Singular Value Decomposition - Applications in Image Processing

Iveta Hnětynková

Katedra numerické matematiky, MFF UK

Outline

1. Singular value decomposition
2. Application 1 - image compression
3. Application 2 - image deblurring

1. Singular value decomposition

Consider a (real) matrix

$$A \in \mathcal{R}^{n \times m}, \;\; r = \text{rank}\,(A) \leq \min\,\{n, m\}\,.$$

A has

m	columns of length n ,
n	rows of lenght m ,
r	is the maximal number of linearly independent columns (rows) of A .

There exists an **SVD decomposition** of A in the form

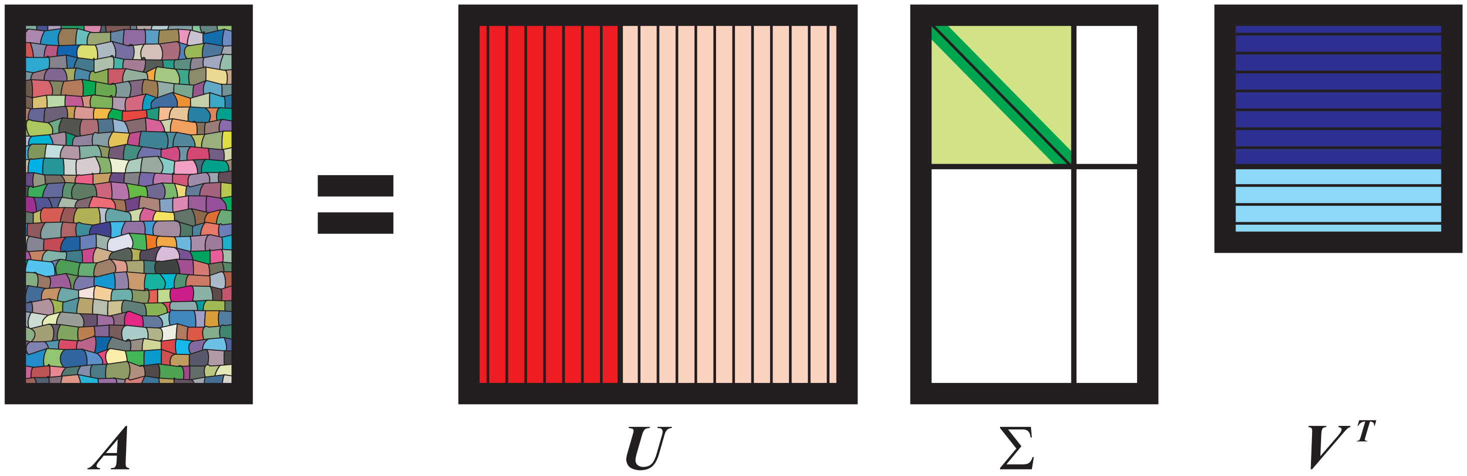
$$A = U \Sigma V^T,$$

where $U = [u_1, \ldots, u_n] \in \mathcal{R}^{n \times n}, V = [v_1, \cdots, v_m] \in \mathcal{R}^{m \times m}$ are orthogonal matrices, and

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{R}^{n \times m}, \quad \Sigma_r = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \in \mathcal{R}^{r \times r},$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0\,.$$

Singular value decomposition – the matrices:



$\{u_i\}_{i=1,\ldots,n}$ are **left singular vectors** (columns of U),

$\{v_i\}_{i=1,\ldots,m}$ are **right singular vectors** (columns of V),

$\{\sigma_i\}_{i=1,\ldots,r}$ are **singular values** of A.

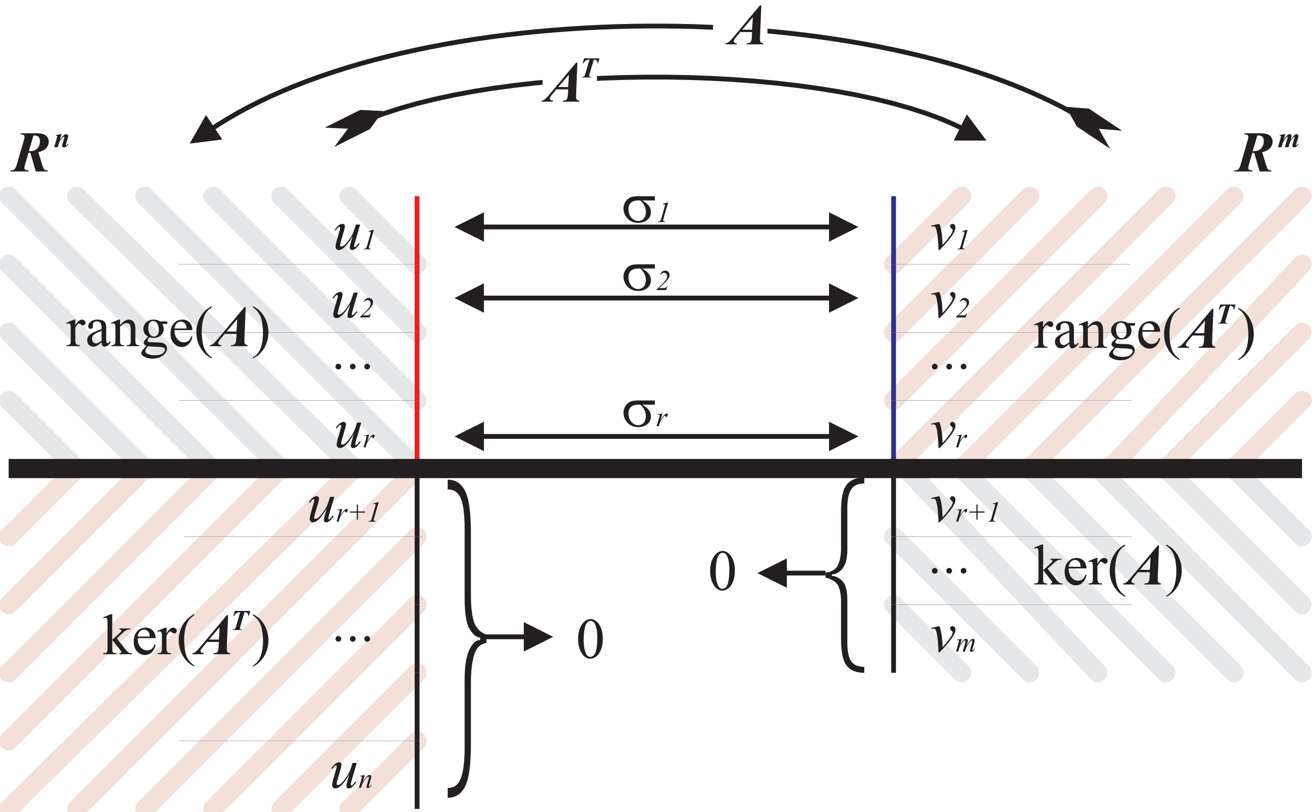
The SVD gives us:

$$\mathrm{span}\,(u_1,\ldots,u_r) \equiv \mathrm{range}\,(\mathrm{A}) \subset \mathcal{R}^n,$$
$$\mathrm{span}\,(v_{r+1},\ldots,v_m) \equiv \ker\,(\mathrm{A}) \subset \mathcal{R}^m,$$

$$\mathrm{span}\,(v_1,\ldots,v_r) \equiv \mathrm{range}\,(\mathrm{A}^{\mathsf{T}}) \subset \mathcal{R}^m,$$
$$\mathrm{span}\,(u_{r+1},\ldots,u_n) \equiv \ker\,(\mathrm{A}^{\mathsf{T}}) \subset \mathcal{R}^n,$$

spectral and Frobenius norm of A, rank of A, ...

Singular value decomposition – the subspaces:



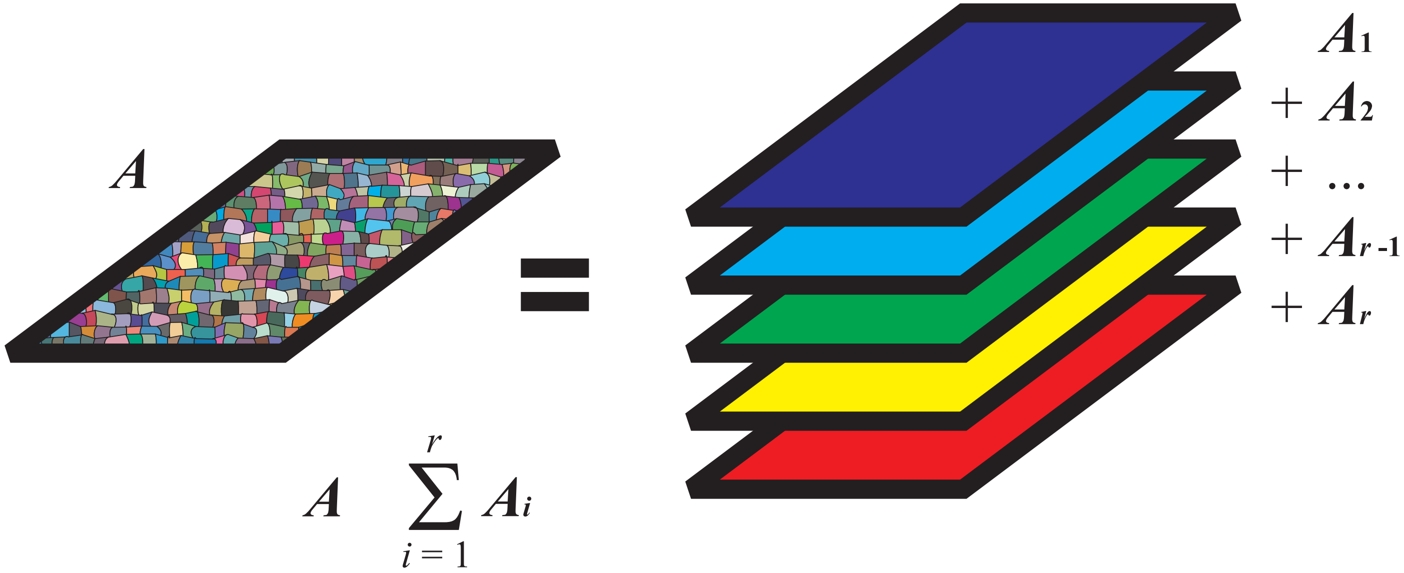
The outer product (dyadic) form:

We can rewrite A as a sum of rank-one matrices in the dyadic form

$$
\begin{aligned}
A &= U\,\Sigma\,V^T \\
&= [u_1, \ldots, u_r] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix} \\
&= u_1\sigma_1 v_1^T \; + \; \ldots \; + \; u_r\sigma_r v_r^T \\
&= \sum_{i=1}^{r} \sigma_i u_i v_i^T \\
&\equiv \sum_{i=1}^{r} A_i \, .
\end{aligned}
$$

Moreover, $\|A_i\|_2 = \sigma_i$ gives $\|A_1\|_2 \geq \|A_2\|_2 \geq \ldots \geq \|A_r\|_2 \, .$

Matrix A as a sum of rank-one matrices:



SVD reveals the dominating information encoded in a matrix. The first terms are the "most" important.

Optimal approximation of A with a rank-k:

The sum of the first k dyadic terms

$$\sum_{i=1}^{k} A_i \equiv \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

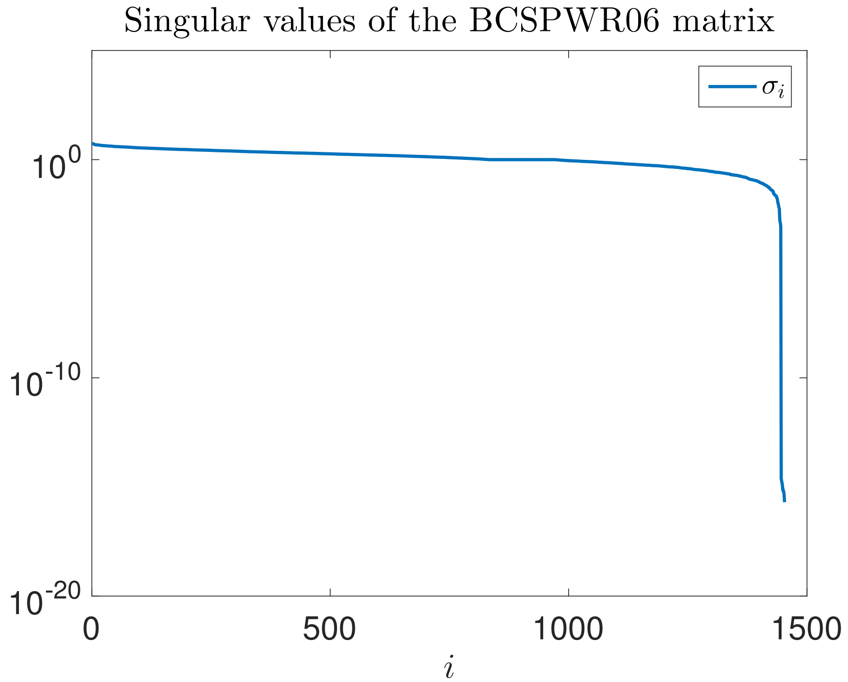
is the best rank-k approximation of the matrix A in the sense of minimizing the 2-norm of the approximation error, i.e.

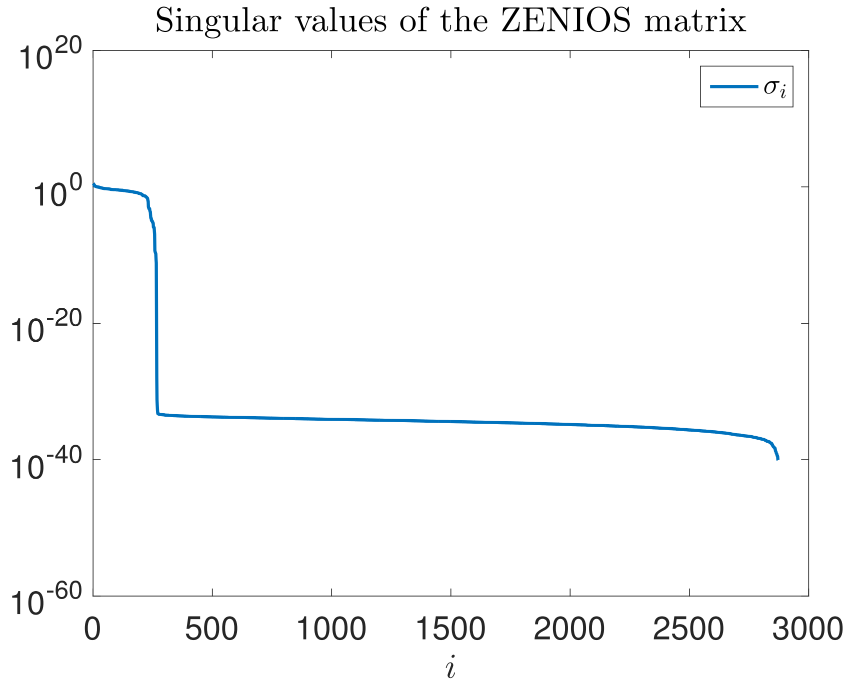
$$\sum_{i=1}^{k} u_i \sigma_i v_i^T = \underset{X \in \mathcal{R}^{n \times m},\, \text{rank}\,(X) \leq k}{\text{argmin}} \{\|A - X\|_2\}.$$

This allows to approximate A with a lower-rank matrix

$$A \approx \sum_{i=1}^{k} A_i \equiv \sum_{i=1}^{k} \sigma_i u_i v_i^T\,.$$

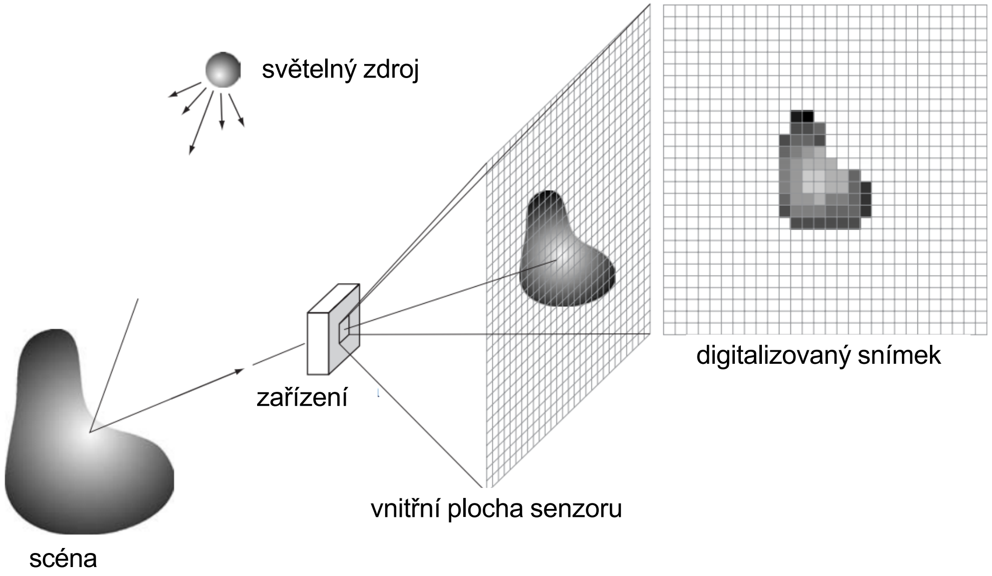
Different possible distributions of singular values:

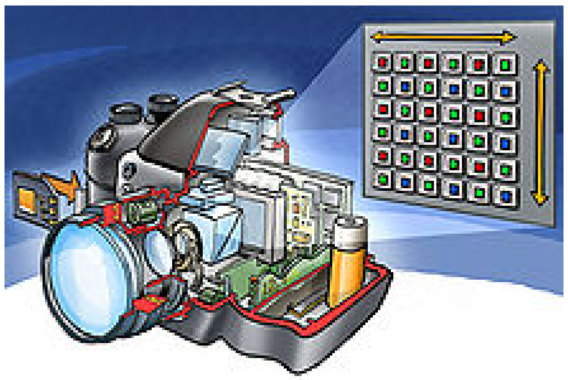




The matrices that are difficult (left) and easy (right) to approximate (BCSPWR06 and ZENIOS from the Harwell-Boeing Collection).

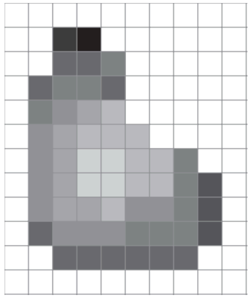
2. Application 1 - image compression





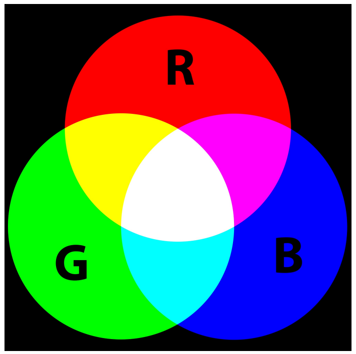
Grayscale image = matrix, each entry represents a pixel brightness.

Grayscale image: scale $0, \ldots, 255$ from black to white



$$
= \begin{bmatrix}
255 & 255 & 255 & 255 & 255 & \ldots & 255 & 255 & 255 \\
255 & 255 & 31 & 0 & 255 & \ldots & 255 & 255 & 255 \\
255 & 255 & 101 & 96 & 121 & \ldots & 255 & 255 & 255 \\
255 & 99 & 128 & 128 & 98 & \ldots & 255 & 255 & 255 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\
255 & 90 & 158 & 153 & 158 & \ldots & 100 & 35 & 255 \\
255 & 255 & 102 & 103 & 99 & \ldots & 98 & 255 & 255 \\
255 & 255 & 255 & 255 & 255 & \ldots & 255 & 255 & 255
\end{bmatrix}
$$

Colored image: 3 matrices for Red, Green and Blue brightness values



MATLAB DEMO: Low rank image approximation

Approximate a grayscale image A using its SVD

$A_k = \sum_{i=1}^{k} A_i$ ... best rank k approximation

Compare storage requirements and quality for different k.

Memory required to store:

an uncompressed image of size $m \times n$: mn values

rank k SVD approximation: $k(m+n+1)$ values

Original image and its approximation by 33% of components:

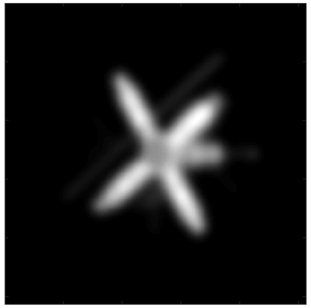
Consequently, in $A = A_1 + A_2 + ... + A_r$:

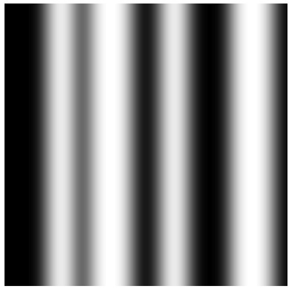
- the first terms represent dominant information
- the last terms represent details (edges)

3. Application 2 - image deblurring



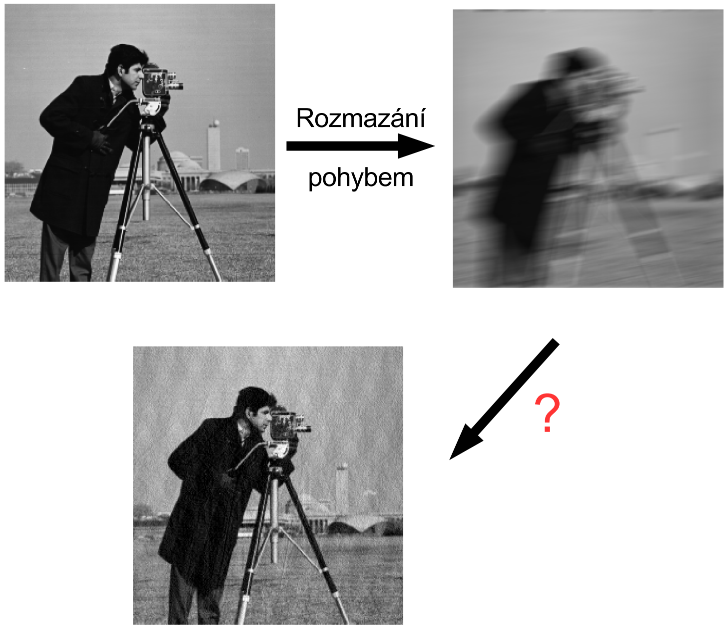




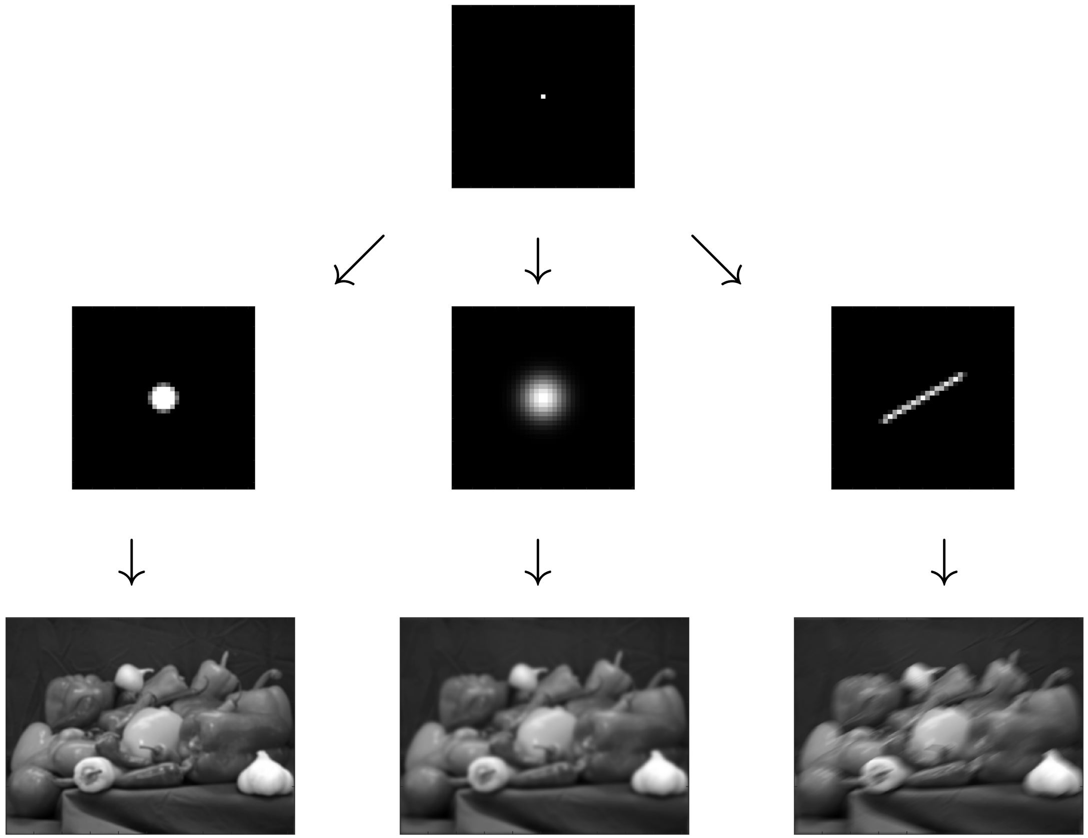


Sources of noise and blurring: physical sources (moving objects, lens out of focus), measurement, discretization, rounding errors, ...

Challenge: Having some information about the blurring process, try to approximate the "exact" image.

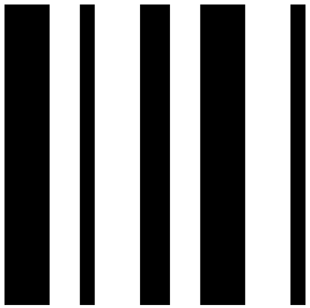
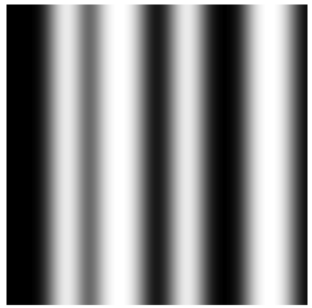


PSF (point spread function) = blurring model for a single pixel



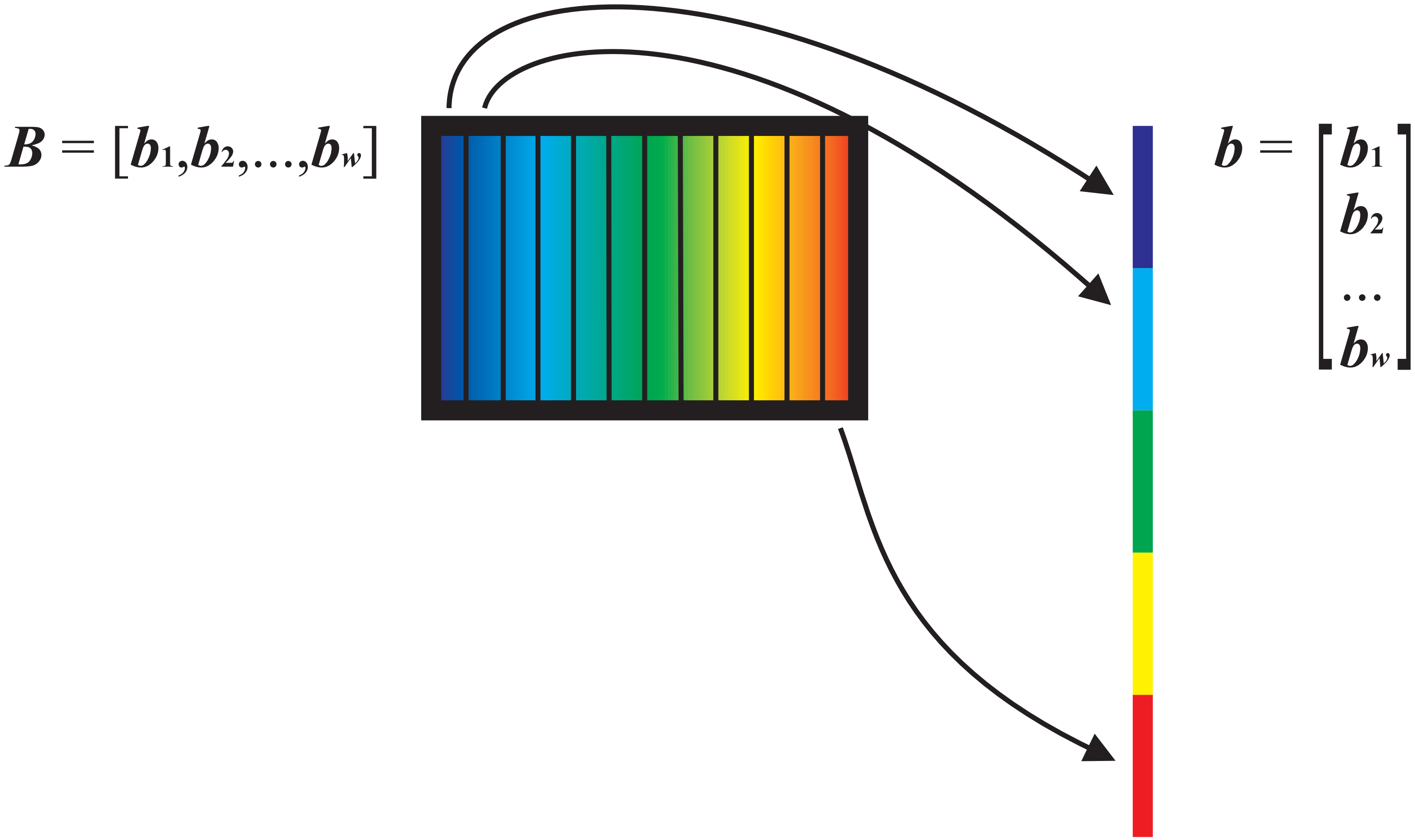
Model of blurring process: CONVOLUTION

Blurred photo:  $\longrightarrow$ 

Barcode scanning:  $\longrightarrow$ 

X(exact image) $\mathcal{A}$(blurring operator) B(blurred noisy image)

Image vectorization $B \rightarrow b = \text{vec}\,(B)$**:**



Obtaining a linear model:

Using some discretization techniques, it is possible to transform this problem to a linear problem

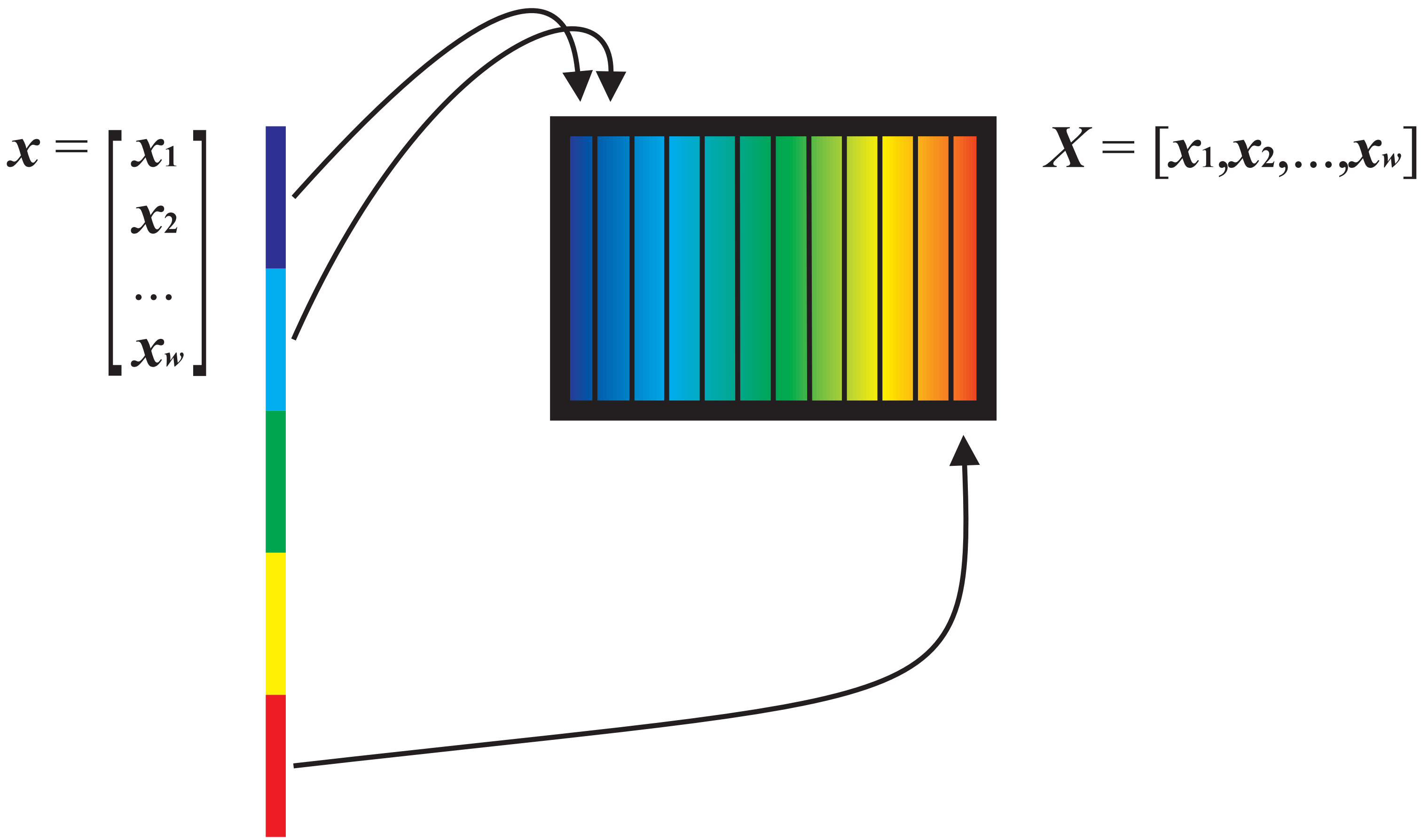
$$Ax = b, \quad A \in \mathcal{R}^{n \times n}, \quad x, b \in \mathcal{R}^n,$$

where

- A is a discretization of $\mathcal{A}$,
- $b = \text{vec}(B)$,
- $x = \text{vec}(X)$.

Size of the problem: $n =$ number of pixels in the image, e.g., even for a low resolution 456 x 684 px we get 311 904 equations.

Solution back reshaping $x = \text{vec}(X) \rightarrow X$:



Solution of the linear problem:

Let A be nonsingular. Then $Ax = b$ has the unique solution

$$x^{\text{naive}} = A^{-1}b.$$

X B naive solution

Why? Because of specific properties of our problem.

The image always contains errors (noise):

 + **NOISE** = 

exact B $\qquad\qquad$ B

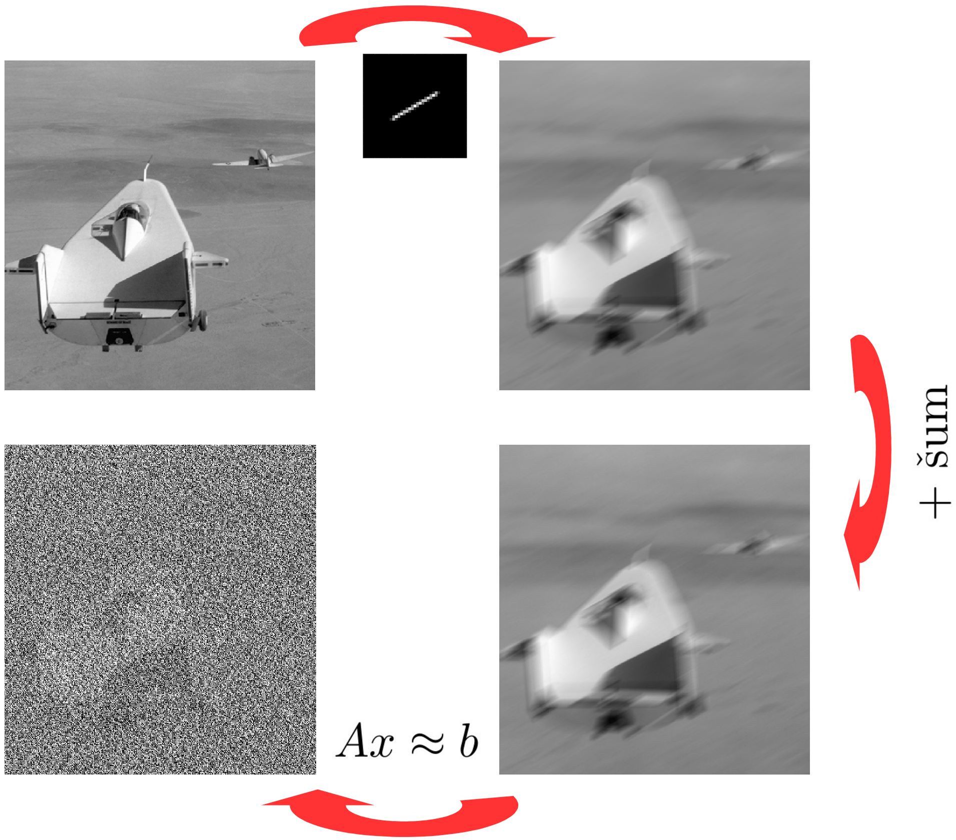
Assuming noise is additive, our linear model is

$$Ax \approx b, \quad b = b^{\text{exact}} + b^{\text{noise}},$$

where

$$\|b^{\text{exact}}\| \gg \|b^{\text{noise}}\| \qquad \text{BUT} \qquad \|A^{-1}b^{\text{exact}}\| \ll \|A^{-1}b^{\text{noise}}\|.$$

Schema of the naive approach: noise amplification



Usual properties of the model $Ax \approx b$:

- ill-posedness = sensitivity of x to small changes in b;

- singular values σ_j of A decay quickly
  $\longrightarrow$ $\sigma_j \approx 0$ for many singular values,
  $\longrightarrow$ A has a large condition number;

- b^{exact} is smooth, and satisfies the discrete Picard condition (DPC);

- b^{noise} is often random and does not satisfy DPC.

SVD components of the naive solution:

From the SVD of A we have

$$
\begin{aligned}
x^{\text{naive}} \equiv A^{-1} b &= \sum_{j=1}^{n} \left(\frac{1}{\sigma_j} v_j \, u_j^T \right) \, b \\
&= \sum_{j=1}^{n} \frac{u_j^T b}{\sigma_j} v_j \\
&= \underbrace{\sum_{j=1}^{n} \frac{u_j^T b^{\text{exact}}}{\sigma_j} v_j}_{x^{\text{exact}} = A^{-1} b^{\text{exact}}} + \underbrace{\sum_{j=1}^{n} \frac{u_j^T b^{\text{noise}}}{\sigma_j} v_j}_{A^{-1} b^{\text{noise}}} .
\end{aligned}
$$

What is the size of the right sum (inverted noise) in comparison to the left one?

Exact data: on average, $|u_j^T b^{\text{exact}}|$ decay faster than σ_j (DPC).

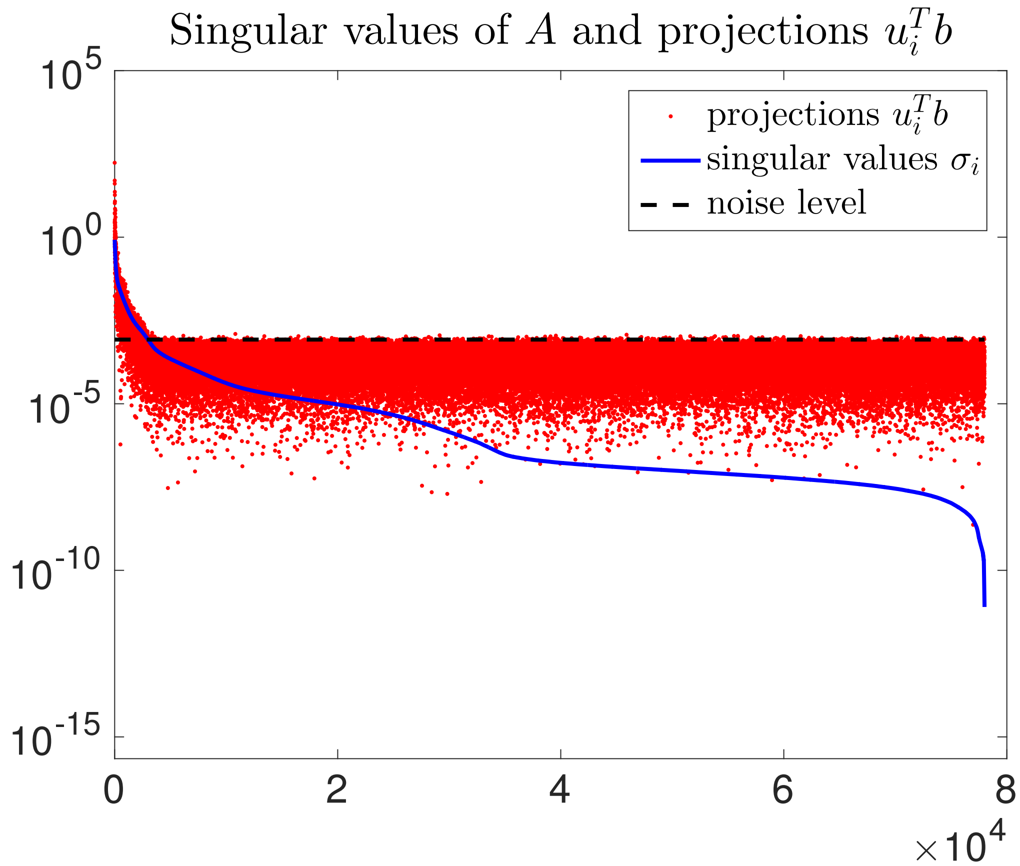
White noise: the values $|u_j^T b^{\text{noise}}|$ do not exhibit any trend.

Thus the coefficients $u_j^T b = u_j^T b^{\text{exact}} + u_j^T b^{\text{noise}}$ are:

- for small j dominated by the exact part,
- for large j dominated by the noisy part.

By the division by σ_j, the noisy components of the naive solution corresponding to small singular values are amplified.

Violation of DPC due to presence of noise in b:



Basic regularization method - Truncated SVD:

Using the dyadic form

$$A = U\,\Sigma\,V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T,$$

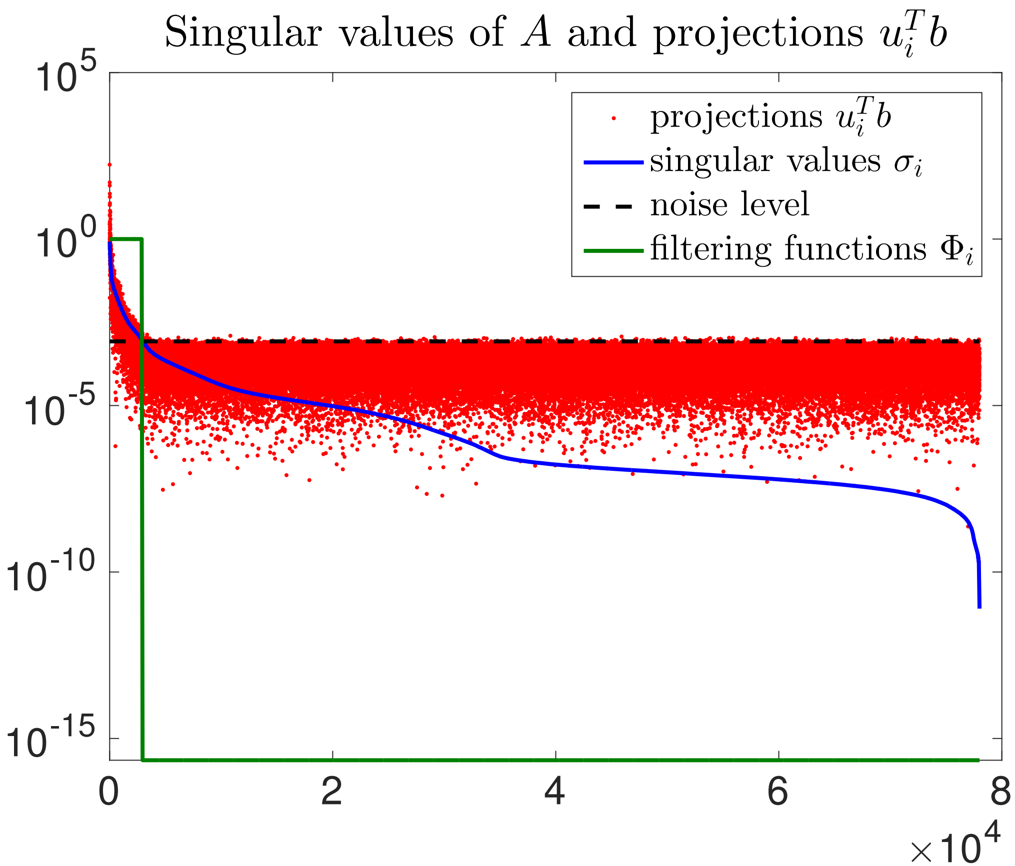
we can approximate A with a rank k matrix

$$A \approx S_k \equiv \sum_{i=1}^{k} A_i = \sum_{i=1}^{k} u_i\,\sigma_i\,v_i^T.$$

Replacing A by S_k gives an TSVD approximate solution

$$x^{(k)} = \sum_{j=1}^{k} \frac{u_j^T b}{\sigma_j} v_j\,.$$

TSVD regularization: removing of troublesome components



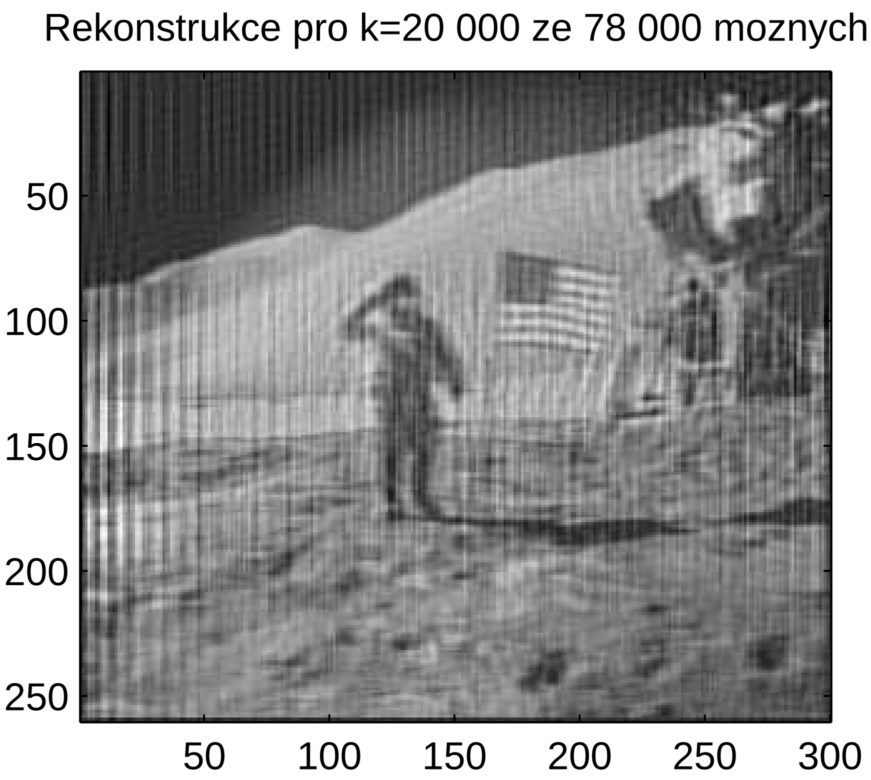
Here the smallest σ_j's are not present. However, we removed also some components of x^{exact}.

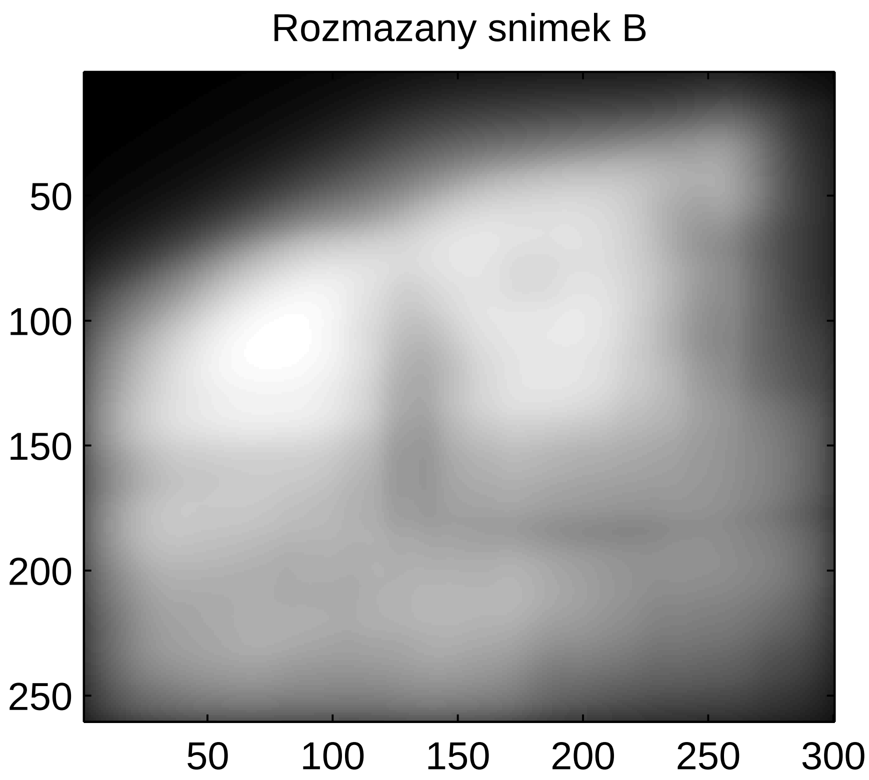
Selection of k: It depends on the amount of noise, image properties, etc. An optimal k has to balance between:

- removing noisy components,
- not losing too many components of the exact solution.

MATLAB DEMO: Compute TSVD regularized solutions for different values of k. Compare quality of the obtained image.

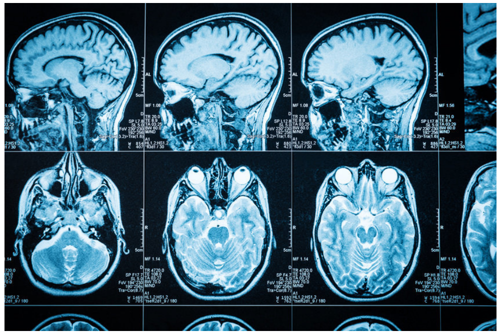
Comparison of blurred noisy image and its TSVD approximation

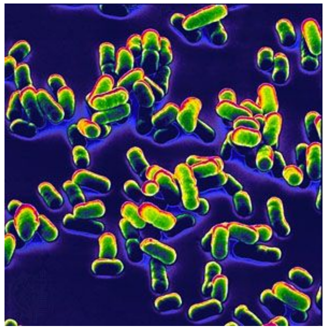




Various sources of images in applications:

CT, MRI, PET, electron microscopy, radar/sonar imaging, ...

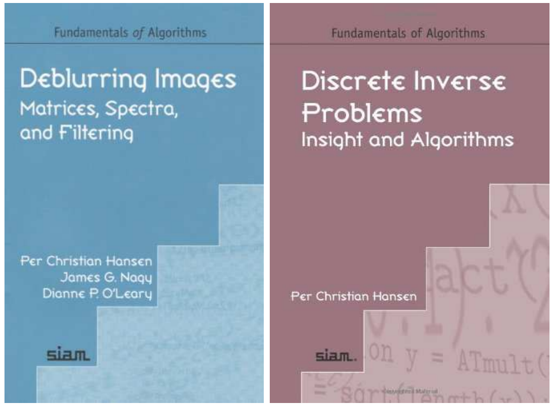




X-Ray application: radiologists selfie

References:

Textbooks:



- Hansen, Nagy, O'Leary: *Deblurring Images, Spectra, Matrices, and Filtering*, SIAM, 2006.
- Hansen: *Discrete Inverse Problems, Insight and Algorithms*, SIAM, 2010.

Software (MatLab toolboxes): on the homepage of P. C. Hansen

- HNO package,
- Regularization tools,
- AIRtools,
- ...