CAMERA MOTION DETECTION USING VIDEO MOSAICING

Masaki Naito, Kazunori Matsumoto, Keiichiro Hoashi and Fumiaki Sugaya

KDDI R&D Laboratories, Inc, 2-1-15 Ohara Fujimino City Saitama 356-8502, Japan

ABSTRACT

In this paper, camera motion detection methods using a background image generated by video mosaicing based on the correlation between feature points on a frame pair are described. In this method, a telop (video caption) removal method, iterative foreground and background image separation method and appropriate frame pair selection from consecutive frames are introduced to generate background images accurately. Parameters indicating the location of each frame on the background image are retrieved and used to detect the camera motion. Except for the simple threshold-based method, a method using Hidden Markov models (HMMs) is introduced to detect variable length camera motion based on the maximum likelihood criterion. The effectiveness of the proposed method is evaluated by using a TRECVID 2005 low-level feature extraction task [1].

1. INTRODUCTION

As digital video becomes popular, effective methods of analyzing video content are becoming increasingly important. Requests for video material from archives often specify the desired or required camera motion. Several methods have been proposed to analyze the camera motion of video, based on analyzing the optical flow between consecutive images [2, 3, 4]. Kanazawa et al proposed a robust method for image mosaicing; progressively estimating the rotation, scale change and projective distortion between feature points on two images by random voting and variable template matching [5].

In this paper, we apply an image mosaicing method based on the correlation between feature points, to detect camera motion from TV programs such as news. TV images have an adverse effect on estimating the correlation between feature points, since they include outliers, such as telops and moving objects. Therefore, we introduce a scheme to detect the telop region and iterative foreground and background image separation to remove these regions from extracting feature points to estimate the correlation between two frames. Furthermore, since the video image has many frames, we select the appropriate frame pair from those possible and generate an accurate background image.

To detect the camera motion, the position of frames on the background image is converted to feature parameters; (1) the coordinate of the center of each quadrangle camera frame and (2) the distance between the center and vertex of the quadrangle frame respectively. Subsequently, camera motion is detected using two types of detection. The first is a simple threshold-based method, where detection of camera motions assesses whether the camera motion parameters exceed or go below a given threshold level. The other is a method using Hidden Markov models (HMMs), enabling the detection of variable length camera motion based on the maximum likelihood criterion. The effectiveness of the proposed method is evaluated using a TRECVID 2005 low-level feature extraction task [1].

2. FEATURE EXTRACTION

2.1 Background image generation using video mosaicing

Firstly, a background image is generated from a consecutive video image; based on the image mosaicing method and progressively estimating the rotation, scale change and the projective distortion between feature points on two images using stratified matching [5].

The outline of background image generation is as follows:

(1) Telop regions in video image are detected by using a moving text detection method [6]. They are then removed from areas where the feature points are extracted in the following steps:
(2) For every frame pair on consecutive video frames, feature points are extracted from both frames using a harris operator [7] and the correlations between the two frames are estimated by using [5].
(3) A temporal background image is generated by using correlations between the frame pairs obtained in step 2
(4) The foreground image is detected based on the method [8] and they are removed from the area to extract feature points on the next iteration.
(5) Steps 2 to 4 are repeated a predetermined number of times.[1]
(6) The position of each video frame is calculated, on the generated background image.

Figure 1 shows an example of the frame pair and extracted feature point. Figure 2 shows separated foreground and background image resulting in step 4. Figure 3 shows the positions of frames on the background image.

[1] These steps are repeated two times, in the following experiments.

Figure 1. Example of frame pair and extracted feature points

Figure 2. Separated foreground and background images; original image (top left), background image (bottom left), foreground image (bottom right) and generated background image (top right).

Figure 3. Position of frames on generated background image.

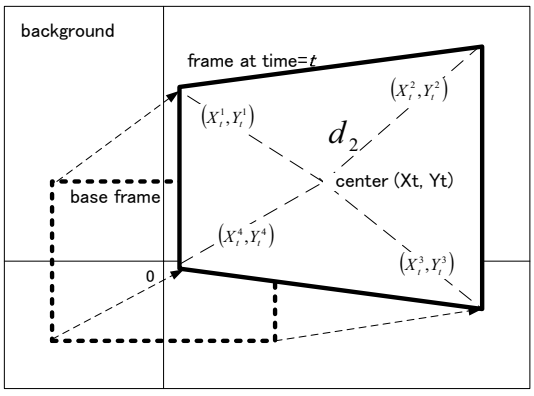

Figure 4. Relation between camera frame position parameters and feature parameters for detection.

2.2 Feature parameters

Firstly, the camera frame position parameters are converted to feature parameters for camera motion detection as follows. In this method, (1) the coordinate of the center of each quadrangle camera frame and (2) the distance between the center and vertex of the quadrangle camera frame are used as feature parameters for detection. Figure 4 illustrates the relation between the frame position and feature parameters for camera motion detection. The center of the quadrangle at the base frame, which is the start frame of each shot, is used as the origin. Then the coordinates of the center of each quadrangle frame (X_t, Y_t) and the distance between the center and vertex of the quadrangle frame z_t were calculated using the following formula:

$$X_t = \sum_{n=1}^{4} X_t^n \Big/ 4 \qquad (1)$$

$$Y_t = \sum_{n=1}^{4} Y_t^n \Big/ 4 \qquad (2)$$

$$d_t = \sum_{n=1}^{4} \left(\left(X_t^n - X_t \right)^2 + \left(Y_t^n - Y_t \right)^2 \right) \qquad (3)$$

, where (X_t^n, Y_t^n) are the coordinates of the vertex of the quadrangle frame at time t, and d_t is the distance between the center and vertex of the quadrangular frame at frame t.

In addition to the static feature parameters, e.g. X_t, we introduce a time derivative parameter, which is called a delta coefficient, to take the speed of motion into account. The delta coefficients are computed using the following regression formula [9]:

$$\Delta C_t = \sum_{\theta=1}^{\Theta} \theta \times \left(C_{t+\theta} - C_{t-\theta} \right) \Big/ 2 \sum_{\theta=1}^{\Theta} \theta^2 \qquad (4)$$

, where ΔC_t is a delta coefficient at frame t computed in terms of the corresponding static feature parameters $c_{t-\Theta}$ to $c_{t+\Theta}$. In the following experiments, Θ was set to 6.[2]

3. CAMERA MOTION DETECTION METHODS

In this paper, a TRECVID low-level feature extraction task is used to evaluate the effectiveness of the proposed method. In this task, shots with one or some of the following 3 low-level features (feature groups) must be detected, (1) pan (left or right) or track, (2) tilt (up or down) or boom and (3) zoom (in or out) or dolly. The following two types of camera motion detection methods, threshold-based and HMM-based respectively, are used to detect these camera motions based on the estimated feature parameters for camera motion detection described in section 2.2.

[2] HMM based camera motion identification experiments with some Θ were conducted using development data and Θ achieving optimal performance is selected.

3.1 Threshold-based camera motion detection

In this method, camera motions are detected whether the feature parameters exceed or go below a given threshold level. With this method, camera motion start point detection begins from the start frame of each shot. Then the start point of camera motion is detected when the feature parameters X_t or ΔX_t exceed given threshold levels Th_X or $Th_{\Delta X}$ for successive L_x^B frames (in the case of pan right). Once a start point is found, end point detection begins. Subsequently, the end point of camera motion is detected when the feature parameter ΔX_t goes below a given threshold level $Th_{\Delta X}$ for successive L_x^E frames. These procedures are repeated until the end of the shot, but only delta coefficients are used for detecting the 2nd and subsequent beginning points in a single shot.
The same procedure is used with a different threshold level to detect pan left. Furthermore, camera motion parameters Y_t and ΔY_t are used to detect tilt up/down and d_t and Δd_t are used for detecting zoom in/out.

3.2 Camera motion detection using Hidden Markov Models

Two-dimensional feature parameters are used in the threshold-based detection method described in section 3.1. Further improvement will be achieved by applying high dimensional feature parameters, such as all six dimensional parameters X_t, Y_t, d_t and their delta coefficients, at once. However, it is difficult to find appropriate threshold values for each dimension when using the threshold-based method. The support vector machines (SVM) has the ability to classify using multidimensional parameters, but this method is problematic for modeling variable length features, such as camera motion. Therefore, we applied a Hidden Markov Model (HMM) with Gaussian mixture to represent camera motions and detect them based on the maximum likelihood criterion. In this method, the video data is separated into sections corresponding to the existence or nonexistence of target camera motion and they are modeled using HMMs. For example, in the case of zoom-in, video data is classified into the following three sections: sections without any camera motion (NOT_CM), sections without zoom-in (NOT_ZOOMIN) and sections during which zoom-in occurs (ZOOM_IN) and HMM for each section is trained.
Using models for these sections, camera motion detection is implemented by finding a camera motion sequence $\hat{C} = [c_1, c_2 \cdots, c_N]$ for the given sequence of feature parameters $Y = [y_1, y_2, \cdots y_T]$ such that

$$\hat{C} = \arg\max_C P(C \mid Y) \qquad (5)$$

from every possible sequence of camera motion C.

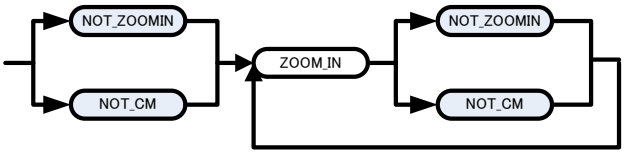

Figure 5. A finite state automaton (FSA) for HMM-based camera motion detection (zoom in)

Table 1. Number of camera motions in development data.

Camera motion	*Number of shots*	
PAN	left=575	right=683
TILT	up=214	down=166
ZOOM	in=826	out=488

Table 2. Number of camera motions in test data.

	Number of shots	
Camera motion	true	false
PAN	587	1159
TILT	210	1159
ZOOM	511	1159

This equation can be modified as follows:

$$\hat{C} = \arg\max_C P(C) P(y \mid C) \qquad (6)$$

where $P(C)$ is the *a priori* probability of the sequence of camera motions C, and $P(y \mid C)$ is the conditional probability of the feature parameters y being observed given a specific sequence of camera motion, C. $P(C)$ were set to be equal for all possible camera motion sequences in the following experiments, but rules that constrain the linkage between each camera motion are used . Figure 5 shows an example of rule described in finite state automaton (FSA). This FSA shows that target camera motion (ZOOM_IN) and section without any camera motion (NOT_CM or NOT_ZOOMIN) alternately appears.
The time period depending on each camera motion (or not) is obtained by using a Viterbi algorithm to find $\hat{C}$ [9]. The start and end points for each camera motion can be decided based on this result. This procedure is performed on six camera motions, namely pan left, pan right, tilt up, tilt down, zoom in and zoom out, respectively. Based on the start and end points of camera motions thus obtained and the given common shot boundary reference, the shots that each camera motion present are detected. Subsequently, the list of shots obtained with camera motion is used to evaluate the effectiveness of the proposed method, using a TRECVID low-level feature extraction task.

4. RESULTS

A TRECVID low-level feature extraction task is used to evaluate the effectiveness of the proposed method. TRECVID data includes about 170 hours of news programs in Arabic, Chinese and English and is divided into development and test data. We randomly selected 26 programs from the TRECVID development data and manually labeled the start and end points of camera motions. The amount of each camera motion in development data is described in Table 1. This development data are used to decide the threshold for the threshold-based method and train HMMs. The test data includes about 30 hours of news programs. Shots with clear examples of existing or non-existing camera motions were selected to evaluate performance. The number of unique shots with each camera motion is described in Table 2.

Table 3 shows the recall, precision, and F-measure[3] of the threshold-based camera motion detection. The results obtained by using feature parameters (1) delta parameters (**delta**) and (2) both static and delta parameters (**2-features**) are described in the table. The parameter for endpoint detection, L_x^B and L_x^E , was set to 10. Experiments with various threshold levels were conducted using development data and the threshold levels achieving optimal F-measure were selected and used in experiments using test data. Table 4 shows the performance of the HMM-based camera motion detection. The results obtained by using (1) static and delta parameters (**2-features**) which were used in threshold-based method and (2) static and delta parameters of all feature parameters X_t, Y_t , d_t (**6-features**) are described in the table. These results show that, even if, under the condition that the same feature parameter is used (**2-features**), the performance of the HMM-based method is superior to that of the threshold-based method. Further improvement is achieved by using the HMM-based method with all feature parameters (**6-features**).

Table 3. Recall, precision and F-measure of threshold-based camera motion detection.

	Feature	Prec.	Recall	F.
PAN	**delta**	0.982	0.652	0.784
	2-features	0.978	0.685	0.806
TILT	**delta**	1.000	0.114	0.205
	2-features	1.000	0.557	0.716
ZOOM	**delta**	0.932	0.270	0.419
	2-features	0.914	0.728	0.810
MEAN	**delta**	0.971	0.346	0.510
	2-features	0.964	0.657	0.781

Table 4. Recall, precision and F-measure of HMM-based camera motion detection.

	Feature	Prec.	Recall	F.
PAN	**2-features**	0.751	0.734	0.742
	6-features	0.892	0.804	0.846
TILT	**2-features**	0.825	0.810	0.817
	6-features	0.833	0.667	0.741
ZOOM	**2-features**	0.800	0.853	0.826
	6-features	0.827	0.840	0.833
MEAN	**2-features**	0.792	0.799	0.796
	6-features	0.851	0.770	0.808

5. CONCLUSION

In this paper, camera motion detection methods using a background image generated by the video mosaicing method are described. The proposed method is evaluated by using a TRECVID low-level feature extraction task and performance of HMM-based method is superior to the one of threshold-based method.

[3] The experimental result is evaluated in terms of the number of shots detected as true and true in the truth (true positives: TP), the number of shots detected as false and false in the truth (true negative: TN), the number of shots detected as true and false in the truth (false positive: FP), and the number of shots detected as false and true in the truth (false negative: FN). Based on these numbers, precision = TP / (TP + FP), recall = TP / (TP + FN) and F-measure = (2 * precision * recall) / (precision + recall) are calculated.

6. REFERENCES

[1] http://www-nlpir.nist.gov/projects/trecvid/

[2] K. Jinzenji et al.: "Algorithm for automatically producing layered sprites by detecting camera movement ", International Conference on Image Processing 1997, pp. 767-770 Vol. 1.

[3] J. Denzler et al.: "Statistical approach to classification of flow patterns for motion detection", International Conference on Image Processing 1996, pp. 517-520 Vol. 1

[4] P. Bouthemy et al.: "A unified approach to shot change detection and camera motion characterization, "IEEE trans. Circuits Syst. Video Technology, No.7, pp. 1030-1044, Vol. 9, Oct. 1999

[5] Y. Kanazawa et al.: "Image mosaicing by stratified matching," *Image and Vision Computing*, Vol. 22, No. 2 (2004-2), pp. 93-103.

[6] C. Harris et al.: "A combined corner and edge detector," *Proceedings of the* 4*th Alvey Vision Conference*, pp. 147-151, 1988.

[7] K. Isogawa et al.: "Recognition Method for Moving Text in Video, " Proc. of FIT2005, pp. 13-16, (in Japanese)

[8] K. JINZENJI et al.: "Global Motion Estimation for Sprite Production and Application to Video Coding, "IEICE D-II Vol. J83-D-II No. 2 pp. 535-544(in Japanese)

[9] S. Young et al.: The HTK Book (for HTK Version 3.3) http://htk.eng.cam.ac.uk/.