ISBN-978-989-53228-5-5
37th ISTANBUL International Conference on "Advances in Science, Engineering & Technology" (IASET-21)
Istanbul (Tutkey) Dec. 16-17, 2021

Singular Value Decomposition on Reducing Required Storage Space of Video Representation

Aini Suri Talita[1], Dewi Anggraini Puspa Hapsari[2] and Sarifuddin Madenda[3]

[1,2,3]Gunadarma University, Indonesia

***Abstract*:** *The needs of data compression as in video compression emerge due to limitation of storage and communication bandwidth. Video is a sequence of digital images, where the digital images can be represented as matrices. One of the matrix factorization that is possible to be implemented on any matrix is Singular Value Decomposition (SVD). On low rank SVD, we choose a smaller value of rank of the new approximation image to reduce its necessary storage capacity. When we save the images in the storage by saving the new representation from SVD results other than the original representation, theoretically with smaller rank of the matrix representation will results in smaller size of file. The experiment result on this research gives a compression ratio 2.6182 on a sample frame of the video data by choosing text file format to save the necessary data.*

Keywords: *SVD, compression, video, image*

1. Introduction

The limitation of storage and communication bandwidth increases the needs of data compression, including videos. Video is a sequence of digital images called frames [1]. On [2], the Discrete Cosine Transform (DCT) coefficients during JPEG compression were modified to simultaneously compress the image and encrypt the data at the same time. The proposed method reduced number of computational steps on data compression and encryption results in increasing the performance of previous methods where compression and encryption steps were computed sequentially.

A digital image can be represented as a matrix with the entries in the form of intensity values for the corresponding pixels from which the visual image can be recovered when needed. This property can be used to implement a matrix factorization on reducing the required storage capacity. One of the matrix factorization that applies to each matrix is Singular Value Decomposition (SVD) [3]. SVD can be used to compress visual information as in digital image, with assuming some singular values are sufficiently small that dropping some terms on the expansion of matrix factorization formula for the image representation produces an acceptable approximation.

SVD was used on generating the authentication information on image blocks on [4]. Shen et al. on [4] proposed a self-embedding fragile image authentication that divided the original image into some non-overlapping blocks. The blocks then divided into upper and bottom part that were concatenated to generate the authentication code by SVD. The proposed method had a great performance that surpassed previous methods in Peak Signal to Noise Ratio (PSNR), False Negative Rate (FNR), and False Positive Rate (FPR).

In [5], Kumar et al. proposed an algorithm based on SVD and Embedded Zero Tree Wavelet (EZQ) to compress Electrocardiogram (ECG) signal. The ECG signal compression deals with huge data of ambulatory system so compression is very much needed. The low rank SVD was used specifically on compressing two dimensional (2-D) ECG data array and then continued with EZW for the final compression stage. The hybrid compression technique was divided into two stages that strengthen the performance with compression ratio on the experiment result reach 24.25:1. The value of Percentage-root-mean square difference (PRD) shows a great quality of signal reconstruction reached 1.89% for ECG signal Rec. 100.

Another work related with ECG that implementing SVD is [6]. Zheng et al. [6] utilized SVD to decompose the ECG signals and then use Support Vector Machine (SVM) and Convolutional Neural Network (CNN) as the classifier on ECG arrhythmia data. By using only 3 singular values on the SVD, the highest accuracy reached more than 96%. For the CNN architecture, the authors designed a 1-dimensional CNN with 2 convolutional layers, 2 pooling layers, and I fully connected. Stochastic Gradient Decent (SGD) method was used to train the model.

On signal processing, compressing and denoising signal is important. On [7], Schanze proposed a compression and signal denoising method that bijectively maps a signal vector into a matrix, which SVD decomposed into singular values and vectors to construct matrices with rank one. The bijective was generated such that the periodic components that present in the signal can be analysed. To determine the size of the mxn image matrix, the author tried several partitions, for different m and n obeying mn = K, where K is the number of data points, and check for an optimal component number p and data compression ratio d.

On this research, we apply SVD on a sequence of frames extracted from an mp.4 video in order to reduce the storage capacity needed to save the data.

2. Singular Value Decomposition

One of the matrix factorization that possible to be implemented on any matrix is SVD [3]. On SVD, an mxn matrix was decomposed to be three matrices U, S, and V. If A is an mxn matrix, then A can be factored as given on (1).

$$A = USV^T \tag{1}$$

where $U = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \dots\ \boldsymbol{u}_k | \boldsymbol{u}_{k+1} \dots \boldsymbol{u}_m]$, $S = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \\ 0 & \sigma_2 & \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots & \boldsymbol{0}_{k\times(n-k)} \\ 0 & 0 & \dots & \sigma_k & \\ & & \boldsymbol{0}_{(m-k)\times k} & & \boldsymbol{0}_{(m-k)\times(n-k)} \end{bmatrix}$, and $V^T = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_k^T \\ \boldsymbol{v}_k^T \\ \boldsymbol{v}_{k+1}^T \\ \boldsymbol{v}_n^T \end{bmatrix}$

with the sizes of U, S and V are mxm, mxn, and nxn respectively. V has the property that it orthogonally diagonalizes A^TA. The non-zero entry on the main diagonal of S is the square root of the non-zero eigen values of A^TA corresponding with the column vectors of V.

The column vectors of V are ordered such that the singular values are in decreasing order, that is $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$. The column vectors of matrix U has property $\boldsymbol{u}_i = \frac{A\boldsymbol{v}_i}{\|A\boldsymbol{v}_i\|} = \frac{1}{\sigma_i} A\boldsymbol{v}_i$, for i= 1, 2, …, k. While $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k\}$ form an orthonormal basis of the column space of A and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \dots, \boldsymbol{u}_m\}$ is an extension of $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k\}$ that form an orthonormal basis of R^m.

Algebraically, on (1), the zero rows and zero columns on matrix S can be ignored since at end of the matrix multiplication, the results are 0. So after deleting the zero blocks on USV factorization, we have a new form of decomposition, reduced singular value decomposition.

On reduced singular value decomposition, the size of each matrices U, S, and V^T is mxk, kxk, kxn respectively. After we remove the block zeros on (1), then A can be represented as in (2) where k is the rank of A, that is the shared dimension of row and column space of A.

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T \tag{2}$$

Assuming that $\sigma_{r+1}, \ldots, \sigma_k$ is small enough so that deleting the corresponding terms on (2) results in acceptable approximation of A, we have another representation of A, its rank r approximation of A (3).

$$A_r = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T \tag{3}$$

When we represent a mxn matrix A with rank k, we need storage with mxn size (we save each entry of the matrix), but if we use rank r representation as in (3), the storage needed becomes:

$$rm + rn + r = r(m + n + 1) \tag{4}$$

for saving values and vectors σ, $\boldsymbol{u}$, and $\boldsymbol{v}$, where $\boldsymbol{u}$ has m entries and $\boldsymbol{v}$ has n entries with assuming $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$.

On this research, the data are color images that we extract from a digital video, which is represented by a tensor of size mxnx3, where 3 is the RGB component of the color image. Without reducing generality, for example, calculated on a component, it takes mxn space allocations to store entries from the image representation matrix. By implementing SVD, the storage needed becomes (4), with "r" is the chosen value for rank of the new image approximation.

3. Results And Discussions

On this research SVD was implemented of an mp.4 video with 1 second duration and size 448 kb. The software that was used is Matlab. The video consist of 47 frames, where the sample the 12th frame is given on Fig. 1. The size of the image on Fig 1. is 661 kb, with dimension 352x640x3.

Fig. 1: Frame -12 of the video

We performed SVD on 3 R, G, B components of the image. For simplicity on the naming of the output file, "A" represent the file to save the original entries of the image, with the number following "1" represents Red component, "2" represents Green components, and "3" represents Blue components. We save them on two different types of files ".mat" which is standard output file on Matlab, and ".txt" text file. "B" representing the entries needed to be save on representation of rank "r" approximation of A, as in (3), the entries are belong to singular value σ's, and vectors/matrix **u**'s and **v**'s. While "C" was used to save the entries on the reconstructed image after we choose the new approximation rank "r". The example of the SVD results implemented on 12th frame of the video data are given on Table 1 and 2.

TABLE I: The Size of Matlab Output Files to Save SVD Results on Frame -12

Each component (Kb)	Total (kb)
A1.mat = 62	
A2.mat = 62	A = 185
A3.mat = 62	
B1.mat = 224	
B2.mat = 224	B = 672
B3. mat = 224	
C1.mat = 1618	
C2.mat = 1618	C = 4857
C3. mat = 1621	

On Table 1, we can see that the size on file B is still larger than the size on file A, so SVD does not help in reducing the storage space needed if we choose to use Matlab output file to save the data. But we can see on Table 2, if we use text file to save the output, the size of file B is smaller than A, which is the point of the data compression. It is possible due to when we save as Matlab ouput file, the history of computation to get the values are also saved by Matlab. While on text file, it is purely contains the entry values that we save. We do not need the history of SVD computation to generate the constructed image (file C), so the more efficient way is by using text file. Based on Table 2, for the 12th frame as an example for the 47 frames of the video data, the compression ratio is 2215/846 = 2.6182.

TABLE II: The Size of Text Files to Save SVD Results on Frame -12

Each component (Kb)	Total (kb)
A1.mat = 744	
A2.mat = 740	A = 2215
A3.mat = 731	
B1.mat = 282	
B2.mat = 282	B = 846
B3. mat = 282	
C1.mat = 1526	
C2.mat = 1522	C = 4576
C3. mat = 1528	

4. Conclusions

On this paper, we implemented Singular Value Decomposition (SVD) to reduce storage space of a sequence of digital images extracted on a mp.4 video. When choosing the smaller rank for the approximation of the image that represent by a matrix for each of its R, G, and B component, we can see that by SVD, on the sample frame, 12th frame, its compression ratio reach 2.6182.

5. Acknowledgement

This work supported by Kemendikbudristek Indonesia in Hibah Penelitian Disertasi Contract Number: 064/E4.1/AK.04.PT/2021 dated July 12, 2021 and Yayasan Pendidikan Gunadarma Jakarta Indonesia, Contract Number: 09.25/LP/UG/VII/2021 dated July 13, 2021.

References

[1] S. Matzner, V.I. Cullinan, and C.A. Duberstein, “Two-dimensional thermal video analysis of offshore bird and bat flight,” *Ecological Informatics*, vol. 30, pp. 20-28, 2015.

https://doi.org/10.1016/j.ecoinf.2015.09.001

[2] S. Maqbool, N. Ahmad, A. Muhammad, and A.M.M. Enriquez, “Simulataneous encryption and compression of digital images based on secure-JPEG encoding,” *J.F. Martinez-Trinidad et al. (Eds.): MCPR* 2016, LNCS 9703, pp. 145-154, 2016. DOI: 10.1007/978-3-319-39393-3_15.

[3] H. Anton and C. Rorres, *Elementary Linear Algebra*, 11th ed. USA: Wiley, 2014, ch. 9, pp. 521-524.

[4] J. J. Shen, C.F. Lee, F.W. Hsu, and S. Agrawal, “A self-embedding fragile image authentication based on singular value decomposition,” *Multimedia Tools and Applications*, vol. 79, pp. 25969-25988, 2020.

https://doi.org/10.1007/s11042-020-09254-1

[5] R. Kumar, A. Kumar, and G.K. Singh, “Hybrid method based on singular value decomposition and embedded zero tree wavelet technique for ECG signal compression,” *Computer Method and Programs in Biomedicine*, vol. 129, pp. 135-148, 2016.

https://doi.org/10.1016/j.cmpb.2016.01.006

[6] L. Zheng, Z. Wang, J. liang, S. Luo, and S. Tian, “Effective compression and classification of ECG arrhythmia by singular value decomposition,” *Biomedical Engineering Advances*, vol. 2, 100013, 2021.

https://doi.org/10.1016/j.bea.2021.100013

[7] T. Schanze, “Compression and noise reduction of biomedical signals by singular value decomposition,” *IFAC*, vol. 51-2, pp. 361-366, 2018.

https://doi.org/10.1016/j.ifacol.2018.03.062