Camera Zoom Motion Detection in the Compressed Domain

Pavan Sandula, Manish Okade
Department of Electronics and Communication Engineering,
National Institute of Technology Rourkela,
Rourkela, India
516ec6004@nitrkl.ac.in, okadem@nitrkl.ac.in

***Abstract*—In this paper we investigate the application of local tetra patterns to the compressed domain camera zoom recognition problem. The primary aim is to separate the zooming frames from the non-zooming (panning, tilting) frames for which the block motion vector orientation information is modeled utilizing a 3$\times$ 3 neighborhood and local tetra patterns. These patterns are binarized followed by feature reduction using the concept of uniform patterns and finally fed to the C-SVM classifier for recognition purposes. Comparative analysis with state-of-the-art methods using ESME and H.264 obtained block motion vectors extracted from standard video sequences show superior performance for the proposed method.**

***Index Terms*—zoom motion, local tetra patterns, uniform patterns, camera motion, C-SVM.**

I. Introduction

The motion dynamics in video sequences arises due to two types of motion viz. object motion also referred as local motion and camera motion also referred as global motion. Camera motion forms an integral component in most video sequences where the video acquisition takes place using a moving camera. Camera motion can be categorized into panning, tilting, zooming, rotation and random depending on the direction of movement of the camera. Panning and tilting are basically translatory behavior of the camera while zooming motion essentially scales the environment under capture. Due to the existence of various types of camera motion in video sequences namely panning, tilting, zooming etc. the first job at hand would be to detect zooming motion which is the focus of this paper. Detecting zooming is important since it can give various clues as to what is going around which is generally useful in surveillance as well as sports video analysis. A zooming-in camera would indicate an important event since the focus would be towards some salient activity. Although translatory behavior i.e. panning/tilting is a well studied problem since its useful for major applications yet zooming camera motion has not been fully established due to various intricacies in the motion vector behavior. A brief review of state-of-the-art methods analyzing and classifying camera motion and exploring its utility for various video processing tasks is presented next.

Jin et al. [1] were the first to carry out the detection of zooming frames in an video sequence by utilizing expectation maximization (EM) with promising results. However, since they used EM which is sensitive to initialization their method suffered along with drop in accuracy when motion vectors were noisy. Improvements to Jin et al. [1] method were later investigated by Guironnet et al. [2], Duan et al. [3] and Okade et al. [4] with each method having a particular application area in focus. Duan's method [3] focussed on video indexing and retrieval while Okade et al. method [4] explored the video stabilization application. Although the listed methods carried out zoom v/s non-zoom detection satisfactorily yet the suffered when noise in motion vectors was significant. However, since their aim was on utilizing zoom detection for some applications like indexing, retrieval, stabilization the accuracy drop as such did not matter much. Based on the review of these methods in our study whose results are part of this paper it was felt that robust zoom detection methods in presence of noise need to be investigated for which the local tetra patterns are explored. The contributions made in this paper are listed below;

1) Firstly, local tetra patterns [5] which are robust image texture descriptors previously utilized for image texture analysis in applications like face recognition are explored for the zoom motion detection problem.
2) Secondly, experiments are carried out to show that zoom motion detection in presence of noise significantly affects state-of-the-art methods while the proposed method is resistant to these noise attacks.

II. Proposed method

Fig. 1 demonstrates the Inter-frame block motion vector field which forms the input to the proposed method. This inter-frame block motion vector field is generated by performing motion estimation between successive frames 'n' and '(n+1)'. Fig. 1 depicts two such frames taken from sequence Tractor. The motion estimation can be carried out either using Exhaustive Search Motion Estimation (ESME) algorithm or by coding/decoding the video sequences using H.264/AVC codec and JM software. Fig. 2 shows the typical ideal zooming pattern scenario where the center is shown and its eight neighbors are oriented as depicted i.e. either diverging or converging. In case this convergence/divergence scenario does not hold then the frame will belong to any of the non-zooming cases i.e. panning, tilting, random etc. Herein comes

This research work is supported by SERB, Government of India under grant number: ECR/2016/000112.

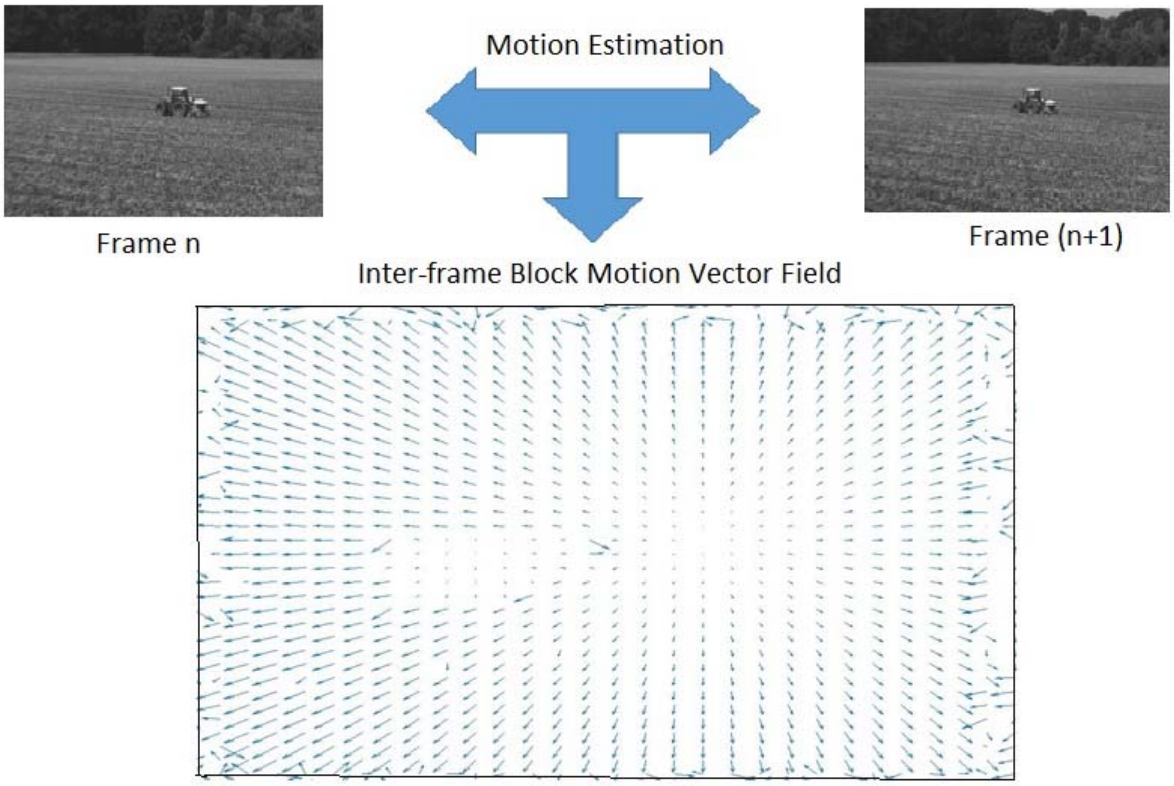

Fig. 1: Inter-frame block motion vector field resulting from motion estimation between two frames of sequence Tractor.

our contribution of using local tetra patterns for studying the local neighborhood with respect to the center orientation value for detecting zooming v/s non zooming camera as described below.

Local Tetra Patterns (LTrP) [5] which earlier found applications in image texture analysis is explored in this paper for video analysis application namely zoom detection. Towards this goal, the orientation of a block motion vector is estimated using $\theta = arctan\left(\frac{MV^Y}{MV^X}\right)$. A 3×3 neighborhood is then chosen for elements in θ array with center orientation denoted by $'c'$. The first order derivatives along 0^o and 90^o directions are estimated followed by quantizing the center direction based on the derivative exceeding or less than 45^o. The choice of 45^o as a threshold is based on the premise that zooming camera vectors are ideally oriented at 45^o as compared to panning and tilting vectors (non-zooming) cases. After the quantizing step the 2^{nd} order LTrP for center orientation is calculated followed by splitting it into 3 binary patterns thereby giving 12 (4 directions $\times$ 3 binary patterns) patterns. Finally, the 13^{th} pattern which is the magnitude of the orientation values is also calculated followed by dimensionality reduction using the concept of uniform patterns [6]. The reduced feature vector is then fed to the C-SVM classifier [7] which is able to separate the zooming v/s non-zooming frames. The detailed step by step description is given below;

1) Estimate the orientation of block motion vectors using Y-component (MV^Y) and X-component (MV^X) of the block motion vector in the range of 0^o to 360^o

$$\theta = arctan\left(\frac{MV^Y}{MV^X}\right) \tag{1}$$

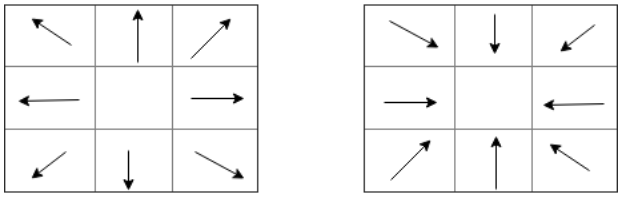

Fig. 2: Zooming Camera Patterns (diverging and converging).

2) Let 'c', 'h', 'v' denote center orientation value, horizontal and vertical neighborhoods respectively as shown below;

$\theta(n_1)$	$\theta(v)$	$\theta(n_3)$
$\theta(n_8)$	$\theta(c)$	$\theta(h)$
$\theta(n_7)$	$\theta(n_6)$	$\theta(n_5)$

The first order horizontal and vertical derivatives of center orientation value are computed using

$$\theta'_{0^o}(c) = \theta(h) - \theta(c) \tag{2}$$

$$\theta^{'}_{90^o}(c) = \theta(v) - \theta(c) \tag{3}$$

3) The first order derivatives are quantized $\theta^{'}_{\text{Dir.}}$ based on the center derivative exceeding or less than 45^o using

$$\theta^{'}_{\text{Dir.}}(c) = \begin{cases} 1, & \text{if } \theta^{'}_{0^o}(c) \geq 45^0 \text{ and } \theta^{'}_{90^o}(c) > 45^0 \\ 2, & \text{if } \theta^{'}_{0^o}(c) < 45^0 \text{ and } \theta^{'}_{90^o}(c) \geq 45^0 \\ 3, & \text{if } \theta^{'}_{0^o}(c) < 45^0 \text{ and } \theta^{'}_{90^o}(c) < 45^0 \\ 4, & \text{if } \theta^{'}_{0^o}(c) \geq 45^0 \text{ and } \theta^{'}_{90^o}(c) < 45^0 \end{cases} \tag{4}$$

4) The second order LTrP [5] is calculated as follows

$$LTrP^2(c) = \Big\{ f_1(\theta^1_{\text{Dir.}}(c), \theta^1_{\text{Dir.}}(n_1)), f_1(\theta^1_{\text{Dir.}}(c), \theta^1_{\text{Dir.}}(n_2)), \ldots, f_1(\theta^1_{\text{Dir.}}(c), I^1_{\text{Dir.}}(n_P)) \Big\}\Big|_{P=8} \tag{5}$$

where 'n' represents the neighborhood around center orientation value

$$f_1(\theta^{'}_{\text{Dir.}}(c), \theta^{'}_{\text{Dir.}}(n_p)) = \begin{cases} 0, & \text{if } \theta^{'}_{\text{Dir.}}(c) = \theta^{'}_{\text{Dir.}}(n_p) \\ \theta^{'}_{\text{Dir.}}(n_p), & \text{elsewhere} \end{cases} \tag{6}$$

5) The second order LTrP is then split up into three binary patterns as follows

$$LTrP^2|_{Dir=2,3,4} = \sum_{p=1}^{P} 2^{(p-1)} \times f_2(LTrP^2(n_c))\big|_{Dir=2,3,4} \tag{7}$$

$$f_2(LTrP^2(n_c))|_{\phi} = \begin{cases} 1, & \text{if } LTrP^2(c) = \phi \\ 0, & \text{elsewhere} \end{cases}$$

where, ϕ indicates $2, 3, 4$.

6) The 13^{th} binary pattern is calculated by utilizing the magnitudes of horizontal and vertical first order derivatives using

$$M_{\theta^1(n_p)} = \sqrt{(\theta^1_{0^o}(n_p))^2 + (\theta^1_{90^o}(n_p))^2} \tag{8}$$

7) This magnitude pattern (MP) is also binarized using

$$MP = \sum_{p=1}^{P} 2^{(p-1)} \times f_3(M_{\theta^1(n_p)} - M_{\theta^1(n_c)}) \tag{9}$$

$$f_3(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{elsewhere} \end{cases}$$

8) Since the dimensionality of the 13 patterns (12 orientation and 1 magnitude) is very high, uniform patterns [6] are used to reduce the dimensionality followed by forming their respective histograms.

9) The 13 uniform pattern histograms are concatenated and fed as feature vector to train the C-SVM classifier.

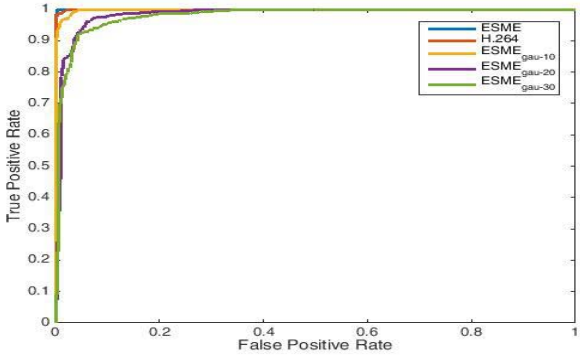

Fig. 3: ROC curves demonstrating the zoom detection (zoom v/s non-zoom) performance.

III. Results

The experimentation is carried out using MATLAB. Standard video sequences namely Tractor, Shields, Stefan, Station, Flowervase, Waterfall, Coastguard and Tempete are used in the study. Inter-frame block motion vectors are extracted from these sequences in two ways; i. Utilizing Exhaustive Search Motion Estimation (ESME) algorithm which depicts the ideal scenario. ii. Utilizing H.264/AVC to encode the above listed sequences followed by extracting the block motion vectors from these sequences which depicts the real codec performance. In both cases block size is maintained at '4×4'. GOP structure for H.264/AVC is IPP... with no B frames. Encoding and decoding for H.264/AVC is carried out using JM 19 software [9]. C-SVM with linear kernel is chosen as the classifier. 40% of zoom and 40% non-zoom samples are picked randomly to train the C-SVM classifier while the left out samples are used for testing. Five fold cross validation is carried out on the training set. Detection accuracy is measured using

$$Accuracy(\%) = \left(\frac{P_{tp} + P_{tn}}{2} \right) \times 100 \tag{10}$$

where P_{tp} is true positive rate and P_{tn} is true negative rate. Research carried out in this paper is reproducible and the MATLAB code can be accessed from https://sites.google.com/site/manishokade so that it help fellow researchers. Comparative analysis is carried out with parametric method [8] where an 8-parameter camera model is used, Duan et al. method [3] which is based on mean shift clustering followed by dominant cluster identification and Okade et al. method [4] where polar angle and magnitude histograms are used for classifying six camera motion patterns. Both Duan et al. method [3] as well as Okade et al. method [4] are non-parametric methods like this work which is also non-parametric.

The proposed method is objectively compared using ROC and AUC metrics with state-of-the-art methods. In order to establish the robustness of the proposed method Gaussian noise is added to the motion vectors to both the horizontal and vertical components under the setting of zero mean and

TABLE I: Accuracy (%) for zoom motion detection.

Block Motion Vector Type	Accuracy (%)			
	parametric method [8]	Duan et al. [3] method	Okade et al. [4] method	proposed method
ESME	54.98	91.27	92.26	99.17
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	51.25	58.41	52.98	96.47
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	50.72	52.96	52.10	91.96
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	48.45	52.25	51.92	84.00
H.264	56.84	81.53	94.81	99.02

varying variance ($\sigma^2 = 10, 20, 30$) so that we get three additional datasets referred as $ESME_{gau-10}$, $ESME_{gau-20}$ and $ESME_{gau-30}$.

Fig. 3 shows the ROC curves obtained for the proposed method for ESME i.e. without noise case, ESME with added Gaussian noise corresponding to three variances and the H.264 case. As observed ESME and H.264 achieves best performance since the tetra patterns are able to classify the zooming and non-zooming frames effectively. This can also be observed from Table I where their accuracy is greater than 98 %. Duan et al. [3] and Okade et al. [4] compete closely with accuracies in the range of 90$\sim$95 %. Parametric method on the other hand achieves very less accuracy in comparison to the three methods. In case of ESME noise added cases it is observed that the ROC curves corresponding to the proposed method experiences a drop in performance with the increase of variance of the added noise. This can also be observed from corresponding AUC Table II and Accuracy Table I. However, the drop in performance is way less than state-of-art-methods i.e. Duan et al. [3] and Okade et al. [4] where the accuracy drop is large with noise being added to the motion vectors. This robustness in performance for the proposed method can be attributed to LTrP feature descriptor which is robust under most noise added cases. One more point worth mentioning is that Duan et al. [3] and Okade et al. [4] methods were designed for a particular application area like indexing/stabilization where these noise added cases would not appear, while this paper is focussed on the generic zoom motion detection problem.

TABLE II: Area Under Curve (AUC) for zoom motion detection.

Block Motion Vector Type	proposed method
ESME	0.9998
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	0.9969
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	0.9837
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	0.9794
H.264	0.9996

IV. Conclusions

This paper dealt with the camera zoom motion detection problem by modeling the orientation information of the block motion vectors utilizing the concept of local tetra patterns. This is a novel application of using local tetra patterns in the domain of video analysis since earlier it had been used only for the image texture recognition problem. Experimental validation carried out utilizing ESME and H.264 obtained block motion vectors extracted from standard video sequences show superior performance for the proposed method in identifying the zooming frames in comparison to existing methods. Future work will concentrate on exploring this detection strategy to video saliency application.

V. Acknowledgement

This work is supported by SERB, Government of India under grant number: ECR/2016/000112.

References

[1] R. Jin, Y. Qi, and A. Hauptmann, "A probabilistic model for camera zoom detection," in *16th IEEE International Conference on Pattern Recognition*, vol. 3, 2002, pp. 859–862.

[2] M. Guironnet, D. Pellerin, and M. Rombaut, "Camera motion classification based on transferable belief model," in *14th European Signal Processing Conference*, Sept 2006, pp. 1–5.

[3] L.-Y. Duan, J. S. Jin, Q. Tian, and C.-S. Xu, "Nonparametric motion characterization for robust classification of camera motion patterns," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 323–340, 2006.

[4] M. Okade, G. Patel, and P. K. Biswas, "Robust learning-based camera motion characterization scheme with applications to video stabilization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 453–466, 2016.

[5] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2874–2886, May 2012.

[6] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107–1118, May 2009.

[7] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[8] Y. M. Chen and I. V. Bajic, "Motion vector outlier rejection cascade for global motion estimation," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 197–200, Feb 2010.

[9] *The H.264 AVC JM Reference Software*. [Online]. Available: http://iphome.hhi.de/suehring/tml/.