

**EE16B**  
**Designing Information**  
**Devices and Systems II**

**Lecture 8A**  
**Singular Value Decomposition (SVD)**

# Recap

---

- Last time
  - Finished outputs and observability
- Today:
  - New module: the Singular Value Decomposition
  - Describe the SVD
  - Show example SVD is useful in applications
- Next:
  - How to calculate the SVD

# Rank 1 Matrix

---

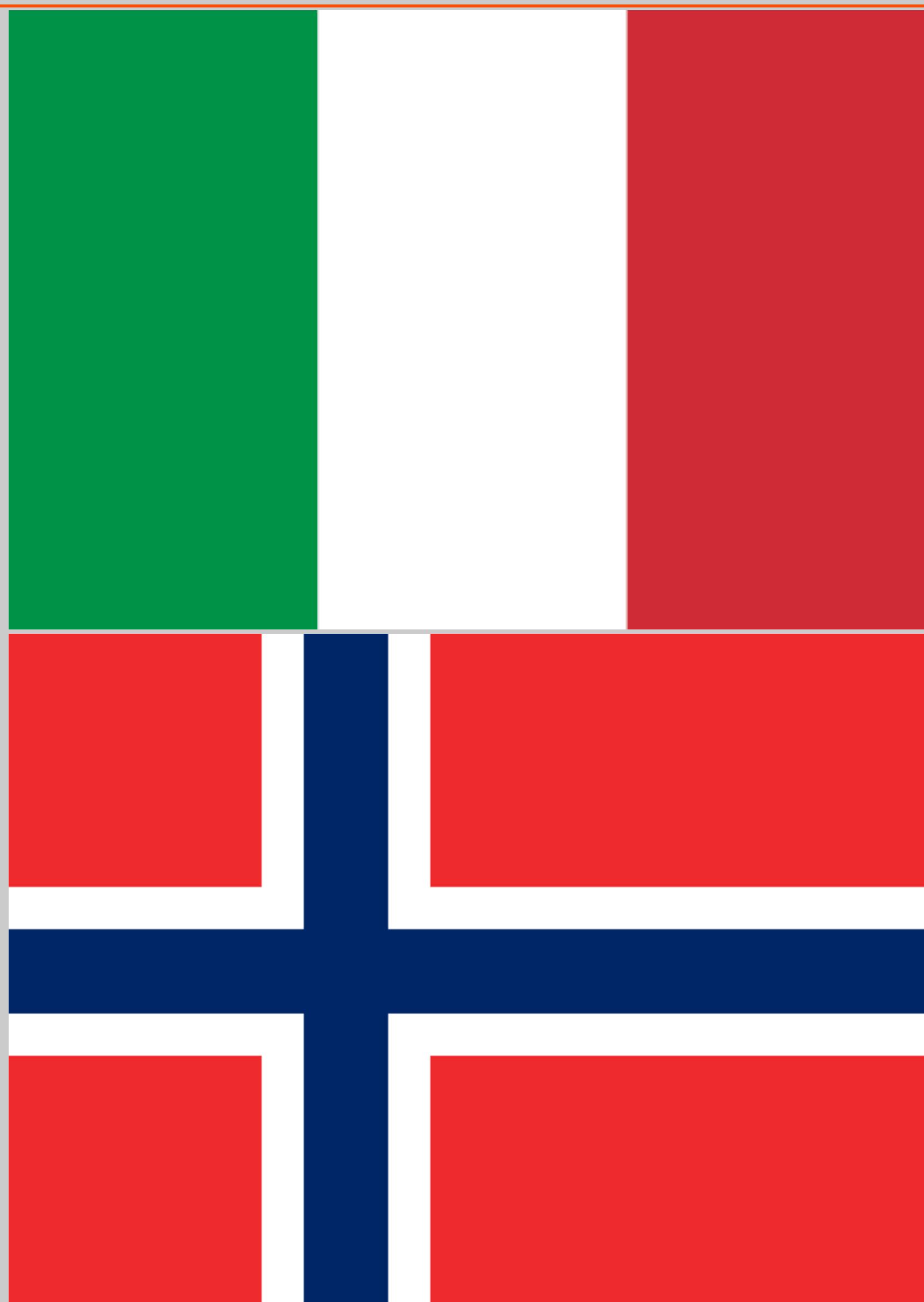
Consider the following matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad \text{Rank} = 1$$

We can decompose a rank-1 matrix as an outer product:

$$\vec{u}\vec{v}^T \in \mathbb{R}^{m \times n} \quad \vec{u} \in \mathbb{R}^m \quad \vec{v} \in \mathbb{R}^n$$
$$\begin{bmatrix} m \times 1 \\ 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 \times n \end{bmatrix}$$

# Flags as low-rank matrices



# SVD

---

SVD decomposes a rank r matrix  $A \in \mathbf{R}^{m \times n}$  into a sum of r rank-1 matrices:

$$A = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T$$

1)  $\vec{u}_i^T \vec{u}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \Rightarrow \|\vec{u}_i\| = 1 \quad \vec{u}_i \perp \vec{u}_j$

2)  $\vec{v}_i^T \vec{v}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \Rightarrow \|\vec{v}_i\| = 1 \quad \vec{v}_i \perp \vec{v}_j$

3)  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$

# SVD

$$A = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T$$

$$m \times n \quad m+n \quad m+n \quad m+n$$

$r(m+n) \leq mn$  If m,n are large and r is small

Typically,  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\hat{r}} \gg \sigma_{\hat{r}+1} \geq \cdots \geq \sigma_r$

$$\begin{matrix} 10 & 8 & 5 & 0.1 & 0.001 \\ \left[ \begin{array}{cccccc} 1.02 & 0.99 & 0.98 & 1.03 & 1.01 & 1 \\ 2 & 1.98 & 2.01 & 2.03 & 1.99 & 1.97 \\ 3.01 & 2.98 & 3 & 2.99 & 3.03 & 3.02 \end{array} \right] \end{matrix}$$

$$A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_1 \vec{v}_1^T + \cdots + \sigma_{\hat{r}} \vec{u}_{\hat{r}} \vec{v}_{\hat{r}}^T$$

# History

---

Known for a long time – but not considered important

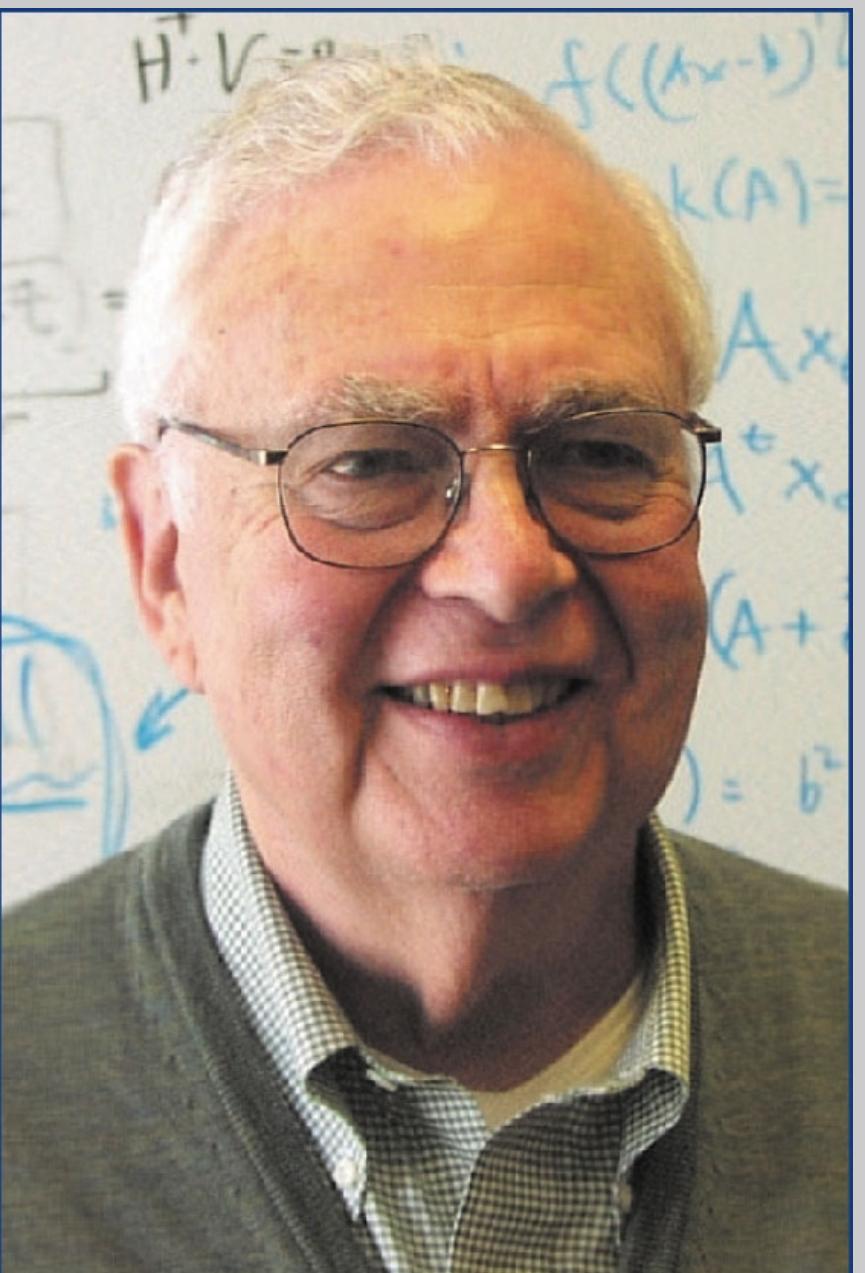
1965 – G. Golub and W. Kahan – practical way of computing the SVD – numerically stable

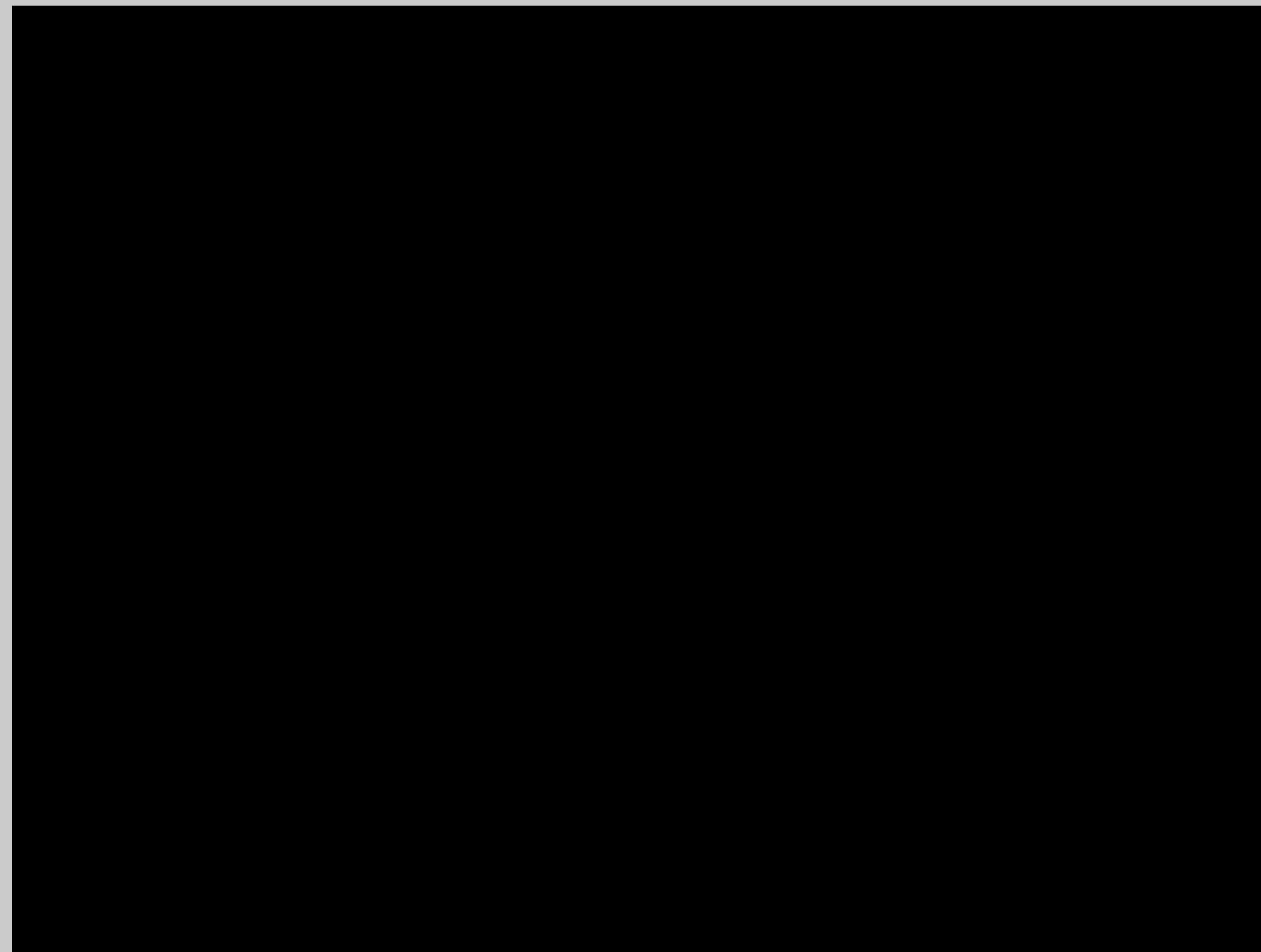
Today, used everywhere!

LAPack implementation –Demmel & Kahan 1990  
(numpy and matlab use LAPack)

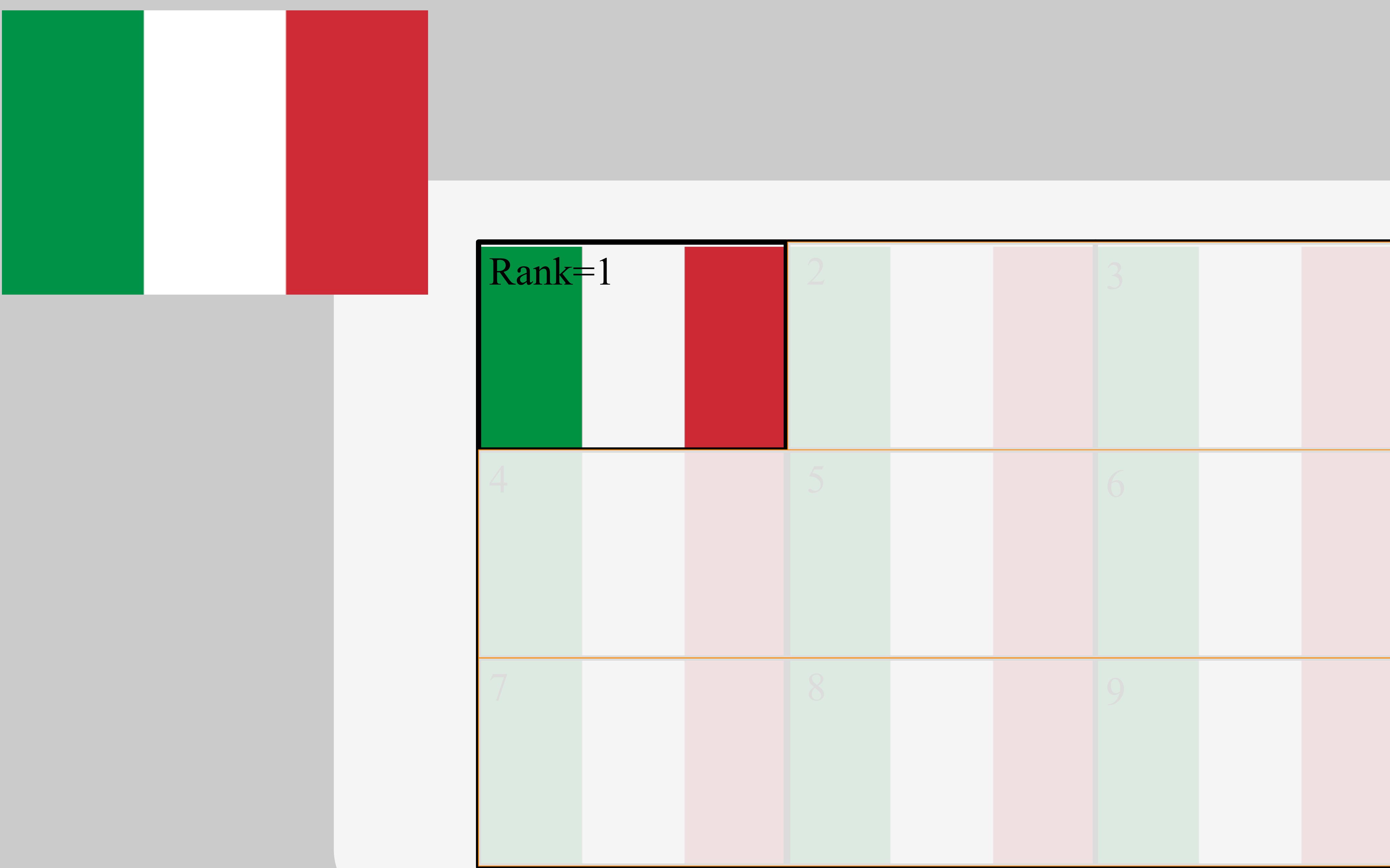


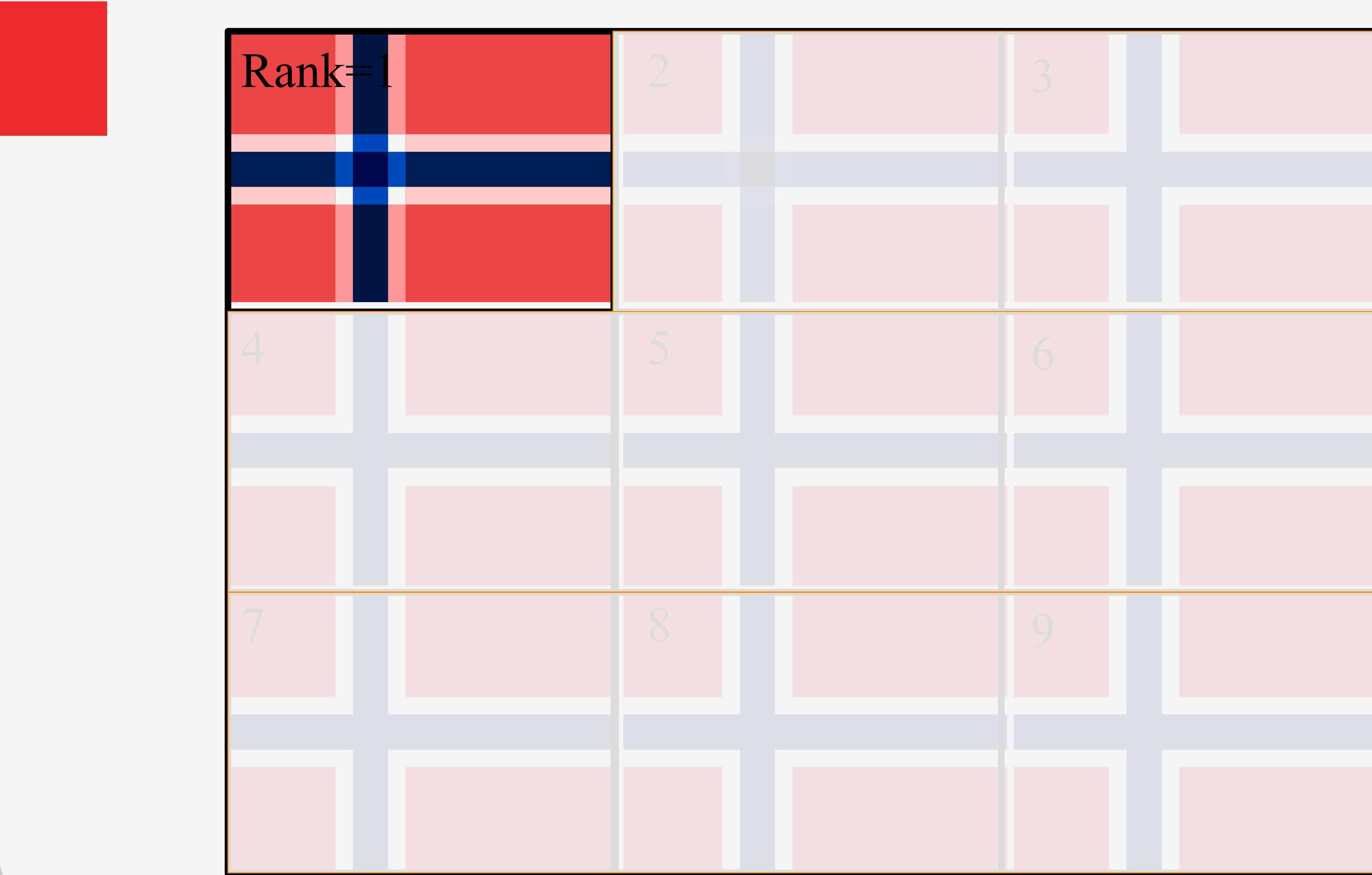
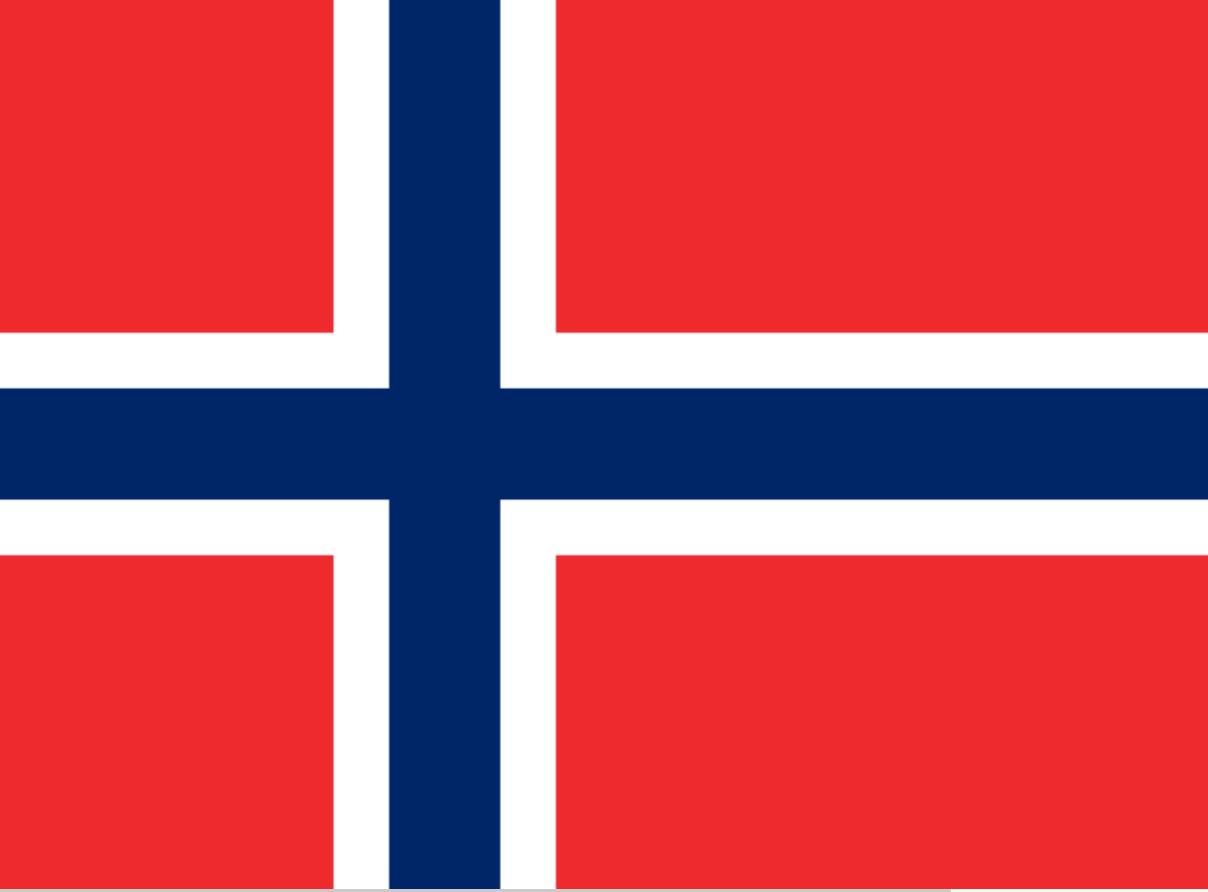
*Gene Golub's license plate, photographed by Professor P. M. Kroonenberg of Leiden University.*

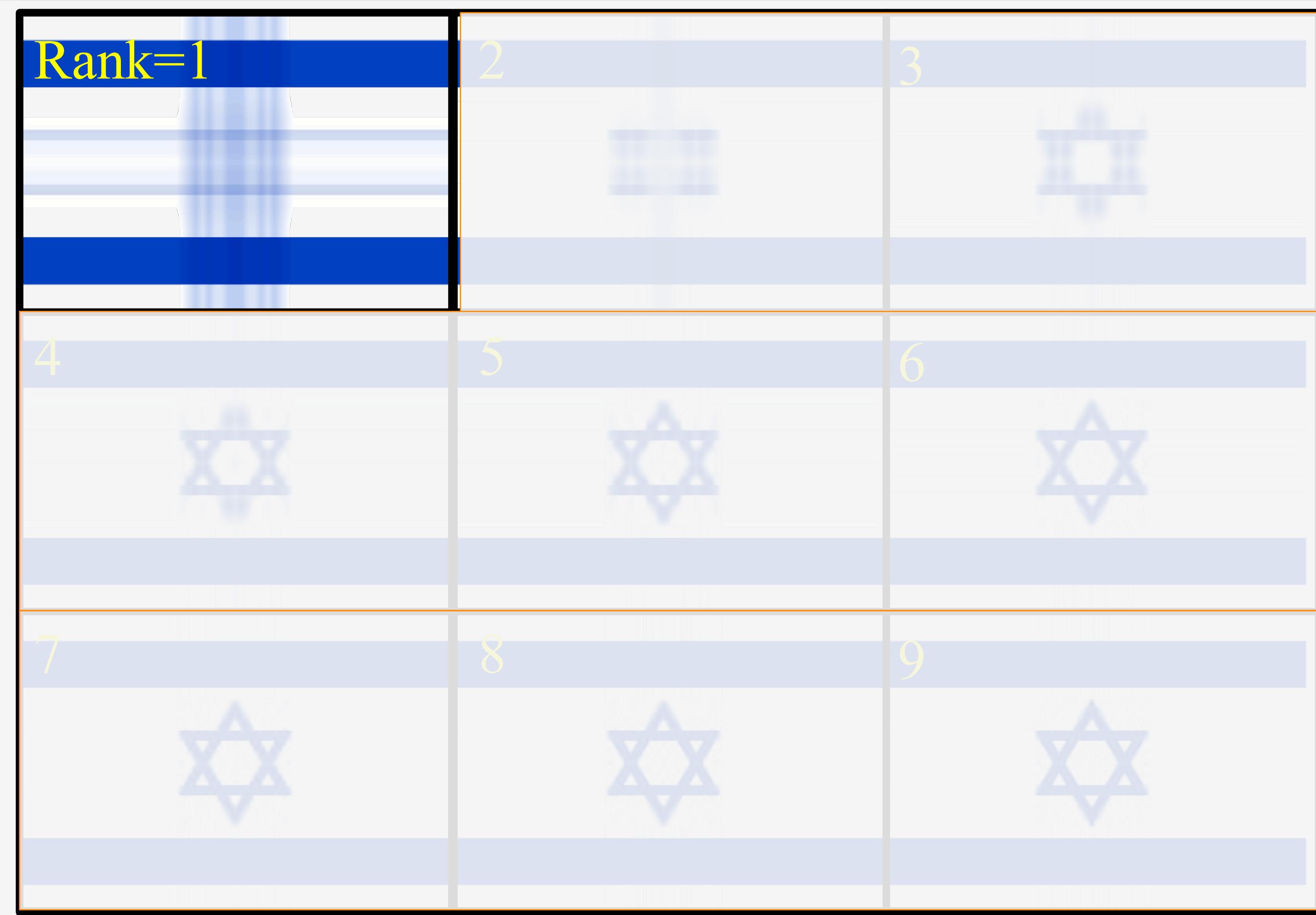


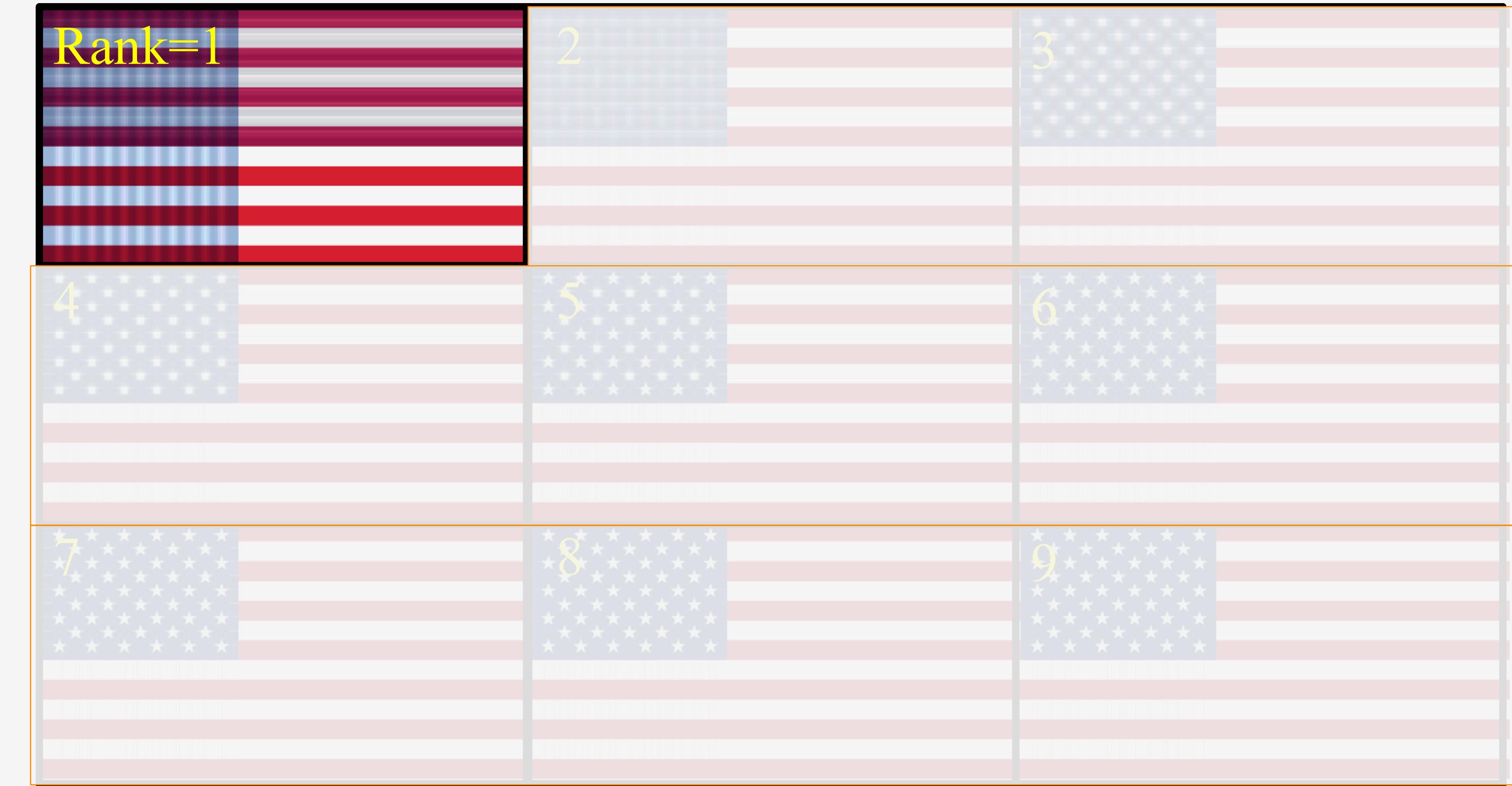


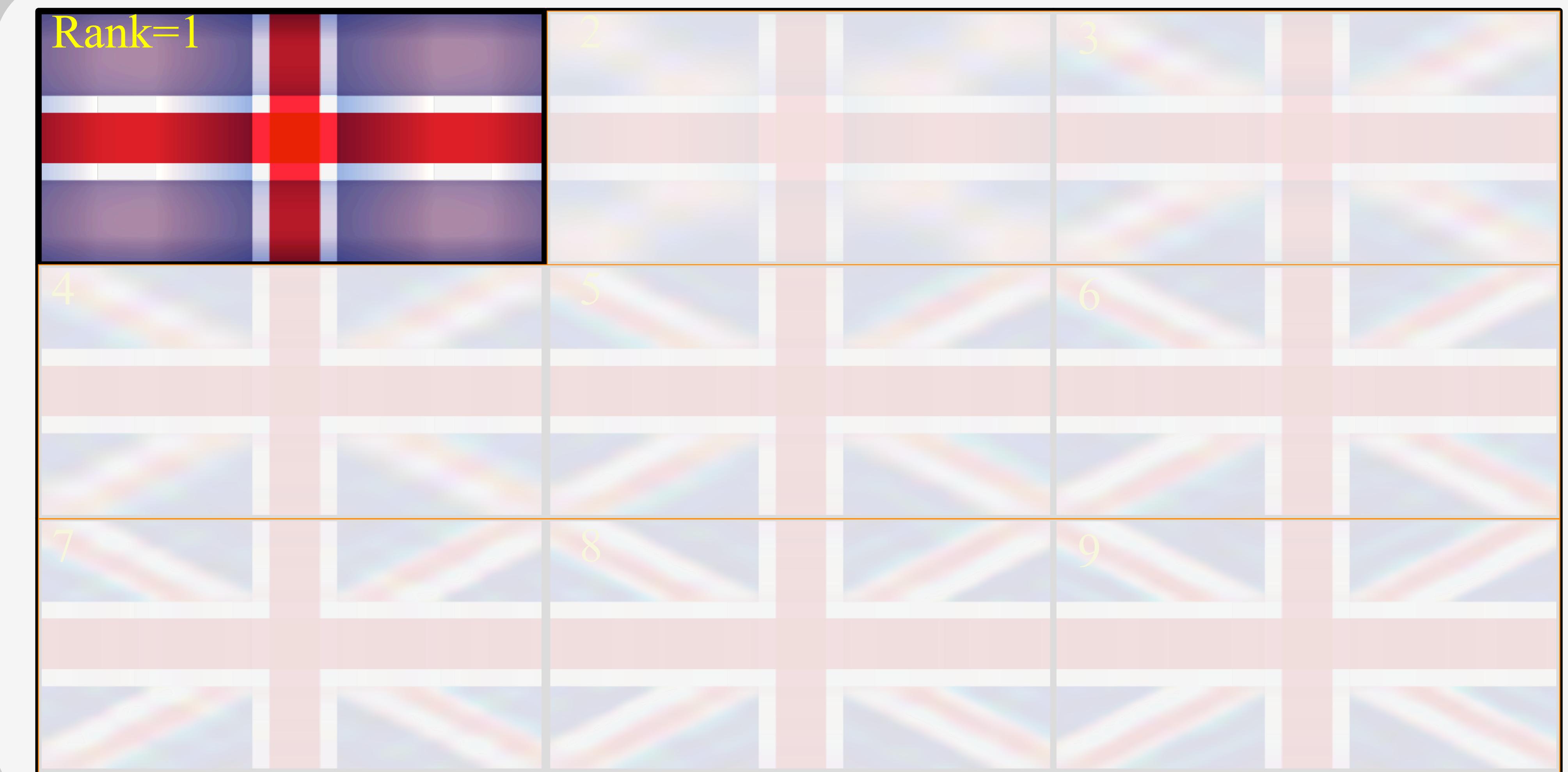
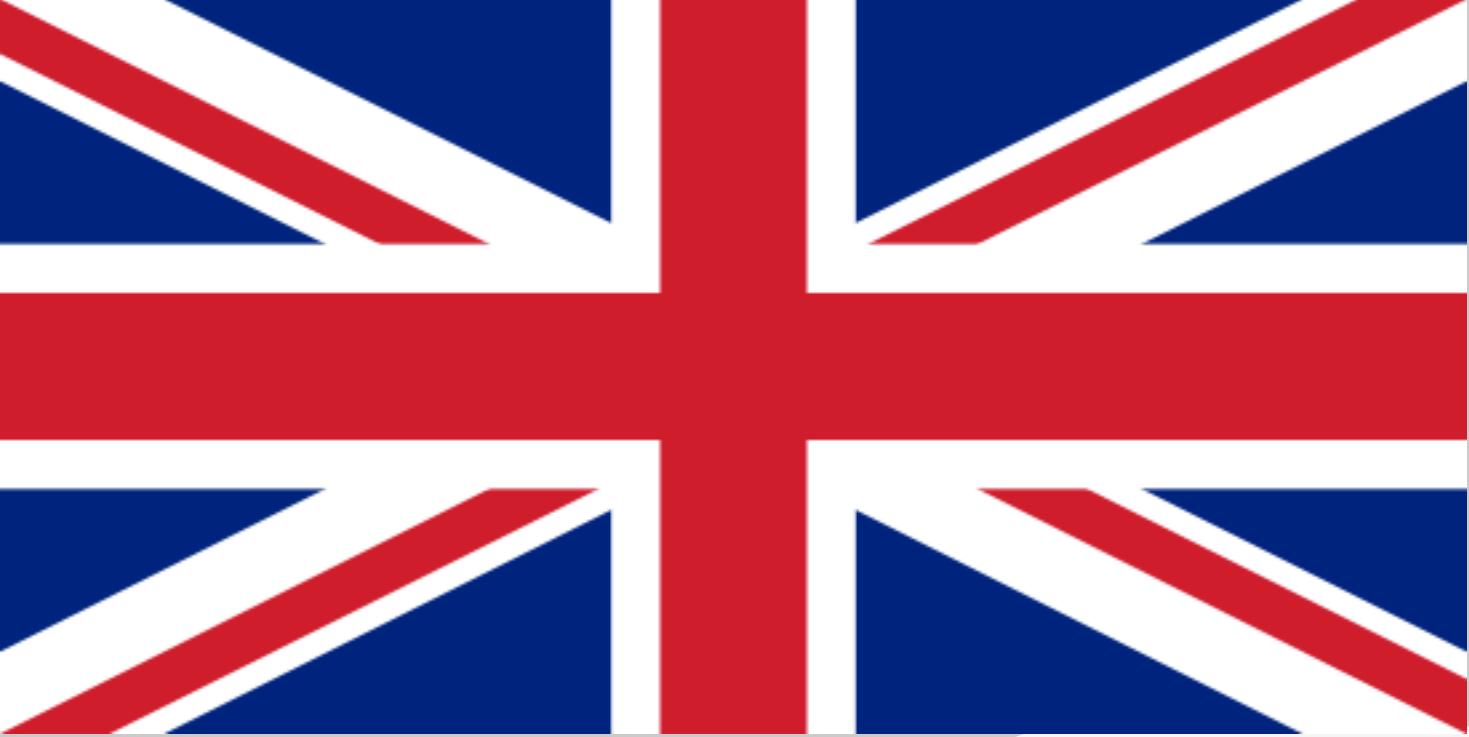
- Los Alamos video 1976 on the SVD. Then Relatively unknown, but today used everywhere.  
3D computer graphics by Cleve Moler (Matlab)
- Lecture by Moler: “SVD Saves the Universe”  
<https://www.mathworks.com/videos/the-singular-value-decomposition-saves-the-universe-1481294462044.html>







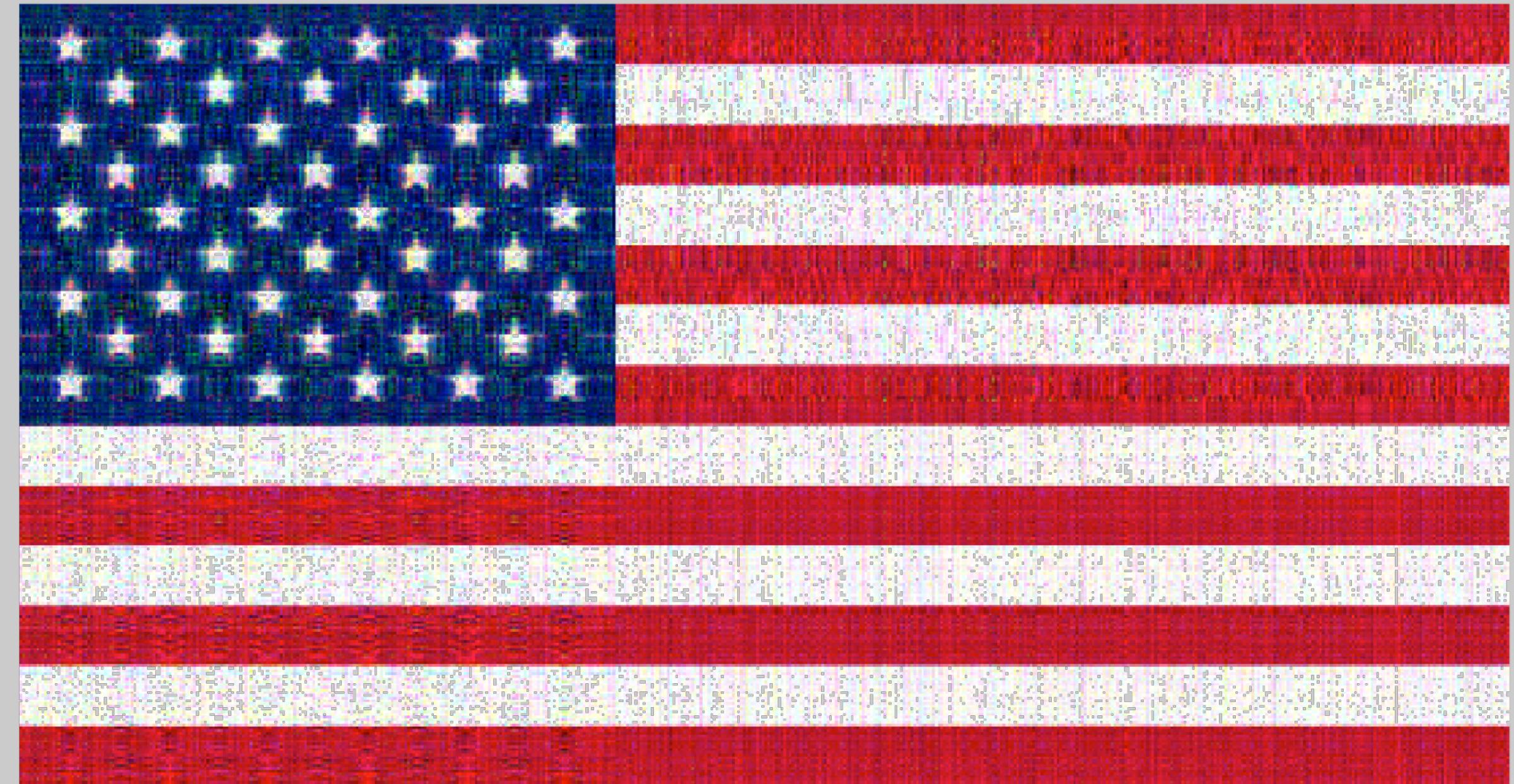
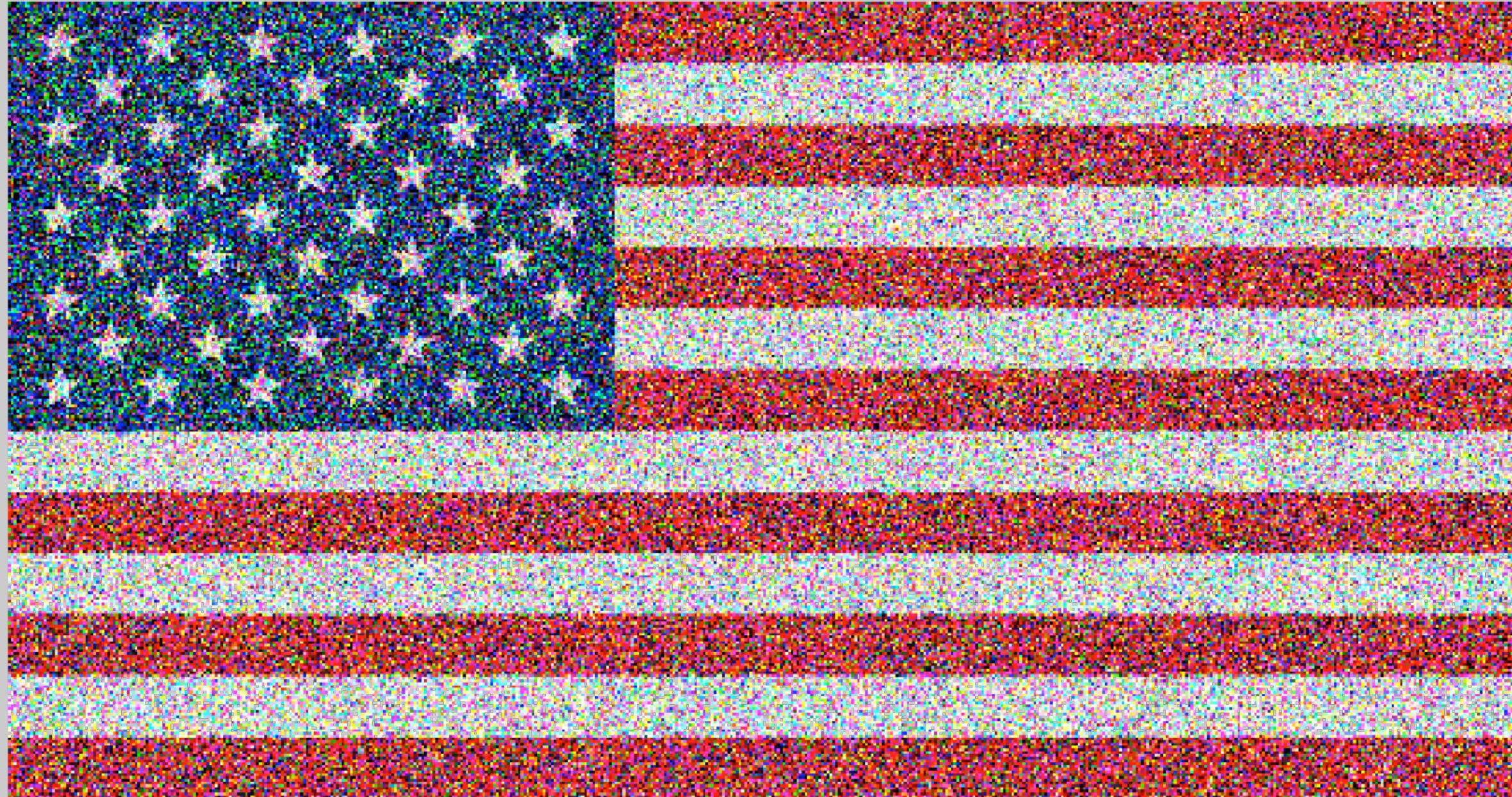




# Noisy Flag

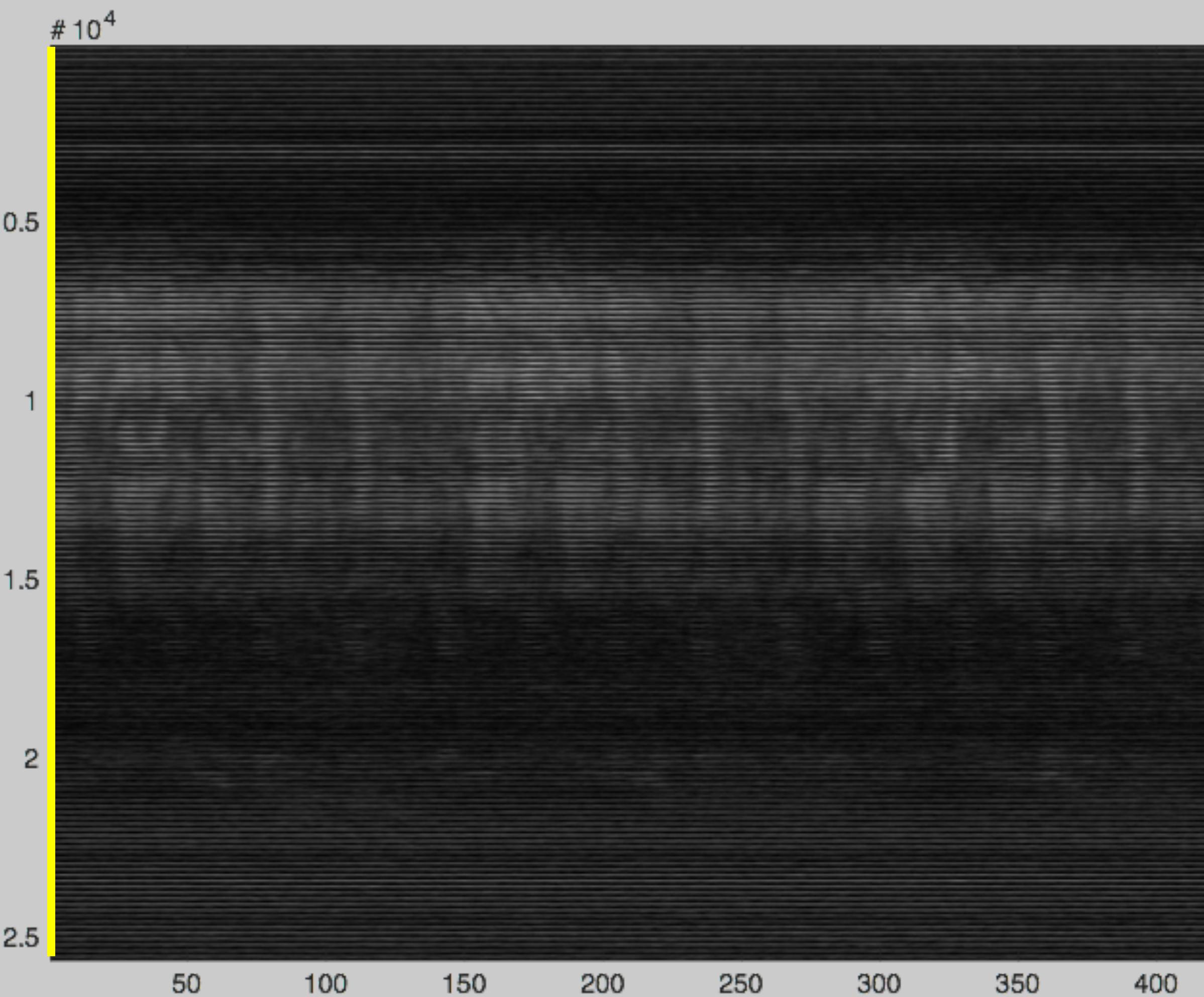
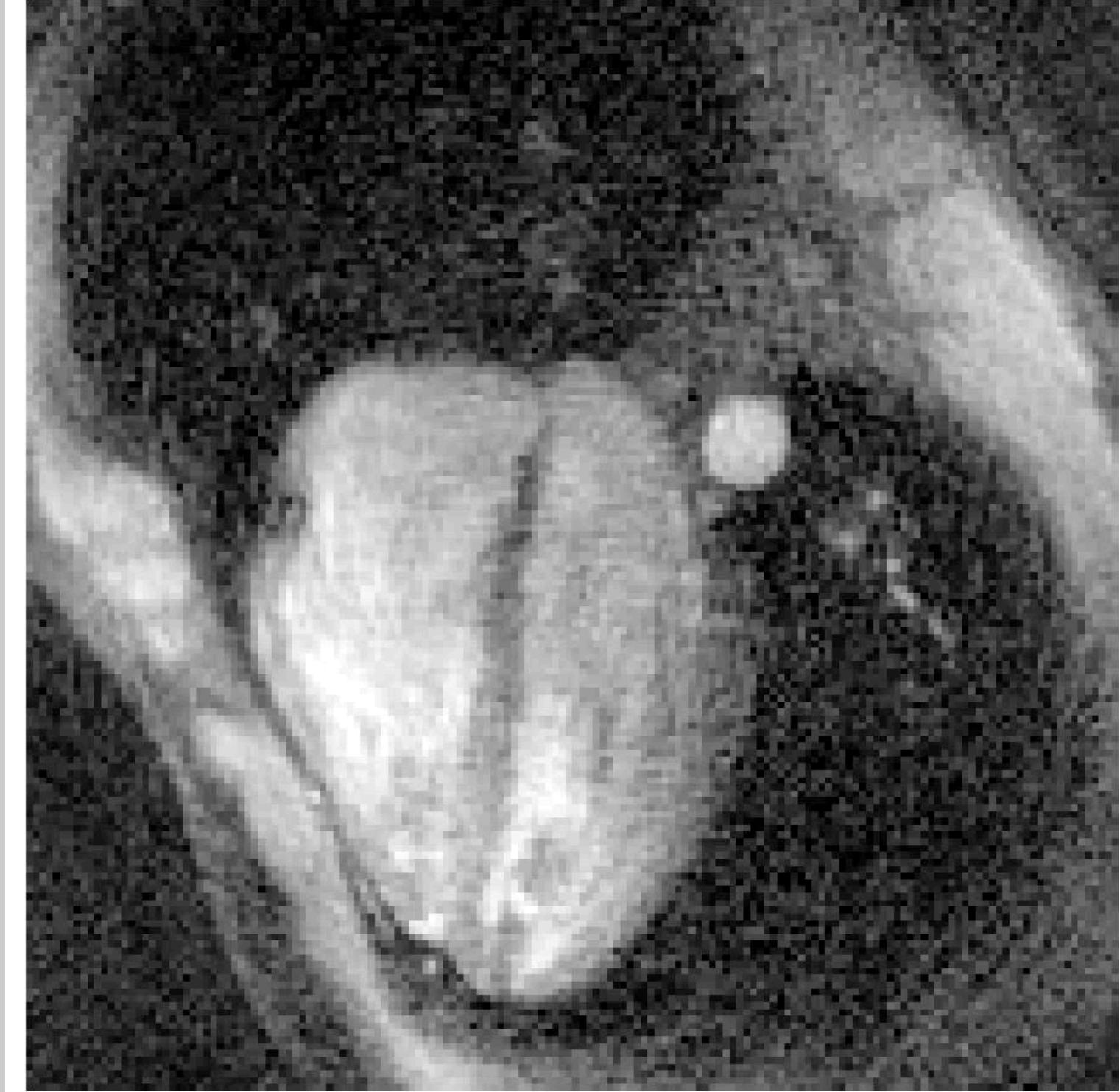
$$\begin{bmatrix} 1.02 & 0.99 & 0.98 & 1.03 & 1.01 & 1 \\ 2 & 1.98 & 2.01 & 2.03 & 1.99 & 1.97 \\ 3.01 & 2.98 & 3 & 2.99 & 3.03 & 3.02 \end{bmatrix}$$

$$\sum_{i=0}^5 \sigma_i \vec{u}_i \vec{v}_i^T$$



# Video of a Heart

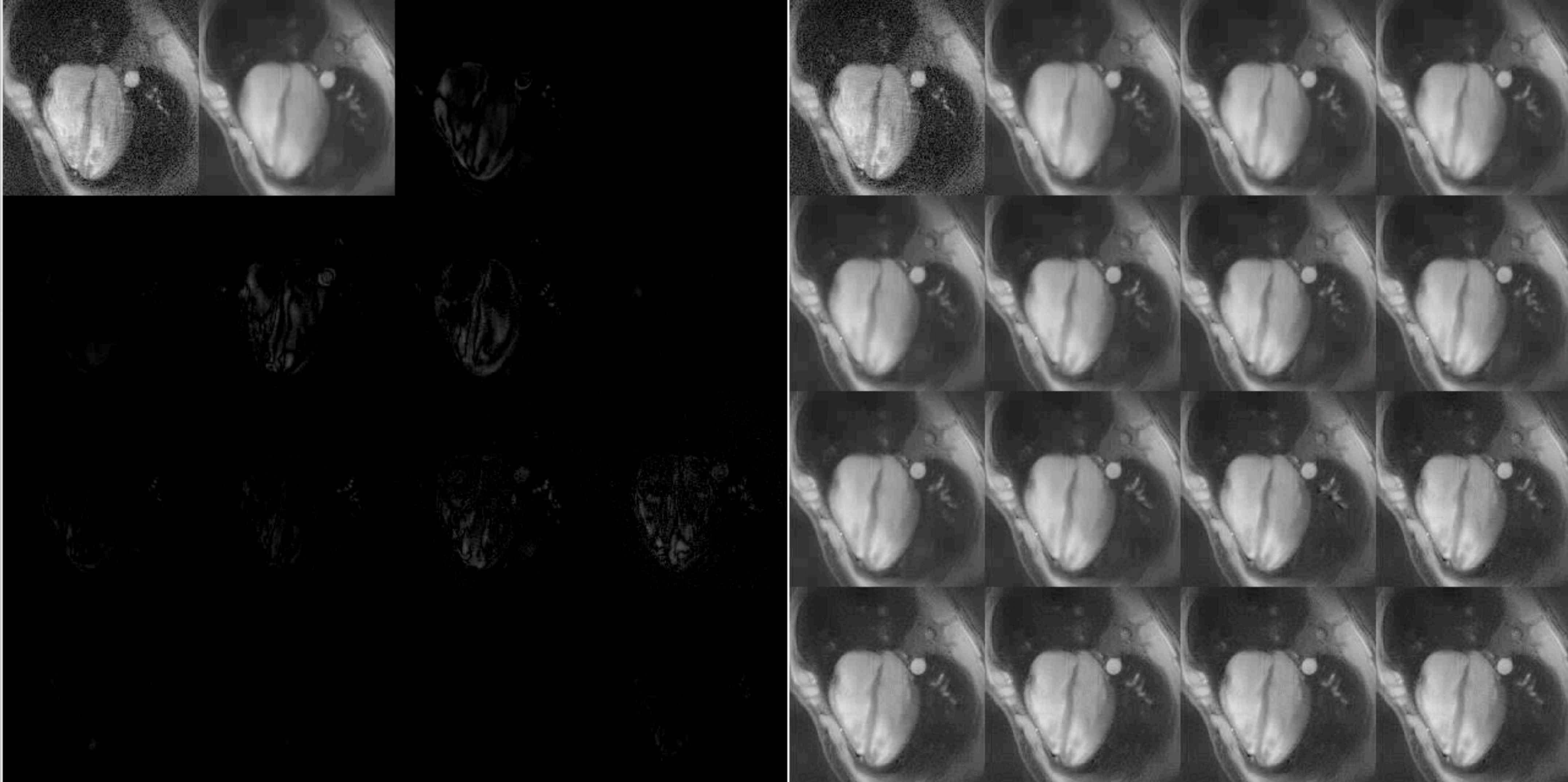
---



# Heart Example

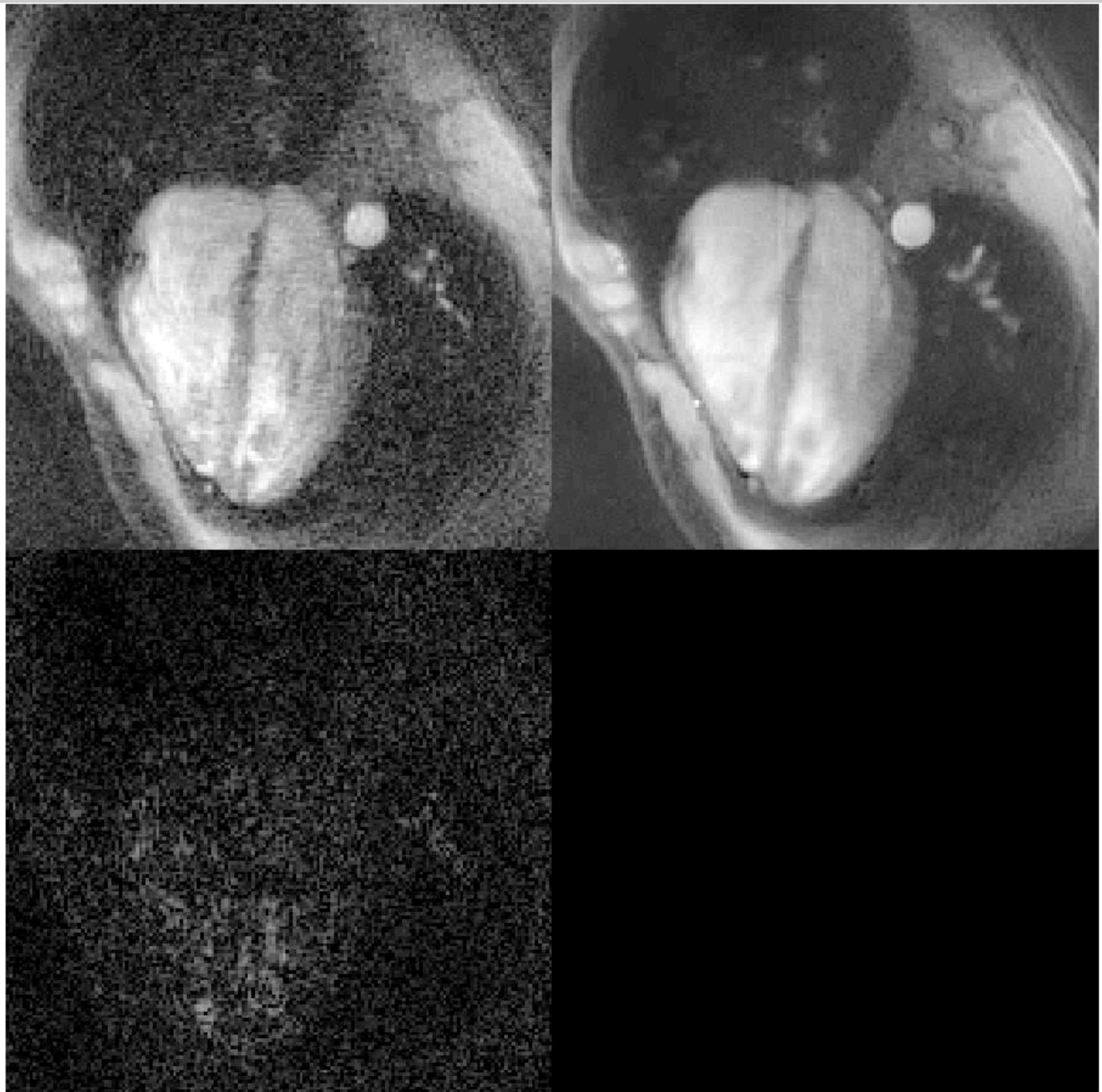
$$\sigma_i \vec{u}_i \vec{v}_i^T$$

$$\sum_{i=0}^r \sigma_i \vec{u}_i \vec{v}_i^T$$



# Heart Example

$$\sum_{i=16}^{417} \sigma_i \vec{u}_i \vec{v}_i^T$$



$$\sum_{i=0}^{15} \sigma_i \vec{u}_i \vec{v}_i^T$$

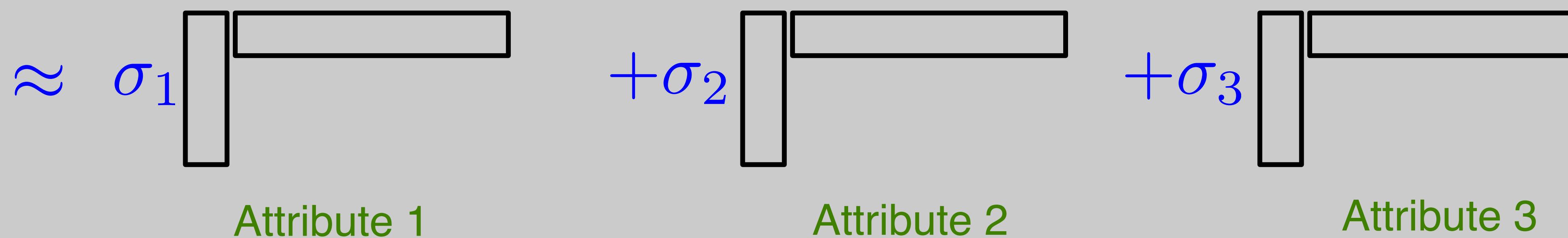
# Data Analysis with SVD

$$A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_1 \vec{v}_1^T + \cdots + \sigma_{\hat{r}} \vec{u}_{\hat{r}} \vec{v}_{\hat{r}}^T$$

n movies

effective

1	2	5	5	1	3	3	3	2	5	5	4	4	3	2	2	5	1	1
5	3	2	1	1	3	3	3	1	1	2	4	5	5	4	4	1	3	2
5	1	2	1	1	2	3	3	1	1	2	1	5	5	3	5	1	1	2
5	3	2	1	1	3	2	3	1	1	2	4	1	1	4	4	5	1	5
1	2	3	2	1	3	2	3	2	1	2	1	1	1	4	4	5	1	5
1	1	1	1	5	3	3	3	1	5	2	4	4	4	2	5	5	1	1



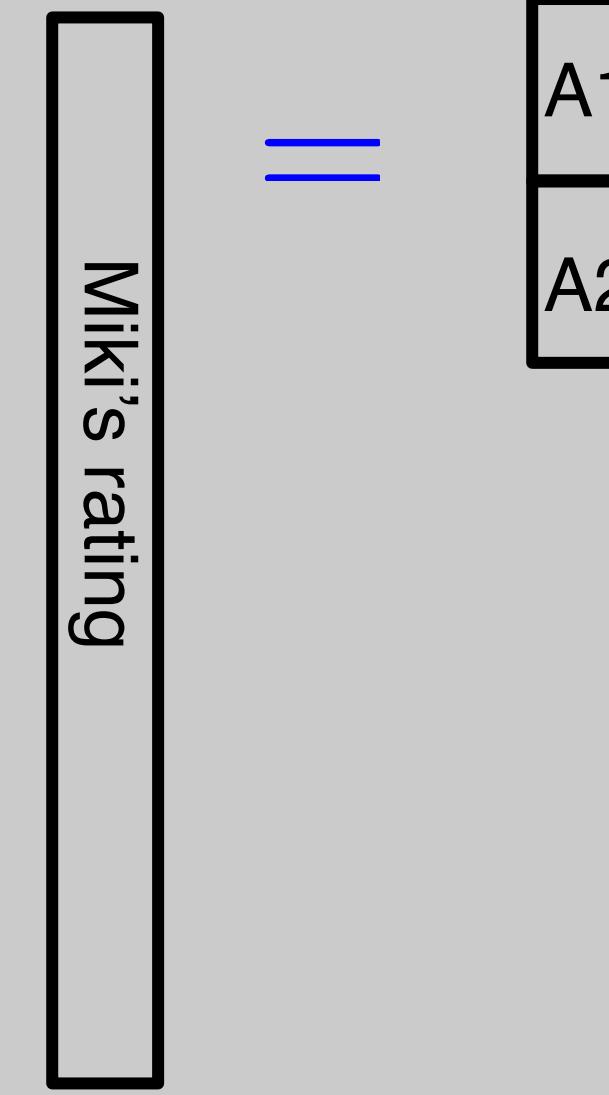
# Classification with SVD

$$v_1^T$$

Movie significance for attribute 1

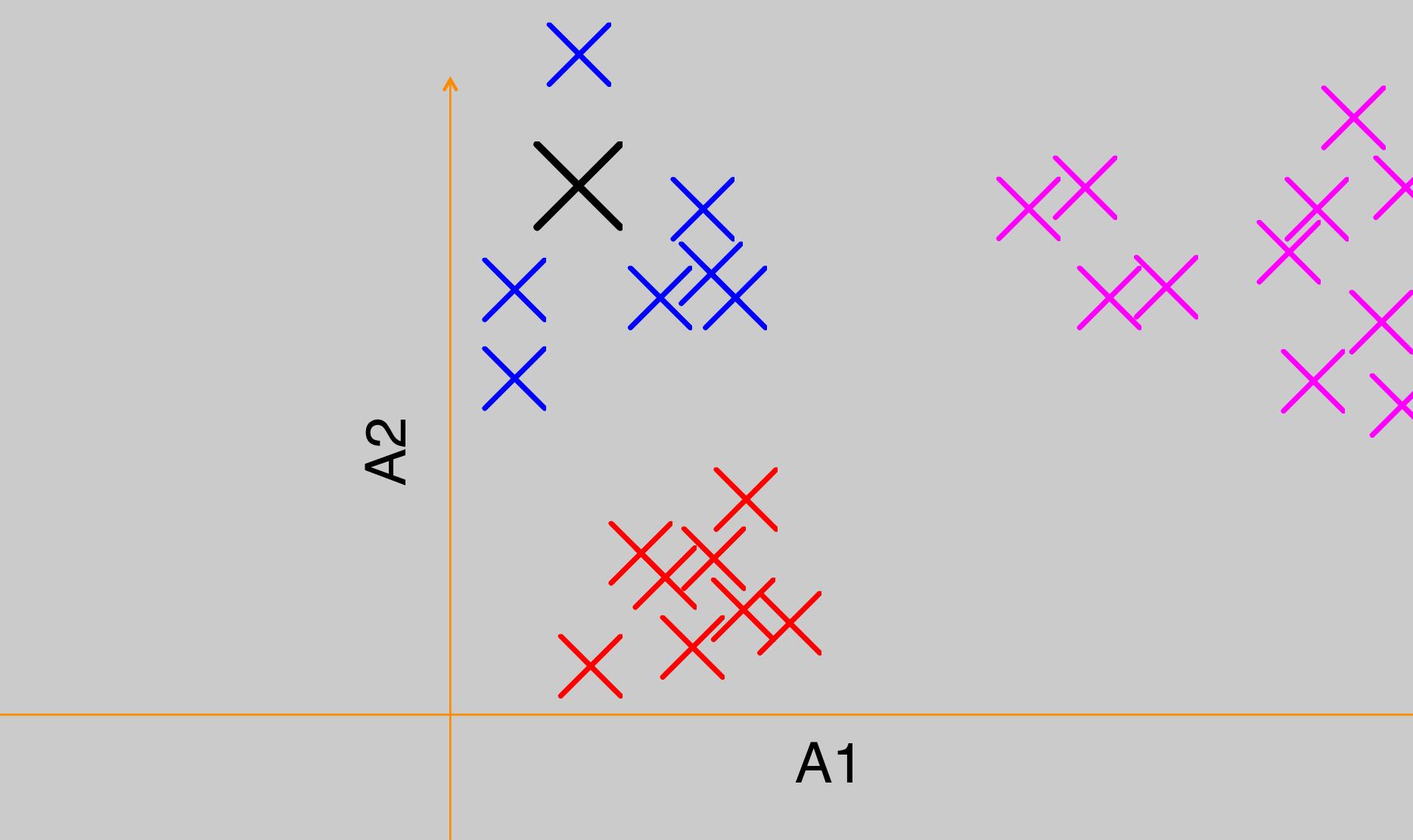
$$v_2^T$$

Movie significance for attribute 2



n movies

1	2	5	5	1	3	3	3	2	5	5	4	4	3	2	2	5	1	1
5	3	2	1	1	3	3	3	1	1	2	4	5	5	4	4	1	3	2
5	1	2	1	1	2	3	3	1	1	2	1	5	5	3	5	1	1	2
5	3	2	1	1	3	2	3	1	1	2	4	1	1	4	4	5	1	5
1	2	3	2	1	3	2	3	2	1	2	1	1	1	4	4	5	1	5
1	1	1	1	5	3	3	3	1	5	2	4	4	4	2	5	5	1	1



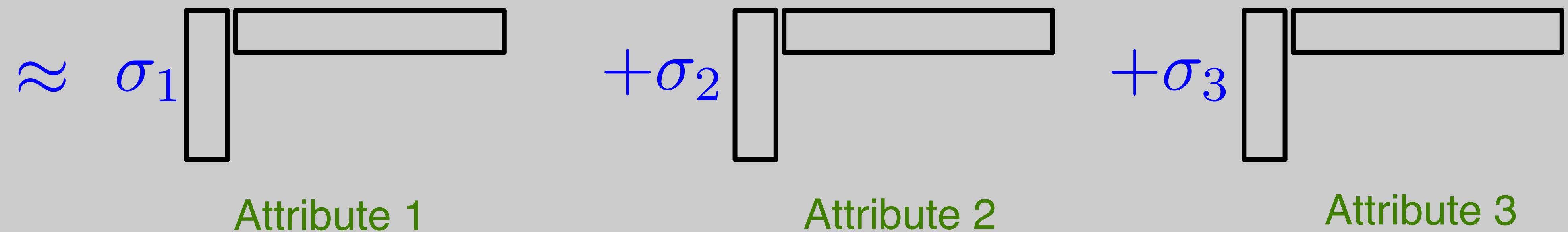
Miki belongs to class: like A2 don't like A1

# Prediction with SVD

Can try to predict preferences of a new customer with few ratings

See homework!

n movies																		
1	2	5	5	1	3	3	3	2	5	5	4	4	3	2	2	5	1	1
5	3	2	1	1	3	3	3	1	1	2	4	5	5	4	4	1	3	2
5	1	2	1	1	2	3	3	1	1	2	1	5	5	3	5	1	1	2
5	3	2	1	1	3	2	3	1	1	2	4	1	1	4	4	5	1	5
1	2	3	2	1	3	2	3	2	1	2	1	1	1	4	4	5	1	5
1	1	1	1	5	3	3	3	1	5	2	4	4	4	2	5	5	1	1
1	?	?	2	?	?	?	?	3	5	1	?	?	?	?	5	2	?	3



# Low-rank Completion

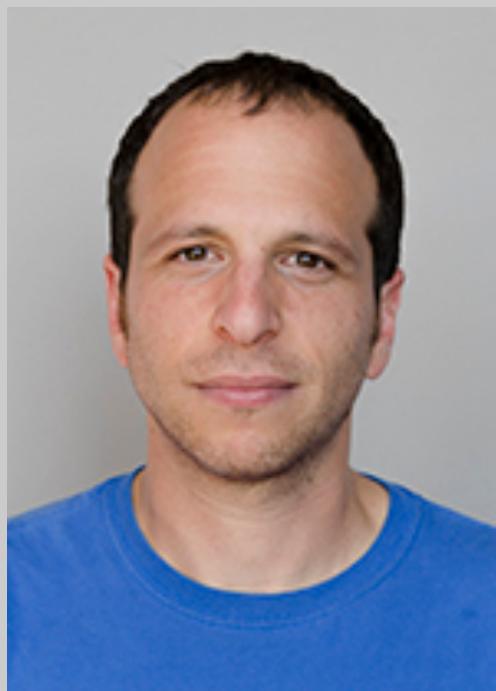
What if my database is full of “holes”?

Should be still low-rank!

n movies																		
m	views	1	5	5	1	3		3	2	5	5	4	3	2	2	5	1	
		5	3		1	1	3	3		1	1	2	5	4	4	1	3	2
		1	2	1	1		3	3		1	2	1	5	5	3	5	1	2
		5	3		1	3	2		1		2	4	1	4	5	5	5	5
		1	3	2	1	3	2	3	2	1		1	4	5	5	1	5	
		1	1		5	3	3		5	2	4		2	5	5	1	1	

Q) Can we complete missing data?

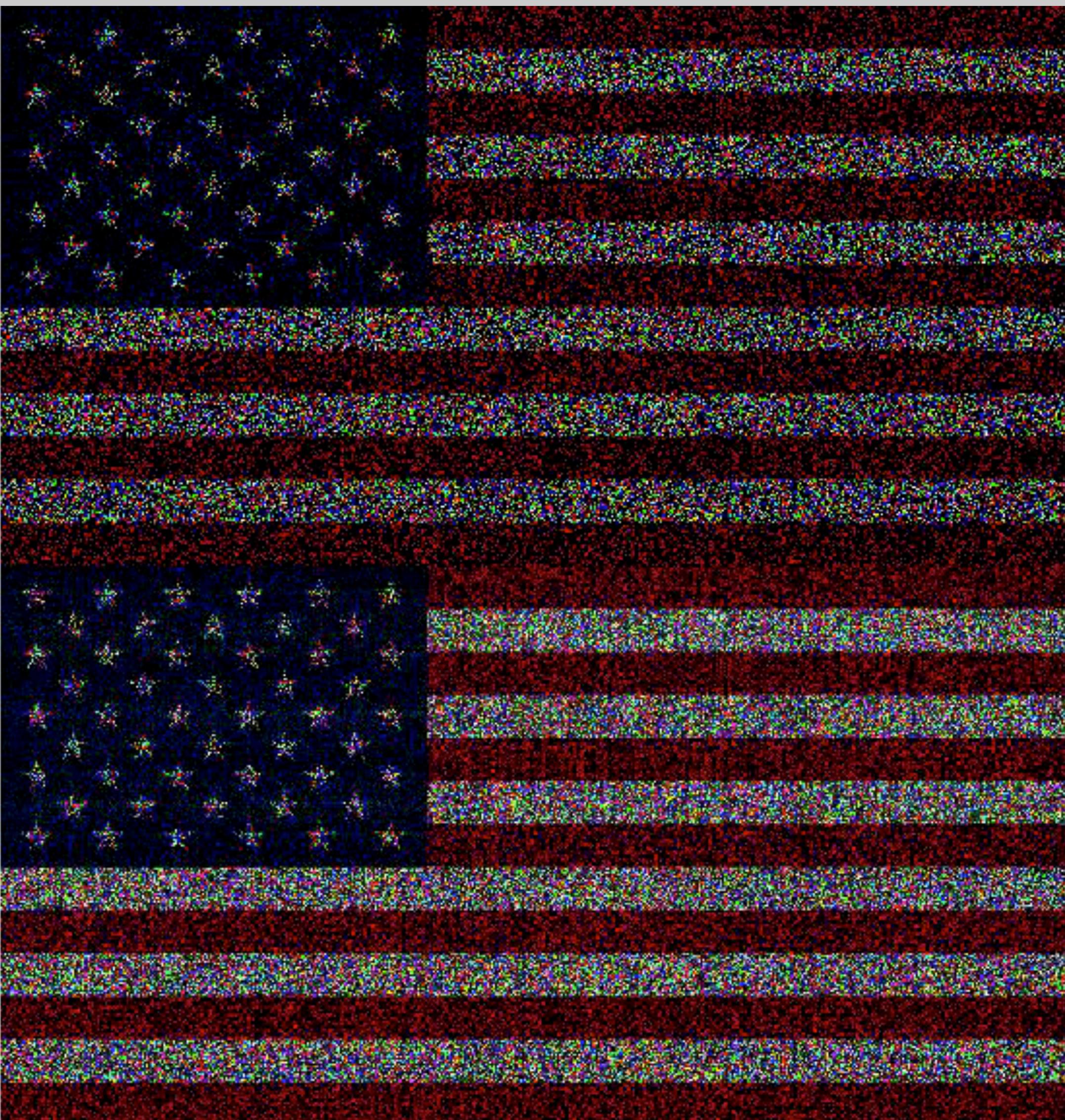
A) Sometimes! Very recent mathematical and practical results show you can.  
Keywords: Compressed Sensing, Low-rank completion, robust PCA

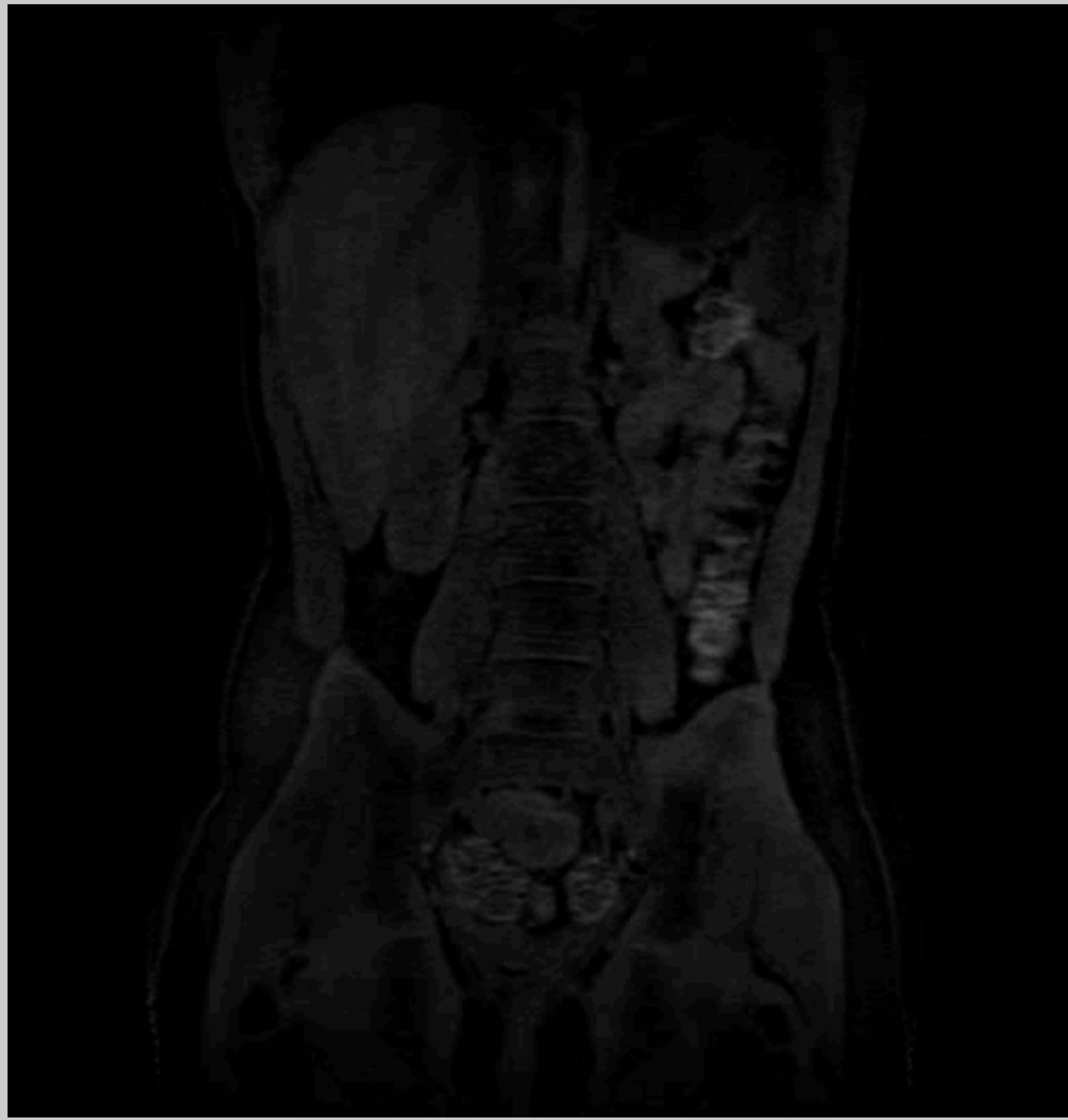


E. Candes and B. Recht, Foundation of Computational Mathematics, 2009;9:717

# Low-Rank Recovery from 20% pixels

- Algorithm for low-rank completion:
  - flag\_hat = flag
  - Compute [U,S,V] = svd(flag\_hat)
  - $$\text{flag}_{\text{hat}} = \sum_{i=0}^6 \sigma_i \vec{u}_i \vec{v}_i^T$$
  - update missing pixels in flag from flag\_hat
  - repeat (250 times here)





T. Zhang et. al, Pediatric Radiology 2015;45(11):1635-43  
EE16B M. Lustig, EECS UC Berkeley

# Input



1x1 (Sparse)

4x4

16x16



**Multi-scale  
Low Rank**