathbf{y}) - \theta)]$. So far, kernel function selection still relays on the experimental method. In Section 4, we'll discuss how to select kernel functions and determine its parameters in detail.

After the preliminary classification, we classify the camera motion into translation and non-translation motions.

3.2 Refine the Camera Motion Categories

Translation Motion Classification. For the translation motion, we identify its motion direction by computing the dominant motion direction histogram, $H^{\mathrm{ori}}(k)$, which represents the percentage of motion vectors at the kth direction, $(-45° + 90°k,\ 45° + 90°k)$.

$$H^{\mathrm{ori}}(k) = \sum_{j=-1}^{1} H^{\mathrm{A}}((3k + j) \bmod 12) \quad k = 0, 1, 2, 3 \tag{6}$$

We classify the translation motion into a specific direction whose bin value has the maximum.

Non-translation Motion Classification. For the non-translation motion, we further classify them into zoom and rotation motion. As discussed in Section 3.1, the motion vectors for zoom-in/zoom-out point inward to/outward from the COF, while the vertical motion vectors for rotation have same characteristic as zoom. So we can identify the zoom and rotation motion categories by detecting COF as follows.

Step 1. As the discussed in Section 2, the surrounding region **B** is composed of 8 subregions. We select one motion vector as key motion vector in each subregion respectively. The key motion vector is the one whose motion direction consists with the dominant motion direction of that subregion. If the number of the motion vectors whose magnitudes are equal to zero is greater than two thirds of the number of the total motion vectors in one subregion, we should not select key motion vector in this subregion. Thus, we get the key motion vector set $\{\mathbf{V}(x_i, y_i)\}$, where $\mathbf{V}(x_i, y_i)$ represents the motion vector of macro block (x_i, y_i).

Step 2. As discussed in Section 3.1, the straight line L_i through point (x_i, y_i) in direction of $\mathbf{V}(x_i, y_i)$ should pass through the COF in the MVF for zoom, so we compute the intersection points formed by pairwise intersection of straight lines (if they intersect) that are determined by the key motion vector set $\{\mathbf{V}(x_i, y_i)\}$.

Step 3. We calculate the centroid of the intersection points. A simply way is to compute the mean for the intersection points' position. We regard the centroid as the COF, and denote it as (x_0, y_0).

Step 4. We calculate the average distance, $dist((x_0, y_0), L_i)$, from (x_0, y_0) to straight line L_i that is determined by the key motion vector $\mathbf{V}(x_i, y_i)$.

$$\bar{Dist} = \frac{1}{N} \sum_{i=1}^{N} dist((x_0, y_0), L_i) \tag{7}$$

where N is the size of key motion vector set $\{\mathbf{V}(x_i, y_i)\}$. If the average distance $\bar{Dist}$ is less than TH_{zoom}, the camera motion is identified as zoom motion. TH_{zoom} is a given threshold and is set as one third of the MVF height.

Step 5. For each key motion vector $\mathbf{V}(x_i, y_i)$, we compute the inner-product between $\mathbf{V}(x_i, y_i)$ and $(x_i - x_0, y_i - y_0)$.

$$O_{\text{zoom}} = \sum_{i} \text{sgn}(dot(\mathbf{V}(x_i, y_i), (x_i - x_0, y_i - y_0))) \tag{8}$$

where sgn() is a sign function, which returns 1 if the element is greater than zero, 0 if it equals zero and -1 if it is less than zero. $dot(,)$ is a inner-product function. If $O_{\text{zoom}} > 0$, the camera motion is zoom in, otherwise is zoom out.

As the vertical MVF for rotation has the same characteristic with the MVF for zoom, we can identify the rotation motion as the same way.

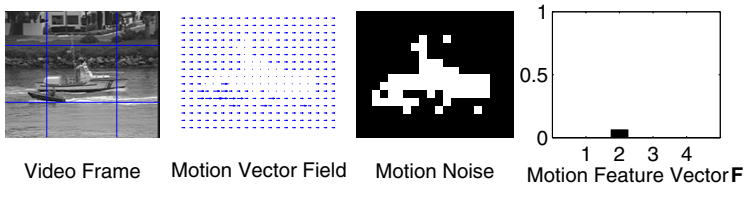
4 Experimental Results

To evaluate the proposed approach for camera motion classification, we collect various video data from MPEG-4 test set and www.open-video.com. The video

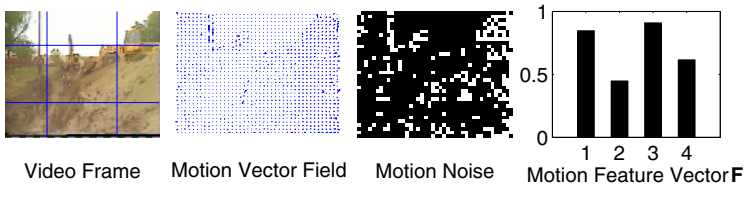
data set consists of *Apo13001*, *Bor10_009*, *Winn001002*, *Rotation*, and *Coastguard*. We analyze the camera motion category every ten frames because there is similar motion between consecutive frames. There are 2214 frames in total.

Fig. 2 gives several examples of motion noise reduction and feature vector extraction. The figures in the first column are the original video frames, and give the attention regions divided by Golden Section spatial composition rule. The figures in the second column are the corresponding MVFs. The figures in the third column give the experimental results of noise reduction, where the white regions indicate the detected motion noises. As the figures shown, the preprocessing step can filter most of abnormal and foreground motion noises effectively. The figures in the fourth column depict the motion feature vectors for various motion categories. The components of the motion feature vector for translation motion are often less than 0.5 and tend to 0, while the components of the motion feature vector for zoom or rotation motion, except for the second component that always changes between 0.4 and 0.5, are often greater than 0.5 and tend to 1.

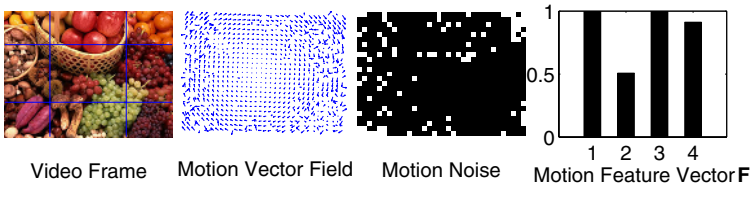
So far, the experimental method is still a main way to select kernel function and its parameters. The optimal kernel function just corresponds to the specific application. In this section, we utilize k-fold Cross Validation method ($k = 5$)



(a) Translation motion extracted from the 84th frame of *Coastguard*



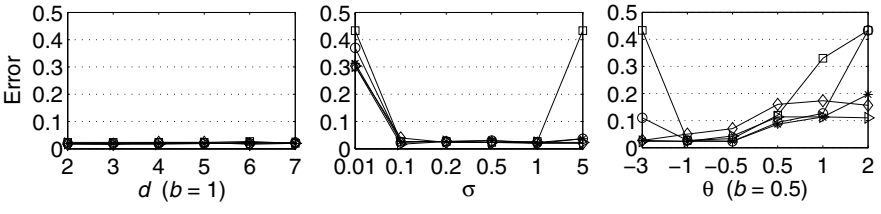
(b) Zoom motion extracted from the 689th frame of *Bor10_009*



(c) Rotation motion extracted from the 12th frame of *Rotation*

Fig. 2. Examples of motion noise reduction and feature vector extraction for various camera motions

to select kernel function, determine its parameters and restriction condition C. Fig. 3 gives the experimental results of parameter selection for various kernel function. In Fig. 3, each value of C corresponds with a curve, which represents the Cross-Validation error along the parameter of kernel function. Different plot symbols, namely, square, circle, star, triangle and diamond, respond with different values of C, 0.1, 0.5, 1, 10 and 100, respectively. We observe the classification performance does not improve obviously with changes in C, while training time increases obviously when the parameter value increases. For polynomial kernel function, parameters b and d have little effect on classification performance. For radial basis function, we achieve better performance when σ changes between 0.1 and 1. For sigmoid kernel function, we achieve the best experimental result when θ is set as -1 and b is set as 0.5. Taking account of the stability of the classifier and classification performance, we select polynomial kernel function ($b = 1$, $d = 2$ and $C = 1$). The experimental results verify the performance of the classifier based on SVM.



(a) Polynomial Kernel (b) Radial Basis Kernel (c) Sigmoid Kernel

Fig. 3. Kernel function and its parameters selection

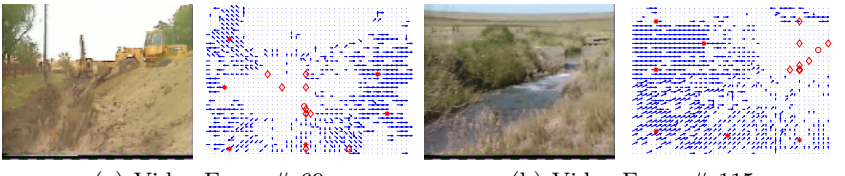
Table 1 gives the experimental results. Here we only consider the number of the correct classification (CC) against the ground truth (GT) for various camera motion categories occurring in each video sequence. F. # is the abbreviation of the number of video frames. P. is the abbreviation of classification precision. The experimental results show that the proposed approach can deal with motion noise robustly and achieve satisfying performance. Although video sequence *Apo13001* and *Winn001002* have poor quality, the proposed approach still gets satisfying results. For video sequences *Bor10_009*, *Rotation* and *Coastguard*, our approach achieves higher precision because these video sequences have stable camera motion and high quality.

In experiment, we find that most of the false detections in the camera motion classification are due to that the video frames have very slight camera motion, and are falsely identified as still motion category. Smooth texture region detection is another problem because the smooth texture region often generates mass abnormal motion noises in the motion vector estimation. For example, video sequence *Winn001002* has a low precision just because some scenes are shoot in the sky. These are our further work.

Besides implementing the camera motion classification, the proposed approach can detect the COF for zoom or rotation motions accurately. Several examples

Table 1. Experimental results for camera motion classification (In the first row, camera motion categories: still, zoom in, zoom out, rotation, panning right, tilting down, panning left and tilting up are denoted as 1, 2, ..., 8.)

Video	F. #	Correct Classification #								P.(%)
			1	2	3	4	5	6	7	8	
Apo13001	962	GT	665	172	60	15	21	16	0	13	79.1
		CC	517	143	50	9	18	13	0	11	
Bor10_009	357	GT	83	48	97	0	129	0	0	0	88.2
		CC	77	39	81	0	118	0	0	0	
Winn001002	511	GT	309	72	50	0	56	18	6	0	80
		CC	259	57	36	0	38	14	5	0	
Rotation	236	GT	31	0	0	74	30	41	27	33	85.6
		CC	25	0	0	61	27	37	23	29	
Coastguard	148	GT	5	0	0	0	106	0	32	5	96
		CC	5	0	0	0	103	0	31	3	



(a) Video Frame# 69 (b) Video Frame# 115

Fig. 4. Experimental results for COF detection

of original frames extracted from *Bor10_009* and their corresponding detection results are illustrated in Fig. 4. In figure, the star indicates the key motion vector in each subregion, the diamond indicates the intersection point determined by the key motion vectors, and the circle indicates the COF estimated by the key motion vectors. As the Fig. 4(a) shown, the object motion (the grab) and smooth texture region (the sky) are eliminated effectively by motion noise reduction, and the COF is correctly identified. When the COF is not at the center of the screen, as Fig. 4(b) shown, the proposed approach can also identify the motion category and detect the COF correctly.

5 Conclusions

We proposed a robust and hierarchical camera motion classification approach in the paper. First, the camera motions were classified into translation and non-translation motions based on the SVM. Then, the rotation and zoom motions were further distinguished, and the translation directions were also identified by analyzing the motion vectors' distribution. The experimental results shown that the proposed approach achieved a good performance. As camera motion can provide an important clue in content-based video parsing and understanding, our

future work is to further improve the performance of camera motion classification, and to apply the camera motion classification into video semantic analysis.

Acknowledgements

This research was supported by Science Foundation of Beijing Jiaotong University (Grant No. 2004SM013).

References

1. Su, Y.P., Sun, M.T., Hsu, V.: Global Motion Estimation From Coarsely Sampled Motion Vector Field and the Applications. IEEE Transaction on Circuits and System Video Technology, Vol. 15, No. 2, (2005) 232 - 242
2. Huang, J.C., Hsieh, W.S.: Automatic Feature-Based Global Motion Estimation in Video Sequences. IEEE Transaction Consumer Electronics, Vol. 50, No. 3, (2004) 911 - 915
3. Kumar, S., Biswas, M., et al.: Global Motion Estimation in Frequency and Spatial Domain. In: Proceedings of IEEE ICASSP, (2004) 17 - 21
4. Zhu, X.Q., Xue, X.Y., et al.: Qualitative Camera Motion Classification for Content-Based Video Indexing. In: Proceedings of IEEE PCM, LNCS, Vol. 2532, (2002) 1128 - 1136
5. Fan, J.C., Mei, C.L.: Data Analysis (Chinese). Science Press, Beijing, China (2002)
6. Millerson, G.: The Technique of Television Production. 12th ed. Focal Publishers (1990)
7. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, UK (2000)