Singular Value Decomposition for Video Background Extraction

Ishan Khare
Stanford University
MATH 104 — Applied Matrix Theory
iskhare@stanford.edu

1. Introduction

Background extraction from a video can be an important tool for various cinematographic applications. For example, background extraction can be utilized to generate backdrops that were previously unattainable due to obstructions in the foreground. This work seeks to take a short video as input and extract the video's background as a single image. This image can then be utilized for a variety of purposes. One distinctive application that comes to mind is being able to obtain a photograph of a background when in real-life there is no moment when the entire background can be captured at once. For example, if someone wants to take a picture of the Taj Mahal without it having random strangers in it, one could instead take a short video of the scene and come back to extract the foreground thus resulting the initial image they sought to capture.

2. Data Acquisition

Three different `.mp4` videos were sourced from the internet for this project. These videos will simply be referred to as Video 1, Video 2, and Video 3 from here on out for the remainder of the paper. Video 1 is a small snippet from a surveillance camera with a duration of 24 seconds. This video was obtained from KairUs, an organization focused on Internet fraud and online scams [6]. Video 1 has a resolution of 720px by 1280px. The next piece of data, Video 2 falls under a Royalty Free license and also under a Creative Commons CC By license. This video was downloaded from YouTube [2]. Video 2 has a duration of 46 seconds and a resolution of 1080px by 1920px. The third video was taken from a 2009 publication on tracking and surveillance [5]. Video 3 has a duration of 50 seconds and a resolution of 240px by 320px. This combination was chosen to ensure the methodology was robust for a greater variety of input videos.

3. Mathematics and Methodology

In the examples for this project, the background of each video is essentially static, *i.e.*, it does not see a lot of movement. Movement takes place in the foreground of the video. In every one of the example videos, the movement in the foreground is people walking through the camera frame. One thing to note is that there is not a single frame in any of the videos where the foreground is not there. That is why extracting the background is a harder task than simply pausing the video when there are no objects in the foreground obstructing the view of the background. This does not occur so we must utilize the techniques proposed here within. An alternative solution is to take screen captures of various parts of the video and manually reconstruct the final background image. However, this project creates a more automated process using linear algebra techniques of low-rank approximations

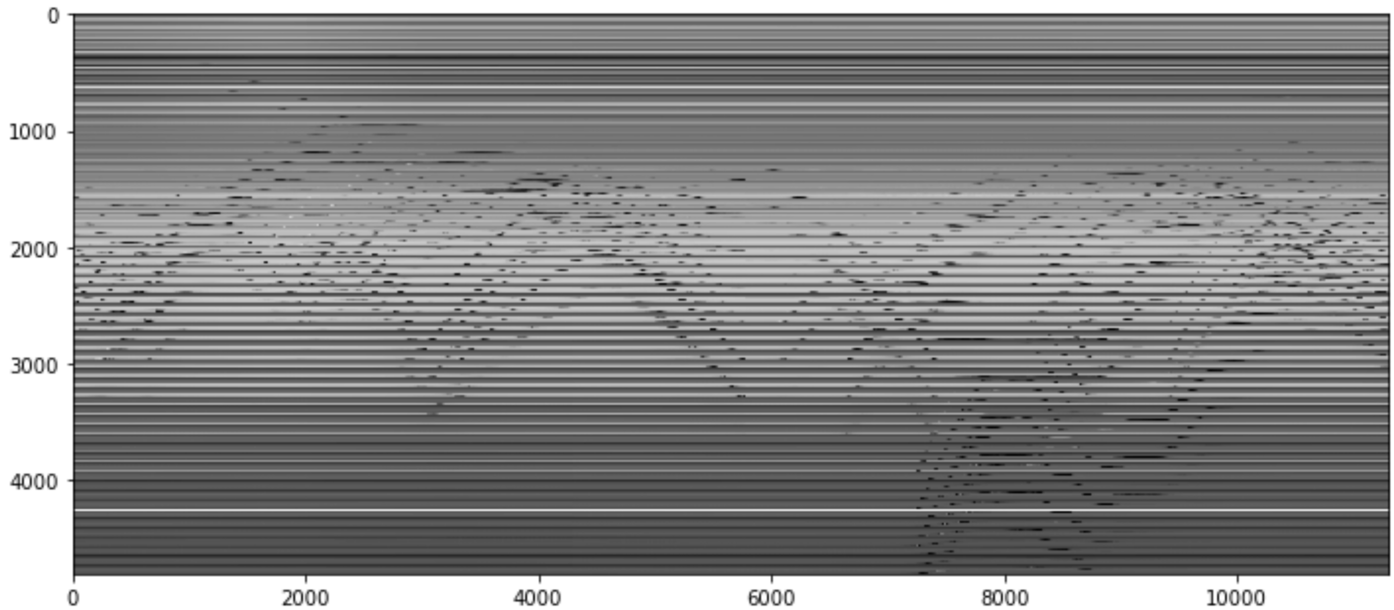

Figure 1. This plot depicts Matrix M which is constructed by randomly sampling frames from Video 3 and flattening these image matrices into arrays. The horizontal lines represent pixel values that do not change over the duration of the video, and the wavy lines represent movement in the foreground.

and singular value decomposition.

In particular, we understand that in order to separate the background from the foreground we must exploit the fact that the background is more or less stationary. First, we construct a matrix M from the input video using the `Open-CV` library [1]. The demonstration in this paper will use Video 3. This matrix M is done by randomly and uniformly sampling frames from the video are regular intervals. These image matrices are then flattened into arrays and are stored as the columns of matrix M. Constructing this matrix for Video 3 gives us the plot shown in Fig. 1. The horizontal lines in Fig. 1 represent pixel values that are constant throughout the video and hence generally correspond with the background. On the other hand, the wavy lines depict movement and thus correspond with the foreground.

This matrix M can be represented as the sum of two matrices: $M = B + F$, where matrix B which is the background and matrix F which is the foreground. Since background matrix B does not have a large variation in pixels, it contains lots of redundancy, meaning there is not a significant amount of unique information. We hypothesize that B is therefore a low-rank matrix. Now, we can simply use SVD or singular value decomposition in this step.

Singular value decomposition deals with decomposing a matrix into the product of 3 matrices as shown: $B = USV^T$. Suppose that matrix B has dimensions $m \times n$. Then, the corresponding matrices will be $U \in \mathbb{R}^{m\times m}$, $S \in \mathbb{R}^{m\times n}$, and $V \in \mathbb{R}^{n\times n}$. Matrix U will be a matrix of the left singular vectors. Moreover, matrix S will be rectangular diagonal matrix of the singular values arranged in a sorted manner with the largest values in the upper left corner. Finally, matrix V will be the matrix of the right singular vectors. Why is this SVD decomposition important to us? Well, it can then be further understood that if B has rank r, then the SVD of B can be expressed as a summation of r rank-one matrices as follows: $B = \sigma_1 \vec{u}_1 \vec{v}_1^\top + \cdots + \sigma_r \vec{u}_r \vec{v}_r^\top$, where $\vec{u}_i$ and $\vec{v}_i$ are respectively the ith columns of U and V. Additionally, σ_i is the ith singular value in matrix S.

Now, we can employ the matrix approximation lemma also know as the Eckart-Young-Mirsky theorem [4]. This theorem states that a best rank-k approximating matrix for B is $B_k = U_k S_k V_k^\top$. Each of the matrices U_k, S_k, and V_k can easily be obtained by retaining only the first k columns of U and V, and exactly the first k singular values of S. This approximation B_k is obtained by computing a matrix that minimizes $||B - B_k||$. Applying Eckart-Young to matrix B essentially gives us B_k, which we use to obtain the final background image which is output by the program. For the purposes of this project, this is a low-rank approximation generally rank-two to be precise.

4. Results and Analysis

First, we begin by extracting the background from Video 1. Since we are unable to show a video in a PDF format, a few sample frames from Video 1 are shown in Fig. 2. We then sample 50 Frames randomly from the video and do this three times to more accurately obtain the extracted background we so desire. Next, Fig. 3 shows the extracted background for Video 1. Upon closer inspection it's clear there has been some foreground extraction to obtain the background. This remnant smearing is a result of noise even after the low-rank approximation. Although this work was not able to completely remove the smearing the sampling size was altered to minimize smearing.

Due to the sake of verbosity and repetition, no further explanations will be given for Videos 2 and 3. Some frames that were sampled from Video 2 are given in Fig. 4 with the corresponding background output in Fig. 5. Similarly, a small subset of frames sampled from Video 3 is given in Fig. 6 with the corresponding background output in Fig. 7.

Figure 2. This figure depicts four frames sampled from Video 1. As is evident, all frames have people in the foreground that obstruct the background we want to extract.

Figure 3. This figure depicts the extracted background from Video 1. Notice the smearing that has occurred as a result of the moving subjects from the foreground.

Figure 4. This figure depicts four frames sampled from Video 2. As is evident, all frames have people in the foreground that obstruct the background we want to extract.

Figure 5. This figure depicts the extracted background from Video 2. Notice the smearing that has occurred as a result of the moving subjects from the foreground.

Figure 6. This figure depicts four frames sampled from Video 3. As is evident, all frames have people in the foreground that obstruct the background we want to extract.

Figure 7. This figure depicts the extracted background from Video 3. Notice the smearing that has occurred as a result of the moving subjects from the foreground.

5. Future Work

In the future it would be of interest to not only extract the background as an image but also to extract the foreground. Additionally, it would be of significant progress to create two output videos from each input: the first output would be the same video with the background removed and the second video would be a video with the foreground removed. This contribution, would allow for a greater amount of use cases in the video editing industry at large.

Another hypothesis that arises is that there might be a manner in which SVD could be applied locally to focus in on the extraction of certain parts of the background more clearly. These local SVD results could then be stitched together for a more complete and high-quality background image retrieval.

Some of the produced background images are also rather low-light and of low quality. To improve the quality of this background image, we could additionally utilize image recovery through the concept of matrix completion. In particular, we can utilize low-rank approximations to restore image quality as demonstrated in the paper by Chen [3].

6. Acknowledgements

I would like to thank Prof. Gene B. Kim and Stanford University to making a tremendous contribution to my love of learning – more specifically towards linear algebra. In addition, I thank my parents for their support throughout my educational journal thus far.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 2

[2] Brollstock.com. People walking past the camera - free stock footage for commerical projects. https://www.youtube.com/watch?v=3FXUw98rrUY/, 2016. 1

[3] Z. Chen. Singular value decomposition and its applications in image processing. *Proceedings of the 2018 1st International Conference on Mathematics and Statistics*, 2018. 5

[4] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936. 3

[5] J. Ferryman and A. Shahrokmi. An overview of the pets2009 challenge. *IEEE International Workshop on Performance Evaluation of Trackig and Surveillance*, pages 25–30, 2009. 1

[6] Linda Kronman and Andreas Zingerle. Suspicious behavior (2020). https://kairus.org/portfolio/suspicious-behavior-2020/, 2020. 1

Appendix A

The codebase for this project can be found at the following link: https://github.com/iskhare/math104-bg-extraction.

A useful note is that much of the code focuses on utilizing library functions that already exist to perform SVD, low-rank approximation, flattening of data into vectors and more. Namely, the code is written in Python 3.10 and relies on the `numpy` and `Open-CV` libraries. The `Open-CV` library in particular was crucial to development as it provides a real-time optimized Computer Vision library, tools, and hardware. Much of the code that would have be re-written to sample frames and work with video input was thus directly utilized from `Open-CV`. Overall, a good chunk of time went into the development of this project – definitely more than any individual assignment for this class.