2009 13th International Machine Vision and Image Processing Conference

Robust Panning Analysis for Slideshow Detection in Video Databases

Zbigniew Zdziarski
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
zdziarze@tcd.ie

Rozenn Dahyot
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
Rozenn.Dahyot@tcd.ie

***Abstract*—We present an algorithm for slideshow detection in video databases such as YouTube or Blip.TV. Our solution is based around feature tracking to extract movement between sequentially captured frames. This movement is then analysed through the use of the Hough Transform and compared against behaviour commonly exhibited by slideshows: still and panning static images. We show experimentally the effectiveness of this novel idea and approach.**

***Keywords*-slideshow detection; video databases; Hough Transform; feature tracking;**

I. Introduction

The rise of digital communication technologies and the availability of cheap video hardware such as webcams or cell-phone cameras have facilitated the creation, recording and distribution of video data between millions of users around the world. Dedicated websites such as YouTube, or Blip.TV allow for an easy sharing of personal images and videos. However, there are few tools offered for computer users to browse these databases efficiently apart from using keyword searching. Word indexing can be efficient to index video content; however, users can be biased in their video descriptions.

Slideshows are videos that consist of a series of selected images (slides) that change after a set time. The images displayed can be still, panning, rotating or zooming. A short period of time may be allocated for transition between selected images. Slideshows may be considered by users as a video type they would rather not view or be discarded in their search results due to their static nature. There may on the other hand be a category of slideshows that users would want to view in their search results. Lectures would be an example of such slideshows. No tools yet exist to subjectively tag a video as being a slideshow.

In this paper we propose a feature tracking and Hough Transform [1] based solution for detecting slideshows in video databases. We use information about movement obtained from feature tracking between adjacently captured frames to analyse and compare against movement commonly exhibited by slideshows. The Hough Transform is used to examine the displacement of feature points between frame pairs. We show experimentally that this approach is successful in detecting slideshows with still or panning images. Slideshows with rotating or zooming images are not considered in this paper, though a solution to this problem is suggested in Section V.

II. Related work

Most research in video classification has focused on identifying entire videos as belonging to several broad categories such as movie genre [2]. Some authors have, however, focused on identifying segments of videos as being either violent [3] or frightening [4] or extracting news segments from an entire news programme [5]. Movies and sports are the most popular videos analysed for classification [2]. Some proposed solutions have focused on identifying a specific sport among a database of other videos [6] or specific informational videos such as news or medical education [7].

Brezeale et al. have proposed a survey of research on automatic video classification [2]. In it they give three sources of features: text, audio and visual.

The survey performed in [2] gives a number of methods for classifying videos (e.g. Support Vector Machines (SVMs) [8] and neural networks classifiers) but two particularly popular methods are the Gaussian Mixture Models (GMMs) and hidden Markov models (HMMs).

Some work with slideshows in video has been performed by Gigonzac et al [9] and Syeda-Mahmood et al. [10]. Gigonzac et al. propose a method to automatically match slides in a recorded presentation with their electronic version to enhance, for example, distance learning applications [9]. Locating the area of video where slides are being displayed is the first phase of their method. Colour matching is used for this. Syeda-Mahmood et al. propose a method to detect topic changes in a recorded presentation by using visual and audio data [10]. Region hashing is used in video analysis to detect slides in a frame sequence. Data from audio analysis is then combined to detect topic changes in the presentation. This area of research is slightly different to slideshow detection: input video in both cases is automatically expected to be a recorded presentation or lecture, and slides in both cases are known to exist in a small area of view in every video sequence. Our field of video analysis encompasses all video types, which gives us less prior knowledge to work with.

978-0-7695-3796-2/09 $26.00 © 2009 IEEE
DOI 10.1109/IMVIP.2009.23

No work to date has been performed to classify slideshow videos in large Internet-based video databases to personalise or optimise searching in such contexts. Our solution only focuses on the visual aspect of videos and is based around the linear transform case of the Hough Transform.

III. Proposed system steps

The algorithm for the proposed approach consists of two parts:

1) the motion analysis algorithm,
2) statistical analysis to ascertain slideshow detection.

The motion analysis algorithm is responsible for analysing the motion between subsequent frame-pairs. All motion analysis obtained from this algorithm is then collated to provide a positive or negative result with respect to slideshow detection.

The motion analysis algorithm contains the following major steps:

1) frame-pair selection algorithm,
2) feature location in the first frame (may have been performed on a previous run),
3) feature tracking calculation from frame 1 to frame 2 of frame pair,
4) motion analysis.

The following sections provide a brief summary of each step.

A. Frame-pair selection algorithm

The frame-pair selection algorithm is responsible for producing a pair of frames to subsequently analyse motion. The first frame in this pair is used as a feature reference frame to the second frame. If motion (or lack of it) pointing to a potential slide is detected in this frame pair, the new frame pair, selected on the next iteration of the algorithm, will be constituted by the first frame of the previous pair and a newly captured frame. This process continues until motion is detected that does not represent typical slideshow behaviour (e.g. erratic movement) - the next frame-pair, then, will consist of the second frame from the last pair (used as the reference frame) and a newly captured frame. If the distance between frames in a frame-pair exceeds 3 frames, the second frame of the last pair becomes the new reference frame. This is done to avoid the loss of a significant amount of feature points when parts of the image pan out of view that are present in the initial reference frame. Newly captured frames are captured at 0.5 second intervals throughout the algorithm. Figure 1 shows an example of the frame-pair algorithm in action.

B. Feature location and tracking

Features are located using the Shi and Tomasi (a.k.a. Kanade-Tomasi) corner detection algorithm [11]. Feature tracking is performed through the use of the iterative Lucas-Kanade method in pyramids [12]. Coordinates of the feature points are calculated on the current video frame (F2 in Figure 1) given their coordinates on the previous frame (F1 in Figure 1). Figure 2 depicts a typical feature tracking example.

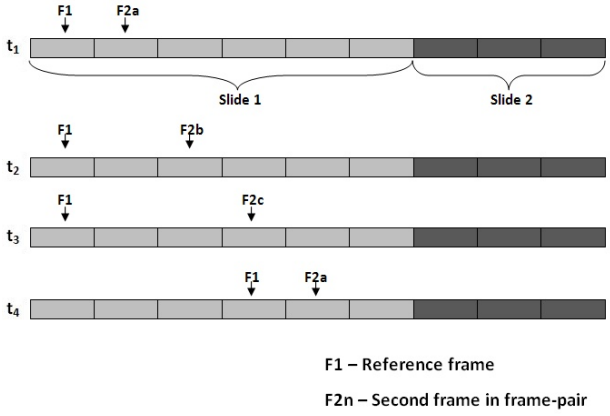

Figure 1. An example of the frame-pair selection algorithm in action

Figure 2. Feature tracking example for a diagonally panning image in a slideshow. Arrows indicate the displacement of features located in the previously captured frame

Our solution to the slideshow detection problem uses the OpenCV [13] implementations for the feature detection and tracking methods mentioned above.

C. Robust estimation of panning

There are two types of slides that interest us: static and panning slides. Lack of motion in a video sequence is considered to indicate a static slide. The linear transform case of the Hough Transform is used to detect panning slides. Each feature point pair defines a straight line $p = x\cos\theta + y\sin\theta$, where $p \in \mathbb{R}$ and $\theta \in [-\pi/2, \pi/2]$ and has a representation as a point (p, θ) in the Hough space. The 2-D histogram in our application is computed over the $(\theta, |n|)$ space, where $|n|$ is the distance between the points of a point pair. The more lines grouped in the same $(\theta, |n|)$ category, the higher the chance that a panning image is present in our series of frames.

The normal equation of a line has by default $\theta \in [-\pi/2, \pi/2]$. In this case, the displacement of a matched pair of points can have only two directions. Therefore, if an object moves in the opposite direction to another object, they will both have the same θ value. To get around this problem, when all the matched pairs of points have been categorised in the accumulator space, their displacement directions are checked. To avoid unnecessary calculations, this is performed only if the specific frame has been classified as potentially belonging to a slide by passing the necessary thresholds.

The length of each segment $|n|$ is considered to avoid confusion in situations in videos where foreground objects move in the same direction as the background. In these cases, the matched pair of points on foreground objects will exhibit smaller or larger displacement $|n|$ values.

It was found that frames should be captured approximately every half a second. This gives enough time for detectable movement to occur in an image sequence.

IV. Training and experimentation results

Two video databases were created to train, optimise and then test the proposed solution. The first database was used to retrieve the various parameters and threshold values needed in such video classification applications. The idea of this database was to set all variables so as to detect 100% of slideshows but minimise false positive and false negative results as much as possible. The second database was used to test the newly learnt application.

Each of the two databases consisted of the following videos:

- 20 slideshows with static slides (over 80 minutes duration in both databases),
- 20 slideshows with panning slides (over 80 minutes duration in both databases),
- 20 videos with very little movement (over 130 minutes duration in both databases),
- 50 randomly chosen videos from various genres (over 130 minutes duration in both databases).

All videos were of 320x240 pixel resolution which is the standard resolution of videos on YouTube.

On average our application takes approximately 2.61 seconds to analyse one minute's worth of video. The total time taken is highly dependent on the number of features found in each frame.

Videos with very little movement were of the following kind: lectures, talks, speeches, stand-up comedy, dramas, concerts and vlogs (video logs). The randomly chosen videos were, among other things, of the following kind: music videos, sports highlights, commercials, documentaries, amateur videos and news stories.

Bin no.	θ thresholds
0	>= -1.326 && <-1.105
1	>= -1.105 && <-0.852
2	>= -0.852 && <-0.576
3	>= -0.576 && <-0.291
4	>= -0.291 && <0.291
5	>= 0.291 && <0.576
6	>= 0.576 && <0.852
7	>= 0.852 && <1.105
8	>= 1.105 && <1.326
9	>= 1.326 to $\pi/2$ <-1.326 to -$\pi/2$

Table I
LIMITS OF THE BINS DEFINED FOR THE VARIABLE θ TO COMPUTE THE $(\theta, |n|)$ ACCUMULATOR

A. Training of the application

Regarding the bin limits for the $(\theta, |n|)$ accumulator, Table I shows the manually trained limit values for the x-axis (θ) bins of this space. The bin limit values of the y-axis ($|n|$) of the $(\theta, |n|)$ space increment by 5 pixels up to 60. Any lengths of segments that exceed 60 pixels are not included in the 2-D $(\theta, |n|)$ histogram. No slides in the test database moved fast enough to produce segment lengths of more than 60 pixels over a period of three frame captures.

It was found that for a frame to be flagged as possibly constituting a slide, at least 80% of matched point pairs needed to fall into the same category in the $(\theta, |n|)$ accumulator or 93% in 2 to 3 neighbouring categories. The error leniency was due to various issues ranging from image quality that affected feature detection, inherent errors in the Shi and Tomasi corner detection algorithm and the iterative Lucas-Kanade method in pyramids, and features being detected in parts of the image that recently panned into view.

Figure 3 shows examples of Hough transform histogram results for a frame pair taken from a slideshow video and a non-slideshow video.

Another set of threshold values was obtained for the amount of features that need to be detected in frame 2 from frame 1 in a frame pair. These are adaptive threshold values because they depend on the sequence number of a slide; parts of the image of a slide disappear per frame for a panning slide the further we are from the reference frame. The faster a slide moves, the more features disappear per frame capture. Since, however, the reference frame changes every 3 iterations, only three such thresholds were required. The threshold values obtained from the training database are shown in Table II.

Of course, the threshold value obtained from F1 - F2c could have been used for all cases but these adaptive thresholds gave us better results on the training database in our application.

A minimum value of 1.5 seconds (three sequentially captured frames) was allowed for slide duration. Any group of frames that obtained lower times than this value were

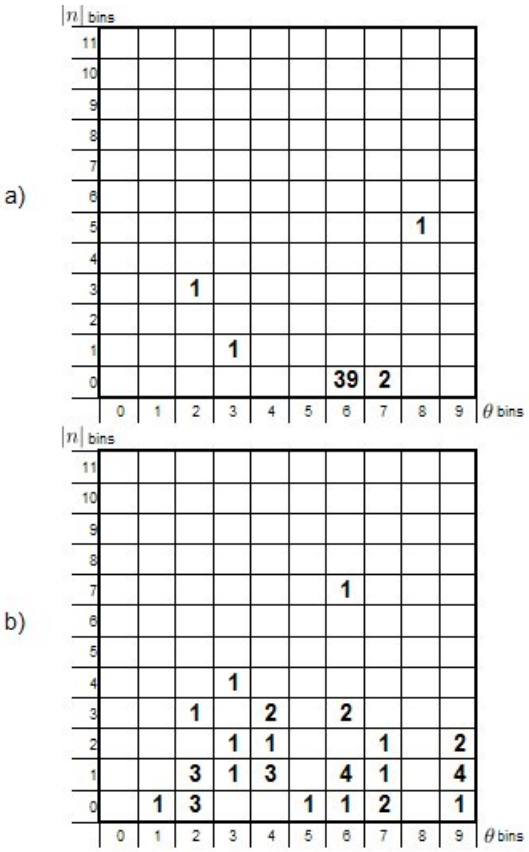

Figure 3. Example Hough transform histogram results: a) 44 matched points for a slide in a slideshow video b) 37 matched points for a random video (a car commercial).

No. of matched pairs	Threshold values (%)
F1 - F2a	89
F1 - F2b	82
F1 - F2c	78

Table II
ADAPTIVE THRESHOLD VALUES FOR THE $(\theta, |n|)$ SPACE

discarded and flagged as not constituting a slide.

A major dilemma in detecting slideshows is how much time to allow for frame transitions for an entire video sequence. The more time you allow, the more slideshows will be detected; however, other videos with smaller amounts of movement will start to be erroneously flagged. To allow for these slide transitions that are often more than a second in duration, it was calculated through the training database that at least half of the video sequence should form still images (panning or not) from a slideshow.

It should be noted that all the threshold and error leniency values mentioned above were obtained through manual tuning to optimise classification results on the training database.

B. Experimental results

It was found that 98% of slideshows were detected in the test database using the proposed algorithm. One slideshow was not detected due to its poor quality. Uncommonly heavy boxing and blurring effects inhibited an accurate detection of features and hence tracking. All the other slideshows were detected successfully.

From the 70 other videos that were not slideshows in the database, 4 were flagged as slideshows. Three of them had a camera at the back of the room or hall like a lecture theatre or concert hall. One was of a speech by Stephen Hawking made on stage. The camera did change angles but little detectable movement was present in the scenes. Any other videos with comparable lack of motion would also be given false positive status here.

The application did detect false positive slides in nearly half of the other videos (in vlogs or commercials, for e.g., when a person stared at the camera and barely moved their lips or when the product of interest was displayed at the end for a short period of time) but overall these 'slides' did not form at least half of the analysed time - sometimes not even a few percent of the total video.

The opposite was also true for some slideshows where certain slides were not detected because they failed to pass one of the thresholds and were therefore discarded. Overall, however, the total time of the other slides constituted at least half of the video length.

V. CONCLUSION AND FUTURE WORK

In this paper we presented a method to detect slideshows consisting of still and panning slides through the use of feature tracking and the Hough transform. We have shown that the method has very high detection rates. Problems with false positives, however, did occur and were caused by videos with very little detectable movement. A number of things could be performed to reduce the number of false positive results. Sound analysis is one such option. For most slideshow cases, music is played in the background of a slideshow. The lack of music could be an indication of another type of video.

Sound analysis could also be used to classify detected slideshows. Music could be an indication of a family's holiday photographs but a voice could be an indication of a lecture. Further sound analysis through word extraction could classify lectures into subjects or break a lecture down into topics as has been proposed in [10].

There exists a time/effectiveness trade-off decision that could decrease false positive results. As was mentioned in Section III-A, our system does not track points between sequential frames but between frames grabbed every half a second. Tracking points in sequential frames would make the application much more precise since, for example, the threshold values in Table II could be made more restrictive. Analysing, however, 25 frames per second instead of 2

(assuming a video of 25 fps ratio), would noticeably affect the running time of the application.

This system is not yet complete for full slideshow detection. Slideshows exist with rotating and zooming slides. A possible solution for this area of research is to use a 6-parameter transformation: $F(x, O) = Ax+d$, where A is a 2 x 2 matrix for affine transformation and d is the displacement vector as described in [14]. Future work will aim at taking into account this larger class of affine motion in slideshows.

ACKNOWLEDGMENT

This work has been supported by a Research Google Award 2007.

REFERENCES

[1] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, January 1972.

[2] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 38, no. 3, pp. 416–430, May 2008.

[3] J. Nam, M. Alghoniemy, and A. H. Tewfik, "Audio-visual content-based violent scene characterization," in *Int. Conf. Image Process. (ICIP)*, 1998, pp. 353–357.

[4] S. Moncrieff, S. Venkatesh, and C. Dorai, "Horror film genre typing and scene labeling via audio analysis," in *Int. Conf. Multimedia Expo (ICME)*, vol. 1, 2003, pp. 193–196.

[5] W. Zhu, C. Toklu, and S.-P. Liou, "Automatic news video segmentation and categorization based on closed-captioned text," in *IEEE Int. Conf. Multimedia Expo (ICME)*, 2001, pp. 829–832.

[6] V. Kobla, D. DeMenthon, and D. Doermann, "Identifying sports videos using replay, text, and camera motion features," in *SPIE Conf. Storage Retrieval Media Databases*, 2000, pp. 332–343.

[7] J. Fan, H. Luo, J. Xiao, and L. Wu, "Semantic video classification and feature subset selection under context and concept uncertainty," in *4th ACM/IEEE-CS Joint Conf. Digit. Libr. (JCDL)*, 2004, pp. 192–201.

[8] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," *Computer Vision Pattern Recognition (CVPR)*, pp. 130–136, June 1997.

[9] G. Gigonzac, F. Piti, and A. Kokaram, "Electronic slide matching and enhancement of a lecture video," *IEE European Conference on Visual Media Production (CVMP'07)*, December 2007.

[10] T. Syeda-Mahmood and S. Srinivasan, "Detecting topical events in digital video," *Proceedings of the eighth ACM international conference on Multimedia*, pp. 85–94, October 2000.

[11] J. Shi and C. Tomasi, "Good features to track," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.

[12] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," OpenCV documentation, Intel Corporation, Microprocessor Research Labs, Tech. Rep., 1999.

[13] *OpenCV Documentation*, http://www.intel.com/technology/computing/opencv/index.htm.

[14] R. Dahyot and A. Kokaram, "Comparison of two algorithms for robust m-estimation of global motion parameters," in *Irish Conference on Machine Vision and Image Processing (IMVIP)*, 2004, pp. 224–231.