Assignment 3, Mathematics for IT
Oct 26, 2020

Background Removal from video using Singular Value Decomposition

Kumar Mayank (MIT2020013) Rishabha Sachan (MIT2020014) Vaibhav Sharma (MIT2020015)
GROUP-5
SEMESTER 1
M.Tech IT (2020-22)
Department of Information technology
Indian institute of information technology, Allahabad

Abstract:

Separating background from a video image is an important step for change detection. Most of the change detection methods depend on intensity and texture variations. These algorithms may not give very satisfactory results because of the illumination variation and presence of noise. In this paper we have attempted to segregate moving foreground from video data set using singular value decomposition (SVD) which is a generalization of the Eigen decomposition which can be used to analyze rectangular matrices. 'CDnet 2012', an open source dataset is used as an input image dataset. We have taken SVD for all input images and singular values of all these images were used for detecting any change present in images. A comparison was made for obtained results with the results of 'CDnet 2012'. It is found that the obtained results are very effective and clearly segregate foreground.

This report is contributed by Kumar Mayank (MIT2020013), Rishabha Sachan (MIT2020014) and Vaibhav Sharma (MIT2020015).

Keywords : Background elimination, Background Subtraction, Singular Value Decomposition, SVD, EigenValues, CDnet 2012.

Introduction:

In linear algebra, the singular value decomposition (SVD) is a factorization of real or complex matrix that generalizes the eigen decomposition of a square normal matrix to any m*n matrix.
Specifically, the singular value decomposition of a real or complex matrix is a factorization of the form USV^T
where U is an m*m unitary matrix
S is an m*n rectangular diagonal matrix with non-negative real numbers on the diagonal
and V is an n*n unitary matrix.

The Singular-Value Decomposition, or SVD for short, is a matrix decomposition method for reducing a matrix to its constituent parts in order to make certain subsequent matrix calculations simpler.

Prior work:

The singular value decomposition was originally developed by differential geometers, who wished to determine whether a real bilinear form could be made equal to another by independent orthogonal transformations of the two spaces it acts on.

Eugenio Beltrami and Camille Jordan discovered independently that the singular values of the bilinear forms, represented as a matrix, form a complete set of invariants for bilinear forms under orthogonal substitutions.

James Joseph Sylvester called the singular values the canonical multipliers of the matrix A.

The first proof of the singular value decomposition for rectangular and complex matrices seems to be by Carl Eckart and Gale J.Young ; they saw it as a generalization of the principal axis transformation for Hermitian matrices.

Erhard Schmidt defined an analog of singular values for integral operators (which are compact, under some weak technical assumptions); it seems he was unaware of the parallel work on singular values of finite matrices.

Background:

The singular value decomposition of a matrix A is the factorization of A into the product of three matrices A = UDV T where the columns of U and V are orthonormal and the matrix D is diagonal with positive real entries. The SVD is useful in many tasks.

First, in many applications, the data matrix A is close to a matrix of low rank and it is useful to find a low rank matrix which is a good approximation to the data matrix. From the singular value decomposition of A, we can get the matrix B of rank k which best approximates A; in fact we can do this for every k.

Also, singular value decomposition is defined for all matrices (rectangular or square) unlike the more commonly used spectral decomposition in Linear Algebra. With eigenvectors and eigenvalue, we need conditions on the matrix to ensure orthogonality of eigenvectors.

In contrast, the columns of V in the singular value decomposition, called the right singular vectors of A, always form an orthogonal set with no assumptions on A. The columns of U are called the left singular vectors and they also form an orthogonal set. A simple consequence of the orthogonality is that for a square and invertible matrix A, the inverse of A is $VD^{-1}UT$.

SVD

A is any m by n matrix, square or rectangular. Its rank is r. We will diagonalize this A, but not by X^{-1} AX. The eigenvectors in X have three big problems: They are usually not orthogonal, there are not always enough eigenvectors, and Ax = λx requires A to be a square matrix. The singular vectors of A solve all those problems in a perfect way.

We have two sets of singular vectors, u's and v's. The u's are in Rm and the v's are in Rn. They will be the columns of an m by m matrix U and an n by n matrix V . SVD will be first described in terms of those basis vectors and then in terms of the orthogonal matrices U and V .

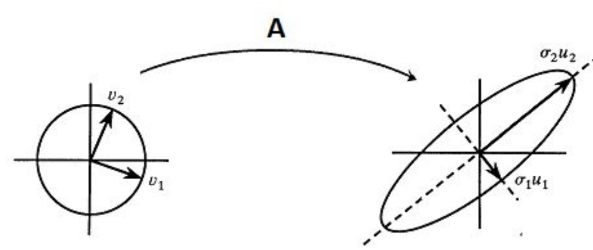


Fig. 1:

On the left side we have a set of vectors (V1, V2, V3,..., Vn) represented in 2-D, contained under a sphere. When V is multiplied by.A (m x n matrix) out vector V is rotated and then stretched, which gives us an eclipse.

Here U1 and U2 are unit vectors which tells us which direction and ***σ***1 and ***σ***2 are singular values which tell us how much in that direction.

$$Av_1 = \sigma_1 u_1 \qquad , \qquad Av_2 = \sigma_2 u_2$$

Those singular values σ1 to σr will be positive numbers : σi is the length of Avi. The σ's go into a diagonal matrix that is otherwise zero. That matrix is Σ.

(using matrices) Since the u's are orthonormal, the matrix U with those r columns has $U^T U$ = I. Since the v's are orthonormal, the matrix V has $V^T V$ = I. Then the equations Avi = σiui tell us column by column that

AVr = UrΣr

$$\begin{array}{c} (m \text{ by } n)(n \text{ by } r) \\ \boldsymbol{AV_r = U_r \Sigma_r} \\ (m \text{ by } r)(r \text{ by } r) \end{array} \quad A\begin{bmatrix} v_1 & \cdot\cdot & v_r \end{bmatrix} = \begin{bmatrix} u_1 & \cdot\cdot & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}$$

We still have

AV = UΣ

Multiplying both sides by V^{-1} , we get

$AVV^{-1} = V^{-1}U\Sigma$

$AI = U\Sigma V^T$

$A = U\Sigma V^T$

This is the equation for reduced singular values decomposition.

SVD will divide A into two unitary matrices which are orthogonal in nature and one diagonal matrix of singular values.

$$\boldsymbol{A = U\Sigma V^T = u_1\sigma_1 v_1^T + \cdots + u_r\sigma_r v_r^T}.$$

When we put the singular values in descending order

σ1 ≥ σ2 ≥ . . . σr > 0,

Theorem: Every matrix A (m x n matrix) has singular value Decomposition.

- Singular values {***σ***j} are uniquely determined and if A square ***σ***j are distinct.
- {Uj} and {Vj} are also unique up to a complex sign.

$A = U\Sigma V^T$

(rotate.stretch.rotate)

$A^T A = (U\Sigma V^T)^T (U\Sigma V^T)$

$A^T A = V\Sigma U^T U\Sigma V^T$

$A^T A = V\Sigma^2 V^T$

Multiplying both side by V

$A^T AV = V\Sigma^2 V^T V$

$A^T AV = V\Sigma^2 V^T V$

$A^T AV = V\Sigma^2$ (solve for V and ***σ***)

This is similar to $Ax = \lambda x$

Again,

$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T$

$AA^T = U\Sigma V^T V\Sigma V^T$

$AA^T = U\Sigma^2 U^T$

Multiplying both side by U

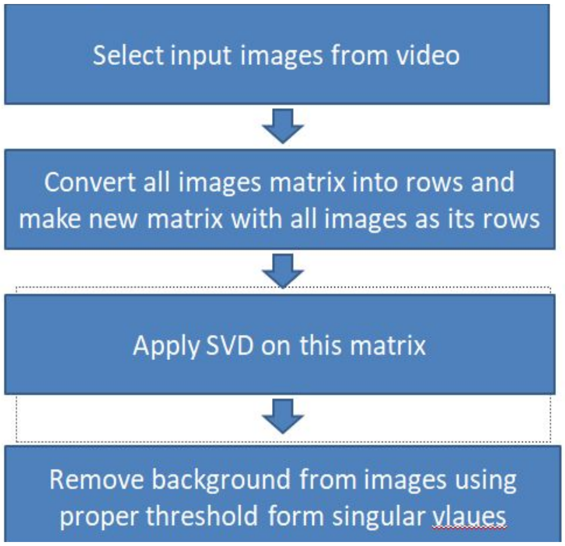
$AA^T U = U\Sigma^2 U^T U$

$AA^T U = U\Sigma^2$ (solve for U and ***σ***)

Matrix AA^T is self adjoint or hermitian matrix which guarantees eigenvalue are real, positive and distinct.

Methodology:

We have taken "CDnet 2012" open source data as our input images. This data set contains a lot of images of different categories. For our experiment we have chosen "highway" datasets. The input data set is selected in such a way that few images will have no moving or changing objects and few image frames will have non-static objects. SVD factorizes a matrix and provides its singular values. The input image matrices are converted into an array and all of these arrays are arranged in a matrix in such a way that all image vectors will be in row. Now a singular value decomposition algorithm is applied on this matrix of images and all singular values are arranged in decreasing order. based on the singular values, we will decide the threshold for segregating foreground from image. These steps are explained in below flow diagram.[2]



Results:

We have used “CDnet 2012” data for our experiment. We have taken 9 images for our experiment from video frames in such a way that in few frames there is no moving object and in other few frames some changes are present. As explained in previous section that SVD provides the Eigen decomposition of input matrix. Following are the input images we have chosen for our experiment –

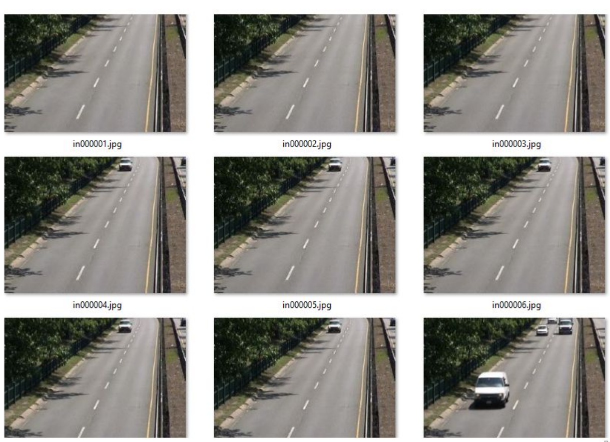


Figure 1 In this figure the first 8 frames are having a static background but in the last frame there is a car.

We have performed SVD for above images and following singular matrix was obtained

Table1

Figure 1 × Figure 2 × Figure 3 × S ×

9×9 double

	1	2	3	4	5	6	7	8	9
1	397.2218	0	0	0	0	0	0	0	0
2	0	27.3866	0	0	0	0	0	0	0
3	0	0	10.4617	0	0	0	0	0	0
4	0	0	0	5.4777	0	0	0	0	0
5	0	0	0	0	3.3746	0	0	0	0
6	0	0	0	0	0	2.6992	0	0	0
7	0	0	0	0	0	0	2.5387	0	0
8	0	0	0	0	0	0	0	2.4339	0
9	0	0	0	0	0	0	0	0	2.4020

From the above plot we can select the first singular value as our background and remaining singular values as our foreground. The processors is explained below-

If A is our input matrix then [3]

$$\boxed{A = USV^T}$$

U is an m × n orthogonal matrix which is made from Eigenvectors of AAT, S is an n × n diagonal matrix with Eigenvalues of AATas diagonal elements in decreasing order and V is an n × n orthogonal matrix which is made from Eigenvectors of ATA.[4] Now, from the above values, for selecting background, we have taken only first singular value. The following equation is used in MATLAB code-

A_back = U(:,1:1) .* S(1:1,1:1) .* V(1:1,:)’;

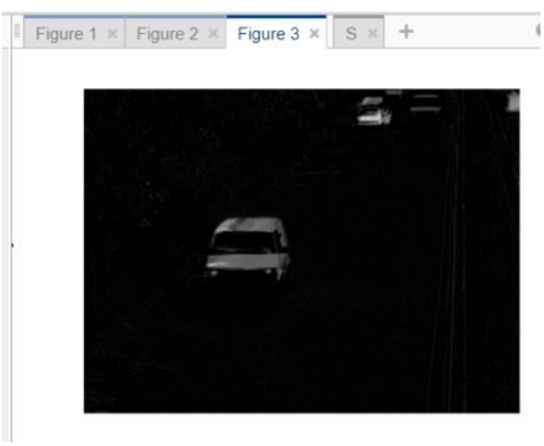
Similarly for foreground we have chosen remaining singular values,

A_fore = U(:,2:9) * S(2:9,2:9) * V(2:9,:)’;

Now, A_back will provide a background image of all frames and A_fore will provide foreground images of all the frames. Following is the segregated background and foreground for the 9th frame of input data-

a

b



c

The first image (a) is the original image and the second image (b) is the background image and third image (c) is the foreground image.

Conclusion:

Many techniques for background segregation and change detection from images are already established. We have attempted to exploit the utility of singular value decomposition to detect the change in any time series image. In our experiment we have implemented SVD using MATLAB software. It can be seen that the results obtained are very good and can be compared with any established change detection and background segregation algorithm. Our technique worked well with the datasets we have chosen. It generates singular vectors for the equivalent to the number of input images. It has been observed from the singular values that only the first singular value provides the background information in all cases. This technique is very fast compared to other established techniques for background segregation from time series data sets.

References:

[1] G.Strang, “The Singular Value Decomposition”, in Introduction to Linear Algebra, 5th edition, 2016

[2] Priyanshi Sachan, Pooja Khanna Amity School Of Engineering ,lucknow,India, Foreground Segmentation and Change Detection Using Singular Value Decomposition,

[3]SingularValuedecomposition,Wolfram MathWorldhttp://mathworld.wolfram.com/SingularValueDecomposition.html

[4] Lili Guo1, Dan Xu, ZhenpingQiang, School of Information and Engineering, Yunnan University. Background Subtraction using Local SVD Binary Pattern.

Annexure 1

```
clear all;
clc;
%selecting input data set
datapath = uigetdir('.\train','select path of training images');

%count no. of images in dataset
D = dir(datapath);  % D is a Lx1 structure with 4 fields as: name,date,byte,isdir of all L files
present in the directory 'datapath'
imgcount = 0;
for i=1 : size(D,1)
    if not(strcmp(D(i).name,'.')|strcmp(D(i).name,'..')|strcmp(D(i).name,'Thumbs.db'))
        imgcount = imgcount + 1; % Number of all images in the training database
    end
end

%creating the image matrix X of column matrix
X = [];
for i = 1 : imgcount
    str = strcat(datapath,'/in00000',int2str(i),'.jpg');%%>>
    img = imread(str);
    img = rgb2gray(img);
    img= im2double(img);%**
    [r, c] = size(img);
    temp = reshape(img',r*c,1);
    X = [X temp];
end

%converting column to row matrix
X=X';

%calculating SVD of X
[U,S,V] = svd(X,'econ');

%background and foreground calculation
vt= V';
A_back = U(:,1:1) .* S(1:1,1:1) .* vt(1:1,:);
A_fore = U(:,2:9) * S(2:9,2:9) * vt(2:9,:);

% original, background and foreground image of 9th image
i_org = reshape(X(9,:), c, r);
figure; imshow(i_org');
```

```
i_back = reshape(A_back(9,:),c,r);
figure; imshow(i_back');
i_fore = reshape(A_fore(9,:),c,r);
figure; imshow(i_fore');
```

Output

Figure 1 | Figure 2 | Figure 3 | S

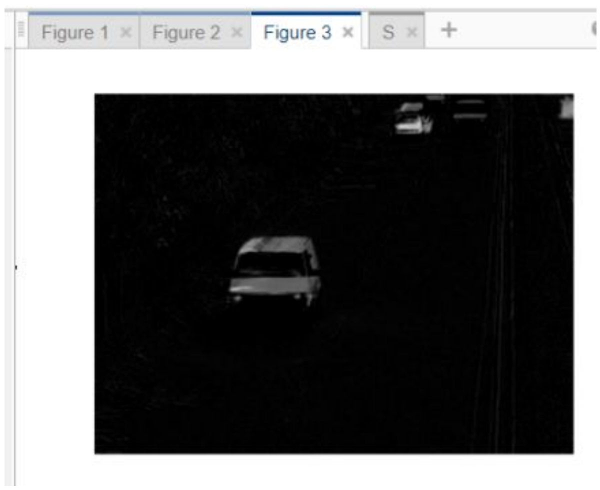
9×9 double

	1	2	3	4	5	6	7	8	9
1	397.2218	0	0	0	0	0	0	0	0
2	0	27.3866	0	0	0	0	0	0	0
3	0	0	10.4617	0	0	0	0	0	0
4	0	0	0	5.4777	0	0	0	0	0
5	0	0	0	0	3.3746	0	0	0	0
6	0	0	0	0	0	2.6992	0	0	0
7	0	0	0	0	0	0	2.5387	0	0
8	0	0	0	0	0	0	0	2.4339	0
9	0	0	0	0	0	0	0	0	2.4020

Singular values matrix of dataset

a: Original 9th image of data set

b: background image wrt 9th image



c: foreground image wrt 9th image