Frequent Temporal Pattern Mining Perspective Of Video Compression

B.Sivaselvan and G.Krishna Sree

Department of Computer Science and Engineering
Indian Institute of Information Technology Design and Manufacturing, Kancheepuram
Off Vandalur-Kelambakkam Road, Melakottaiyur Village
Chennai, Tamil Nadu 600127

Abstract—**This project explores the techniques of data mining which focus on discovering similar items, frequently occurring patterns in the domain of lossy video compression. Both spatial and temporal redundancy in a video are removed to an extent till where the loss is not perceived.The main motive of this system is to reduce these redundancies by employing data mining and encoding techniques. The temporal redundancy is removed by applying mining algorithm that considers sequential patterns with time extended data base.The spatial redundancy is removed by applying clustering and closed frequent sequence mining. Simulations show significant reduction in redundant parts of the video and compression is achieved close to the existing compression codecs.**

Index Terms—**Video compression, Frequent Sequence Mining,Temporal pattern mining, K-means Clustering.**

I. Introduction

There is a popular saying, "We are living in the information age" [1], however, we are actually living in the data age. Terabytes or petabytes of data can be seen in everyday life. This explosive growth of data is a result of the computerization of our society and the fast development of powerful data collection and storage tools. The list of sources that generate huge amounts of data is endless. The widely available, and gigantic body of data, makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining. Data mining refers to discovering information from data which are hidden, useful and non-trivial. This process is also known as Knowledge Discovery from Data(KDD) [1].

In Data Mining, five perspectives were observed by Naren Ramakrishnan et. al. which are Compression, Search, Induction, Approximation, and Querying [2]. The perspective of Data mining as a compression technique is of interest in this study. The process of data mining focuses on generating a reduced(smaller) set of patterns(knowledge) from the original DB, which can be viewed as a compression mechanism. A few of the mining techniques such as Classification, Clustering, Association Rule Mining(ARM) may be explored from the compression perspective. In this study, the knowledge Frequent Sequence Mining(FSM) is used to achieve efficient compression.

Compression techniques are broadly classified into lossless and lossy compression. Lossless compression reduces the size without reducing the quality of the source. Compression techniques like Shannon-Fano Encoding, Huffman Encoding, Arithmetic Encoding, Lempel-Ziv-Welch(LZW), Bit-Plane Coding, Lossless Predictive Coding, PNG, DEFLATE perform lossless compression [3], [4], [5], [6], [13], [14]. Lossy compression,on the other hand, reduced the size of the video by removing or reducing the irrelevancy in addition to redundancy, thereby achieving better compression than lossless compression. Compression techniques like H.264 [7], HEVC [8], MPEG-2 [9] perform the task of lossy video compression.

Frequent Sequence Mining(FSM) is the process of mining sequences that occur more frequently more than a minimum support count. FSM could be classified into two based on the representation of the DB, horizontal and vertical. An example of horizontal mining is GSP [10] which is similar to Apriori algorithm, and SPADE [11] and SPAM are examples of vertical mining algorithms.

II. Related Work

In the area of lossy video compression, MPEG is one of the most common and widely used formats. Some of its versions that are widely used are MPEG-1, MPEG-2 and MPEG-4 part 10. MPEG-1, one of the early versions of MPEG is a standard for video lossy compression. This was design and created to compress analog video(VHS) to 1.5 Mbit/s with less quality loss, making it suitable for broadcasting through digital cable/satellite TV and digital audio broadcasting. MPEG-2 is an extension to MPEG-1 with a much higher bit rate, suitable for digital TV. MPEG-2 part 10 which is widely used in Youtube and Google video, compresses the video to almost half the size of that of MPEG-1 and MPEG-2 without compromising on the quality of the video. It is also known as H.264/AVC [12]. The major difference between H.264 and MPEG-1/2 is due to the support of arithmetic coding in addition to Huffman coding.

Video compression is the process of reducing the memory needed to represent a video and is based on the fact pixel neighboring pixel values are correlated within a frame and other frames. Video compression based on pixels from a frame and its surrounding frames is known as spatial compression or inter-frame compression. If it is based on neighboring pixels in the same frame then it is known as temporal compression or inter-frame compression.

III. Temporal Redundancy Removal

One major problem that arises during the mining process is treating data with temporal feature i.e. the attributes related with the temporal information present in the database. This temporal attribute require a different procedure from other kinds of attributes. However, most of the data mining techniques tend to treat temporal data as an unordered collection of events, ignoring its temporal information. The ultimate goal of temporal data mining is to discover hidden relations between sequences and subsequence of events. An efficient approach to mining casual relations is sequence mining. The discovery of relations between sequences of events involves mainly three steps [15]:

- Representation and modelling : In this sequence of the temporal data are transformed into a suitable form
- Similarity measure: Definition of similarity measures between sequences.
- Mining Operation: Application of models and representations to the actual mining problems.

Temporal redundancy is present in video signals when there is significant similarity between successive video frames.Pixels in two video frames that have the same values in the same location. Exploiting temporal redundancy is one of the primary techniques in video compression. [16].

A. Preprocessing of the video

The given video is prior processed to store as a time extended sequential database. Each frame in video is considered and is split into nine parts.

This split is performed because a part of a frame can be in similar with other rather than comparing two frames. All these split parts from the frames are given as an input to the K means clustering algorithm. K means clustering is implemented using the euclidean distance metric between the corresponding pixel values. Each split frame in its corresponding cluster number is considered as an item for mining. The frames in one shot, the shot boundary is calculated by algorithm 1, is considered as one transaction in the time extended data base.

B. Sequence pattern mining with time extended database

Sequential pattern mining is an important data mining method with broad applications that can extract frequent sequences while maintaining their order. However, it is important to identify item intervals of sequential patterns extracted by sequential pattern mining. For example, a sequence <A, B >with a 1-day interval and a sequence <A, B >with a one-year interval are completely different; the former sequence may have some association, while the latter may not. To adopt item intervals, two approaches have been proposed for integration of item intervals with sequential pattern mining; (1) constraint-based mining and (2) extended sequence-based mining.

Algorithm 1 Shot Boundary Detection [18]

```
    Input:
Modified Video file $V$
    Output:
Shot boundary frame numbers
    procedure
2:      $fc \leftarrow frame\_count(video)$
        for  $i \in$ [1,fc]  do
          Split each frame into 9 sub blocks
4:          for $i \in$ [1,9]  do
                $histogram(i,j) \leftarrow calchist(video(i,j))$
6:          end for
        end for
8:      $dist \leftarrow 0$
        for $i \in$ [0, fc-1] do
10:         $dist(i) \leftarrow dist(i) + distance(hist(i), hist(i+1))$
        end for
          kmeans(dist, num_of_clusters = 2)
12: Shot_boundary_frames = frames in cluster 1
    end procedure
```

For example, when users set the maximal time interval to one day, they extract frequent sequences whose items occur within 1 day. Although constraint-based mining algorithms escape extraction of sequences with non-interest time intervals such as too long intervals, however, it is difficult for users to specify an optimal constraint related to item interval, and so the user must re-execute the algorithm after changing the constraint value. This results in a decrease in usability.

To extract frequent sequences consisting of the same items but satisfying different item interval constraints by one time execution, extended sequence-based mining algorithms have been proposed. These algorithms extract frequent sequential patterns with item intervals by converting item intervals to pseudo items. However, as they cannot adopt any constraints related to item intervals, they may extract meaningless patterns, such as sequences with too long item intervals.

A generalised sequential pattern mining algorithm with item intervals is explained in this paper [17]. The output of this algorithm will be the frequently occurring sequences within the time interval constraint specified by the user. A sequence of length one is considered because,as a central tendency measure is used to represent the frequent sequences, reconstruction quality would be diminished if two items of different clusters are considered. Here the minimum time interval mentioned is 4 units and maximum time interval mentioned is 6 units. These frequently occurring sequences are the parts of the frames that are sliced in earlier step. Now all the frequently occurring sequences are replaced by their corresponding median value as a measure of central tendency. Now all the parts of the frames that are split are combined and given as input to the spatial reduction process.

ID	Sequences
1	(0, 1, 2), (1, 1, 3), (2, 1, 2, 3), (3, 1, 4, 5), (4, 2, 3)
2	(5, 3, 4), (6, 2, 3), (7, 4, 5)
3	(8, 4, 5), (9, 3, 6)

TABLE I
TIME EXTENDED SEQUENTIAL DATA BASE SAMPLE

If the sequential data base is represented as shown in table I, the output of the frequent temporal mining algorithm with the parameters given are, minimum support is 60 %, the minimum time interval allowed between first itemset and last itemset is 4,the maximum time interval allowed between first itemset and the last itemset is 6 units, the maximum number of items to be present between items is 0 as for the video compression the input items are cluster identifiers and one cluster can't be compared with other. So the output given by the algorithm is (1,1,1,1) from the first sequence and these four parts of the frames are represented by same number which is the median of all the pixel values. Median is used as a measure of central tendency because it is less susceptible to outliers than mean.

Algorithm 2 Temporal redundancy removal

```
    Input:
1) Video file: $V$
2)Time intervals: $\beta 1$, $\beta 2$
Output:
Modified video frames removing temporal redundancy.
 1: procedure
 2:
 3:     $(S_1, S_2, ..., S_t) \leftarrow Shot\_boundary\_detection(V)$
 4:
 5:     $(k_1, k_2, ..., k_p) \leftarrow split\_frames(V)$
 6:
 7:     $(C_1, C_2, ..., C_n) \leftarrow clustering(k_1, k_2, ..., k_p)$
 8: Find and Replace candidate items with median of corresponding pixels
 9: Combine all the frames and shots to get modified video.
10: end procedure
```

IV. SPATIAL REDUNDANCY REMOVAL

To remove more redundant data present in the frames of the video, the following process is followed. Here the algorithm devised for video compression by using clustering,frequent sequence patterns is presented. The input video is converted to a form suitable for mining. The given modified input video is split into its corresponding components R,G,B and the 3d video matrix is converted to 2d and given as input to clustering. After clustering the pixels are replaced by cluster identifiers [19]. And cluster identifiers are given as input to the mining algorithm.After the mining process is over the sequences are encoded using Huffman encoding.

A. Modified Video

The Video S, that is modified after removing temporal redundancy is chosen in the RGB color space and each component is converted to the form $m{\times}n{\times}f$, where m is the width, n is the height and f is the number of frames. Cell {i,j,k} in this matrix represents the color value of the pixel in the i^{th} column, j^{th} row, and k^{th} frame. The three color components of the video are represented by R, G and B. The three components are independent of each other and are processed in parallel. Consider the matrix representation of each component S_i. The $m{\times}n{\times}f$ 3D matrix is converted to a 2D matrix with ($m{\times}n$) rows and f columns. Each column in the matrix represents the values of each pixel for all the f frames of the video.

Each row in the matrix is considered as a transaction for mining. For example, consider a video of dimensions 320×240 and number of frames f is 205, then the 2D matrix to which it is converted contains 320×240 = 76,800 rows and 205 columns. If we split this matrix into blocks of size b = 100(for example), we get n_b = 3 shots. The dimensions of these blocks will be 76800×100, 76800×100 and 76800×5.

B. Compressor

The compressor pseudo code is shown in algorithm 3. Each block is given to clustering by k-means. Every pixel is considered as a data object. The first step in K-means is to maximize the intra cluster similarity and the second is to minimize the inter cluster similarity. K-means clustering defines k centroids, one for each cluster. The centroids should be chosen in such a way that, they are far from each other. Next to this, every data object is associated with the cluster whose centroid is the nearest to the object. After every association, the cluster centroid is updated. Once all the objects are clustered, the global centroids are found and every data object is re associated to the cluster whose centroid is the nearest, to minimize the inter cluster similarity. This process is repeated until no more re-associations are done. K-means clustering is applied in parallel to all the blocks. The pixels in each block are replaced with their respective cluster identifiers using the cluster identifier tables.
These blocks which are modified i.e which are replaced by the cluster identifiers is given to a mining algorithm.A frequent sequence mining algorithm is to be used for discovering closed frequent patterns in sequence databases. An overview of how frequent pattern mining algorithm works, a sequence of transactions is considered, in which each transaction has some itemsets. To the input, minimum support is given and based on this the frequent items are generated.
The cluster centroids are stored so that during decompression the pixels belonging to that cluster are replaced by their cluster centroids.

Algorithm 3 Spatial redundancy removal

```
    Input:
1) Video file: $V$
2) Block size $b$ 3) Minimum Support $\alpha$
4) Number of clusters $k$
    Output:
1) $C_I$, the compressed video
2) Code Table of each component $CT_1$,$CT_2$,$CT_3$
1:  procedure
2:  Split input video $V$ into $V_1$, $V_2$, $V_3$ corresponding to Red, Green, Blue components respectively.
3:  Number of blocks $n_b = \frac{f}{b}$
4:
5:      $i \leftarrow 1$
6:  Parallel for $i$ in [1,3] do
7:  Split $V_i$ into $n_b$ blocks $B_{i_1}$, $B_{i_2}$,..., $B_{i_{n_b}}$
8:  Perform K-means Clustering in parallel on ($B_{i_1}$, $B_{i_2}$,..., $B_{i_{n_b}}$) resulting in $k$ clusters.
9:  [$T_{B_1}$, $T_{B_2}$,..., $T_{B_{n_b}}$] = Replace each pixel value in every block with the cluster identifier of the cluster to which the pixel belongs.
10: $S_{i_1}$,..., $S_{i_{n_b}}$ = prefix-span ([$T_{B_1}$, $T_{B_2}$,..., $T_{B_{n_b}}$])
11: $CT_i$ = Huffman-coding($S_{i_1}$,..., $S_{i_{n_b}}$)
12: end procedure
```

C. Huffman Encoding

Huffman encoding [3] is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code. The variable-length codes assigned to input characters are Prefix Codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bit stream. Using the sequences in *M* with their modified support values, Huffman coding is performed to generate the binary codes for each sequence. The sequences and their corresponding binary codes are stored in Code Table *CT*.

D. Decompressor

Algorithm 4 performs decompression of the clustering approach mentioned above. In the decompression process, the code tables of each component are decoded. For each component, cluster identifier are replaced with the corresponding mean pixel value. Then, all the components are merged to form the input video.

The above presented algorithm is tested and compared with many videos on compression ratio and visual quality on various metrics.The results are presented in the later section.

Algorithm 4 Decompressor for clustering based video compression

```
    Input:
1) Compressed Video $C_I$
2) Number of blocks $l$
3) Code Table of each component $CT_1$,$CT_2$,$CT_3$
4) Cluster Identifier Table of each block in each component $T_{S_1}$,..., $T_{S_K}$
    Output: Decompressed Video $D_I$

1: procedure
2: Decode each component of the compressed video $C_I$ using the code tables $C_{T_i}(1 \leq i \leq 3)$, to get $V_i$.
3: For each component, replace each cluster identifier in $V_i$ with the corresponding mean pixel value.
4: Merge all the components to form the decompressed video $D_I$.
5: end procedure
```

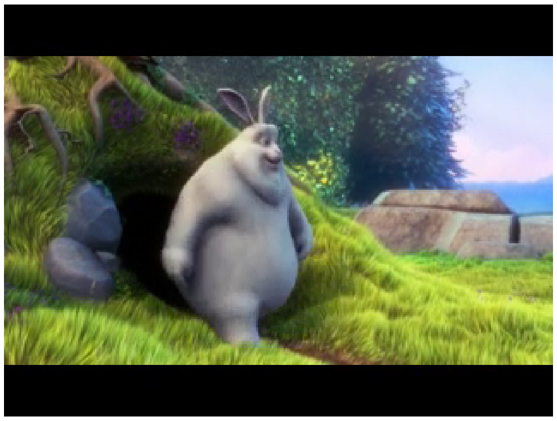
Fig. 1. Frame 100 of sample original video

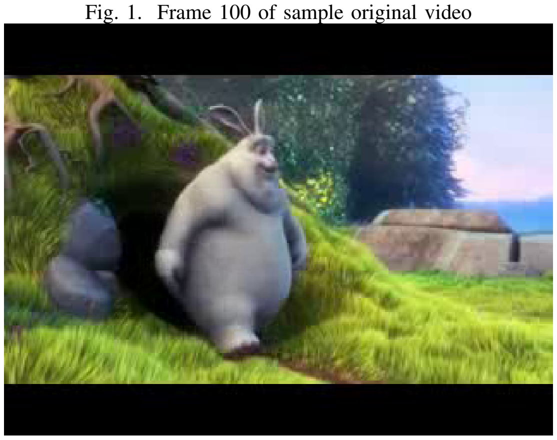
Fig. 2. Frame 100 of reconstructed video

V. Results

The simulation is performed on the standard test videos. The algorithm is implemented in python and is simulated on intel core i5-3210M CPU @ 2.50GHz clock speed and 4 GB Main memory running on linux ubuntu 16.04. The efficiency of the compression in temporal redundancy removal is effected by number of the cluster and the time constraints in temporal mining given by the user. Whereas the efficiency in spatial redundancy removal is affected by paramenters block size b, number of clusters K, minimum support α.

A. Compression metrics

Data compression ratio is defined as the ratio between the uncompressed size and compressed size.

$$C_r = \frac{\text{Uncompressed size of the video}}{\text{Compressed size of the video}}$$

Thus a representation that compresses a 10 MB file to 2 MB has a compression ratio of 10/2 = 5. The higher the compression ratio , the better is the compression algorithm.

B. Visual quality metrics

The video quality is measured using the metrics Peak Signal to Noise Ratio(PSNR). PSNR is measured using Mean Square Error(MSE).
PSNR is defined as follows:

$$\text{MSE} = \frac{1}{XY}\sum_{m=0}^{X-1}\sum_{n=0}^{Y-1}[I(m,n) - I^{'}(m,n)]^2$$

$$PSNR = 20log_{10}\frac{I_{max}}{MSE}$$

Here X, Y are the dimensions of a video frame indicating the number of rows and columns. I(m, n) and $I^{'}$(m, n) indicate the pixel value of the decompressed video frame and the original video frame at location (m, n) respectively. I_{max} denotes the maximum possible pixel value of the video frame I. If the average of errors(MSE) is low, PSNR will be high.

The temporal redundancy removal algorithm is given the inputs as number of clusters as twenty, and the time constraint as 4 units of time i.e if the items occur in the interval of 4 units it is considered in the count of support. And the minimum support is chosen to be 75%. And the parameters of the spatial redundancy removal are varied and compared.

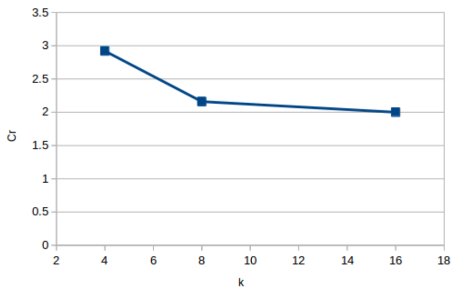

Fig. 3. C_r vs K

Based on the above comparison, the cluster value is chosen to be 8 clusters for spatial redundancy removal.

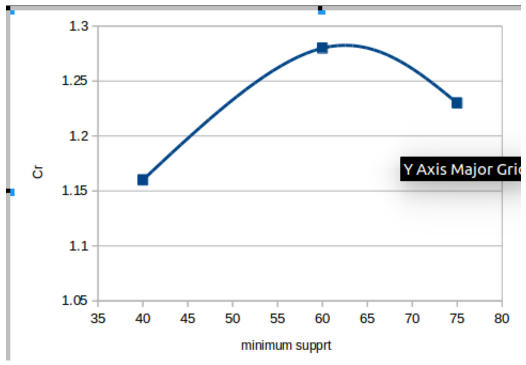

Fig. 4. C_r vs *min_sup*

Data set of the 10 videos are taken from shutter stock and are compared with existing video compression techniques on visual quality and compression ratio. In table 6 comparison between the proposed compression technique and existing video compression codecs is done. The table 7 for 1 Mb sized video and table 8 for 3 Mb, compares the visual quality metrics of the proposed algorithm and the existing video compression techniques. The visula quality metrics compared are PSNR, SSIM [20], VQM, MSAD, Netflix VMAF.

SSIM
Structural information of an image can be defined by those characteristics that represent the structure of the objects in the scene, independently of the mean brightness and contrast. These components are based on measurement of three components: luminance comparison, contrast comparison and structure comparison [20]. The SSIM index can gain values from 0 to 1 where value of 1 represents maximum quality.

VQM
This metric takes into account of the spatial temporal property of human's visual perception. The values of VQM start from 0 and in real situations can reach around 12. VQM value of 0 represents minimum distortion and maximum quality.

Netflix VMAF
Video multi method assessment function(Vmaf), is a video quality metric developed by Netflix. The values lies between 0 to 100 where, 100 represents a better reconstructed video.

Video Id	Original size (in KB)	H-264 (in KB)	H-265 (in KB)	Mpeg-2 (in KB)	Mpeg-4 (in KB)	proposed (in KB)
1	1190	570	621	1112	950	564
2	1260	674	538	1230	1010	556
3	3440	3240	3400	3335	3450	2960
4	1260	792	955	1021	1207	864
5	3160	1170	1245	2720	2190	1960
6	1180	641	639	1050	963	756
7	2050	731	771	2000	1420	842
8	1000	512	513	556	663	895
9	5000	4190	4785	4850	4670	3820

Fig. 5. Comparison of the sizes between proposed and existing video codecs

Video Id	H-264	H-265	Mpeg-2	Mpeg-4	proposed
1	2.08	1.91	1.07	1.25	2.04
2	1.86	2.34	1.02	1.4	2.26
3	1.06	1.011	1.03	1.15	1.004
4	1.59	1.31	1.23	1.04	1.45
5	2.7	2.53	1.16	1.44	1.612
6	1.84	1.843	1.12	1.22	1.56
7	2.80	2.65	1.025	1.44	2.43
8	1.95	1.94	1.79	1.50	1.117
9	1.19	1.044	1.03	1.07	1.30

Fig. 6. Compression Ratio Comparison

Video codec	PSNR (in Db)	SSIM (No units)	VQM (No units)	MSAD (No units)	Netflix VMAF (No units)
H-264	42.4	0.9895	0.476	0.965	95.64
H-265	43.51	0.990	0.42	0.876	96.40
Mpeg-2	44.90	0.990	0.332	0.861	98.67
MPEG-4	44.18	0.989	0.355	0.920	97.90
Proposed	39.00	0.980	0.770	1.39	90.313

Fig. 7. Visual quality metrics comparison for 1MB video

Video codec	PSNR (in Db)	SSIM (No units)	VQM (No units)	MSAD (No units)	Netflix VMAF (No units)	MSSIM
H-264	41.817	0.978	0.6577	1.465	92.66	0.9912
H-265	43.353	0.9835	0.5591	1.2248	94.43	0.993
Mpeg-2	42.82	0.9791	0.5731	1.351	95.20	0.9916
MPEG-4	42.02	0.9758	0.6053	1.463	93.54	0.9899
Proposed	41.923	0.9782	0.645	1.438	93.07	0.9911

Fig. 8. Visual quality metrics comparison for 3MB video

VI. Conclusion and Future Work

This work presents an approach for video compression using temporal mining and other data mining techniques to remove the spatial and temporal redundancy.And the algorithm is tested with benchmark video data sets and compared with the existing video compression techniques. The comparison between the proposed and existing video codecs confirm that the proposed work is in par with the compression ratio and in visual quality. So, as part of future work focus should be on improving compression ratio while achieving better quality and less compression time.

References

[1] J.Han and J.Pei and M.Kamber, "Data mining: concepts and techniques," , Elsevier, 2011.

[2] N.Ramakrishnan and A.Y.Grama, Data mining: From serendipity to science, 32nd ed., vol. 8. Computer, 1999, pp.34–37.

[3] D.A.Huffman, "A method for the construction of minimum-redundancy codes," Proceedings of the IRE, vol. 40(9), 1952, pp. 1098–1101.

[4] D. Salomon, " Data compression: the complete reference," Springer Science and Business Media, 2004.

[5] C.E.Shannon, "Communication theory of secrecy systems," Bell Labs Technical Journal, vol. 28(4), 1949, pp. 656–751.

[6] I.H.Witten, R.M.Neal, J.G.Cleary, "Arithmetic coding for data compression," Communications of the ACM, vol. 30(6), 1987, pp. 520–540.

[7] J. Ostermann, J. Bormans , P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video coding with h. 264/avc: tools, performance, and complexity," IEEE Circuits and Systems magazine, vol. 4(1), 2004, pp. 7–28.

[8] G.J.Sullivan, J. Ohm, W.J.Han, T. Wiegand, " Overview of the high efficiency video coding (hevc) standard ," IEEE Transactions on circuits and systems for video technology, vol. 22(12), 2012, pp. 1649–1668.

[9] B.G.Haskell, A.Puri, A.N.Netravali , " Digital video: an introduction to MPEG-2 ," Springer Science and Business Media, 1996.

[10] R. Srikant, R. Agrawal , " Mining sequential patterns: Generalizations and performance improvements ," ICDE, 1995,, pp. 1–17.

[11] M.J.Zaki , " Spade: An efficient algorithm for mining frequent sequences ," Machine learning, vol. 42(1), 2001, pp. 31–60.

[12] D. Marpe , T. Wiegand, G.J. Sullivan, " The h. 264/mpeg4 advanced video coding standard and its applications ,"IEEE communications magazine, vol. 44(8), 2006, pp. 134–143.

[13] T. Boutell, " PNG(Portable Network Graphics) specification version 1.0," 1997.

[14] L.P.Deutsch, " Deflate compressed data format specification version 1.3," 1996.

[15] Mohd Shahnawaz, Ashish Ranjan, and Mohd Danish. "Temporal datamining: an overview". In:International Journal of Engineering and Ad-vanced Technology1.1 (2011), pp. 2249–8958

[16] Richard Mayer and Richard E Mayer.The Cambridge handbook of multi-media learning. Cambridge university press, 2005

[17] u Hirate and Hayato Yamana. "Generalized Sequential Pattern Mining-with Item Intervals." In:JCP1.3 (2006), pp. 51–60

[18] Gautam Pal et al. "Video shot boundary detection: a review". In:EmergingICT for Bridging the Future-Proceedings of the 49th Annual Conventionof the Computer Society of India CSI Volume 2. Springer. 2015, pp. 119–127

[19] M Karthik, C Oswald, and B Sivaselvan. "Frequent Sequence Min-ing Ap-proach to Video Compressio". In:International Conference on Computa-tional Intelligence, Cyber Security, and Computational Mod-els. Springer.2017, pp. 87–97

[20] M Vranjes, Snježana Rimac-Drlje, and Drago Zagar. "Objective video-quality metrics". In:ELMAR 2007. IEEE. 2007, pp. 45–49

[21] A.K.Jain, " Data clustering: 50 years beyond k-means. Pattern Recognition Letters ,"19th International Conference in Pattern Recognition (ICPR), vol. 31(8), 2010, pp. 651–666.

[22] M. Einasto1, L.J. Liivamagi1 et al, " Principal component analysis ,"Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2(4), 2010, pp. 433–459.

[23] A. Gomariz, M. Campos et al, " ClaSP: An efficient algorithm for mining frequent closed sequences ,"Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2013, pp. 50–61.

[24] P. Fournier-Viger, A. Gomariz et al, " Fast Vertical Mining Sequential Pattern Mining Using Co-occurrence Information ,"Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2014, pp. 40–52.