Frequent Sequence Mining approach to Video Compression

Sudheera Daggumalli - COE14B006
Inchara.K.M - COE14B010
Sirisha Kona- COE14B017

Indian Institute of Information Technology,
Design and Manufacturing Kancheepuram, Chennai

Project Guide: Dr. B. Sivaselvan

April 26, 2018

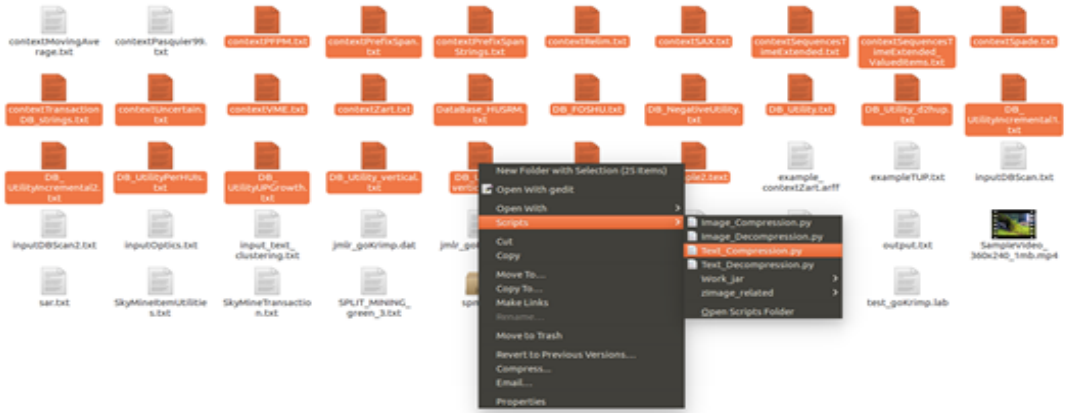
Project Details

The Project consists of the following work:

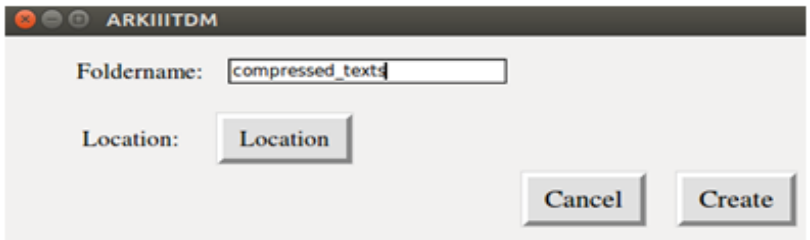
- ⋆ A Graphical User Interface based Compression tool is designed and developed for Text and Image Compression.
- ⋆ Closed Frequent Sequence Mining Approach to Video Compression
- ⋆ Clustering based approach to Video Compression
- ⋆ Dimensionality reduction based approach to Video Compression

GUI for Text Compression

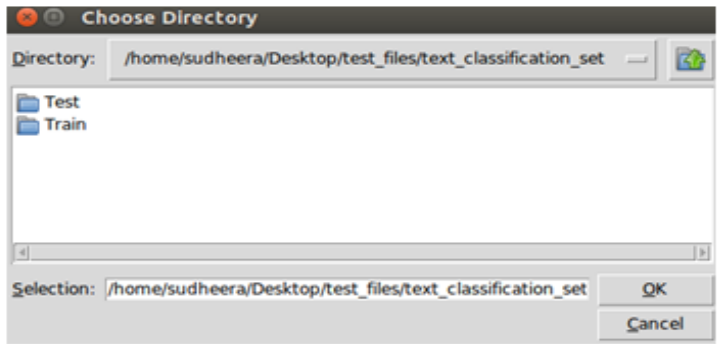
1. Choose text files for compression



2. Right click to choose compression

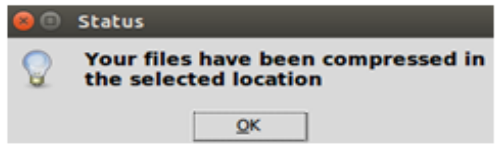


3. Enter name for compressed folder



4. Choose location to save compressed files

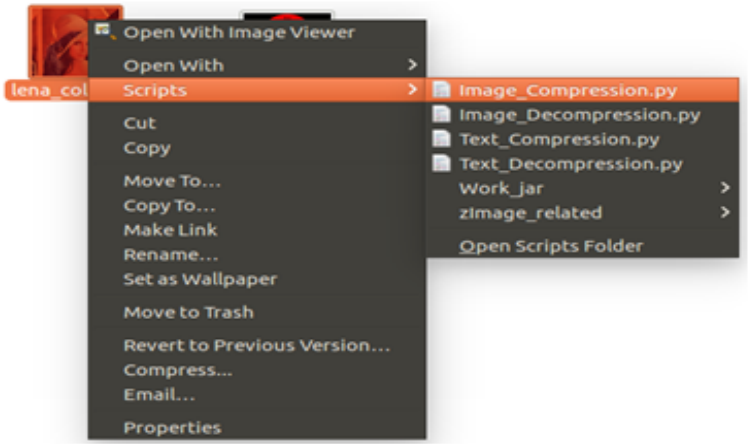
5. Loading bar displaying the progress



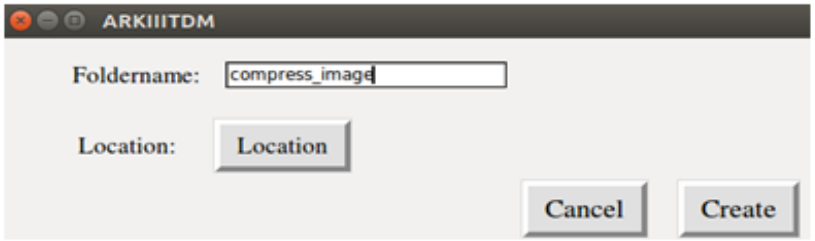
6. Message is displayed after compression is over

7. Compressed file folder

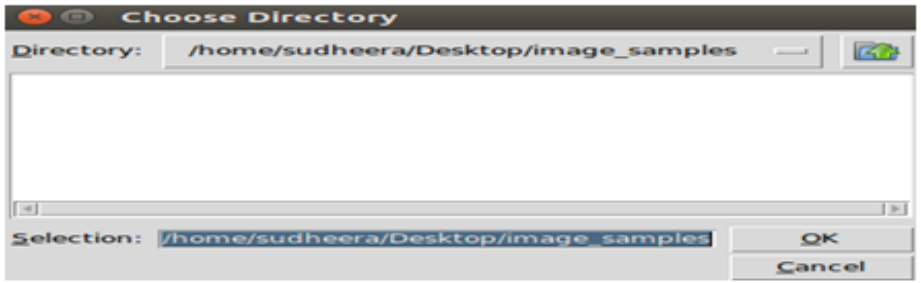
GUI for image Compression



1. Right click on image to choose compression

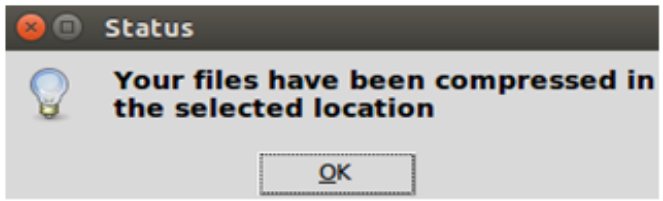


2. Enter name for compressed folder



3. Choose location to save compressed files

6. Compressed file folder



5. Message is displayed after compression is over

4. Loading bar displaying the progress

Closed Frequent Sequence Mining Approach to Video Compression

1. Source Video
   - ⋆ The input video V is chosen in the RGB color space and each component is converted to the form m (width)×n (height)×f (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 V_1, V_2 and V_3 and are independent of each other.
   - ⋆ The m×n×f 3D matrix is converted to a 2D matrix with (m×n) rows and f columns. Each row in the matrix represents the values of each pixel for all the f frames of the video.
   - ⋆ V_i is divided column wise based on the block size b. The number of blocks is represented using n_b and the blocks are represented as $B_i = \{B_{i_1}, B_{i_2}, ..., B_{i_{n_b}}\}$.
   - ⋆ Each row in each block is considered as a transaction for mining.

Example

Input Video V

- ⋆ Dimensions = 320×240
- ⋆ number of frames(f) = 205

Modified Video

- ⋆ 2D matrix Dimensions = 76,800 rows and 205 columns
- ⋆ If block size(b) = 100, we get $n_b = 3$ blocks.
- ⋆ The dimensions of these blocks will be 76800×100, 76800×100 and 76800×5.

Example

$$V_i = \begin{bmatrix} 2 & 12 & 133 & 24 & 25 & 26 & 7 & 11 & 92 & 10 \\ 5 & 12 & 3 & 14 & 15 & 11 & 7 & 128 & 9 & 102 \\ 5 & 12 & 3 & 1 & 1 & 1 & 7 & 218 & 19 & 12 \\ 5 & 12 & 3 & 4 & 5 & 11 & 7 & 217 & 7 & 100 \end{bmatrix}$$

$$B_{i_1} = \begin{bmatrix} 2 & 12 & 133 & 24 \\ 5 & 12 & 3 & 14 \\ 5 & 12 & 3 & 1 \\ 5 & 12 & 3 & 4 \end{bmatrix} \quad B_{i_2} = \begin{bmatrix} 25 & 26 & 7 & 11 \\ 15 & 11 & 7 & 128 \\ 1 & 1 & 7 & 218 \\ 5 & 11 & 7 & 217 \end{bmatrix} \quad B_{i_3} = \begin{bmatrix} 92 & 10 \\ 9 & 102 \\ 19 & 12 \\ 7 & 100 \end{bmatrix}$$

Mining Frequent Sequences

- The blocks B_i mentioned in the previous section is the input for the mining operation.
- Each row in B_i is considered as a single transaction and frequent sequences are mined.

A simple trace is given below:

SDB=

SID	Sequences
1	$\langle\{1\}, \{2\}, \{3\}, \{1\}\rangle$
2	$\langle\{2\}, \{3\}, \{2\}, \{1\}\rangle$
3	$\langle\{2\}, \{2\}, \{3\}, \{1\}\rangle$

1	
SID	Itemsets
1	1,4
2	4
3	4

2	
SID	Itemsets
1	2
2	1,3
3	1,2

3	
SID	Itemsets
1	2
2	2
3	3

Mining Frequent Sequences

F_1		
PID	Sequence	Support
1	$\langle\{1\}\rangle$	3
2	$\langle\{2\}\rangle$	3
3	$\langle\{3\}\rangle$	3

F_2		
PID	Sequence	Support
1	$\langle\{1\},\{1\}\rangle$	0
2	$\langle\{1\},\{2\}\rangle$	1
3	$\langle\{1\},\{3\}\rangle$	0
4	$\langle\{2\},\{1\}\rangle$	1
5	$\langle\{2\},\{2\}\rangle$	1
6	$\langle\{2\},\{3\}\rangle$	2
7	$\langle\{3\},\{1\}\rangle$	2
8	$\langle\{3\},\{2\}\rangle$	1
9	$\langle\{3\},\{3\}\rangle$	0

F_3		
PID	Sequence	Support
1	$\langle\{2\},\{3\},\{1\}\rangle$	2
2	$\langle\{2\},\{3\},\{2\}\rangle$	1
3	$\langle\{2\},\{3\},\{3\}\rangle$	0
4	$\langle\{3\},\{1\},\{2\}\rangle$	0
5	$\langle\{3\},\{1\},\{2\}\rangle$	0
6	$\langle\{3\},\{1\},\{3\}\rangle$	0

Mining Frequent Sequences

F_4		
PID	Seq	α
1	$\langle\{2\},\{3\},\{1\},\{1\}\rangle$	0
2	$\langle\{2\},\{3\},\{1\},\{2\}\rangle$	0
3	$\langle\{2\},\{3\},\{1\},\{3\}\rangle$	0

PID	Seq	α
1	$\langle\{2\},\{3\}\rangle$	2
2	$\langle\{3\},\{1\}\rangle$	2
3	$\langle\{2\},\{3\},\{1\}\rangle$	2

Compressor

Input:

- ⋆ Video file V
- ⋆ Block size b
- ⋆ Minimum Support α

Output:

- ⋆ C_I, the compressed video
- ⋆ Code Table of each component CT_1, CT_2, CT_3

Pseudocode:

- ⋆ Split V_i into n_b blocks.
- ⋆ Perform Mining on each block using Frequent sequence mining approach
- ⋆ Perform Huffman encoding on sequence identifiers.

Clustering based approach to reduce Video Compression ratio

K-means Clustering

- ⋆ partitions a given set of data objects into k clusters
- ⋆ maximizes the intra cluster similarity
- ⋆ minimize the inter cluster similarity
- ⋆ Every data object is associated with the cluster whose centroid is the nearest to the object
- ⋆ Process is repeated until no more reassociations are done

Compressor

Input:

- Video file V
- Block size b, Minimum Support α
- Number of clusters k

Output:

- C_I, the compressed video
- Code Table of each component CT_1, CT_2, CT_3

Pseudocode:

- Split V_i into n_b blocks.
- Perform Clustering on each block using K-means.
- Perform Mining on Cluster Identifiers using Frequent sequence mining approach
- Perform Huffman encoding on sequence identifiers.

Observations for Approach II

Figure: Frame 195 of sample video

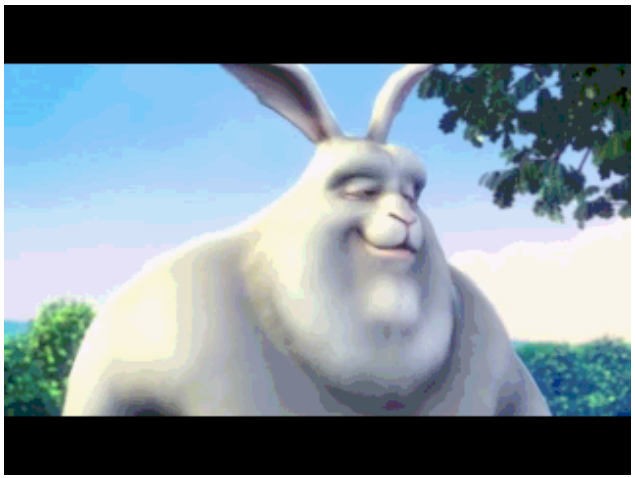


Figure: Frame 195 of reconstructed Video for k=16

Observations for Approach II

Table: Comparison of Proposed Algorithm with MPEG4 for 1.1MB test video

k	Compressed size(kB)	MPEG size(kB)	Proposed C_r	MPEG C_r	Proposed PSNR	MPEG PSNR
16	1740	332	0.605	3.171	42.54	24.11
8	1331	332	0.791	3.171	38.23	24.11

Dimensionality reduction based approach to Video Compression Ratio

Principle Component Analysis

- Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set.
- It is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components.
- The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

Compressor

Input:

- Video file V
- Block size b, Minimum Support α
- Number of clusters k

Output:

- C_I, the compressed video
- Code Table of each component CT_1, CT_2, CT_3

Pseudocode:

- Split V_i into n_b blocks.
- Perform Dimensionionality reduction using PCA on each block.
- Perform Clustering on each block's principle components.
- Perform Mining on Cluster Identifiers using Frequent sequence mining approach
- Perform Huffman encoding on sequence identifiers.

Observations for Approach III

Figure: Frame 195 of sample video

Figure: Reconstructed video of frame 195 using Approach III at pca value 90

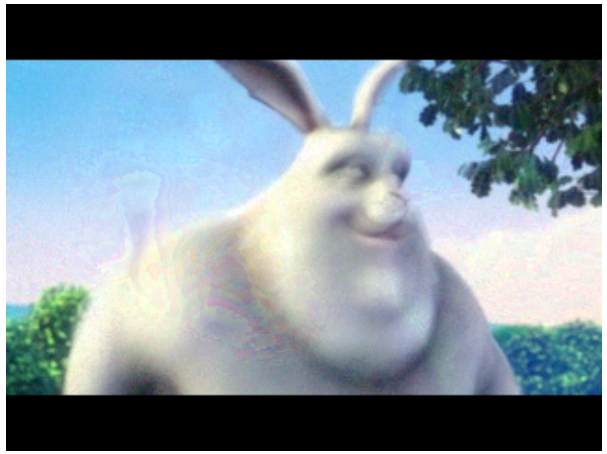


Figure: Reconstructed video of frame 195 using Approach III at pca value 96

Observations for Approach III

Table: Variation of C_r for different values of PCA, k, α for 1.1MB test video

PCA	k	α	Original size(kB)	Compressed size(kB)	C_r
90	4	50	1053.65	248.55	4.242
90	8	50	1053.65	302.11	3.487
90	16	50	1053.65	334.97	3.145
95	8	50	1053.65	506.04	2.082
95	16	50	1053.65	586.83	1.796
95	32	70	1053.65	671.325	1.57
96	32	70	1053.65	853.26	1.235
96	32	80	1053.65	855.11	1.232
96	32	40	1053.65	879.9	1.189
96	16	50	1053.65	681.29	1.546

Table: Variation of C_r for different values of PCA, k, α for 2MB test video

PCA	k	α	Original size(kB)	Compressed size(kB)	C_r
96	32	50	2100.39	1406.24	1.494
97	32	70	2100.39	2228.6	0.942
97	8	50	2100.39	1438.2	1.46
97	16	50	2100.39	1955.04	1.074

Table: PSNR value for frame 197 in 1.1MB video file

PCA	k	α	PSNR
90	16	50	30.02
95	32	70	37.08
96	32	80	36.88

Observations for Approach III

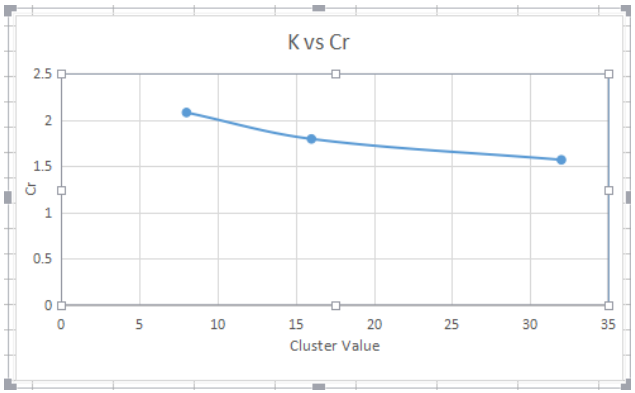


Figure: C_r vs K at pca = 95

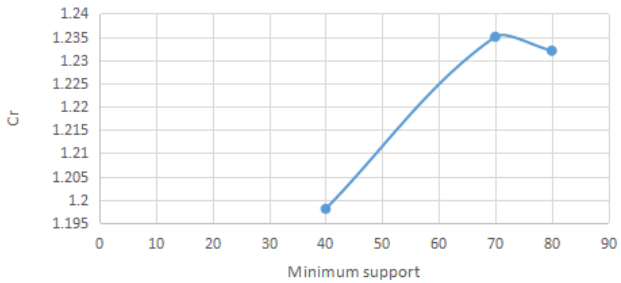


Figure: C_r vs α at pca = 96 and K = 32

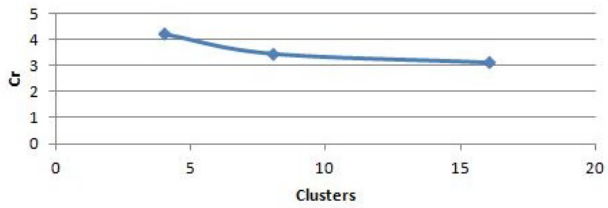


Figure: C_r vs K at pca = 90

Conclusions

- ⋆ Cluster Based mining approach achieves better quality than that of MPEG but has poor compression ratio.
- ⋆ Manageable quality and better compression ratio were observed using pca based approach.
- ⋆ In PCA based approach higher quality leads to lower compression ratio.
- ⋆ The developed GUI compression tool has high degree of usability and is platform independent and interactive.

Future Work

- ⋆ Improving compression ratio while achieving better quality
- ⋆ Aim to conduct extensive simulations on standard datasets with large size to investigate several other parameters including the compression time
- ⋆ Extend the GUI compression tool for videos

Thank you