

# Singular Value Decomposition (SVD) for Image Processing

AM 205

Jovana Andrejevic and Catherine Ding

September 29, 2021

## Table of Contents

Motivation

Definition of SVD

Full and reduced forms

Low-Rank Approximation

Compression

Denoising

Connection to PCA

More reconstruction applications

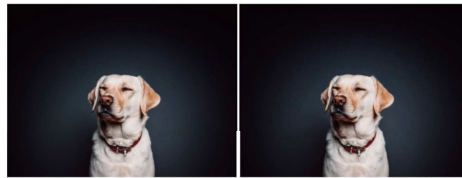
Steganography

3D Reconstruction

## Motivation

Singular Value Decomposition (SVD) has been applied in a wide range of fields:

- ▶ Computer vision: image compression and denoising



Original (28KB)

Lossy Compression (14KB, 50%)

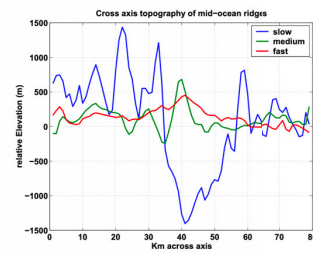
- ▶ Computer vision: steganography



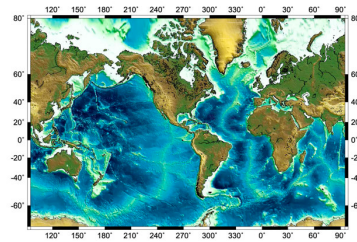
image credits: google image

## Motivation

- Scientific computing: 3D reconstruction



(a) Topography data



(b) 3D reconstruction

image credits: <http://www.columbia.edu/>

## Motivation

- ▶ Machine learning: feature extraction



image credits: <https://mathematicaforprediction.wordpress.com/>

## Singular Value Decomposition

The SVD of a matrix  $A \in \mathbb{R}^{m \times n}$  is a factorization  $A = \hat{U} \hat{\Sigma} V^T$  where

- ▶  $\hat{\Sigma} \in \mathbb{R}^{n \times n}$  is a diagonal matrix of **singular values** sorted in descending order,  $\sigma_1 \geq \sigma_2 \geq \dots \sigma_n$
- ▶  $\hat{U} \in \mathbb{R}^{m \times n}$  has orthonormal columns - **left singular vectors**
- ▶  $V \in \mathbb{R}^{n \times n}$  has orthonormal columns - **right singular vectors**

## Singular Value Decomposition

In applications, we will often think of  $A$  as a tall, thin matrix, representing relatively few  $n$  samples in a high  $m$ -dimensional space, though the SVD is defined for *any* matrix. For  $m > n$ , the columns of  $\hat{U}$  can be padded with  $m - n$  arbitrary orthonormal vectors to obtain a full  $m \times m$  matrix  $U$ , and  $\hat{\Sigma}$  padded with rows of zeros to null the contribution of these columns.

The diagram illustrates the Singular Value Decomposition (SVD) equation:  $A = U \Sigma V^T$ . It features three blue rectangular boxes representing matrices. The first box on the left is labeled  $A$  and is tall and thin. An equals sign follows it. The second box is labeled  $U$  and is also tall and thin. To its right is a third box labeled  $V^T$ , which is shorter and wider. A diagonal line connects the right side of the  $U$  box to the left side of the  $V^T$  box. Above the  $U$  box is the label  $\hat{U}$ , and above the  $V^T$  box is the label  $\hat{\Sigma}$ .

## Singular Value Decomposition

Reduced SVD of  $A$ :

$$\begin{matrix} A \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix} = \begin{matrix} U \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix} \begin{matrix} \Sigma \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix} \begin{matrix} V^T \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix}$$

The diagram shows the reduced SVD of matrix  $A$  as the product of matrix  $U$ , matrix  $\Sigma$ , and matrix  $V^T$ . Matrix  $\Sigma$  is represented by a blue box, and a diagonal line with the label  $\hat{\Sigma}$  connects the  $\Sigma$  box to the  $V^T$  box.

Full SVD of  $A$ :

$$\begin{matrix} A \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix} = \begin{matrix} U \\ \left[ \begin{array}{c|c} \text{blue box} & \text{light blue box} \\ \hline U & U^\perp \end{array} \right] \end{matrix} \begin{matrix} \Sigma \\ \left[ \begin{array}{c|c} \text{blue box} & \text{light blue box} \\ \hline \text{light blue box} & 0 \end{array} \right] \end{matrix} \begin{matrix} V^T \\ \left[ \begin{array}{c} \text{blue box} \end{array} \right] \end{matrix}$$

The diagram shows the full SVD of matrix  $A$  as the product of matrix  $U$ , matrix  $\Sigma$ , and matrix  $V^T$ . Matrix  $U$  is represented by a blue box and a light blue box, with the label  $U^\perp$  below the light blue box. Matrix  $\Sigma$  is represented by a blue box and a light blue box, with the label  $0$  below the light blue box. A diagonal line with the label  $\hat{\Sigma}$  connects the  $\Sigma$  box to the  $V^T$  box.



## Singular Value Decomposition

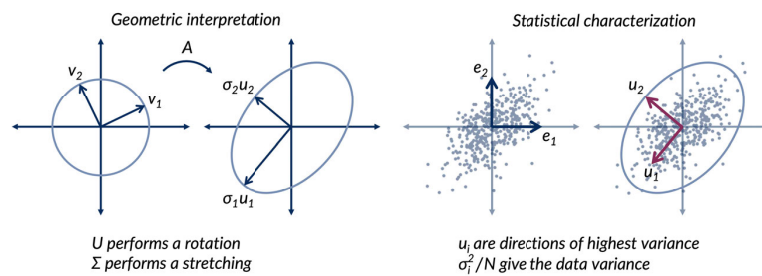
In Python:

```
1 import numpy as np
2 A = np.random.rand(20, 5)
3 U, s, Vt = np.linalg.svd(A) # full SVD
4 U, s, Vt = np.linalg.svd(A, full_matrices=False) # reduced SVD
```

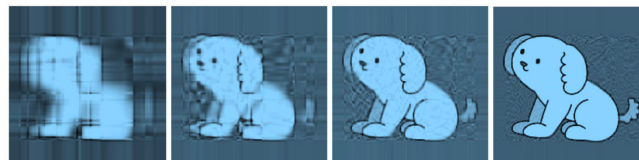
In MATLAB:

```
1 A = randn(20,5);
2 [U,S,V] = svd(A); % full SVD
3 [U,S,V] = svd(A,'econ'); % reduced SVD
```

## SVD viewed under different lenses



Optimal approximation



Compress an image:  $A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T$

## Low-Rank Approximation

The SVD provides a natural hierarchy of approximations we can make to  $A$ , expressed as a sum of rank-one matrices. If

$$A = \sum_{j=1}^n \sigma_j u_j v_j^T,$$

where each  $u_j v_j^T$  is a rank-one matrix whose columns are all scalar multiples of each other, then a **rank- $r$  approximation**  $A_r$  of  $A$  is

$$A_r = \sum_{j=1}^r \sigma_j u_j v_j^T.$$

## Low-Rank Approximation

Reduced SVD of  $A$ :

$$\begin{matrix} A \\ \left[ \begin{array}{c} \phantom{A} \end{array} \right] \end{matrix} = \begin{matrix} U \\ \left[ \begin{array}{c|c} \phantom{U} & \phantom{U} \end{array} \right] \end{matrix} \begin{matrix} \Sigma \\ \left[ \begin{array}{c} \hline \phantom{\Sigma} \hline \end{array} \right] \end{matrix} \begin{matrix} V^T \\ \left[ \begin{array}{c|c} \phantom{V^T} & \phantom{V^T} \end{array} \right] \end{matrix}$$

The diagram illustrates the reduced SVD of matrix  $A$ . Matrix  $A$  is represented by a single blue vertical rectangle. It is equal to the product of three matrices:  $U$ ,  $\Sigma$ , and  $V^T$ . Matrix  $U$  is shown as a vertical rectangle with a dark blue left portion labeled  $U_r$  and a lighter blue right portion. Matrix  $\Sigma$  is a square matrix with a dark blue top-left portion labeled  $\Sigma_r$  and a lighter blue bottom-right portion. Matrix  $V^T$  is a horizontal rectangle with a dark blue top portion labeled  $V_r^T$  and a lighter blue bottom portion. A diagonal line with arrows at both ends connects the  $\Sigma_r$  block in  $\Sigma$  to the  $U_r$  block in  $U$  and the  $V_r^T$  block in  $V^T$ .

Rank- $r$  approximation of  $A$ :

$$\begin{matrix} A_r \\ \left[ \begin{array}{c} \phantom{A_r} \end{array} \right] \end{matrix} = \begin{matrix} U_r \\ \left[ \begin{array}{c} \phantom{U_r} \end{array} \right] \end{matrix} \begin{matrix} \Sigma_r \\ \left[ \begin{array}{c} \phantom{\Sigma_r} \end{array} \right] \end{matrix} \begin{matrix} V_r^T \\ \left[ \begin{array}{c} \phantom{V_r^T} \end{array} \right] \end{matrix}$$

The diagram illustrates the rank- $r$  approximation of matrix  $A$ . Matrix  $A_r$  is represented by a single blue vertical rectangle. It is equal to the product of three matrices:  $U_r$ ,  $\Sigma_r$ , and  $V_r^T$ . Matrix  $U_r$  is a vertical rectangle with a dark blue color. Matrix  $\Sigma_r$  is a square matrix with a dark blue color. Matrix  $V_r^T$  is a horizontal rectangle with a dark blue color. A diagonal line with arrows at both ends connects the  $\Sigma_r$  block in  $\Sigma_r$  to the  $U_r$  block in  $U_r$  and the  $V_r^T$  block in  $V_r^T$ .

## Low-Rank Approximation

Note that by the orthogonality of the columns of  $u$ ,

$$u_k^T A = u_k^T \left( \sum_{j=1}^n \sigma_j u_j v_j^T \right) = \sigma_k v_k^T,$$

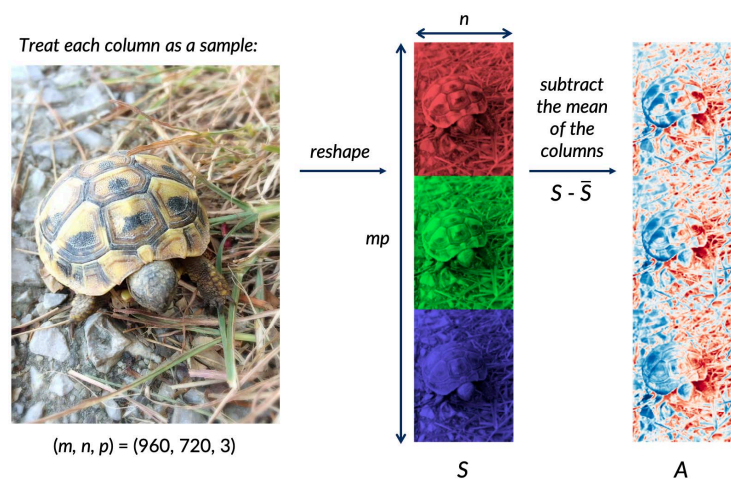
so for a particular data point  $a_i$  that is the  $i^{th}$  column of  $A$ ,

$$u_k^T a_i = \sigma_k v_{ki}^T \rightarrow (u_k^T a_i) u_k = \sigma_k u_k v_{ki}^T$$

is the **projection** of  $a_i$  onto the  $k^{th}$  left singular vector  $u_k$ . So another way to think about the low-rank approximation is that it is a sum of projections onto a limited number of left singular vectors.

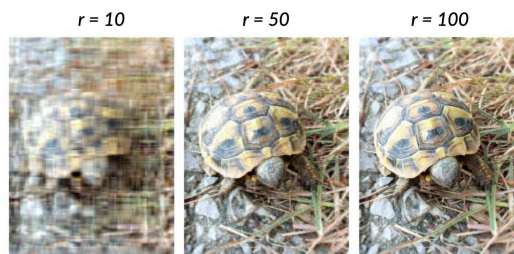
## Image Compression

The low-rank approximation gives us a useful algorithm for [compressing](#) data and images.



## Image Compression

We factor the centered  $A = S - \bar{S}$ , where  $\bar{S} = \frac{1}{n} \sum_{j=1}^n S_j$ .  
Then,  $S_r = \bar{S} + \sum_{j=1}^r \sigma_j u_j v_j^T$ .



## Image Compression

The following are two possible metrics we can use to quantify the fraction of our image reconstructed by a rank- $r$  approximation, as well as the fraction of storage space required.

- Explained variance ratio:

$$\rho(r) = \frac{\sum_{j=1}^r \sigma_j^2}{\sum_{j=1}^n \sigma_j^2}$$

- Compression ratio:

$$c(r) = \frac{r \overbrace{(1 + 3m + n)}^{\sigma, u, v^T} + \overbrace{3m}^{\bar{s}}}{\underbrace{3mn}_S}$$



## Image Denoising

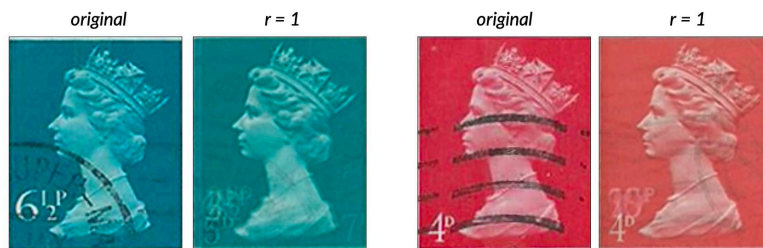
Retaining a low-rank approximation of an image can also be a technique for [denoising](#).

Consider the set of  $N = 22$  stamps below, which all have similar features, but are obscured by different black mail markings.



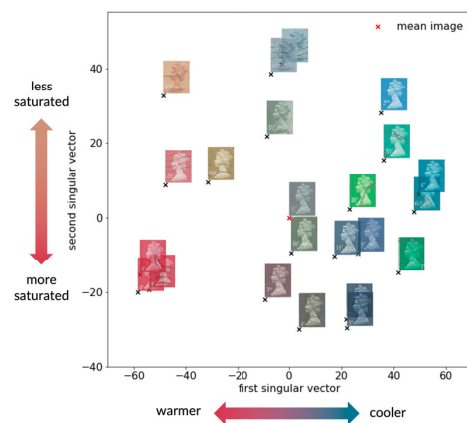
## Image Denoising

In this example, we consider each  $m \times n \times p$  color image as a single sample of length  $mnp$ , where  $p = 3$ , and assemble a matrix  $S \in \mathbb{R}^{3mn \times N}$ .



## Image Denoising

The singular vectors can be used to construct a lower-dimensional space that captures the most significant features of the data. This forms the basis of PCA and can be used to uncover features such as clustering in an unsupervised way.



## Principal Component Analysis

The left singular vectors  $u_1, u_2, \dots, u_n$  form a rotated, orthonormal basis for the  $m$ -dimensional space occupied by the columns of  $A$ ,  $a_1, a_2, \dots, a_n$  (data points).

This new basis is oriented such that  $u_1$  points in the direction along which the data has the largest variance,  $u_2$  points along the direction of next-largest variance orthogonal to  $u_1$ , and so on.

How?

Consider the **covariance matrix**  $C = \frac{1}{n}AA^T \in \mathbb{R}^{m \times m}$ , where

- ▶ the diagonals  $c_{ii} = \frac{1}{n} \sum_{j=1}^n a_{ij}^2$  give the variance of the data along the  $i^{th}$  axis
- ▶ the off-diagonals  $c_{ik} = \frac{1}{n} \sum_{j=1}^n a_{ij}a_{jk}$  give the covariance along the  $i^{th}$  and  $k^{th}$  axes.

## Principal Component Analysis

If  $A = \hat{U}\hat{\Sigma}V^T$ , then

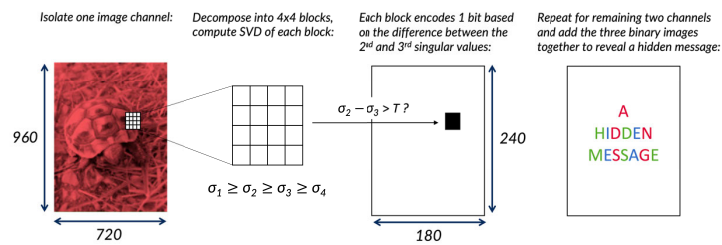
$$\begin{aligned} C &= \frac{1}{n}(\hat{U}\hat{\Sigma}V^T)(\hat{U}\hat{\Sigma}V^T)^T = \frac{1}{n}\hat{U}\hat{\Sigma}(V^TV)\hat{\Sigma}^T\hat{U}^T \\ &= \frac{1}{n}\hat{U}\hat{\Sigma}\hat{\Sigma}^T\hat{U}^T = \hat{U}\left(\frac{\hat{\Sigma}^2}{n}\right)\hat{U}^T \end{aligned}$$

Since  $Cu_j = \frac{1}{n}\sigma_j^2 u_j$  for any  $u_j$ , the columns of  $\hat{U}$  are the **eigenvectors** and the diagonals of  $\hat{\Sigma}^2/n$  the **eigenvalues** of the covariance matrix.

The eigenvalues once again represent the variance of the data, now along the rotated axes  $u_1, \dots, u_n$ . These axes are called **principal components** in PCA, but they are the same as **left singular vectors**!

## Steganography

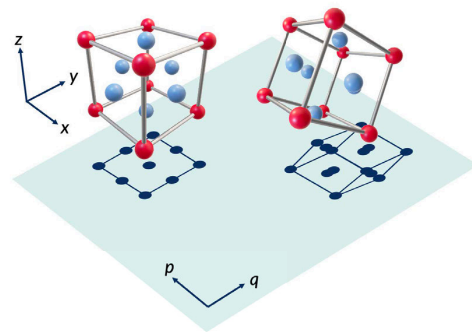
**Steganography** is the art of concealing hidden messages within non-secret data. The first exercise will feature decoding a hidden message encoded in the singular values of an image.



**Key idea:** Since the information contained in later singular vectors, which have correspondingly smaller singular values, is less important, this part of the decomposition can be manipulated to encode hidden messages in plain sight!

## 3D Reconstruction

SVD can also be used to perform 3D reconstruction from a sequence of 2D projections<sup>1</sup>.



Here we will consider a rotating object characterized by  $N$  control points on its surface.

---

<sup>1</sup>Reference: Muller, N. *et al.* (2004). Singular value decomposition, eigenfaces, and 3D reconstructions. *SIAM review*, 46(3), 518-545. [↗](#) [↖](#) [↕](#) [↔](#) [↻](#) [↺](#) [↻](#)

### 3D Reconstruction

The object's state in 3D can be expressed as an **object matrix**:

$$O \in \mathbb{R}^{3 \times N} = \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(N)} \\ y^{(1)} & y^{(2)} & \dots & y^{(N)} \\ z^{(1)} & z^{(2)} & \dots & z^{(N)} \end{bmatrix}$$

Its tracked motion is captured by the product of a time-varying rotation  $R_t$  and the orthographic projection  $P_z$  in the **motion matrix**:

$$M_t \in \mathbb{R}^{2 \times 3} = P_z R_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_t$$

The coordinates of the control points in the 2D projected space are thereby captured by the **measurement matrix**:

$$A_t \in \mathbb{R}^{2 \times N} = M_t O = \begin{bmatrix} q_t^{(1)} & q_t^{(2)} & \dots & q_t^{(N)} \\ p_t^{(1)} & p_t^{(2)} & \dots & p_t^{(N)} \end{bmatrix}$$



## 3D Reconstruction

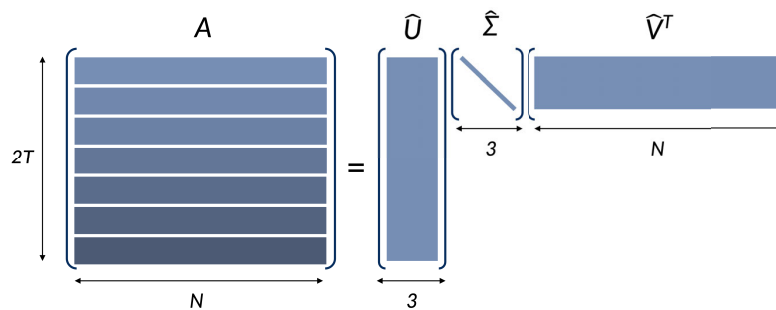
Measurements across  $T$  rotations can be stacked to form a combined measurement matrix:

$$A \in \mathbb{R}^{2T \times N} = MO = \begin{bmatrix} q_0^{(1)} & q_0^{(2)} & \cdots & q_0^{(N)} \\ p_0^{(1)} & p_0^{(2)} & \cdots & p_0^{(N)} \\ q_1^{(1)} & q_1^{(2)} & \cdots & q_1^{(N)} \\ p_1^{(1)} & p_1^{(2)} & \cdots & p_1^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{T-1}^{(1)} & q_{T-1}^{(2)} & \cdots & q_{T-1}^{(N)} \\ p_{T-1}^{(1)} & p_{T-1}^{(2)} & \cdots & p_{T-1}^{(N)} \end{bmatrix}$$

where  $M \in \mathbb{R}^{2T \times 3}$  is a stacked series of motion matrices.

## 3D Reconstruction

- Our objective is to deduce the object matrix  $O$  given only  $A$ . Since  $\text{rank}(O) = 3$ , we expect for a general 3D rotation that  $\text{rank}(A) = 3$ .
- This means that we can represent  $A$  by a truncated SVD  $A = \hat{U}\hat{\Sigma}\hat{V}^T$ , where  $\hat{U} \in \mathbb{R}^{2T \times 3}$ ,  $\hat{\Sigma} \in \mathbb{R}^{3 \times 3}$ , and  $\hat{V}^T \in \mathbb{R}^{3 \times N}$ .



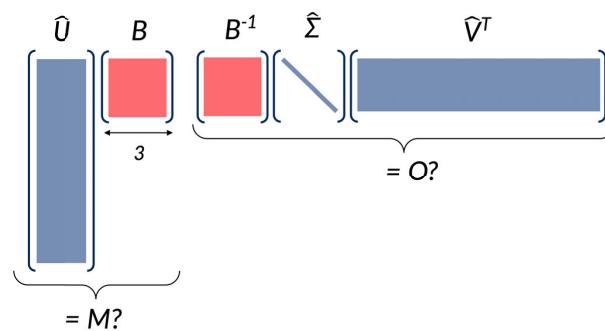
## 3D Reconstruction

- ▶ How can we obtain the factorization  $A = MO$ ? Naively, we can propose  $M = \hat{U}$ ,  $O = \hat{\Sigma} \hat{V}^T$ :

Diagram illustrating the question: Is the product of two matrices,  $U$  and  $V^T$ , equal to  $O$ ?

### 3D Reconstruction

- ▶ However, this is not a unique factorization; we could just as easily introduce  $M = \hat{U}B$ ,  $O = B^{-1}\hat{\Sigma}\hat{V}^T$ , to obtain  $MO = \hat{U}BB^{-1}\hat{\Sigma}\hat{V}^T = \hat{U}\hat{\Sigma}\hat{V}^T$  for some matrix  $B \in \mathbb{R}^{3 \times 3}$ .



- ▶ We'd like to use this additional freedom with  $B$  to impose constraints in our factorization.

### 3D Reconstruction

- Recall that each pair of rows in  $M$  is given by

$$M_t = P_z R_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_t.$$

- The two rows of  $P_z$  pick out the first two rows of  $R_t$ , which is orthogonal as required for rotation matrices. Thus, we expect that the pairs of rows in  $M = \hat{U}B$  to be orthogonal. If  $m_i$  and  $u_i$  denote the  $i^{\text{th}}$  row of  $M$  and  $\hat{U}$ , respectively,

$$\begin{aligned} m_{2i}^T m_{2i} &= (u_{2i}^T B)(B^T u_{2i}) = 1 \\ m_{2i+1}^T m_{2i+1} &= (u_{2i+1}^T B)(B^T u_{2i+1}) = 1 \\ m_{2i}^T m_{2i+1} &= (u_{2i}^T B)(B^T u_{2i+1}) = 0 \end{aligned}$$

for  $i = 0, 1, \dots, T-1$ .

## 3D Reconstruction

- ▶ The 6 unknowns in the symmetric matrix  $S = BB^T$  can be solved for by setting up a least squares problem from these orthogonality relations.

$$\begin{aligned} \left( \begin{array}{c} u_{2i}^T \\ \hline \end{array} \right) \left[ \begin{array}{c|c} S & \\ \hline \end{array} \right] \left( \begin{array}{c} u_{2i} \\ \hline \end{array} \right) &= 1 \\ \underbrace{\left( \begin{array}{c|c} \hline \hline \end{array} \right)}_{\begin{array}{cc} B & B^T \\ \left( \begin{array}{c|c} \hline \hline \end{array} \right) \end{array}} & \end{aligned} \quad \begin{aligned} \left( \begin{array}{c} u_{2i+1}^T \\ \hline \end{array} \right) \left[ \begin{array}{c|c} S & \\ \hline \end{array} \right] \left( \begin{array}{c} u_{2i+1} \\ \hline \end{array} \right) &= 1 \end{aligned} \quad \begin{aligned} \left( \begin{array}{c} u_{2i}^T \\ \hline \end{array} \right) \left[ \begin{array}{c|c} S & \\ \hline \end{array} \right] \left( \begin{array}{c} u_{2i+1} \\ \hline \end{array} \right) &= 0 \end{aligned}$$

## 3D Reconstruction

- ▶ If  $S = Q\Lambda Q^T$  is the eigendecomposition of  $S$  with orthogonal eigenvectors in  $Q$  and diagonal matrix of eigenvalues  $\Lambda$ , then  $B$  can be determined as  $B = Q\Lambda^{1/2}$ .

The diagram shows two equations. The first equation is  $S = Q \Lambda Q^T$ , where  $S$  is a yellow square with a white diagonal line,  $Q$  is a purple square,  $\Lambda$  is a yellow square with a white diagonal line, and  $Q^T$  is a purple square. The second equation is  $B = Q \Lambda^{1/2}$ , where  $B$  is a red square,  $Q$  is a purple square, and  $\Lambda^{1/2}$  is a red square with a white diagonal line.

## 3D Reconstruction

- ▶ However,  $B$  is still not unique! Multiplication by an arbitrary rotation such as  $B = Q\Lambda^{1/2}R$  still results in  $BB^T = (Q\Lambda^{1/2}R)(R^T\Lambda^{1/2}Q^T) = Q\Lambda Q^T = S$  by the orthonormality of columns of a rotation matrix.
- ▶ It's acceptable to simply take  $R = I$ , but we acknowledge that our final solution for the object matrix  $O$  will be **unique up to a rotation**.
- ▶ Thus, our final factorization is

$$M = \hat{U}B = \hat{U}(Q\Lambda^{1/2}R)$$
$$O = B^{-1}\hat{\Sigma}\hat{V}^T = (R^T\Lambda^{-1/2}Q^T)\hat{\Sigma}\hat{V}^T$$

and the 3D object is recovered in  $O$ .



## References

1. Brunton, Steven L., and J. Nathan Kutz. "Chapter 1: Singular Value Decomposition." *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.
2. Chanu, Yambem Jina, Kh Manglem Singh, and Themrichon Tuithung. "A robust steganographic method based on singular value decomposition." *Int. J. Inf. Comput. Technol* 4.7 (2014): 717-726.
3. Muller, Neil, Lourenço Magaia, and Ben M. Herbst. "Singular value decomposition, eigenfaces, and 3D reconstructions." *SIAM review* 46.3 (2004): 518-545.