# Mathematical Models for Image Processing: Linear and Nonlinear Approaches

Tibor Lukić

Full Professor of Theoretical and Applied Mathematics

Faculty of Technical Sciences

University of Novi Sad, Novi Sad, Serbia

SSIP 2026

Timisoara, Romania

# NOVI SAD, Serbia



# NOVI SAD









# NOVI SAD / PETROVARADIN FORTRESS



# UNIVERSITY OF NOVI SAD



UNS campus

*Faculty of Technical Sciences*

1200 employee

2450 first year students



# SUMMER SCHOOL ON IMAGE PROCESSING 2017, NOVI SAD







# SSIP23, NOVI SAD









# SSIP23, NOVI SAD

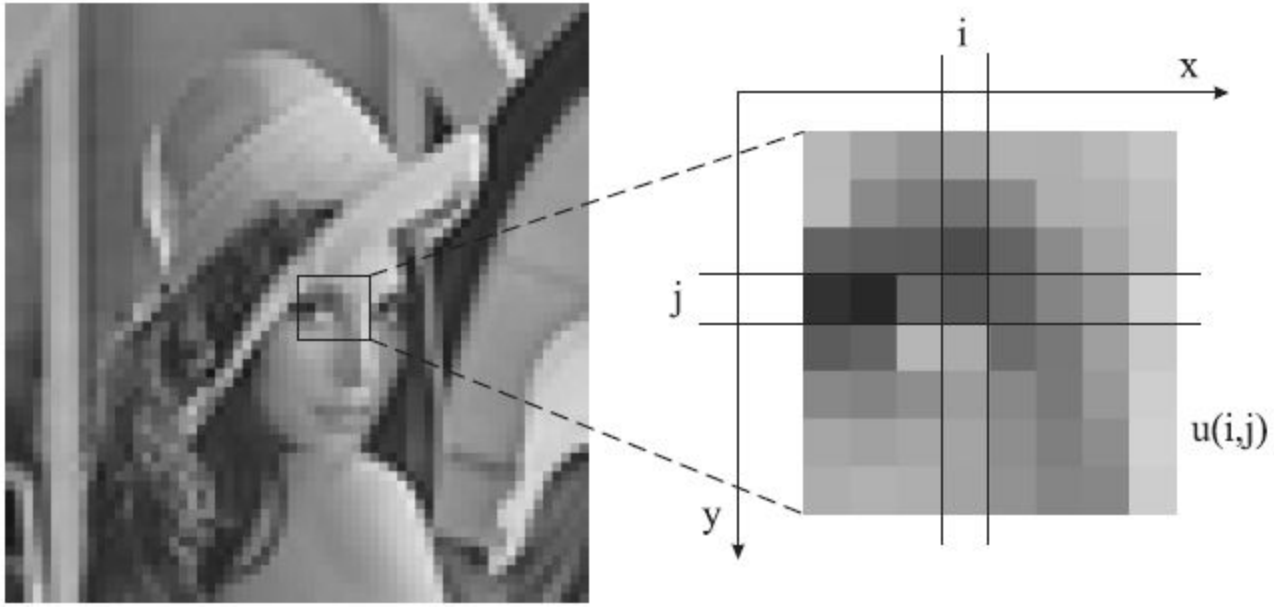


# OUTLINE

LINEAR MODELS

NON-LINEAR MODELS

# DIGITAL IMAGE




$$u : \Omega \longrightarrow \mathbb{R}$$

$$u(i,j) \longrightarrow \Omega \subset \mathbb{Z}^2$$

Digital image function

**LINEAR MODELS**

$$u(x,y) \longrightarrow \Omega \subset \mathbb{R}^2$$

Continuously differentiable function

NON-LINEAR MODELS

# LINEAR TRANSFORMATION

## Linearity

Additivity: $\mathcal{L}(f+g) = \mathcal{L}(f) + \mathcal{L}(g), \quad f, g -$ digital images

Homogeneity: $\mathcal{L}(\alpha f) = \alpha \mathcal{L}(f), \quad \alpha \in \mathbb{R}$

$$\mathcal{L}(f) = \begin{bmatrix} \text{MATRIX} \\ \text{OPERATOR} \end{bmatrix} \bullet \begin{bmatrix} f \end{bmatrix}$$

# LINEAR TRANSFORMATIONS

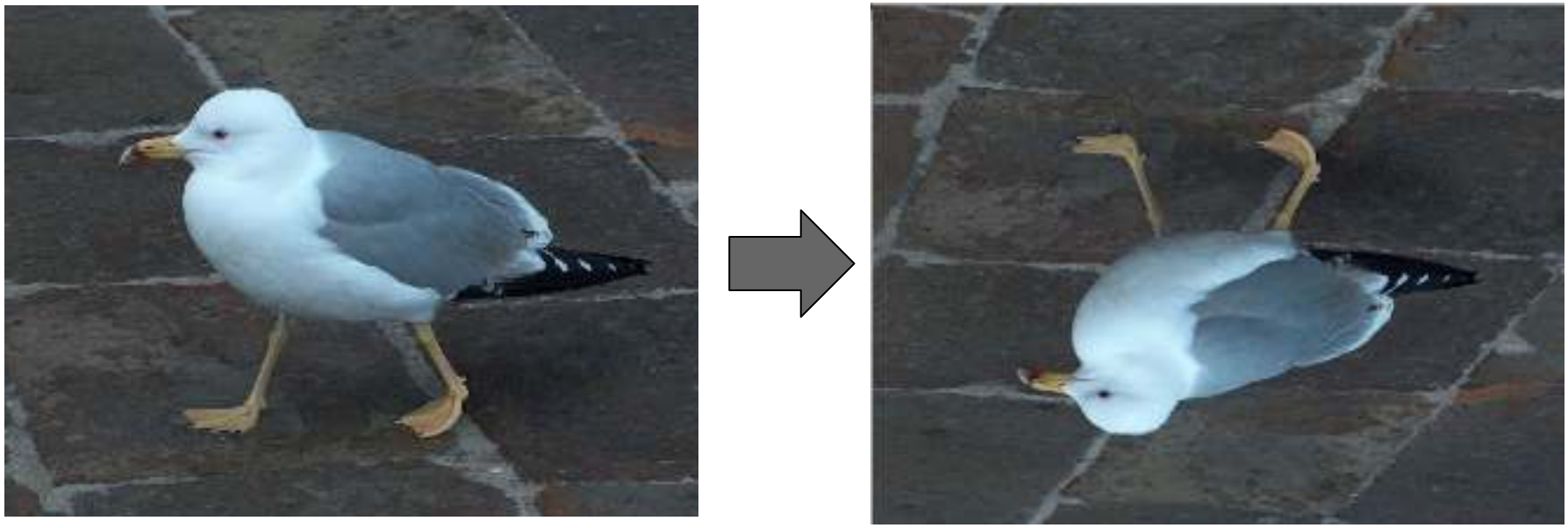
$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- Scaling
- Rotation
- Shearing
- Reflection

It maps straight lines to straight lines and the origin to the origin.

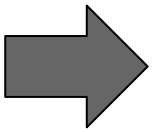
# LINEAR TRANSFORMATIONS



$$T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

# LINEAR TRANSFORMATION







$$T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

# LINEAR TRANSFORMATION

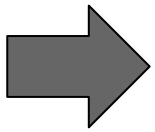




$$T = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$$

# LINEAR TRANSFORMATIONS







$$T = \begin{bmatrix} \cos(30^0) & -\sin(30^0) \\ \sin(30^0) & \cos(30^0) \end{bmatrix}$$

# LINEAR TRANSFORMATION





$$T = \begin{bmatrix} 2.1 & 0.1 \\ 0 & 1 \end{bmatrix}$$

# LINEAR TRANSFORMATION



$$T = \begin{bmatrix} -1 & 0 \\ 0.4 & 0.5 \end{bmatrix}$$

# LINEAR TRANSFORMATION



# CONVOLUTION

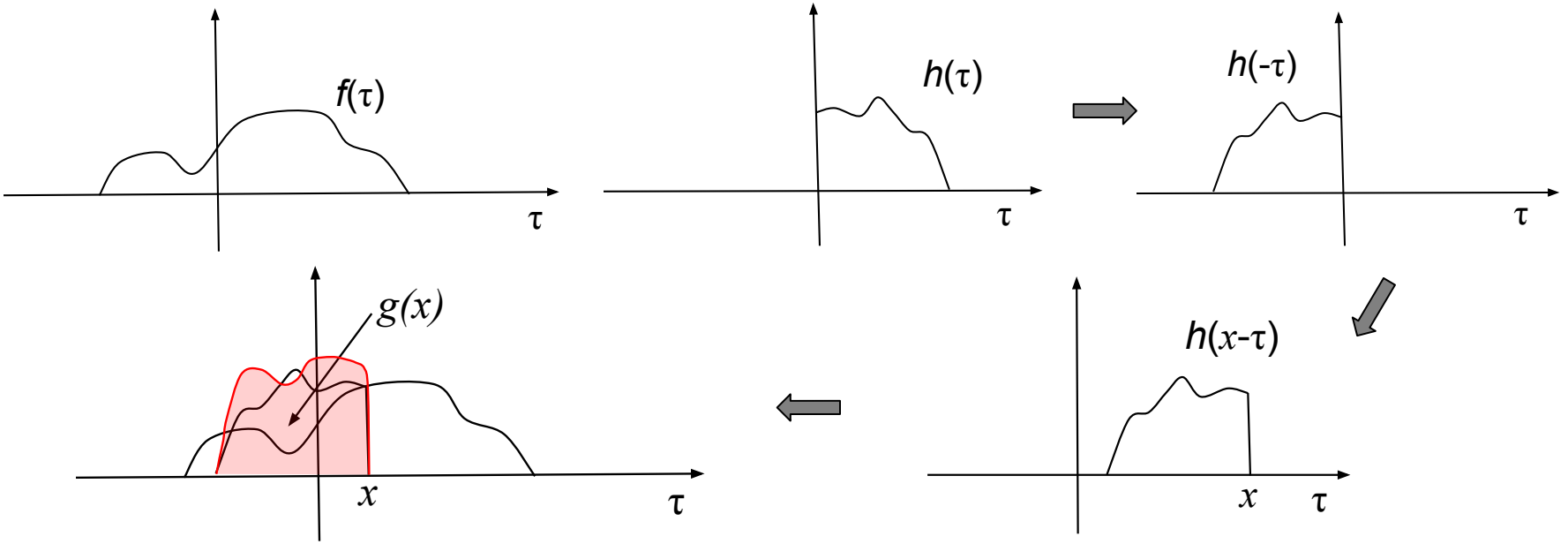
## 1D CASE

Convolution of two functions $f(x)$ and $h(x)$

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$

$h(x)$ – filter (kernel, impulse response)

Convolution is linear.



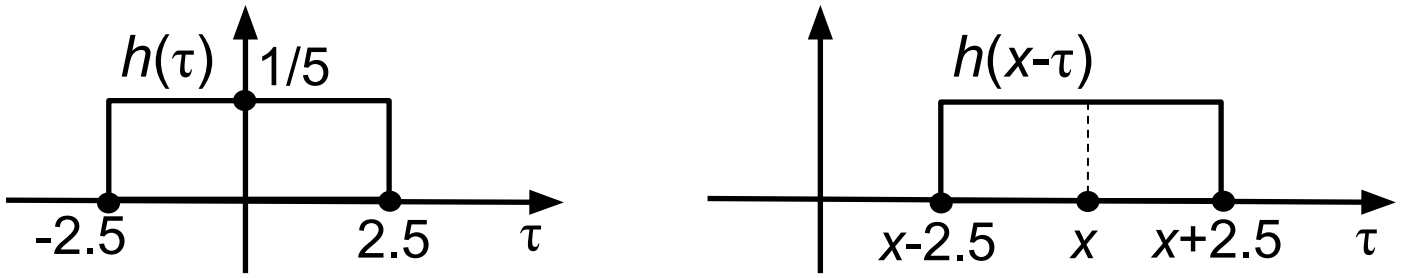
## 1D CASE

Recall.

$$\frac{1}{b-a}\int_a^b f(x)\,dx \;=$$

Mean value of *f(x)* at the interval [a, b].

Smoothing (averaging) filter example.




$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$

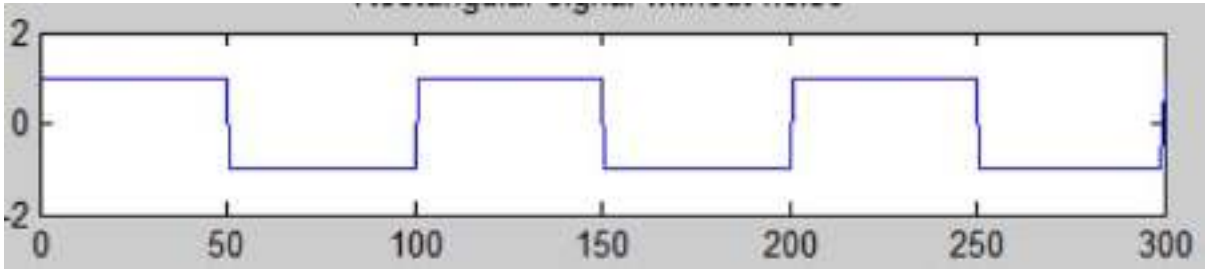
$$= \frac{1}{5}\int_{x-2.5}^{x+2.5} f(\tau)\,d\tau =$$

Mean value of *f(x)* at the interval [x-2.5, x+2.5].

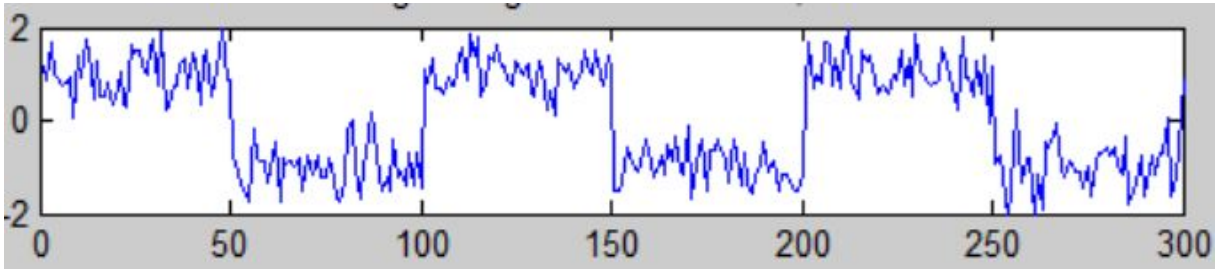
# CONVOLUTION

## 1D CASE

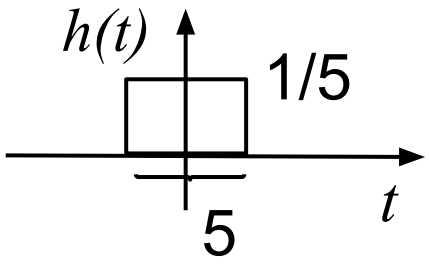
Original signal

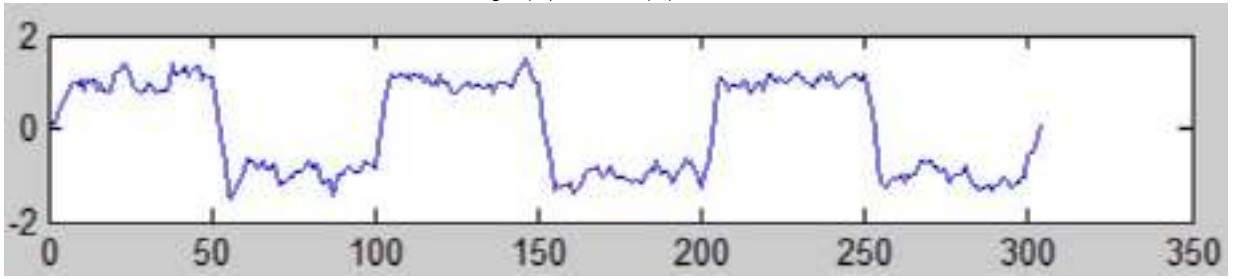


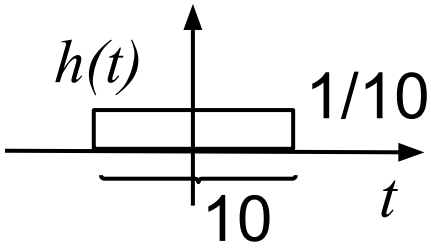

Noisy signal, *f(t)*

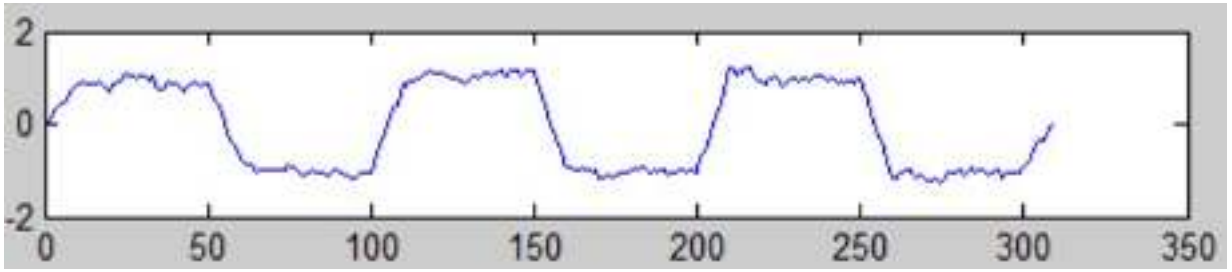



*f(t)*h(t):*



# CONVOLUTION

## 2D CASE

$$g(x,y) = f(x,y) \ * \ h(x,y) = \int\int_{-\infty}^{\infty} f(\tau,\mu)h(x-\tau,y-\mu)\,d\tau d\mu$$

$h(x,y)$ – "filter", "mask", "kernel", "Point Spread Function" (PSF)

$$g(i,j) = f(i,j) \ * \ h(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N} f(i,j)h(i-m,j-n)$$

| | | |
|---|---|---|
| $h_{1,1}$ | $h_{1,2}$ | $h_{1,3}$ |
| $h_{2,1}$ | $h_{2,2}$ | $h_{2,3}$ |
| $h_{3,1}$ | $h_{3,2}$ | $h_{3,3}$ |

3x3

| | | | | |
|---|---|---|---|---|
| $h_{1,1}$ | $h_{1,2}$ | $h_{1,3}$ | $h_{1,4}$ | $h_{1,5}$ |
| $h_{2,1}$ | $h_{2,2}$ | $h_{2,3}$ | $h_{2,4}$ | $h_{2,5}$ |
| $h_{3,1}$ | $h_{3,2}$ | $h_{3,3}$ | $h_{3,4}$ | $h_{3,5}$ |
| $h_{4,1}$ | $h_{4,2}$ | $h_{4,3}$ | $h_{4,4}$ | $h_{4,5}$ |
| $h_{5,1}$ | $h_{5,2}$ | $h_{5,3}$ | $h_{5,4}$ | $h_{5,5}$ |

5x5

Convolution masks

# CONVOLUTION



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $u_{1,1}$ | $u_{1,2}$ | $u_{1,3}$ | $u_{1,4}$ | $u_{1,5}$ | $u_{1,6}$ | $u_{1,7}$ | $u_{1,8}$ |
| $u_{2,1}$ | $u_{2,2}$ | $u_{2,3}$ | $u_{2,4}$ | $u_{2,5}$ | $u_{2,6}$ | $u_{2,7}$ | $u_{2,8}$ |
| $u_{3,1}$ | $u_{3,2}$ | $u_{3,3}$ | $u_{3,4}$ | $u_{3,5}$ | $u_{3,6}$ | $u_{3,7}$ | $u_{3,8}$ |
| $u_{4,1}$ | $u_{4,2}$ | $u_{4,3}$ | $u_{4,4}$ | $u_{4,5}$ | $u_{4,6}$ | $u_{4,7}$ | $u_{4,8}$ |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | $u_{8,8}$ |

$*$

| | | |
|---|---|---|
| $h_{1,1}$ | $h_{1,2}$ | $h_{1,3}$ |
| $h_{2,1}$ | $h_{2,2}$ | $h_{2,3}$ |
| $h_{3,1}$ | $h_{3,2}$ | $h_{3,3}$ |

$= \quad g(i,j)$

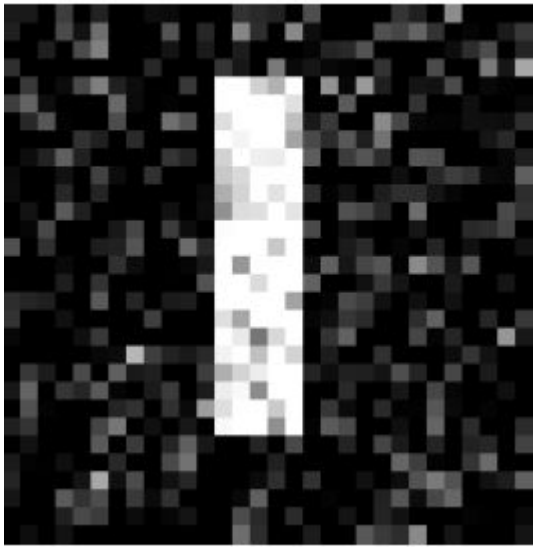
$$g(2,2) = u_{1,1} \cdot h_{1,1} + u_{1,2} \cdot h_{1,2} + u_{1,3} \cdot h_{1,3} + u_{2,1} \cdot h_{2,1} \ldots + u_{3,3} \cdot h_{3,3}$$

$$g(2,6) = u_{1,5} \cdot h_{1,1} + u_{1,6} \cdot h_{1,2} + u_{1,7} \cdot h_{1,3} + u_{2,5} \cdot h_{2,1} \ldots + u_{3,7} \cdot h_{3,3}$$

Border problem: $g(1,1) =?,\ g(1,2) =?,\ g(2,1) =?, \ldots$

# CONVOLUTION

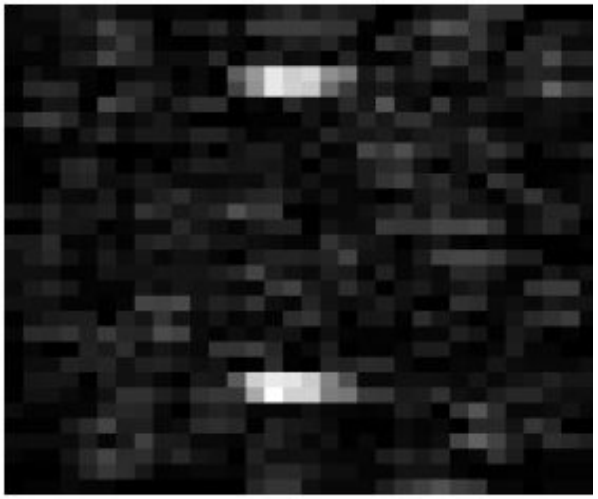
## 2D CASE

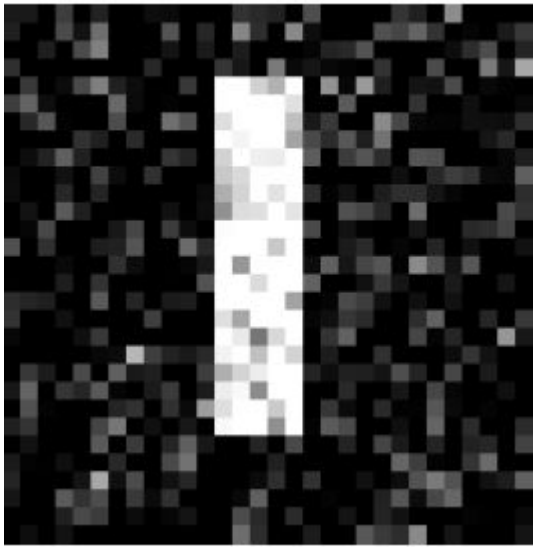


*

| | | |
|---|---|---|
| −1 | −1 | −1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Prewitt mask
horizontal edge

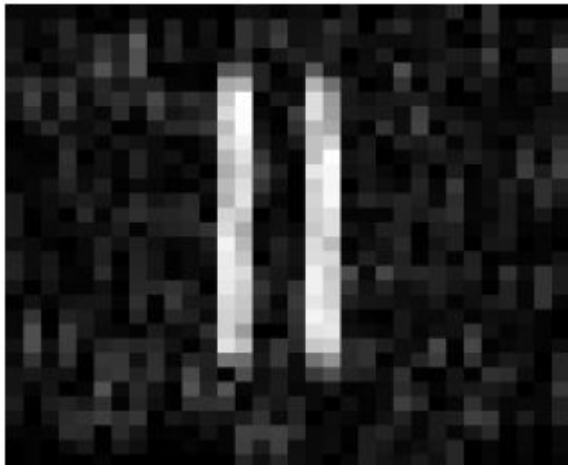




*

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −1 | 0 | 1 |
| −1 | 0 | 1 |

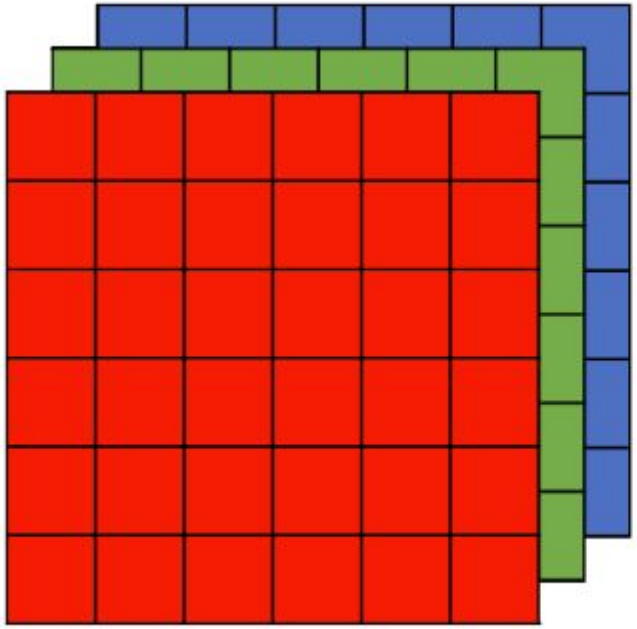
Prewitt mask
vertical edge



# CONVOLUTIONS

## 2D CASE

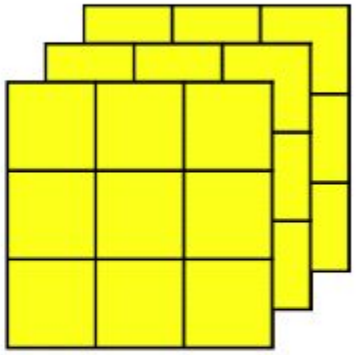
### Convolutions of RGB color images



$6 \times 6 \times 3$

3 channel RGB image

*



$3 \times 3 \times 3$

The convolution kernel is applied separately to the Red, Green, and Blue channels, and then the results are recombined into a new RGB image

# CONVOLUTIONS

## 2D CASE



3 channel RGB image

*

| | | |
|---|---|---|
| −1 | −1 | −1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Prewitt mask
horizontal edge



*

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −1 | 0 | 1 |
| −1 | 0 | 1 |

Prewitt mask
vertical edge



# CONVOLUTIONS

## 2D CASE



*

| | | |
|---|---|---|
| −1 | −1 | −1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Prewitt mask
horizontal edge



*

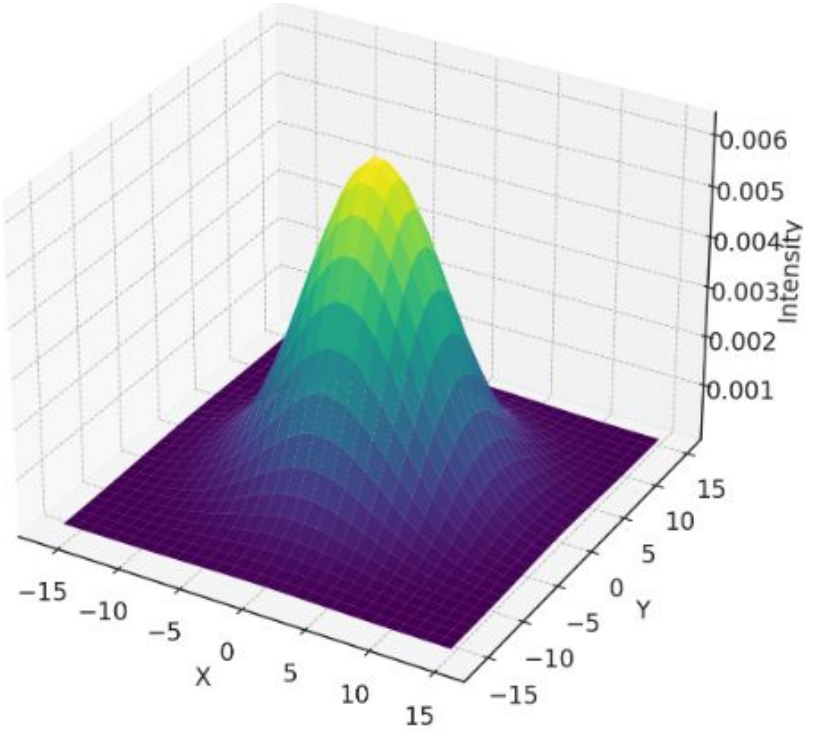
| | | |
|---|---|---|
| −1 | 0 | 1 |
| −1 | 0 | 1 |
| −1 | 0 | 1 |

Prewitt mask
vertical edge



# CONVOLUTION

2D Gaussian image filter




$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ is the standard deviation (controls the spread or "blurriness").

Effect: smoothing or blurring an image.

# DENOISING BY CONVOLUTION

## 2D CASE Example



Original noisy image

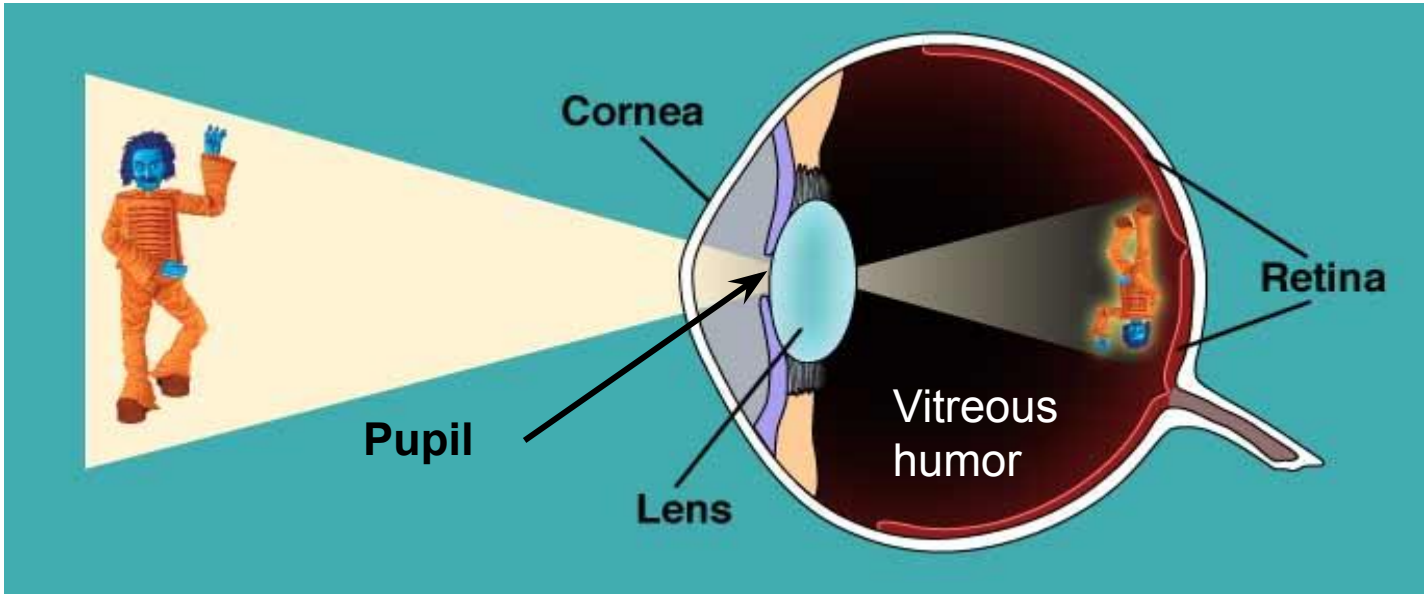


Gaussian, linear filter



Bilateral, non-linear filter

## Human visual system - image formation

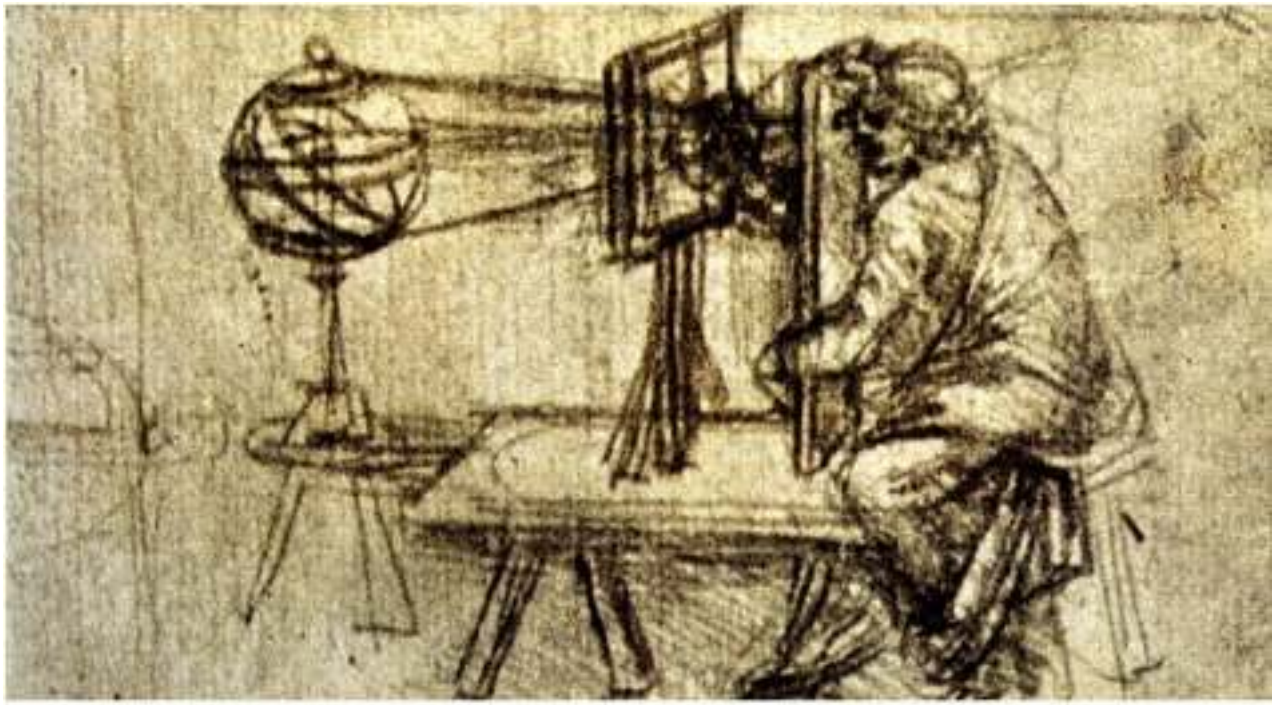



Vision is the **perception** of 3D world from its 2D partial projections:

$R^3$ (world) ⟶ $R^2$ (image plane)

Vision is an **ill-posed inverse problem**, because the depth and range information are lost.

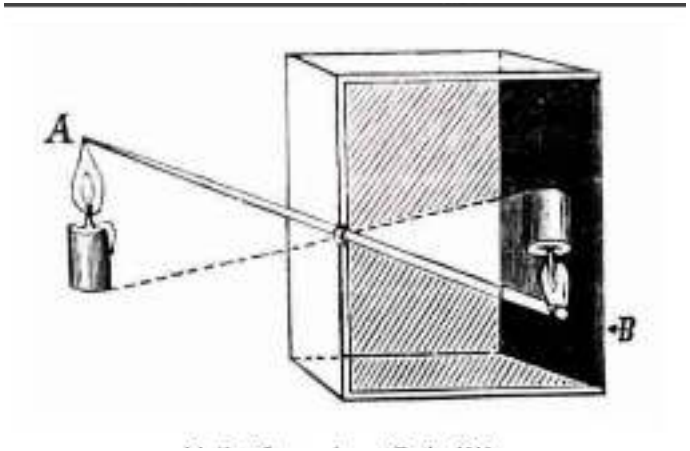
# COMPUTER VISION: PROJECTIVE CAMERA MODEL



The **camera obscura** sketched by **Leonardo da Vinci** in Codex Atlanticus (1515).

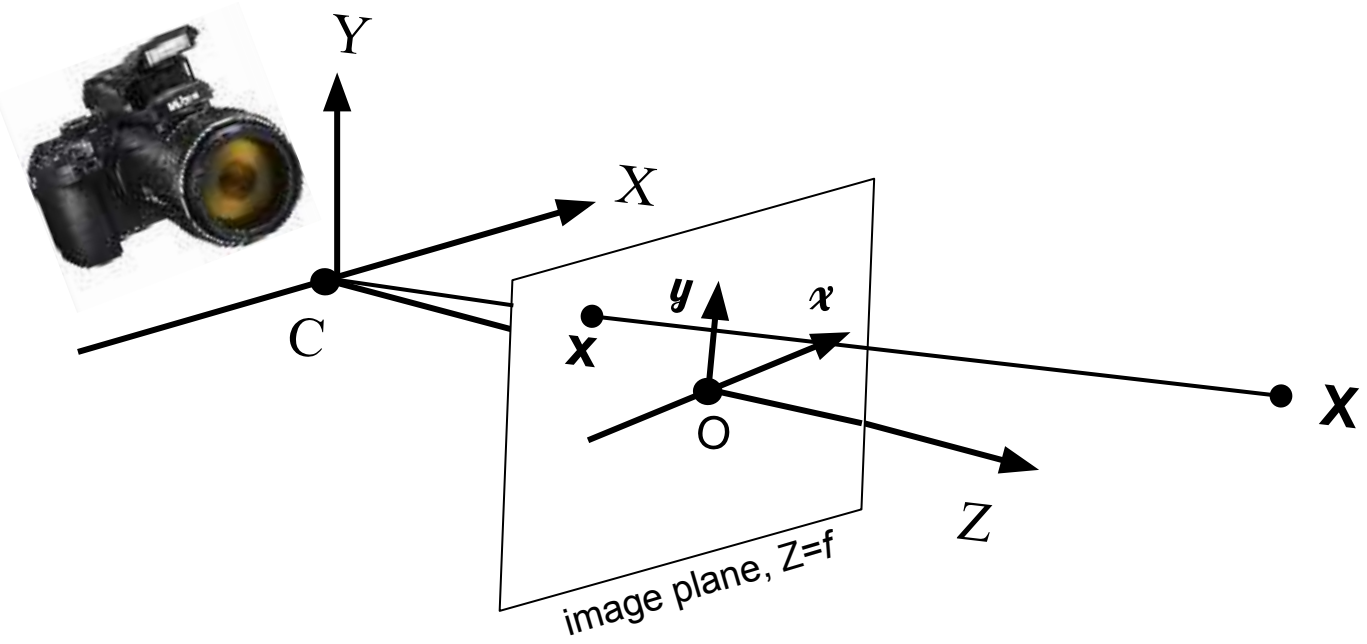
camera = room/chamber

obscura = dark

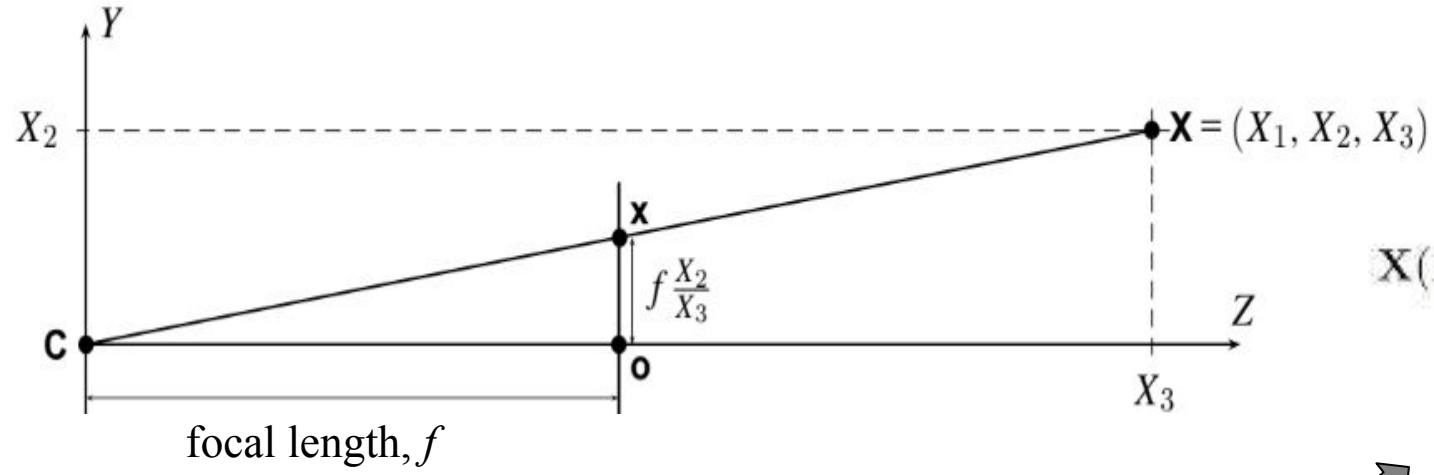



The camera obscura or the **pinhole camera,** Fizyka (1910).

# COMPUTER VISION: PROJECTIVE CAMERA MODEL




Projection of the 3D point onto the image plane.




$$\mathbf{X}(X_1, X_2, X_3) \rightarrow \mathbf{x}(x_1, x_2)$$

$$x_1 = f\frac{X_1}{X_3}$$

$$x_2 = f\frac{X_2}{X_3}$$

This transformation is **not linear**!

# COMPUTER VISION: PROJECTIVE CAMERA MODEL

Therefore, switching to **homogeneous coordinates**.

$$(x_1, x_2) \rightarrow (x_1, x_2, 1)$$

$$(X_1, X_2, X_3) \rightarrow (X_1, X_2, X_3, 1)$$

Scaling is unimportant:

$$(x_1, x_2, 1) = \alpha(x_1, x_2, 1)$$
$$(X_1, X_2, X_3, 1) = \alpha(X_1, X_2, X_3, 1), \quad \alpha \neq 0$$

The projection transformation is now **linear**!

Image point in hom. coord.

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \mathbf{x_3} \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$

World point in hom. coord.

Camera matrix

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \mathbf{x_3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{\mathbf{x_1}}{\mathbf{x_3}} \\ \frac{\mathbf{x_2}}{\mathbf{x_3}} \\ \mathbf{1} \end{bmatrix} \rightarrow \begin{aligned} x_1 &= \frac{\mathbf{x_1}}{\mathbf{x_3}} = f\frac{X_1}{X_3} \\ x_2 &= \frac{\mathbf{x_2}}{\mathbf{x_3}} = f\frac{X_2}{X_3} \end{aligned} \qquad x_3, X_3 \neq 0$$

Conversion of homogeneous coordinates back to Euclidean coordinates

# COMPUTER VISION: PROJECTIVE CAMERA MODEL

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Camera matrix

**Calibration matrix**

$$K = \begin{bmatrix} s_x f & a & s_x o_x \\ 0 & s_y f & s_y o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Calibration matrix in real applications

$f$ – focal length

$(o_x, o_y)$ – image center (principal point)

$s_x, s_y$ – scale factors (pixels/mm)

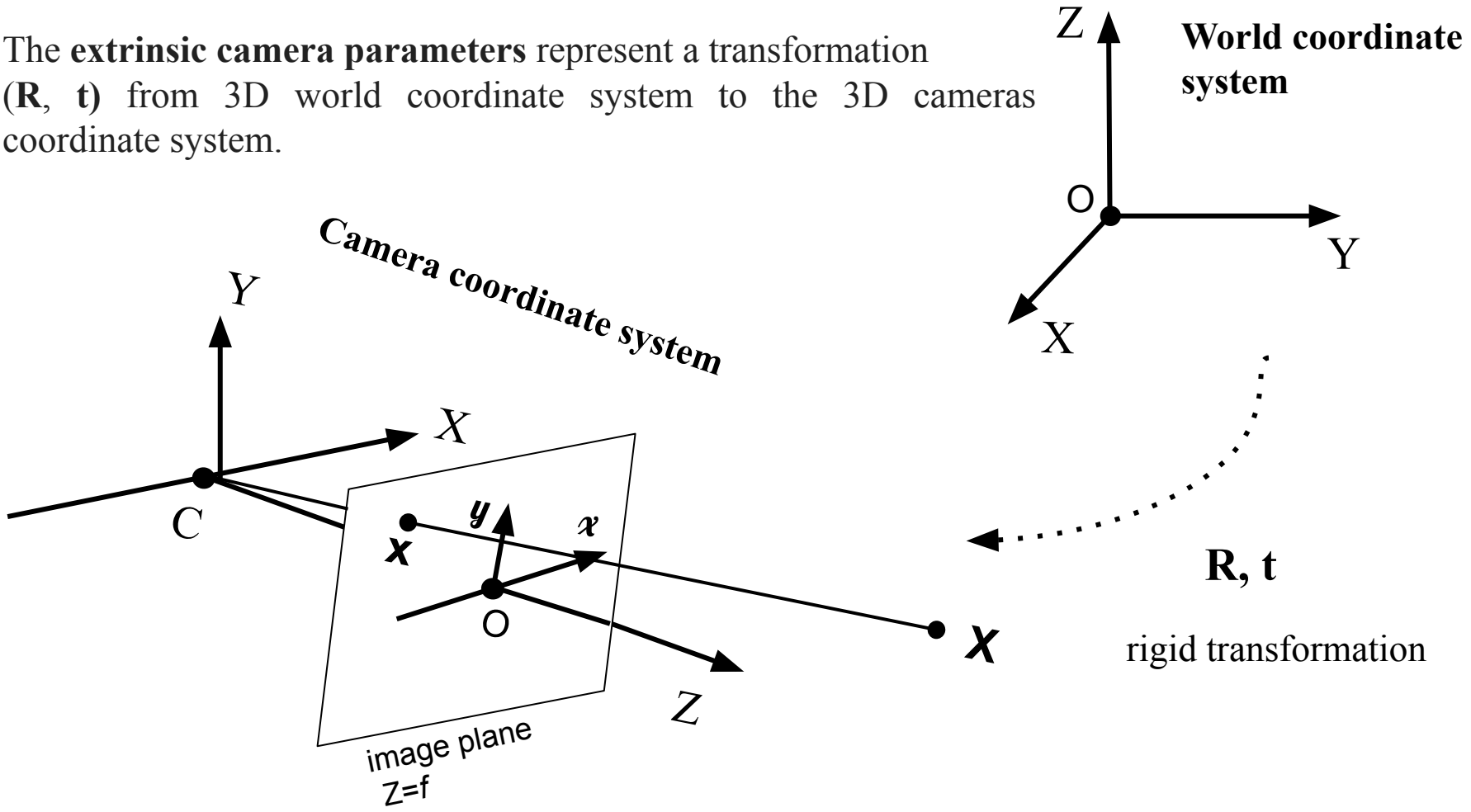
$a$ – skew coefficient

Nonzero elements (5) of the K matrix represent the **intrinsic camera parameters**.

These parameters depend solely on the cameras internal properties, and are not affected by the cameras position or direction.

# COMPUTER VISION: PROJECTIVE CAMERA MODEL

The **extrinsic camera parameters** represent a transformation **(R, t)** from 3D world coordinate system to the 3D cameras coordinate system.




**R - ROTATION MATRIX**

**t - TRANSLATION VECTOR**

$$R \cdot X^{World} + \mathbf{t} = X^{Camera}$$

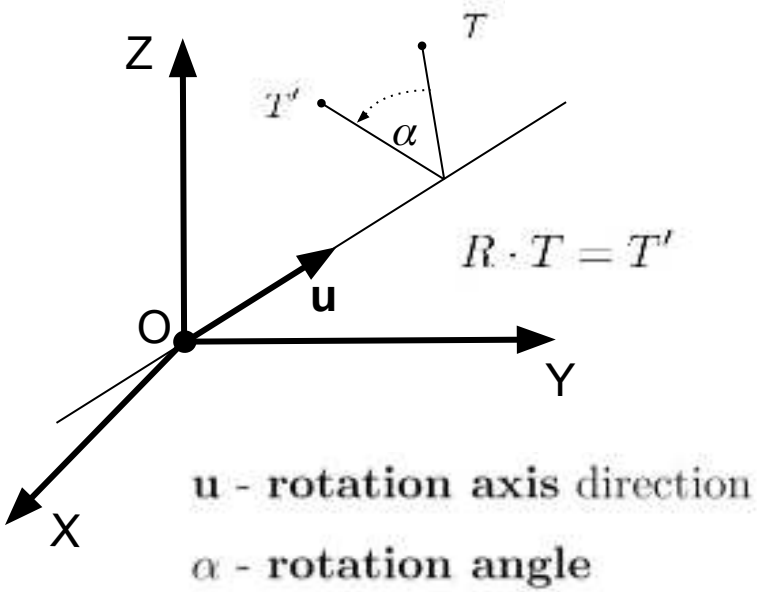
# COMPUTER VISION: PROJECTIVE CAMERA MODEL

Properties of the general **rotation matrix**:

1) $R = [\ \ ]_{3\times 3}$

2) $R^T = R^{-1}$ (orthogonal matrix)

3) $\det(R) = 1$




If $R$ is given:

- Determine the axis of rotation, i.e., the direction $\mathbf{u}$.

  $R \cdot \mathbf{u} = \mathbf{u}$ (homogeneous system with infinitely many solutions)

  $R \cdot \mathbf{u} = \lambda \mathbf{u},\ \lambda = 1$ ( $\mathbf{u}$ is eigenvector of $R$)

- Determine the rotation angle $\alpha$.
  Let $\mathbf{v}$ is a vector prependicular onto $\mathbf{u}$, then

  $$\alpha = \angle(\mathbf{v}, R \cdot \mathbf{v}).$$

# COMPUTER VISION: PROJECTIVE CAMERA MODEL

If **R** and **t** are given, we can determine the **CAMERA POSE.**

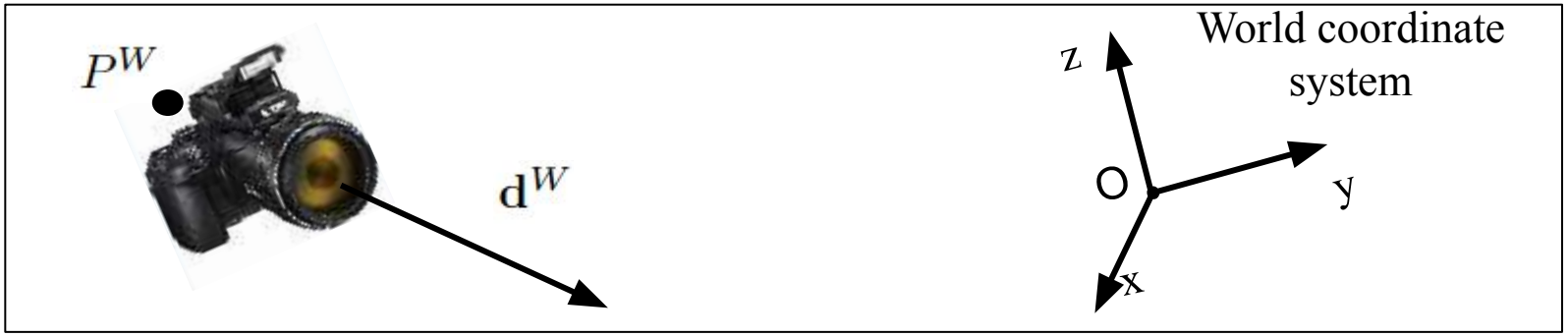
**CAMERA POSE = CAMERA ORIENTATION + CAMERA POSITION**
relative to the world coordinate system

- CAMERA ORIENTATION, vector $\mathbf{d}^W$

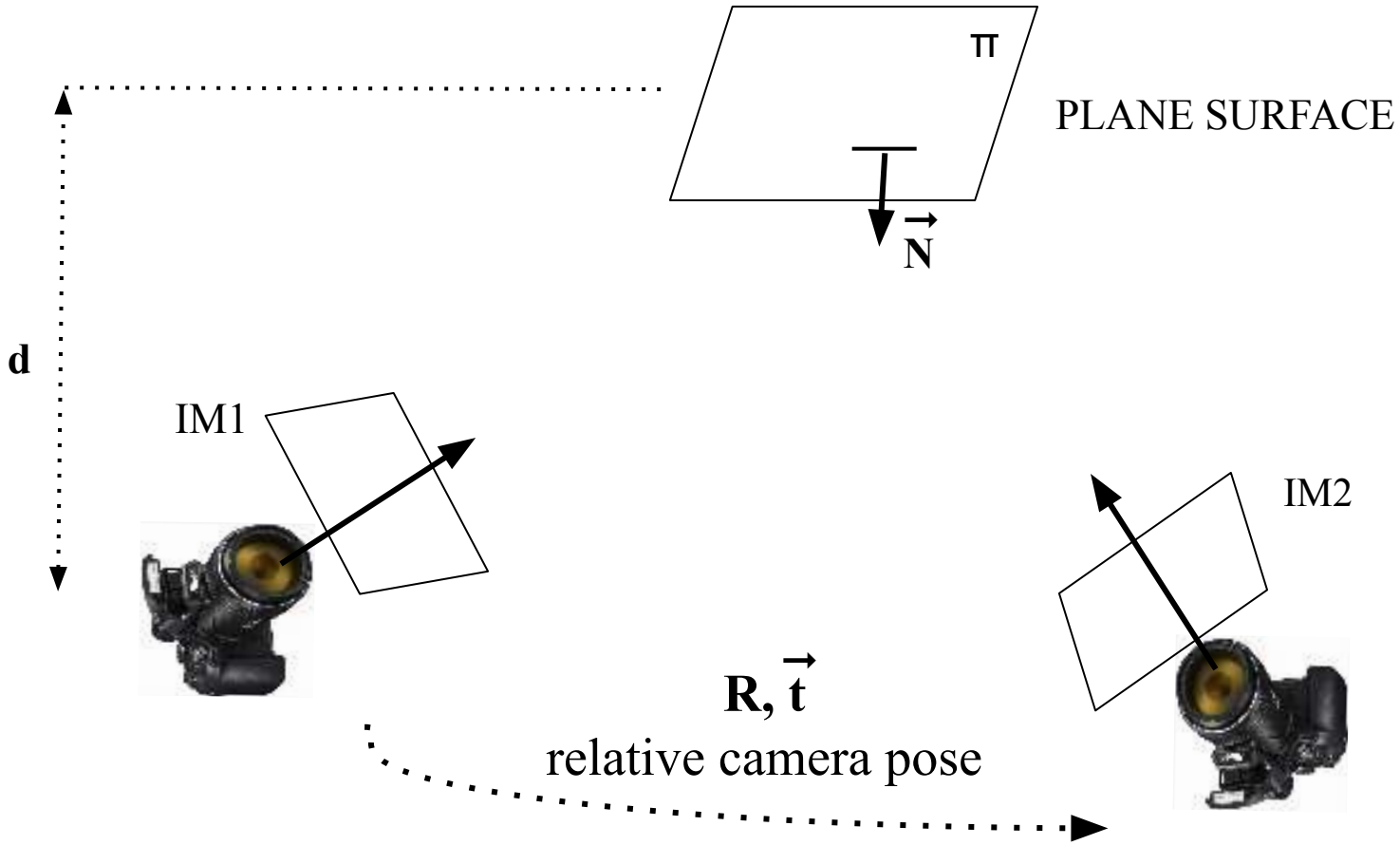
$$R \cdot \mathbf{d}^W = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \Leftrightarrow \mathbf{d}^W = R^{-1} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

- CAMERA POSITION, point $P^W$

$$R \cdot P^W + \mathbf{t} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \Leftrightarrow P^W = -R^{-1} \cdot \mathbf{t}$$



# COMPUTER VISION: HOMOGRAPHY




$$H = R + \frac{\vec{t} \cdot \vec{N}^T}{d} \qquad \text{IM1} \rightarrow \text{IM2}$$

A **homography** is a projective transformation that establishes a one-to-one correspondence between points in two planes.

# COMPUTER VISION: HOMOGRAPHY

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_2' \\ y_2' \\ \omega_2' \end{bmatrix} \simeq \begin{bmatrix} \frac{x_2'}{\omega_2'} \\ \frac{y_2'}{\omega_2'} \\ 1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

H IM1 IM2

where, g and h are projective sheare parameters.

H has 8 unknown parameters; therefore, we can determine them using (at least) **4 image point correspondences**.



IM1



IM2

# COMPUTER VISION: HOMOGRAPHY



H



# COMPUTER VISION: HOMOGRAPHY



H



# COMPUTER VISION: HOMOGRAPHY

**Panorama stitching**

When we are far from the observed objects:

$$H = \lim_{d \to \infty} \left( R + \frac{\vec{t} \cdot \vec{N}^T}{d} \right) = R$$

H







# COMPUTER VISION: HOMOGRAPHY

In case there is no translation of the camera:

$$\vec{t} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow H = \left( R + \frac{\vec{t} \cdot \vec{N}^T}{d} \right) = R$$

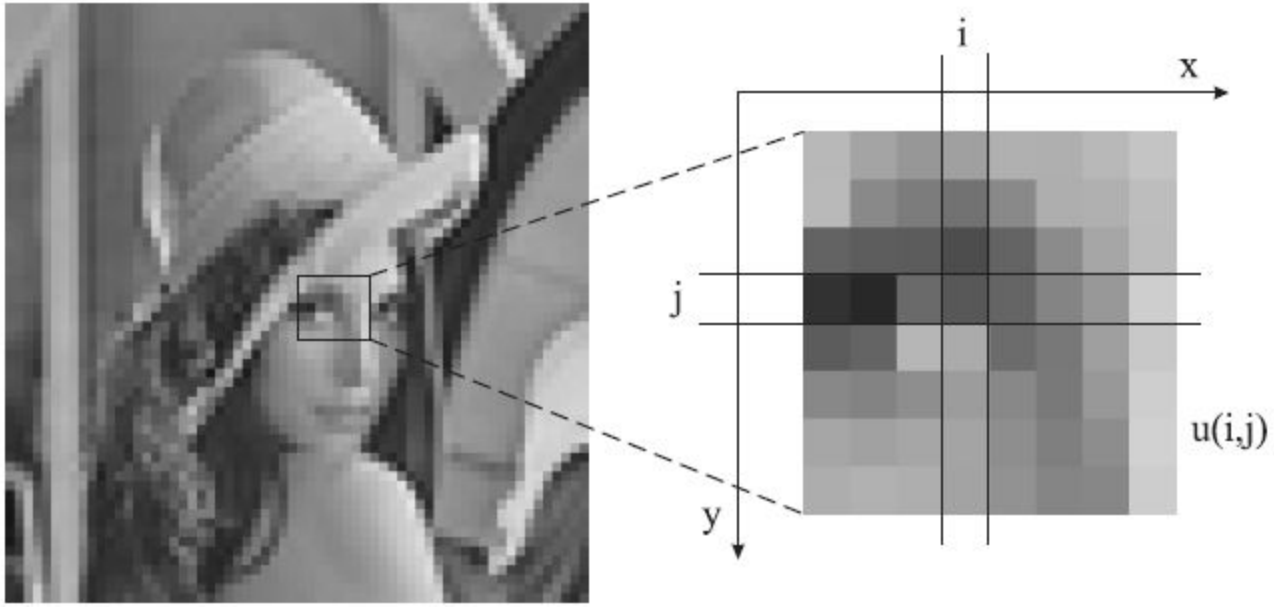


H





# DIGITAL IMAGE




$$u : \Omega \longrightarrow \mathbb{R}$$

$$u(i,j) \longrightarrow \Omega \subset \mathbb{Z}^2$$

Digital image function

LINEAR MODELS

$$u(x,y) \longrightarrow \Omega \subset \mathbb{R}^2$$

Continuously differentiable function

**NON-LINEAR MODELS**

# GRADIENT

Problem: how to determine the solution $x^* \in \mathbb{R}^n$ of the given energy-minimization model:

$$x^* = \arg\min_x f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an energy or objective function.

The most frequently used methods for this task belongs to **gradient based numerical optimization** approaches.

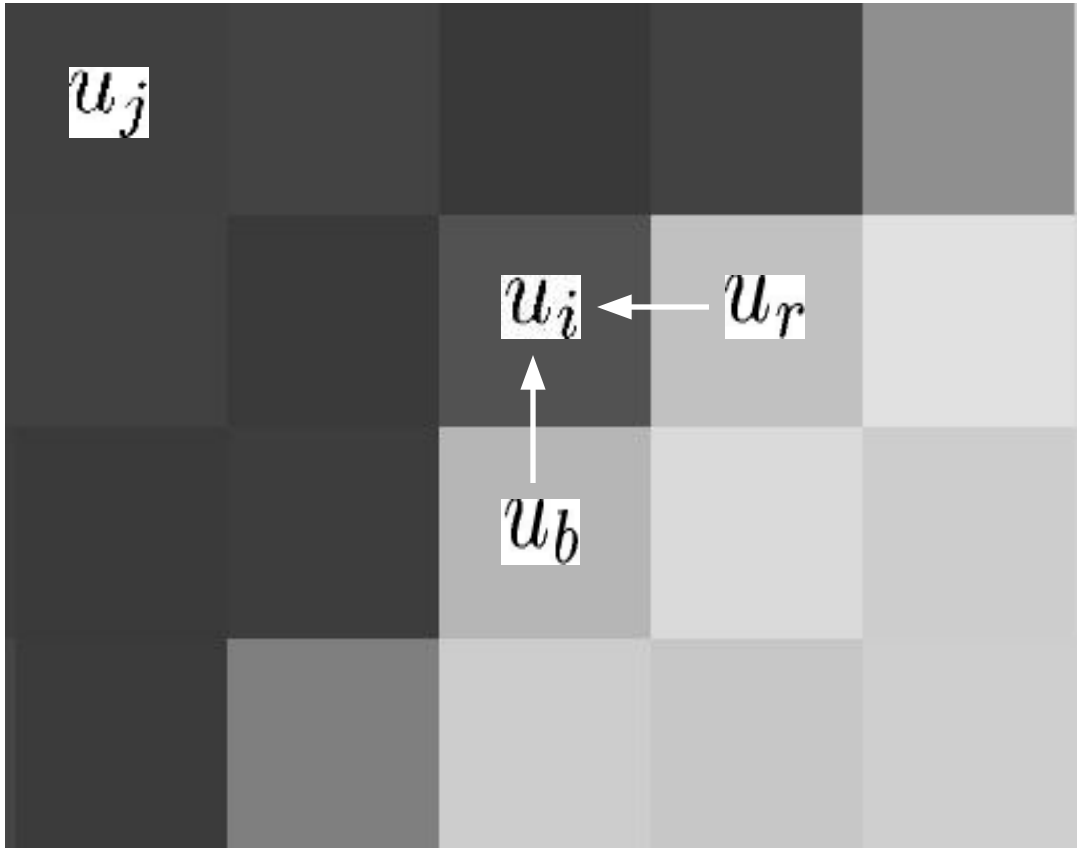
**GRADIENT operator**

$$\nabla f(x) = \operatorname{grad} f(x) = \left( \frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right)$$

# GRADIENT

*Discrete gradient*

$$\nabla(u_i) = [u_r - u_i, u_b - u_i]^T$$




$$\|\nabla(u_j)\| \leq \|\nabla(u_i)\|$$

# IMAGE GRADIENT



$u(x, y)$



$\|\nabla u(x, y)\|$



$g(\|\nabla u(x, y)\|)$

# GRADIENT

In continuous case, we can consider the **directional derivative**:

$$\frac{\partial u}{\partial \vec{l}} = \nabla u \cdot \vec{l} = \|\nabla u\| \cdot \|\vec{l}\| \cdot \cos \sphericalangle(\nabla u, \vec{l}), \quad \|\vec{l}\| = 1$$

$$\max_{\vec{l}} \left| \frac{\partial u}{\partial \vec{l}} \right| = \|\nabla u\|$$

Decreasing the gradient magnitude we achieve smoothing effect in all directions around the pixel.

# TOTAL VARIATION

Example. Rudin et al. (1992) introduced the *Total Variation (TV)* based **regularization** for denoising problem, where

$$\Psi(u) = TV(u) = \iint_{\Omega} \|\nabla u\| \, dxdy \approx \sum_{i=1}^{N} \|\nabla(u_i)\|$$

# TOTAL VARIATION



$u(x, y)$

Aracs Deserted Church
Aracsi pusztatemplom
~1230

$$\min_u \iint_\Omega \|\nabla u\| \, dxdy$$



# TOTAL VARIATION



$$\min_u \iint_\Omega \|\nabla u\|^2 \, dxdy$$

$$u_{t+1} = u_t + \Delta t \, \mathrm{div}\,(\nabla u_t)$$

**ISOTROPIC** smoothing (diffusion)

$$\min_u \iint_\Omega \|\nabla u\| \, dxdy$$

$$u_{t+1} = u_t + \Delta t \, \mathrm{div}\left(\frac{\nabla u_t}{\|\nabla u_t\|}\right)$$

**ANISOTROPIC** smoothing (diffusion)

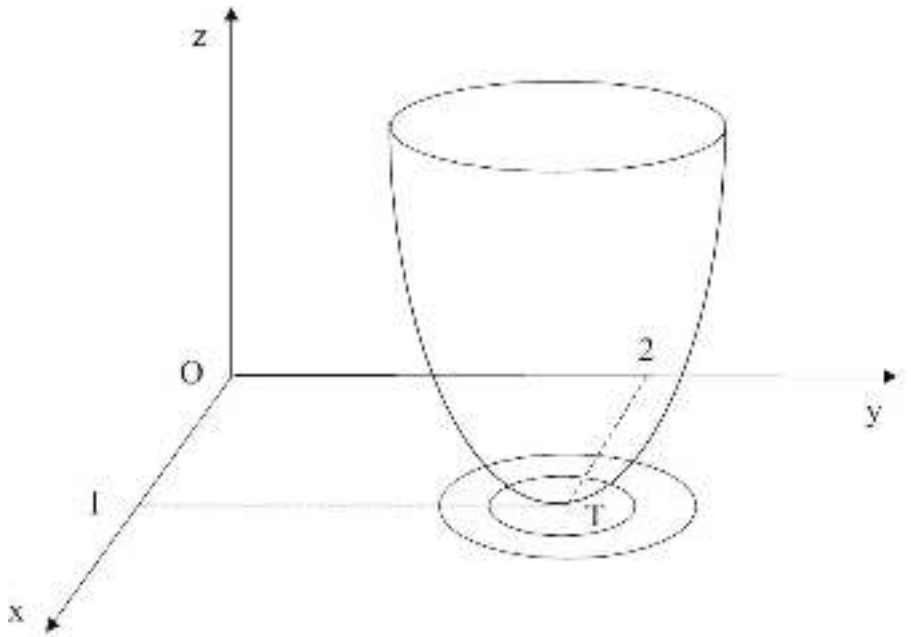


# GRADIENT BASED OPTIMIZATION

**LEVEL CURVES (SETS)**

$$f(x) = const$$

Paraboloid $z = (x-1)^2 + (y-2)^2$




Level curves (sets):

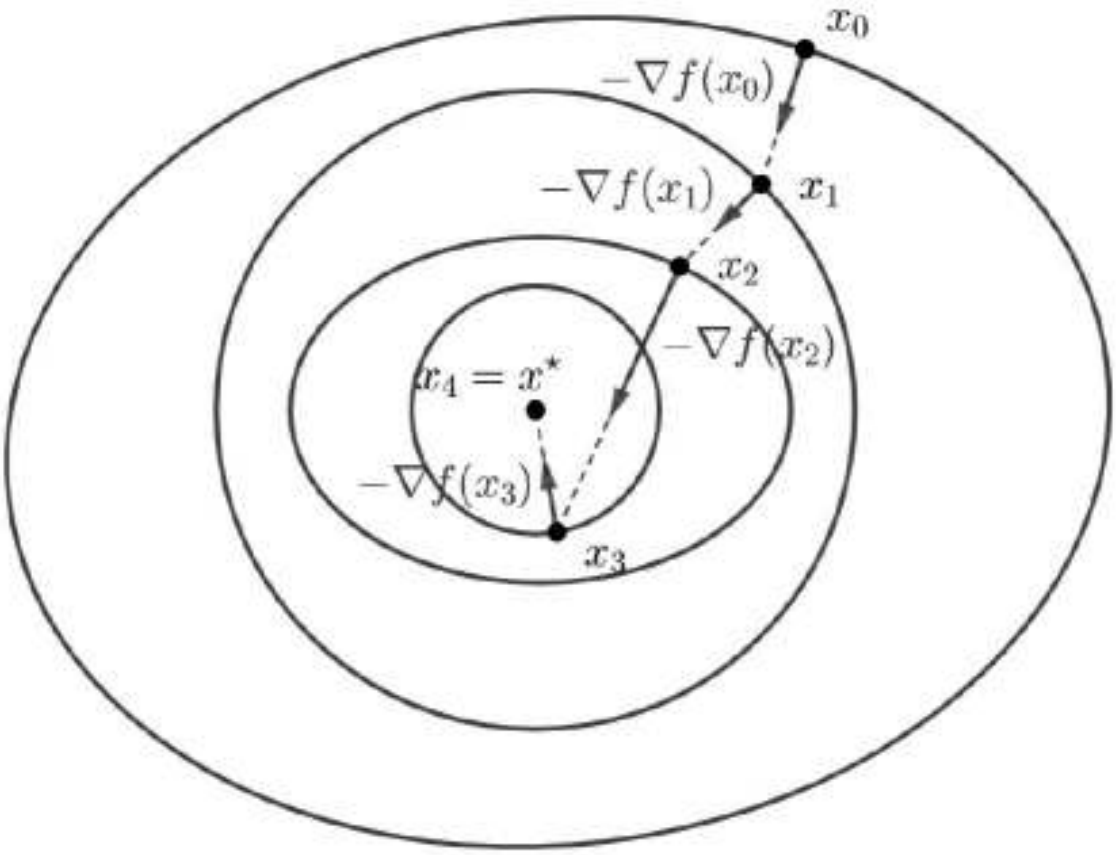
$$(x-1)^2 + (y-2)^2 = c$$

# GRADIENT BASED OPTIMIZATION

Gradient descent iterative algorithm:

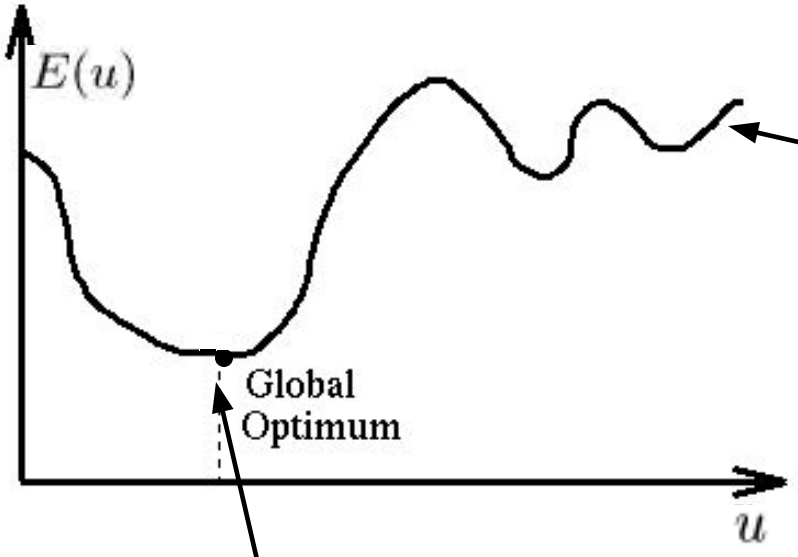
$$x^* = \arg\min_x f(x)$$

$$x_{t+1} = x_t - \Delta t \nabla f(x_t), \quad t = 0, 1, 2, \ldots$$



# ENERGY-MINIMIZATION PROBLEM

Image reconstruction example








* Model design: $\min_u E(u)$

* Minimization process

# TOMOGRAPHIC IMAGE RECONSTRUCTION

**Tomography** deals with the reconstruction of images, or slices of 3D volumes, from a number of projections obtained by penetrating waves through the considered object.

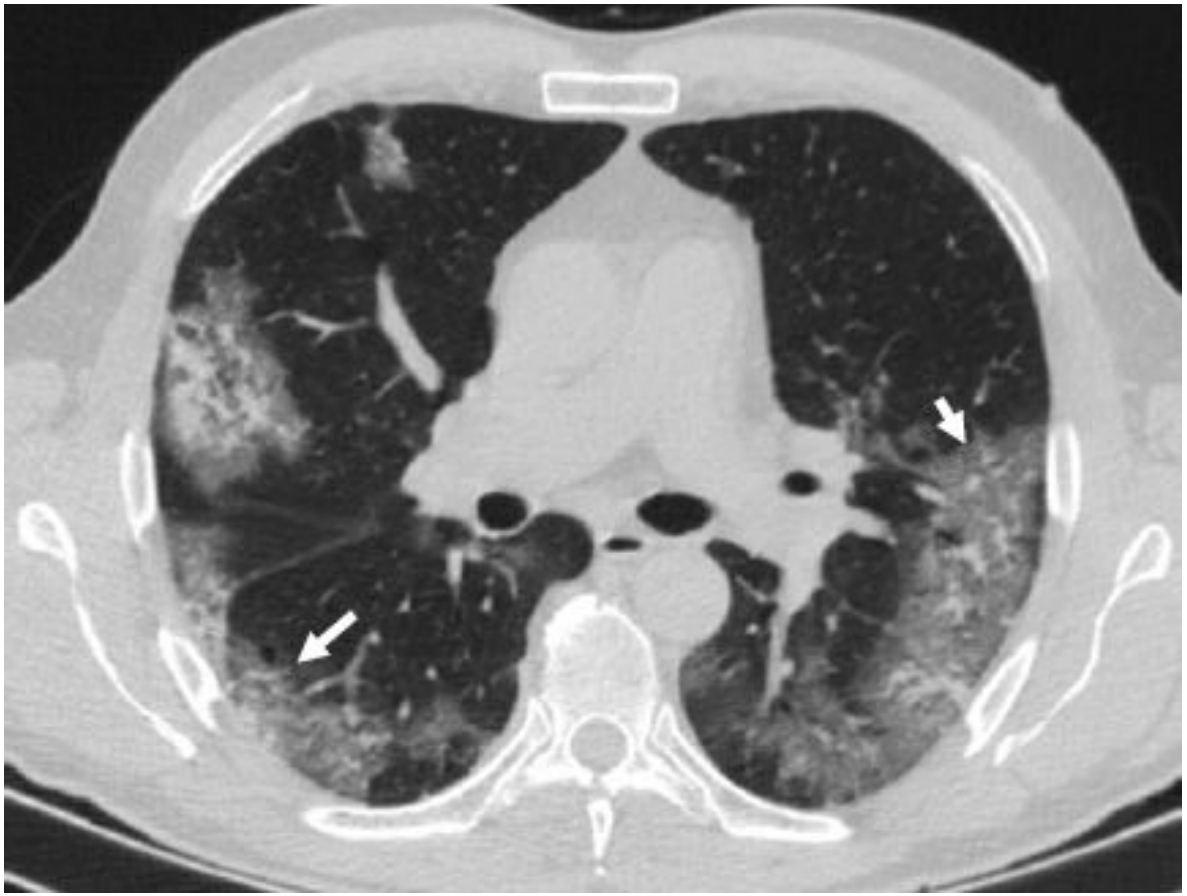
Applications in radiology, industry, airport baggage security, archeology etc.

Advantages: fast acquisition process, robustness and high image resolution.



CT scanner in human medicine

# TOMOGRAPHIC IMAGE RECONSTRUCTION



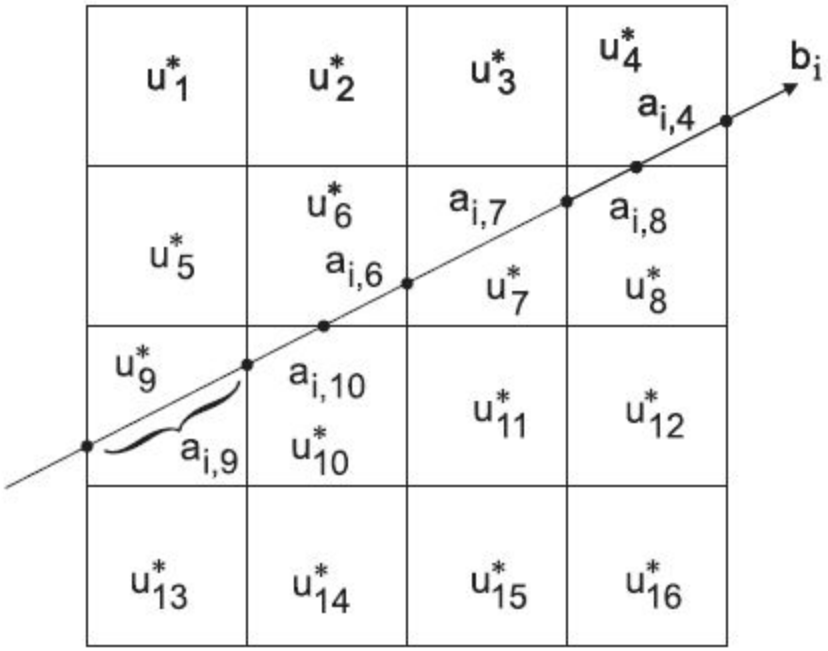
Lung CT scan in case of COVID-19 infection. The grey stains are consequences of the infection.

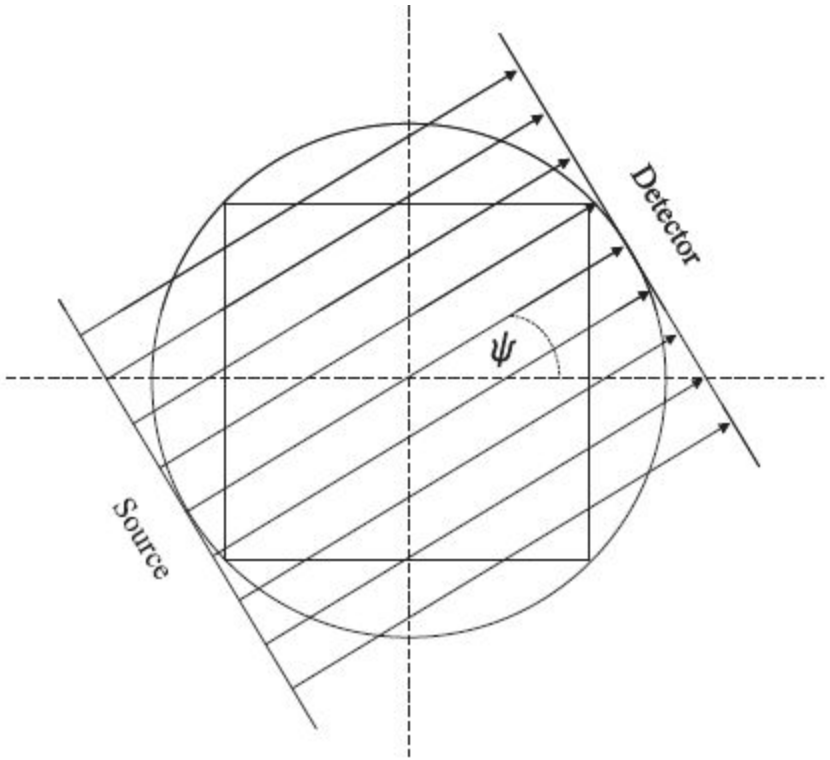
# HISTORY OF THE TOMGORAPHIC IMAGE RECONSTRUCTION

- The first CT scanners in the 1970s used iterative reconstruction algorithms, but limited computational power prevented their clinical use.
- Filtered back projection (FBP) became the standard for decades due to its speed, despite limitations in noise modeling and image quality at low radiation doses.
- In 2009, iterative reconstruction (IR) algorithms became commercially available (clinical use), leading to rapid adoption and significant improvements in image quality and dose reduction.
- IR methods have become indispensable in clinical CT, enabling significant radiation dose reductions (23–76%) without compromising diagnostic quality.
- AI-based methods are not yet standard in clinical practice but are expected to play a major role in the near future.

# TOMOGRAPHIC IMAGE RECONSTRUCTION

Tomography reconstruction of an image using **parallel beam geometry**.




$$b_i = a_{i,4}u_4^* + a_{i,6}u_6^* + a_{i,7}u_7^* + a_{i,8}u_8^* + a_{i,9}u_9^* + a_{i,10}u_{10}^*$$

Reconstruction problem: $Au = b$ , where the projection matrix $A \in \mathbb{R}^{M \times N}$ and vector $b \in \mathbb{R}^M$ are given.

# TOMOGRAPHIC IMAGE RECONSTRUCTION

$$
\begin{aligned}
a_{1,1}u_1 + a_{1,2}u_2 + \cdots + a_{1,n}u_n &= b_1,\\
a_{2,1}u_1 + a_{2,2}u_2 + \cdots + a_{2,n}u_n &= b_2,\\
&\vdots\\
a_{m,1}u_1 + a_{m,2}u_2 + \cdots + a_{m,n}u_n &= b_m
\end{aligned}
$$

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n}\\
a_{2,1} & a_{2,2} & \cdots & a_{2,n}\\
\vdots & \vdots & \ddots & \vdots\\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{bmatrix}
\cdot
\begin{bmatrix}
u_1\\ u_2\\ \vdots\\ u_n
\end{bmatrix}
=
\begin{bmatrix}
b_1\\ b_2\\ \vdots\\ b_m
\end{bmatrix}
$$

or in the compact form,

$$
A\,u = b, \quad A \in \mathbb{R}^{m\times n},\ u \in \mathbb{R}^n,\ b \in \mathbb{R}^m,
$$

# RECONSTRUCTION MODEL

$$\min_u \left[ \|Au - b\|^2 + \alpha_1 R_1(u) + \alpha_2 R_2(u) + \alpha_3 R_3(u) \right]$$

$$\alpha_i \in \mathbb{R}$$

$R_i$ : regularizációs függvények

Main goal: To improve image quality and reducing the harmful X-ray radiation dose.

A priori informations:

- **orientation** *Lukic & Balazs 2016*
- **compactness/circularity** *Lukic & Balazs 2020*
- **centroid** *Lukic & Balazs 2022*
- **geometrical moments** *Lukic & Balazs 2024*

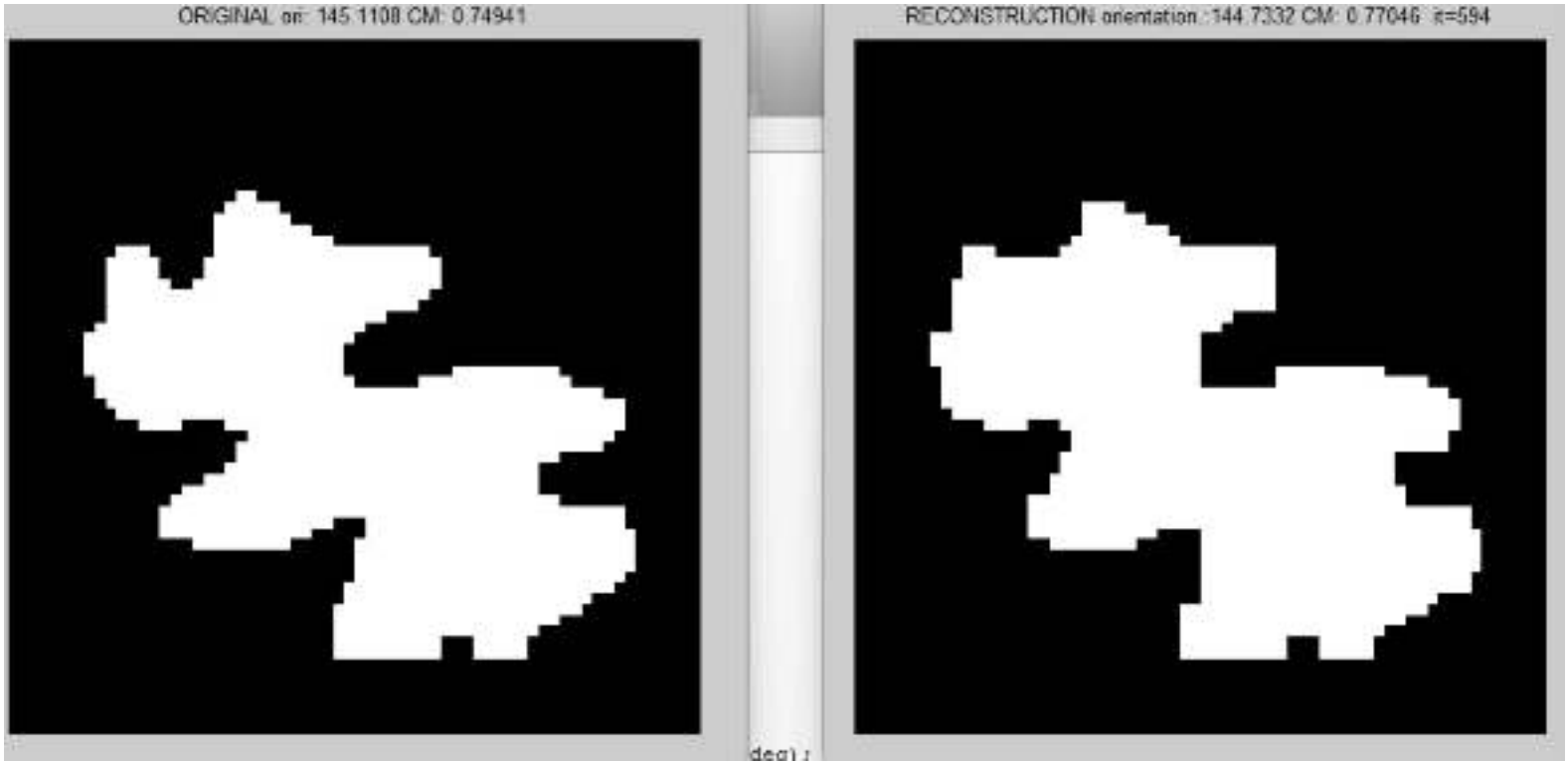
# BINARY TOMOGRAPHY

Binary tomography image reconstruction example.

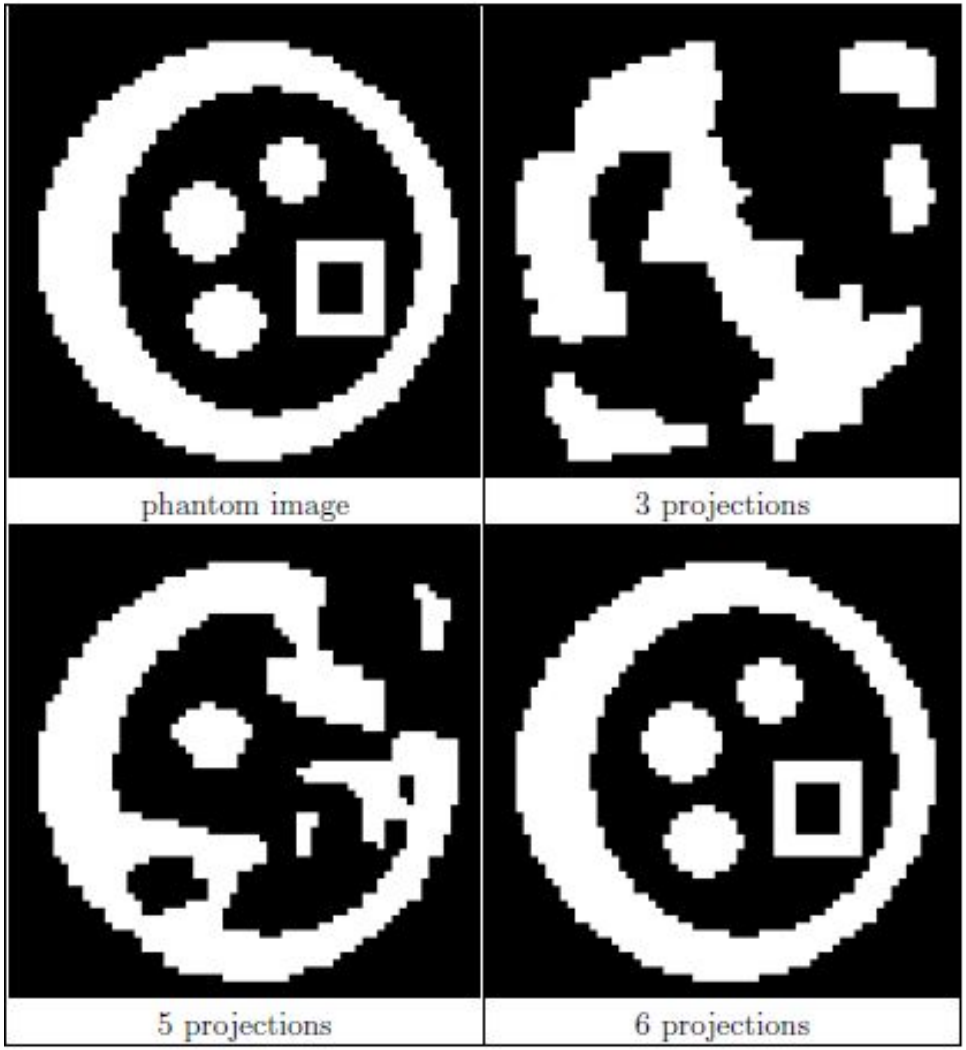
Method: *Spectral Projected Gradient* (SPG) deterministic iterative optimization.

Projective directions: $0^0$, $45^0$ and $90^0$.

Regularization: TV + convex-concave.



# BINARY TOMOGRAPHY

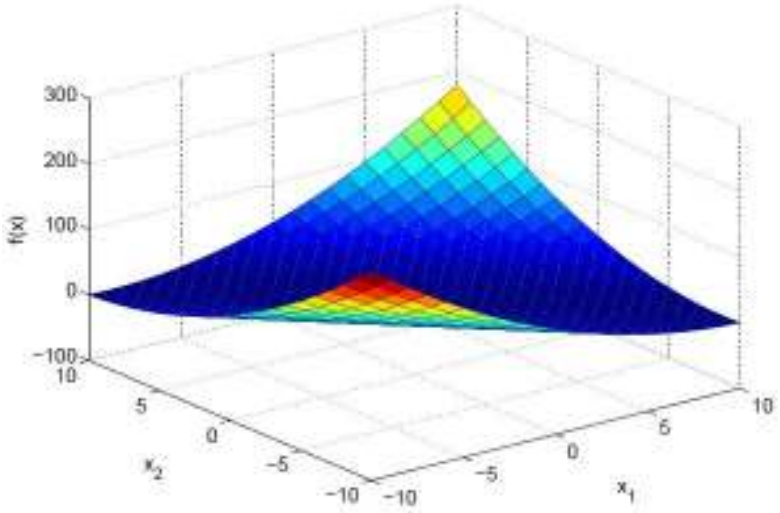


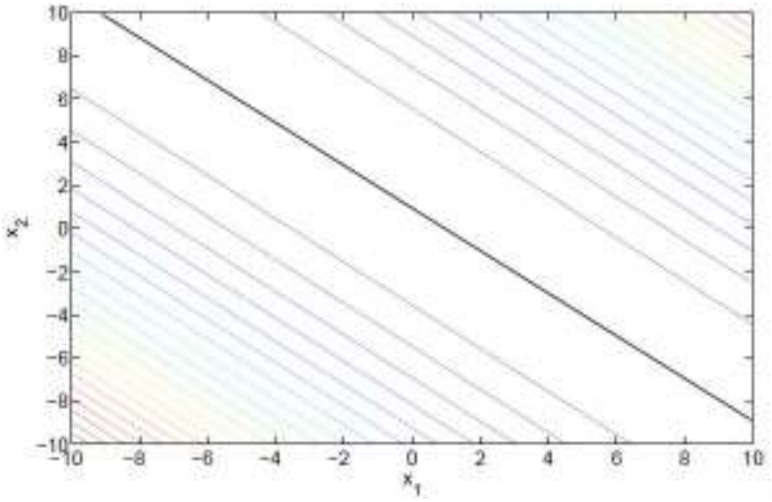
# SUM OF SQUARES

$$f(x) = \frac{1}{2}||Ax - b||^2$$

Quadratic function, convex, but often not strictly convex.

2D example:



# Nonconventional Image Grid: Triangular Grid

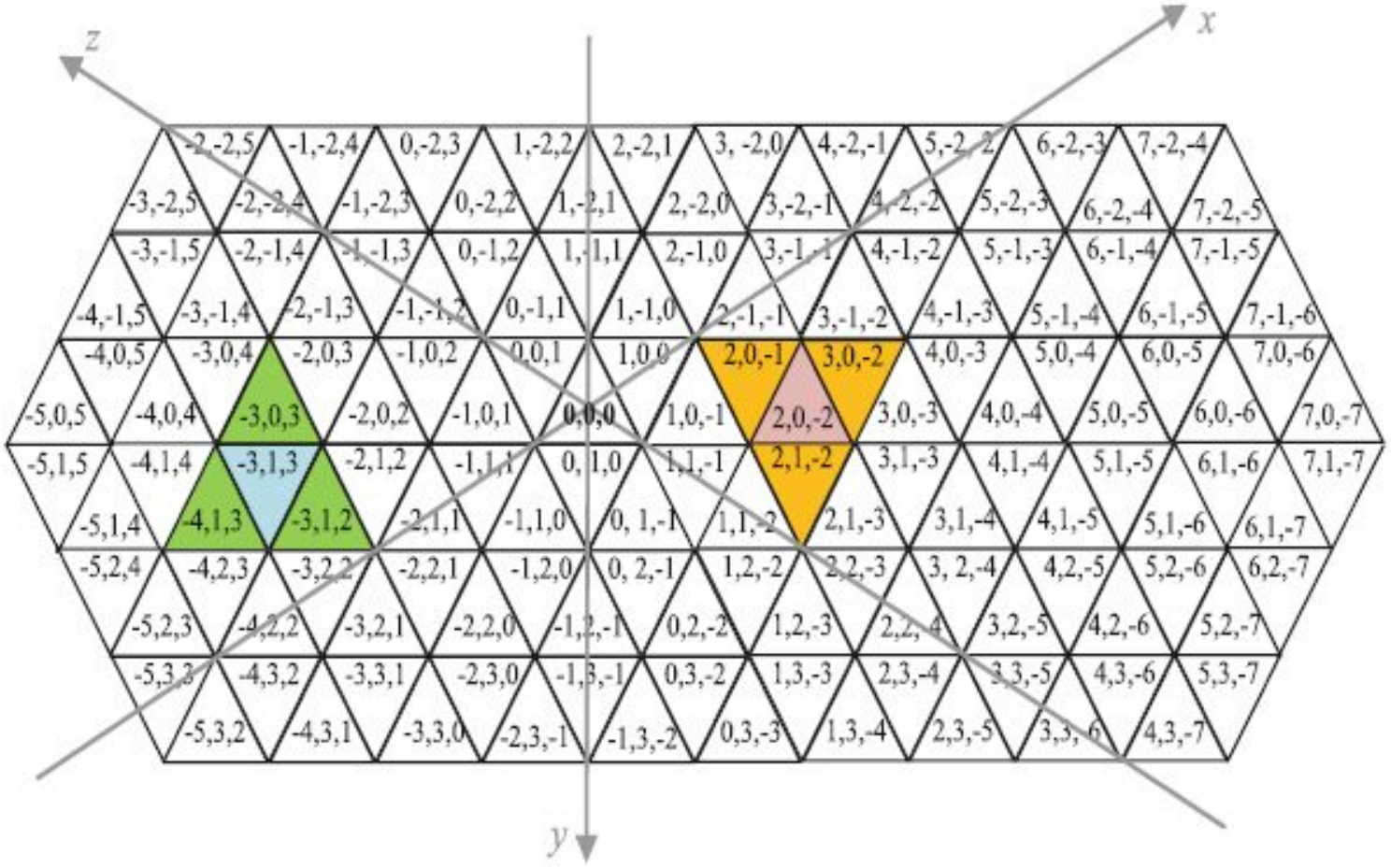



**Fig. 1.** The symmetric coordinate system for the triangular grid, with an even and an odd pixel and their closest neighbors.

3 “natural” projections: x=const, y=const and z=const.

# Shifted Projection Approach on the Triangular Grid

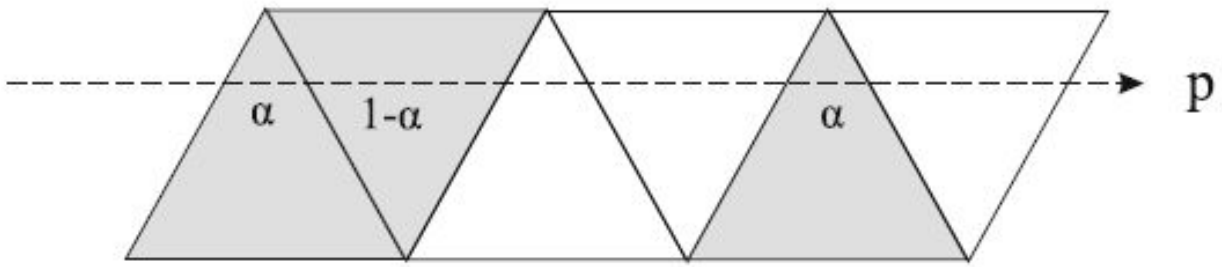



**Fig. 2.** Shifted projection ray penetrating trough a triangular grid lane.

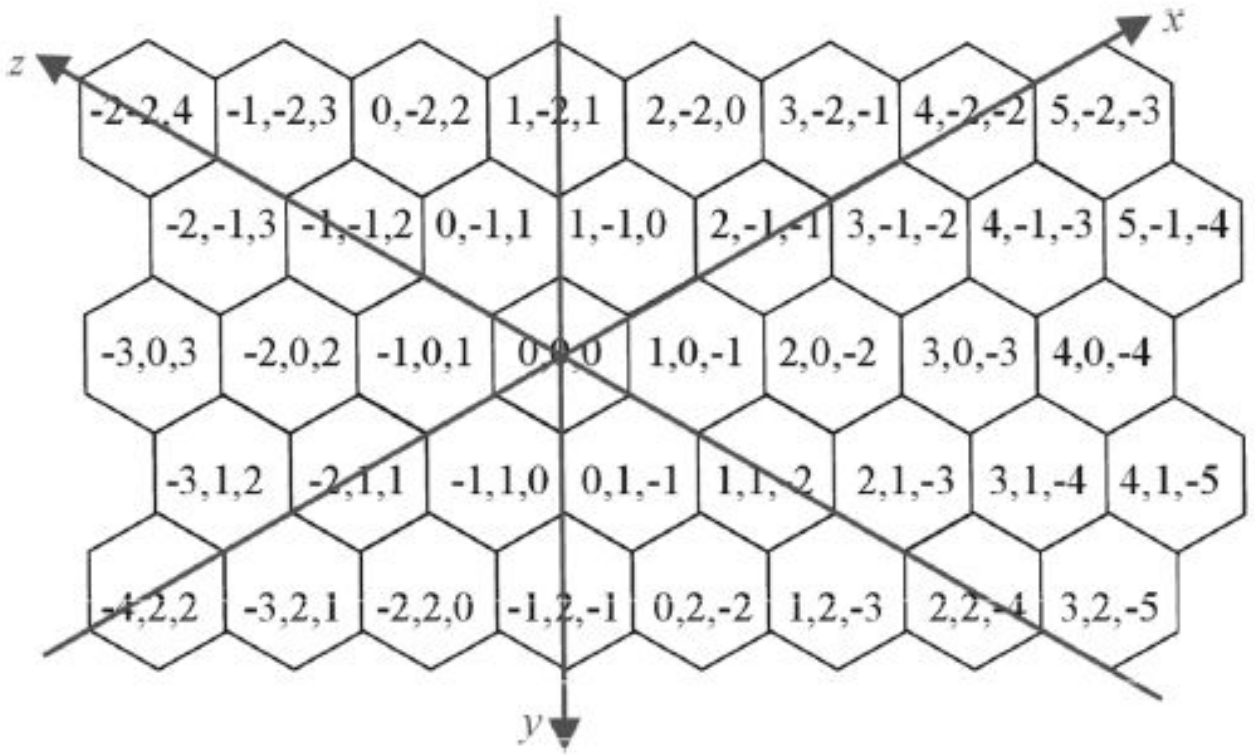
$\alpha$ is selected from interval $(0, 1)$, where $\alpha \neq 0.5$

$$s \cdot (1 - \alpha) \quad + \quad l \cdot \alpha \quad = \quad p$$

The number of nabla and delta type pixels per lane are determined by the given projection value *p.*

[Benedek Nagy and Tibor Lukic, Binary tomography on the isometric tessellation involving pixel shape orientation, IET Image Processing, 14(1), 25–30, 2020.]

# Nonconventional Image Grid: Hexagonal Grid




**3.** The symmetric coordinate system for the hexagonal grid [1

[Tibor Lukic and Benedek Nagy, Regularized binary tomography on the hexagonal grid, Physica Scripta, IOP Publishing, Vol. 94, 025201 (9pp), 2019.]

# Cairo Pattern Image Grid

Cairo pattern type tessellation in modern bathrooms.



# Cairo Pattern Image Grid

Cairo pattern type tessellation in modern bathrooms.



THANK YOU FOR YOUR ATTENTION!