Camera Zoom Detection and Classification Based on Application of Histogram Intersection and Kullback Leibler Divergence

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*—This paper presents a novel compressed domain technique for detecting zooming camera in video sequences and its further classification into zoom-in camera and zoom-out camera. The inter-frame block motion vector field serves as the input to the proposed system which is partitioned into four representative quadrants for analysis purposes. The histograms of these four quadrants are analyzed utilizing histogram intersection feature for zoom motion detection while the cumulative histogram of these four quadrants are analyzed utilizing Kullback-Leibler divergence feature for zoom motion classification purposes. Experimental validation carried out utilizing block motion vectors extracted using Exhaustive Search Motion Estimation algorithm as well as H.264 decoded block motion vectors demonstrate superior performance in comparison to existing techniques.**

***Index Terms*—zoom motion, histogram intersection, Kullback-Leibler divergence, camera motion, compressed domain, support vector machine, block motion vectors.**

I. Introduction

Motion in video sequences occurs due to either object motion, camera motion or due to combination of object as well as camera motions. The camera dynamics occur mainly due to the movement of the camera and needs to be recognized for video analysis purposes since it has various applications like autonomous navigation [1], video saliency estimation [2], video indexing and retrieval [3] to name a few. The motion of the camera can be translational wherein the motion is either in horizontal (referred as pan) direction or vertical (referred as tilt) direction or it can be zooming in nature where the environment under capture is brought near (referred as zoom-in) or taken away (referred as zoom-out) from the camera.

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 and later separate it into either zooming-in or zooming-out camera motion types which is the objective of the current work presented in this paper. Major focus on zoom motion in video sequences has been from the video coding domain particularly the motion compensated prediction problem [4]–[6] for compression applications. However, zoom motion has also wide applications from video analysis point of view with applications ranging from indexing [7], retrieval [8], saliency estimation [9] to name a few. Zoom v/s non-zoom detection utilizing expectation maximization (EM) was carried out by Jin et al. [10]. However, since EM was utilized it had issues with initialization and convergence which affected the accuracy. Duan et al. [3] proposed an non parametric scheme for classifying camera motion categories with applications for video indexing and retrieval. They utilized mean shift clustering for identifying dominant motion clusters which was finally used for camera motion recognition. Since they used features namely cluster size, cluster number along with histograms of projected positions for identification purposes their method had shortcomings since these features were not able to bring out the underlying relationship of the block motion vectors. This method was improved in [11] wherein polar angle and magnitude histograms were used using a learning based scheme for identifying six camera motion types including zooming camera. However, in their work the zoom motion classification into zoom-in and zoom-out was left as future work since their focus was on translational camera for video stabilization applications. In [12] a transferable belief parametric model was utilized for the camera motion recognition problem. However, it utilized the Motion2D software to carry out the initial estimation of parameters, thereby not making it a stand alone algorithmic entity. In this paper, both the zoom motion detection as well as its further classification into zoom-in and zoom-out is carried out utilizing the concept of histogram intersection and Kullback-Leibler divergence [13] by analyzing the orientation histograms obtained by dividing the block motion vectors into four representative quadrants. Our results show superior performance in comparison to existing methods when tested using Exhaustive Search Motion Estimation (ESME) as well as H.264 compressed videos. Zoom motion detection and its classification into zoom-in/zoom-out plays a vital role in object localization which has applications in surveillance and autonomous navigation. Rest of the paper is organized as follows. Section II highlights

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

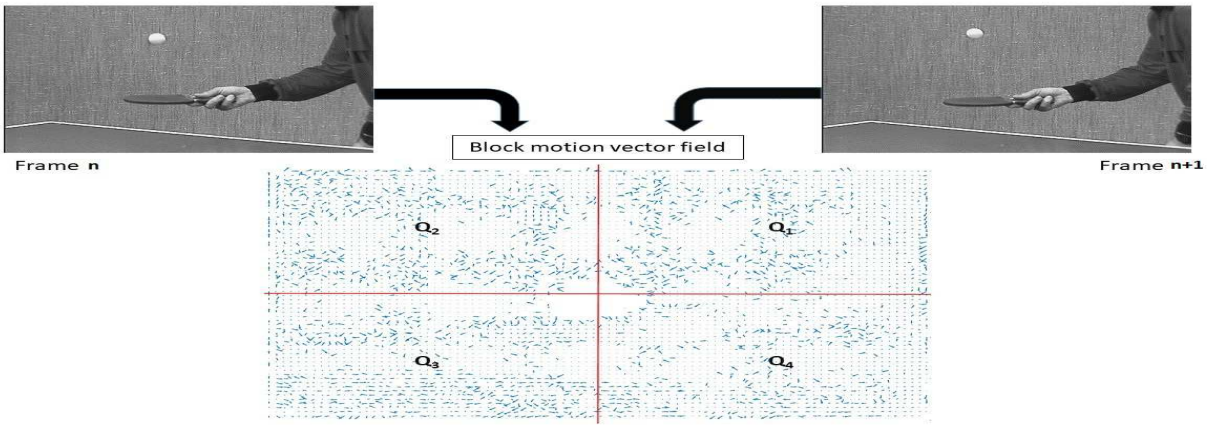


Fig. 1: Inter-frame Block Motion Vector Field.

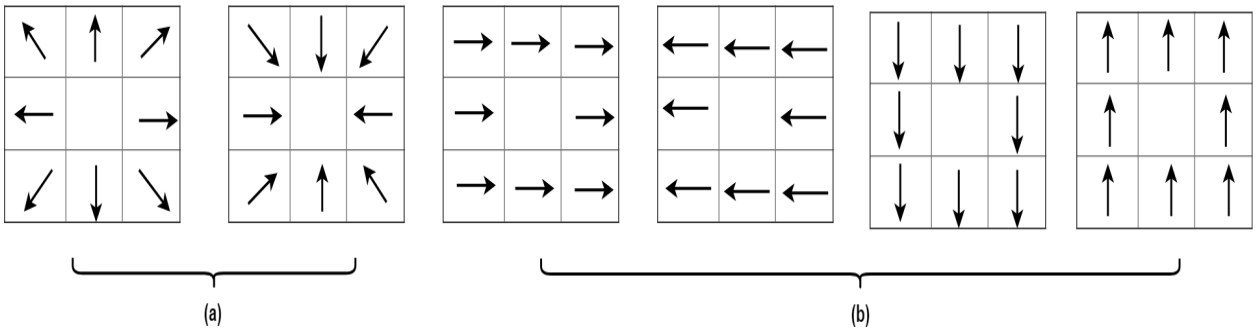


Fig. 2: Motion Vector Patterns corresponding to (a) zooming and (b) non-zooming camera.

the key contributions of the paper. Section III describes the proposed zoom motion detection and classification technique while Section IV gives the experimental results and finally in Section V we draw the conclusions.

II. Key Contributions

The contributions made in this work are two-fold. Firstly, zoom motion is recognized in video sequences i.e. identified from other camera motions like pan, tilt etc. utilizing the concept of histogram intersection between the quadrant histograms. Secondly, the identified zooming frames are further classified into zooming-in camera type and zooming-out camera type using the KL divergence between cumulative histogram of quadrants.

III. Proposed method

A. Zoom motion detection

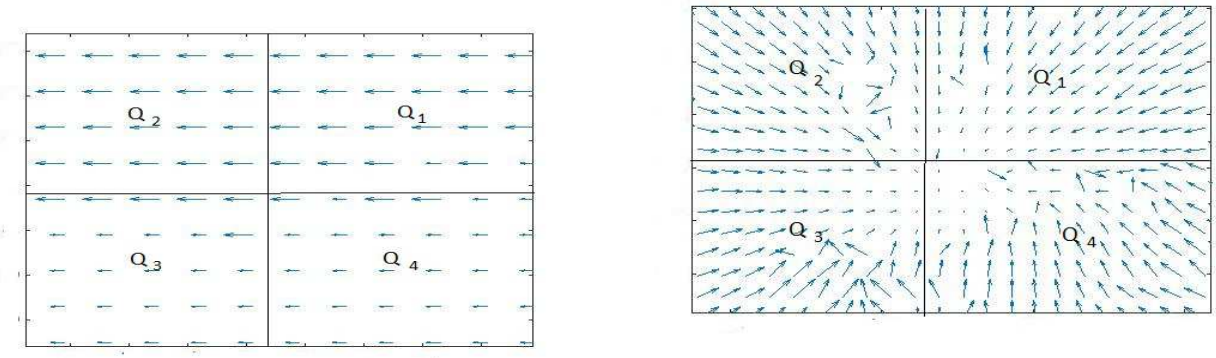
The proposed method utilizes the block motion vectors extracted from the compressed bitstream. Fig. 1 shows the inter-frame block motion vector field between two frames (frame # 33 and frame # 34) of sequence table tennis. As observed the block motion vector field shows various orientations corresponding to the nature of block motion vectors. The knowledge of nature of orientation pattern in case of zooming and non-zooming block motion vector fields will aid in detecting and separating the zooming camera from non-zooming camera. Fig. 2(a) shows the orientation patterns corresponding to the zooming camera motion pattern and Fig. 2(b) shows the orientation patterns corresponding to non-zooming category. As observed, the orientation pattern nature is different for the zooming camera and non-zooming camera which is exploited in this study by partitioning the block motion vector field into quadrants. The inter-frame block motion vector field between two frames is partitioned into 4 quadrants (Q_1, Q_2, Q_3, Q_4) for analysis purpose as shown in Fig. 4. The orientation histogram for the quadrants are estimated separately followed by calculating the histogram intersection between the quadrants to arrive at the feature vector which is utilized to train the C-SVM [14] classifier for separating the zooming frames from non-zooming frames. The detailed step by step description is given below;

1) Estimate the orientation of block motion vectors utilizing

$$MV_{ori} = arctan\left(\frac{MV^Y}{MV^X}\right) \quad 0 \leq MV_{ori} < 360 \quad (1)$$

where, $MV^Y \rightarrow$ vertical component of block motion vector and $MV^X \rightarrow$ horizontal component of block motion vector

2) Partition the inter-frame block motion vector field of size $N_1 \times N_2$ into 4 quadrants (Q_1, Q_2, Q_3, Q_4) as shown in



(a) (b)

Fig. 3: Block motion vector field pattern for (a) Panning camera (b) Zooming camera.

Fig. 4, followed by estimating the orientation histogram of individual quadrants utilizing

$$H_{Q_i}(l) = \frac{1}{\frac{N_1}{4} \times \frac{N_2}{4}} \sum_{j=1}^{R} \sum_{k=1}^{S} f_1(Q_i(j,k); l) \quad (2)$$

where, $l \in [0^o, 360^o]$, $R \times S$ is size of individual quadrant (Q_i) with $R = \frac{N_1}{4}$ & $S = \frac{N_2}{4}$ and

$$f_1(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

3) Estimate the histogram intersection (HI) between quadrants (Q_1 and Q_2), (Q_1 and Q_3), (Q_2 and Q_4) and (Q_3 and Q_4) utilizing

$$HI_{m,n} = \frac{\sum_{q=1}^{l} min(H_{Q_m}(q), H_{Q_n}(q))}{\sum_{q=1}^{l} H_{Q_n}(q)} \quad (4)$$

where, m $\in (1,1,2,3)$, n $\in (2,3,4,4)$

4) Concatenate the histogram intersection of the quadrants estimated earlier to form the Feature Vector (FV)

$$FV = [HI_{(1,2)}, HI_{(1,3)}, HI_{(2,4)}, HI_{(3,4)}] \quad (5)$$

5) Train the C-SVM classifier with linear kernel utilizing the feature vector formed for separating zooming frames from non-zooming frames.

The histogram intersection between two quadrants finds the amount of overlap between the orientation bins of the respective quadrants. If the orientation bins are similar for the quadrants under analysis as observed from Fig. 3 (a) then Eq. (4) tends towards 1 (i.e maximum overlap) signifying non-zoom motion (i.e pan, tilt). On the other hand for quadrants possessing dissimilar orientation bins as observed from Fig. 3 (b), Eq. (4) tends towards 0 (i.e least overlap) signifying zooming camera motion. This concept is exploited in the current work to distinguish between a zooming camera and a non-zooming camera. The rationale for computing histogram intersection between only specific pairs of quadrants is based on exploiting the concept of similarity between block motion vector orientations. As observed from Feature Vector formed in Eq. (5) adjacent quadrants (1, 2) & (3, 4) and diagonally opposite quadrants (1, 3) & (2, 4) are utilized to estimate the histogram intersection. In case of zooming frames these quadrants (adjacent and diagonal) result in least overlap due to dissimilar orientation types while in case of non-zooming frames (pan/tilt etc.) these quadrants result in maximum overlap due to similar orientation types.

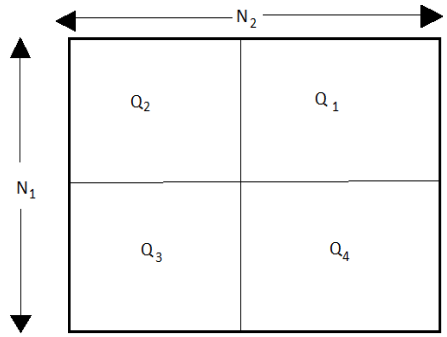
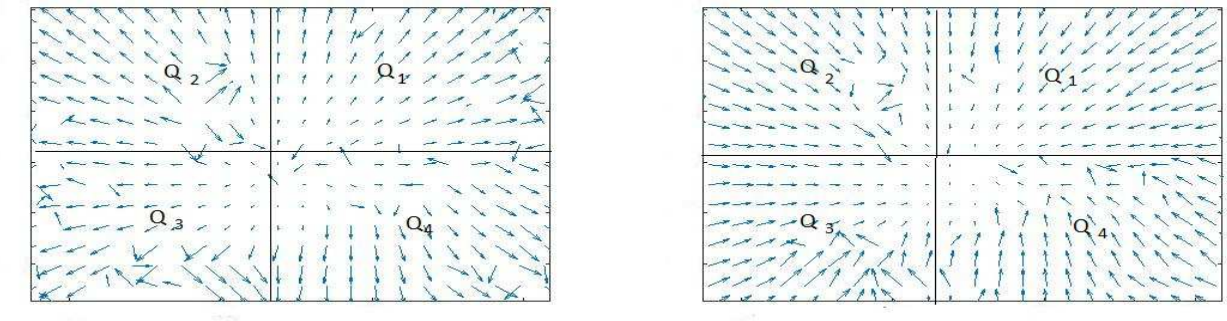


Fig. 4: Inter-frame block motion vector field (size $N_1 \times N_2$) depicting the partitioning into 4 quadrants.

B. Zoom motion classification

Once the zooming frames have been detected the next task would be to classify them into zooming-in camera and zooming-out camera. Fig. 5(a) and Fig. 5 (b) show the block motion vector field for zooming-in and zooming-out camera motion types, respectively. As observed, Zooming-in camera has motion vectors pointing outward from the center of frame



(a) (b)

Fig. 5: Block motion vector field pattern for (a) Zooming-in camera (b) Zooming-out camera.

(i.e diverging field) while Zooming-out camera has motion vectors pointing towards the center of the field (i.e converging field). In-order to classify them, we utilize the Kullback-Leibler (KL) divergence between diagonal and adjacent quadrants as described below;

1) Estimate the orientation of block motion vectors utilizing Eq. (1).
2) Partition the inter-frame block motion vector field into 4 quadrants utilizing Eq. (2)
3) Estimate the cumulative histogram of individual quadrants utilizing

$$CH_{Q_i}(q) = \sum_{j=1}^{q} H_j \tag{6}$$

where, Q_i is the ith quadrant with i $\in (1, 2, 3, 4)$ and q $\in (1 \dots 360)$

4) Estimate the Kullback-Leibler (KL) divergence between the cumulative histograms of quadrants (CH_{Q_1} and CH_{Q_3}), (CH_{Q_1} and CH_{Q_4}), (CH_{Q_2} and CH_{Q_3}), (CH_{Q_2} and CH_{Q_4}) utilizing

$$D_{KL}(CH_{Q_m}||CH_{Q_n}) = \sum_{q=1}^{360} CH_{Q_m}(q) \log\left(\frac{CH_{Q_m}(q)}{CH_{Q_n}(q)}\right) \tag{7}$$

where, m $\in (1, 1, 2, 2)$, n $\in (3, 4, 3, 4)$

5) Concatenate the KL-divergence of the quadrants to form the Feature Vector (FV)

$$FV = [D_{KL}(CH_{Q_1}||CH_{Q_3}), D_{KL}(CH_{Q_1}||CH_{Q_4}), D_{KL}(CH_{Q_2}||CH_{Q_3}), D_{KL}(CH_{Q_2}||CH_{Q_4})] \tag{8}$$

6) Train the C-SVM classifier with linear kernel utilizing the feature vector formed for separating zooming-in frames from zooming-out frames.

The KL divergence measures the amount of similarity between two distributions. In the current scenario, the cumulative histograms of quadrants are utilized to study the behaviour of block motion vectors in case of Zooming-in camera and Zooming-out camera cases followed by estimating the KL-divergence between the cumulative histograms of the quadrants. Cumulative histogram (distribution) is chosen since it provides information of how the orientation patterns vary in each quadrant for zoom-in/zoom-out pattern types. We have not normalized the cumulative histogram before its application in Eq. (7) and plan to do it when we exploit the proposed zoom motion classification scheme for saliency application in future. In case of Zooming-in camera the KL divergence between the diagonally opposite and adjacent quadrants will be large since motion type is divergence (i.e vector pointing outwards) while for Zooming-out camera the KL divergence between diagonally opposite and adjacent quadrants will be relatively small since motion type is convergence (i.e vectors pointing inwards). This concept is utilized to separate the Zooming-in and Zooming-out camera motion types. The horizontally adjacent quadrants are excluded while estimating the KL divergence since it has been observed in our experimental simulation that including it in the feature vector does not significantly change the accuracy. This is due to the fact that whilst recognizing zoom-in v/s zoom-out camera types the maximum change in orientation will occur in vertically adjacent quadrants i.e. (1, 4) & (2, 3) and diagonally opposite quadrants i.e. (1, 3) & (2, 4).

IV. Results

MATLAB R2016a is utilized for experimentation. Sequences available at https: //media.xiph.org/video/derf and https://nsl.cs.sfu.ca/wiki/index.php/ Video_Library_and _Tools which are standard in video analysis studies namely Tractor, Shields, Stefan, Station, Flowervase, Waterfall, Coastguard and Tempete are used. The zooming and non-zooming frames used in training and testing are manually labeled. Inter-frame block motion vectors are generated from these sequences by utilizing Exhaustive Search Motion Estimation (ESME) algorithm with block size '4×4', search range [-12 12] and cost function set to

TABLE I: Accuracy (%) for zoom motion detection at false positive rate set to 1%.

Block Motion Vector Type	Accuracy (%)		
	[3]	[11]	proposed method
ESME	91.08	92.25	96.71
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	57.41	51.01	85.43
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	51.25	50.16	70.00
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	50.41	49.83	61.11
H.264	81.53	94.81	97.94

TABLE II: Area Under Curve (AUC) for zoom motion detection demonstrating the performance on various block motion vector types.

Block Motion Vector Type	proposed method
ESME	0.9958
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	0.9289
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	0.8297
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	0.7338
H.264	0.9987

MAD. H.264/AVC obtained block motion vectors are also used to demonstrate the performance on a real codec by encoding them using JM19 encoder [15] (Software) with GOP IPP... and block size '4×4' to maintain consistency in comparison with ESME block size. Block motion vectors extracted from these encoded sequences (using Idecode.exe in JM-19) form the practical block motion vector case. Comparative studies is carried out with method proposed in [3] where dominant motion clusters were identified utilizing mean shift clustering followed by extracting features from the dominant clusters for camera motion recognition as well as method proposed in [11] where a learning based camera motion characterization scheme based on polar angle and magnitude histograms was utilized for recognizing six camera motion types.

A. Classifier Details

C-SVM with linear kernel is utilized for carrying out the classification studies. For each training and testing pair 40% of zoom and 40% of non-zoom samples are picked up randomly to train the C-SVM classifier and the remaining samples were used for testing. The above procedure is repeated thirty times using five fold cross validation on the training set. The cost parameter 'C' is trained and is used to obtain optimum cost in range $\{i|i \in \{0.1, 0.5..10\}\}$. 2000 frames from each class type are utilized for training the C-SVM and the frames from each class type which are not utilized for training are picked for testing. Same combination of sample size and classifier type is used in classifying zoom-in and zoom-out camera types. The detection accuracy is taken as the average of probability of true positive rate (P_{tp}) and true negative rate (P_{tn}) and this is averaged over 30 random experiments utilizing

$$Accuracy(\%) = \left(\frac{P_{tp} + P_{tn}}{2}\right) \times 100 \tag{9}$$

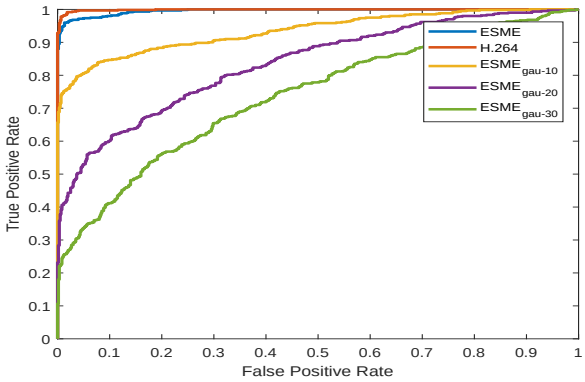


Fig. 6: ROC curves depicting the zoom detection (zoom v/s non-zoom) performance on various block motion vector types.

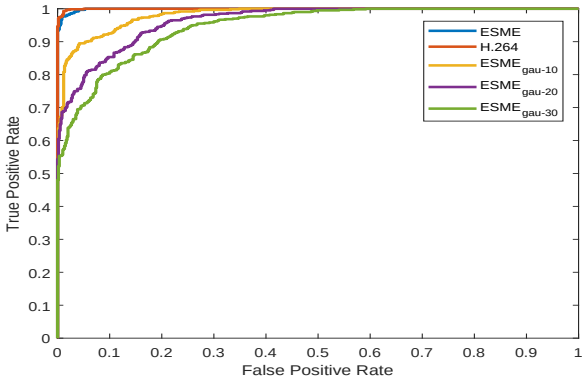


Fig. 7: ROC curves depicting the zoom classification (zoom-in v/s zoom-out) on various block motion vector types.

B. Objective Evaluation

Objective evaluation for the proposed technique is carried out in two ways: 1) ROC and AUC analysis which signifies the general detection performance and 2) by measuring detection accuracy at a low probability of false positives rate setting. The false positive rate is set to 1%, since this is the most widely used setting in classification studies. In order to analyze the robustness of the proposed method we add gaussian noise to both horizontal as well as vertical components of the block motion vector with zero mean and varying variance ($\sigma^2 = 10, 20, 30$) thereby generating 3 additional datasets for the experimental studies which we refer as $ESME_{gau-10}$, $ESME_{gau-20}$ and $ESME_{gau-30}$. Fig. 6 shows the ROC curves for ESME and its noise added variants and as observed

TABLE III: Accuracy (%) for zoom motion classification at false positive rate set to 1%.

Block Motion Vector Type	Accuracy (%)	
	[3]	proposed method
ESME	76.03	98.25
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	63.53	89.24
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	52.82	83.30
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	50.50	76.56
H.264	60.31	98.55

.

TABLE IV: Area Under Curve (AUC) for zoom motion classification demonstrating the performance on various block motion vector types.

Block Motion Vector Type	proposed method
ESME	0.9991
ESME corrupted with gaussian noise ($\sigma^2 = 10$)	0.9822
ESME corrupted with gaussian noise ($\sigma^2 = 20$)	0.9636
ESME corrupted with gaussian noise ($\sigma^2 = 30$)	0.9468
H.264	0.9994

ESME achieves the best detection performance followed by drop on its noise added variants where it is noted that detection performance drops with increase in the variance of the added gaussian noise i.e. ($ESME_{gau-30} < ESME_{gau-20} < ESME_{gau-10}$). The corresponding AUC values are shown in Table II. The performance evaluation for Zoom motion classification is shown in Fig 7 which shows similar trend for ESME and its noise added variants. The corresponding AUC values are shown in Table IV.

Next, the detection accuracy at FPR $< 1\%$ obtained by the proposed method is shown in Table I and Table III. As observed the performance for the proposed method is better for all cases in comparison to existing methods thereby signifying the robustness of the proposed method which is due to the fact that the quadrant analysis using measures like histogram intersection for zoom motion detection and KL divergence for zoom motion classification is better able to capture the mutual relationship between the orientation of block motion vectors. It is observed from Fig. 6 and Fig. 7 that H.264 case achieves nearly same performance as ESME case in both zoom detection as well as zoom classification scenarios since H.264/AVC uses the concept of "skipped" motion inference wherein a skipped area of a predictively coded (P) frame infers motion content and aids in the detection as well as classification process which is very useful while coding video containing camera (global) motion.

V. Conclusions

This paper investigated the zoom motion detection as well as its further separation into zoom-in and zoom-out camera in case of compressed domain videos. The first motive was to detect zooming frames from non-zooming frames which was carried out utilizing the histogram intersection between quadrants as a feature. Once the zooming frames were detected, the next task was to separate them into zooming-in and zooming-out types which was carried out utilizing the KL divergence between quadrants as a feature. C-SVM classifier was utilized for training/testing purposes. Comparative analysis with existing methods using ESME as well as H.264 obtained block motion vectors showed very good performance for the proposed method. Our future work is focussed on exploiting the zooming cue for estimating salient regions in video sequences.

References

[1] S. Ghosh and J. Biswas, "Joint perception and planning for efficient obstacle avoidance using stereo vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1026–1031.

[2] Y. Fang, W. Lin, Z. Chen, C. Tsai, and C. Lin, "A video saliency detection model in compressed domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 27–38, Jan 2014.

[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] L.-M. Po, K.-M. Wong, K.-W. Cheung, and K.-H. Ng, "Subsampled block-matching for zoom motion compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1625–1637, 2010.

[5] H.-S. Kim, J.-H. Lee, C.-K. Kim, and B.-G. Kim, "Zoom motion estimation using block-based fast local area scaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1280–1291, 2012.

[6] H. Yuan, Y. Chang, Z. Lu, and Y. Ma, "Model based motion vector predictor for zoom motion," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 787–790, Sept 2010.

[7] K. Schoeffmann, M. Taschwer, and L. Boeszoermenyi, "Video browsing using motion visualization," in *IEEE International Conference on Multimedia and Expo*, June 2009, pp. 1835–1836.

[8] W. Pan and F. Deschenes, "Interpreting camera operations in the context of content-based video indexing and retrieval," in *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, June 2006, pp. 7–7.

[9] G. Abdollahian, Z. Pizlo, and E. J. Delp, "A study on the effect of camera motion on human visual attention," in *15th IEEE International Conference on Image Processing*, Oct 2008, pp. 693–696.

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

[11] 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.

[12] 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.

[13] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 03 1951.

[14] 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.

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