# Vision and Image Processing: Camera Models, Homographies

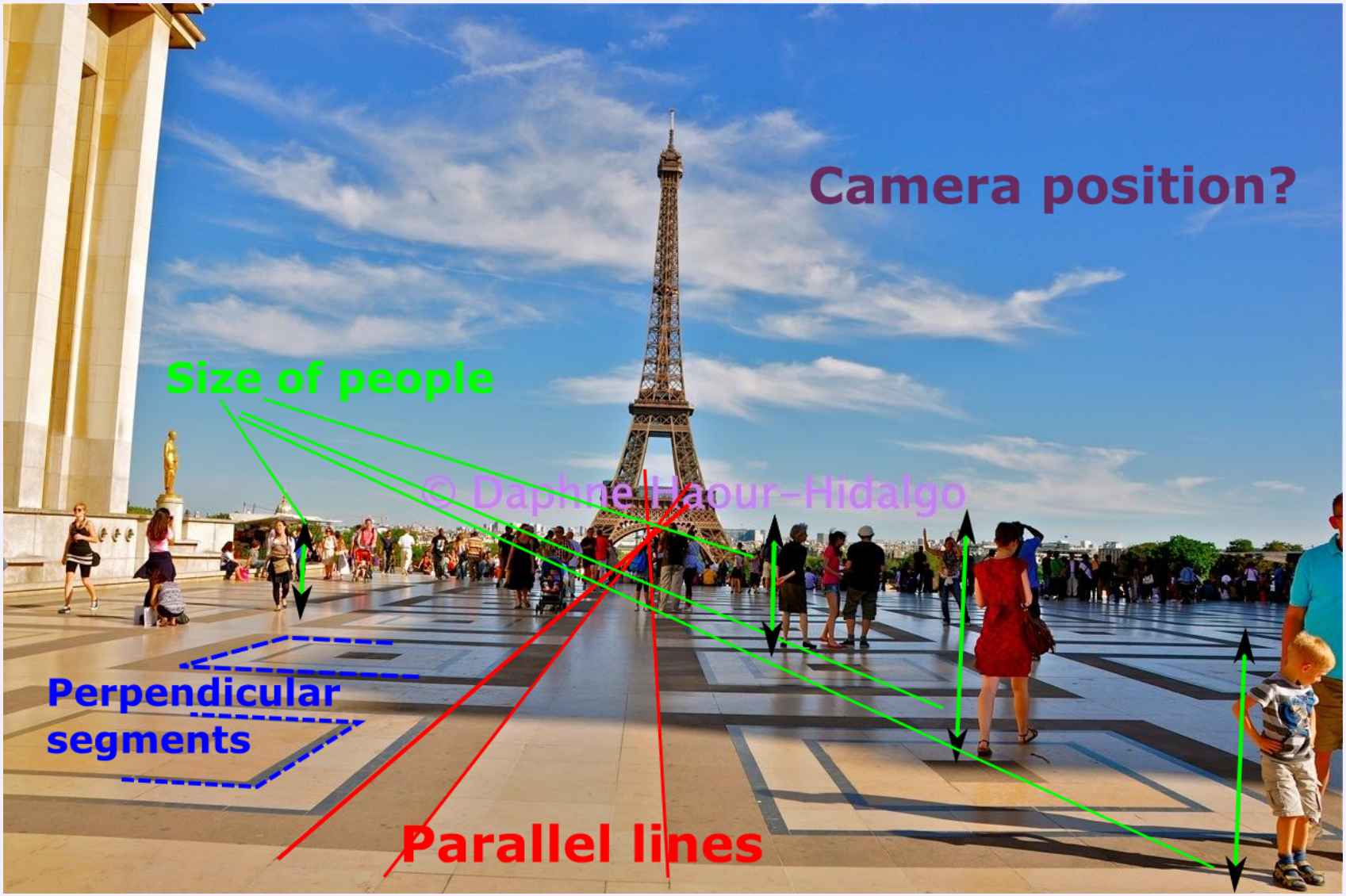
Søren Olsen

Department of Computer Science
University of Copenhagen

# Plan for today

- Introduction to Camera Models, specifically the pinhole model and the perspective transformation
- Homogeneous coordinates, Homographies
- Camera matrices and parameters
- Camera calibration

# Cameras and projections

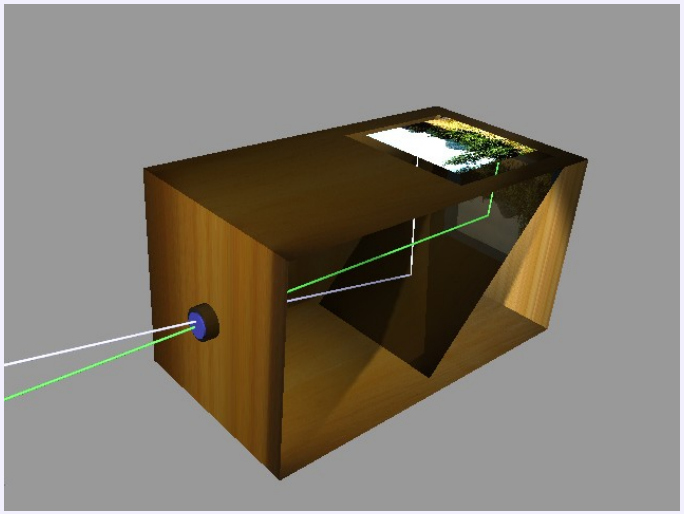


# Questions

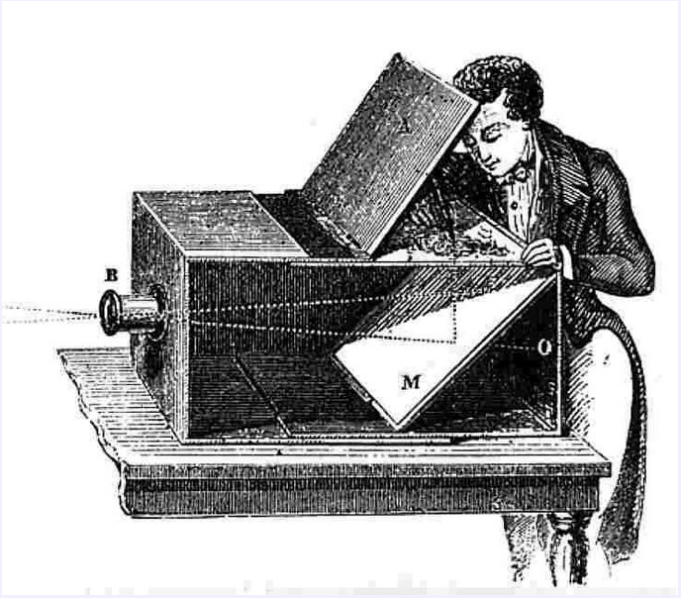
Previous picture raises some questions about:

- Lines?
- Parallelism?
- Angles / orthogonality?
- Sizes?
- Camera position / Horizon?

# Camera Obscura



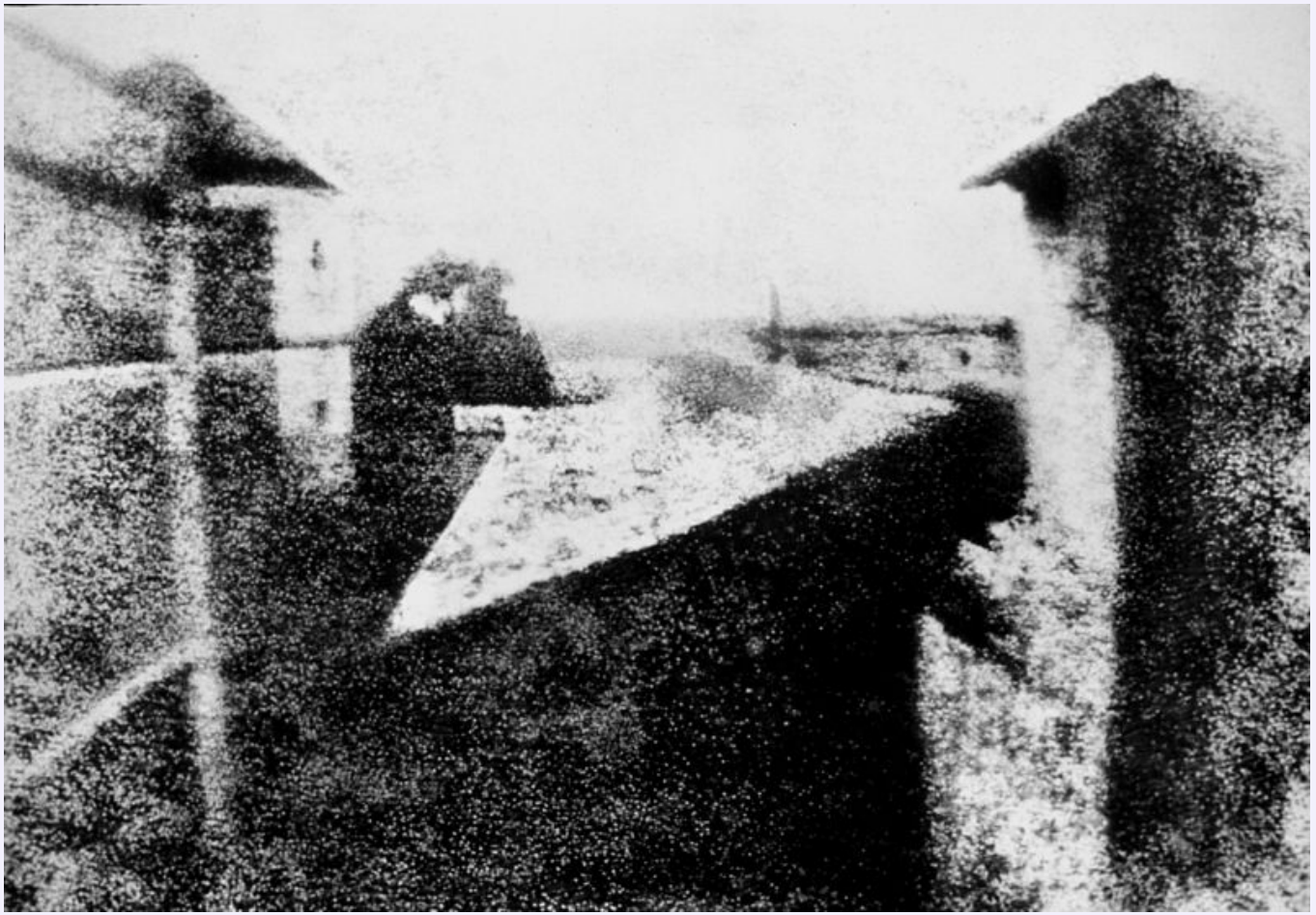
Principle of Camera Obscura




18th Century Camera Obscura

- Known from old chines writings
- Mentioned by Aristotle
- Plaque with photosensitive material: Photographic camera!

# The Very First Photography, 1826

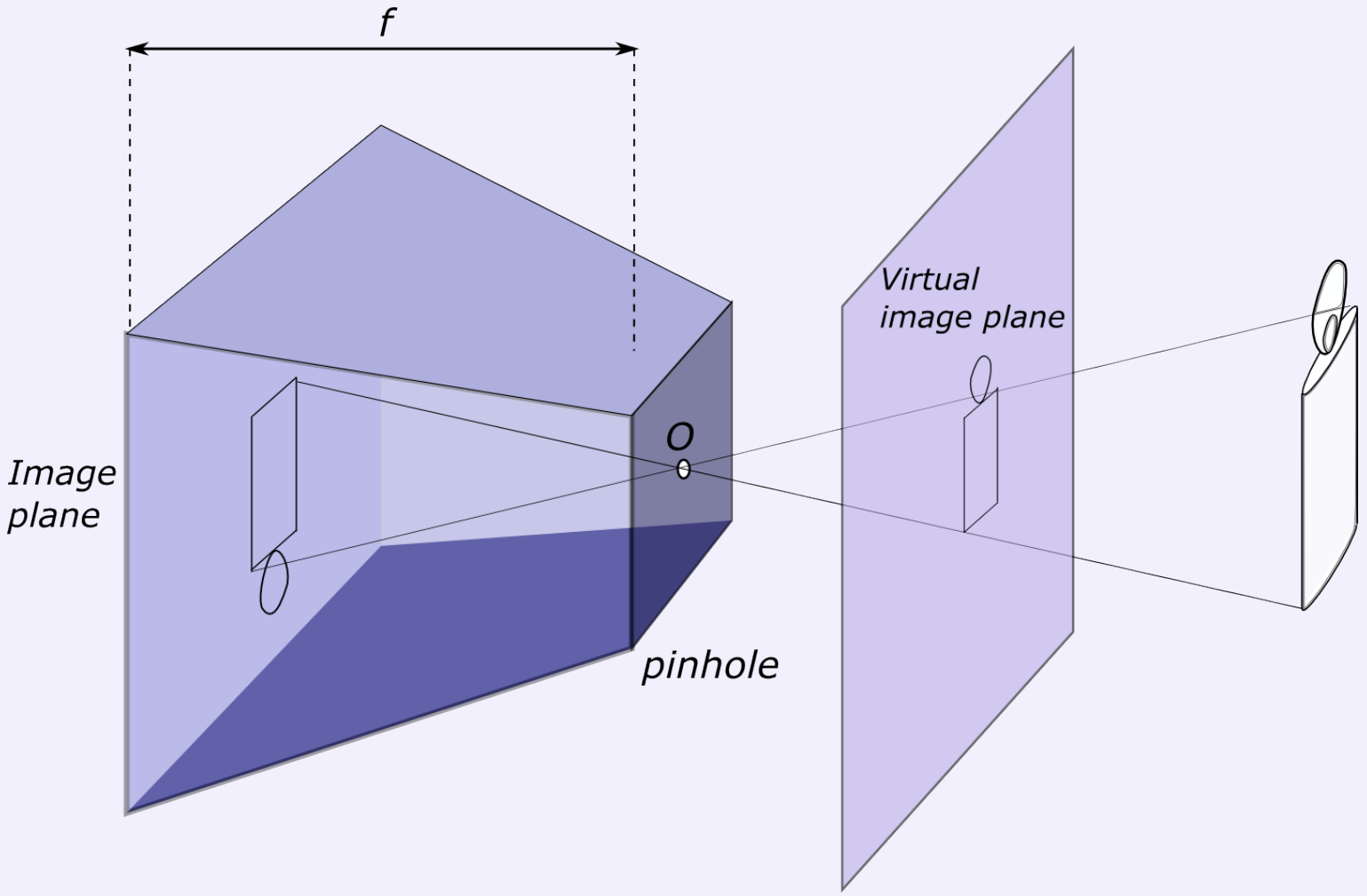


J.N. Niépce, View from the window at Le Gras, Saint Loup de Varennes, France – Now at University of Texas at Austin.

# The pin-hole camera

Magnesium light was used to make light enough enter the pin-hole box.

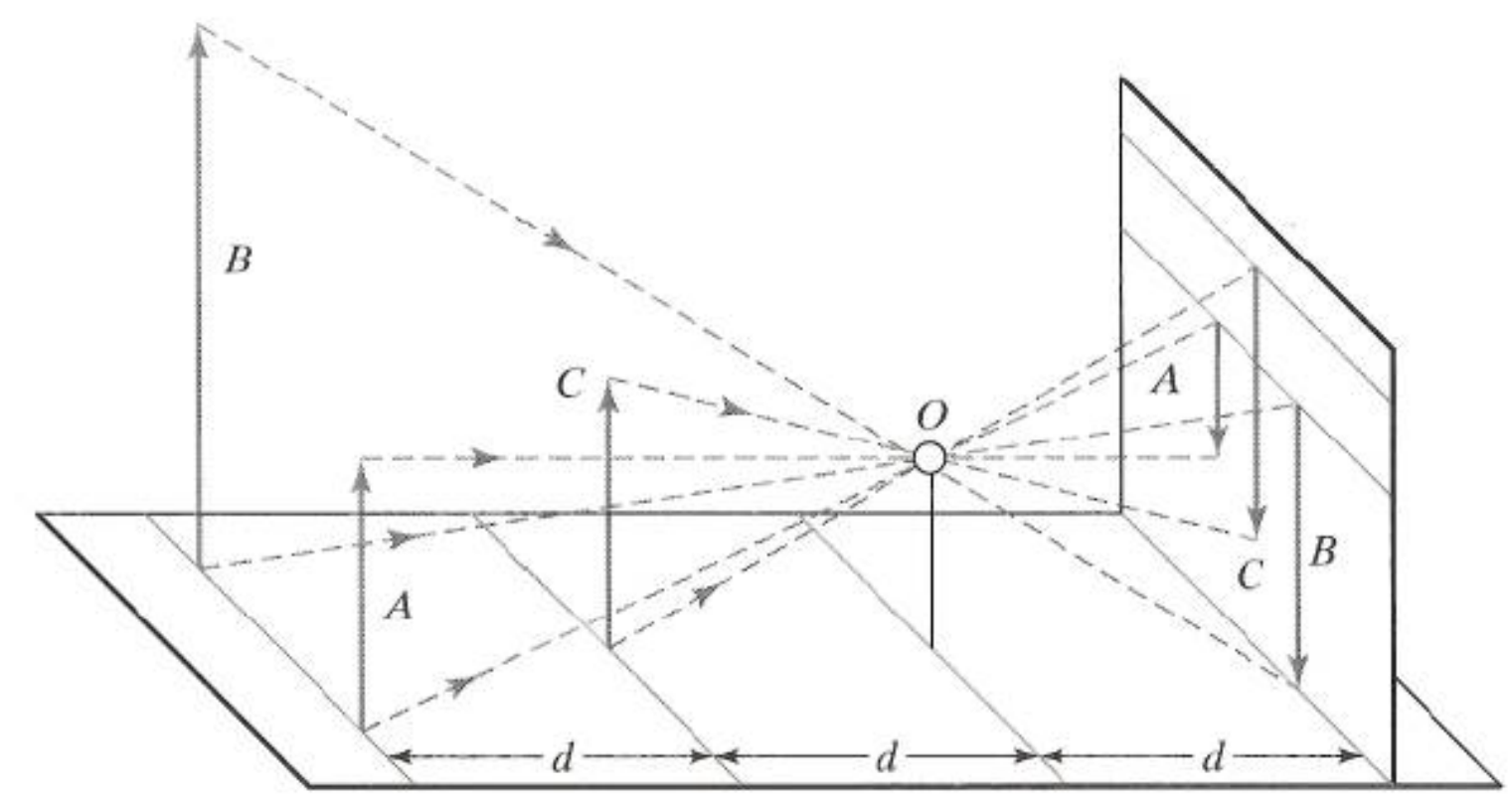
# The Pinhole Camera Model




- *f* is the focal length,
- *O* is the camera center.

# Perspective Effects

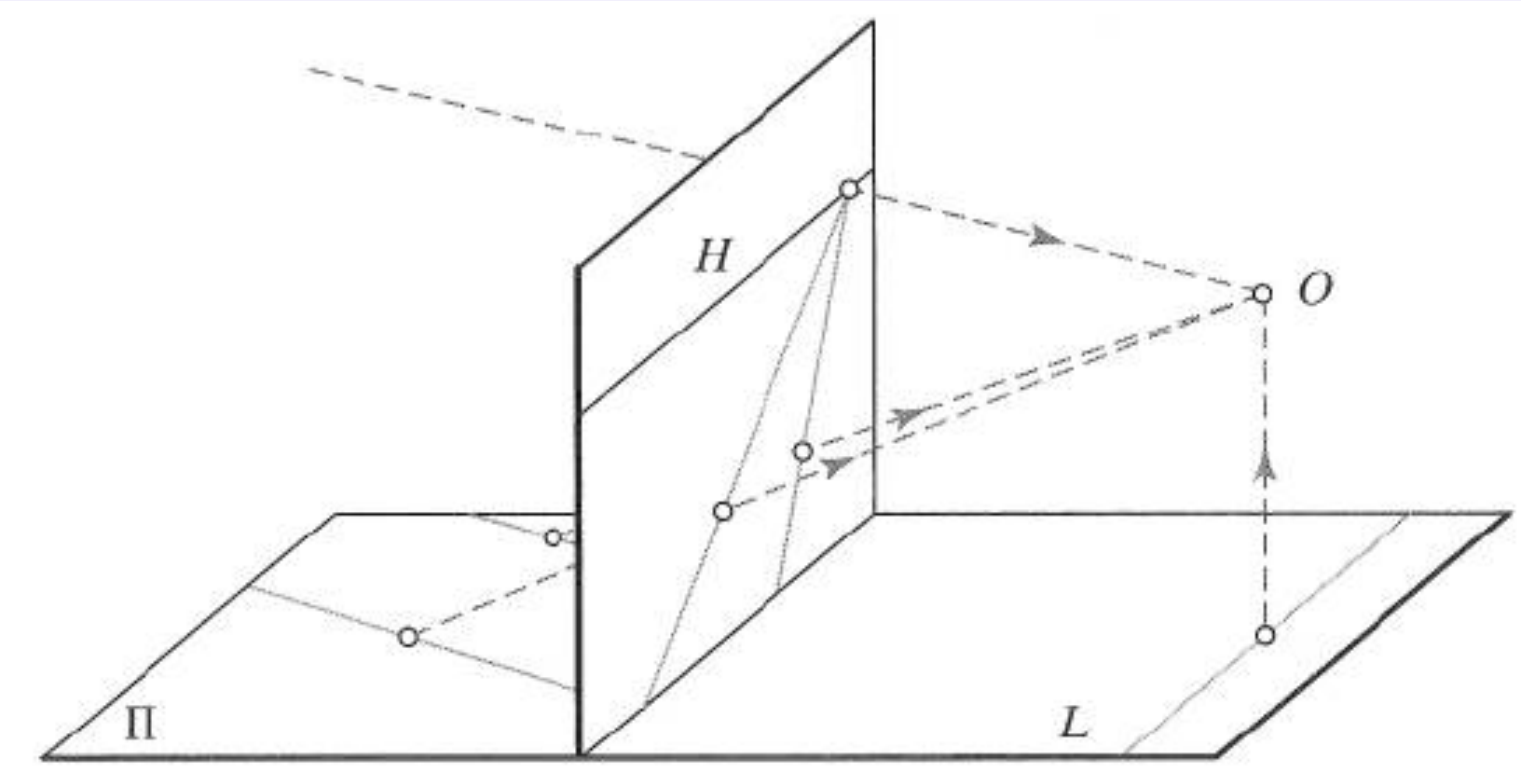
Remember from first lecture




Far objects appear smaller that close ones.

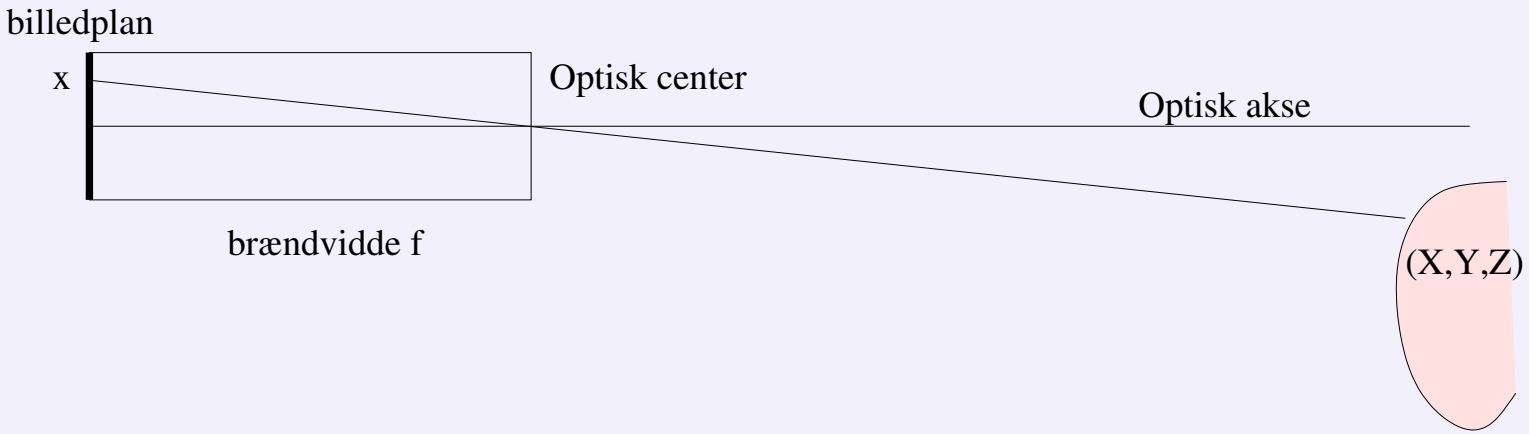
# Perspective Effects

Remember from first lecture again




Images of parallel lines intersect at the horizon (virtual image plane).

# Projection Equations




- Project $P(X, Y, Z)$ onto $(x, y, -f)$. Remember: similar triangles:

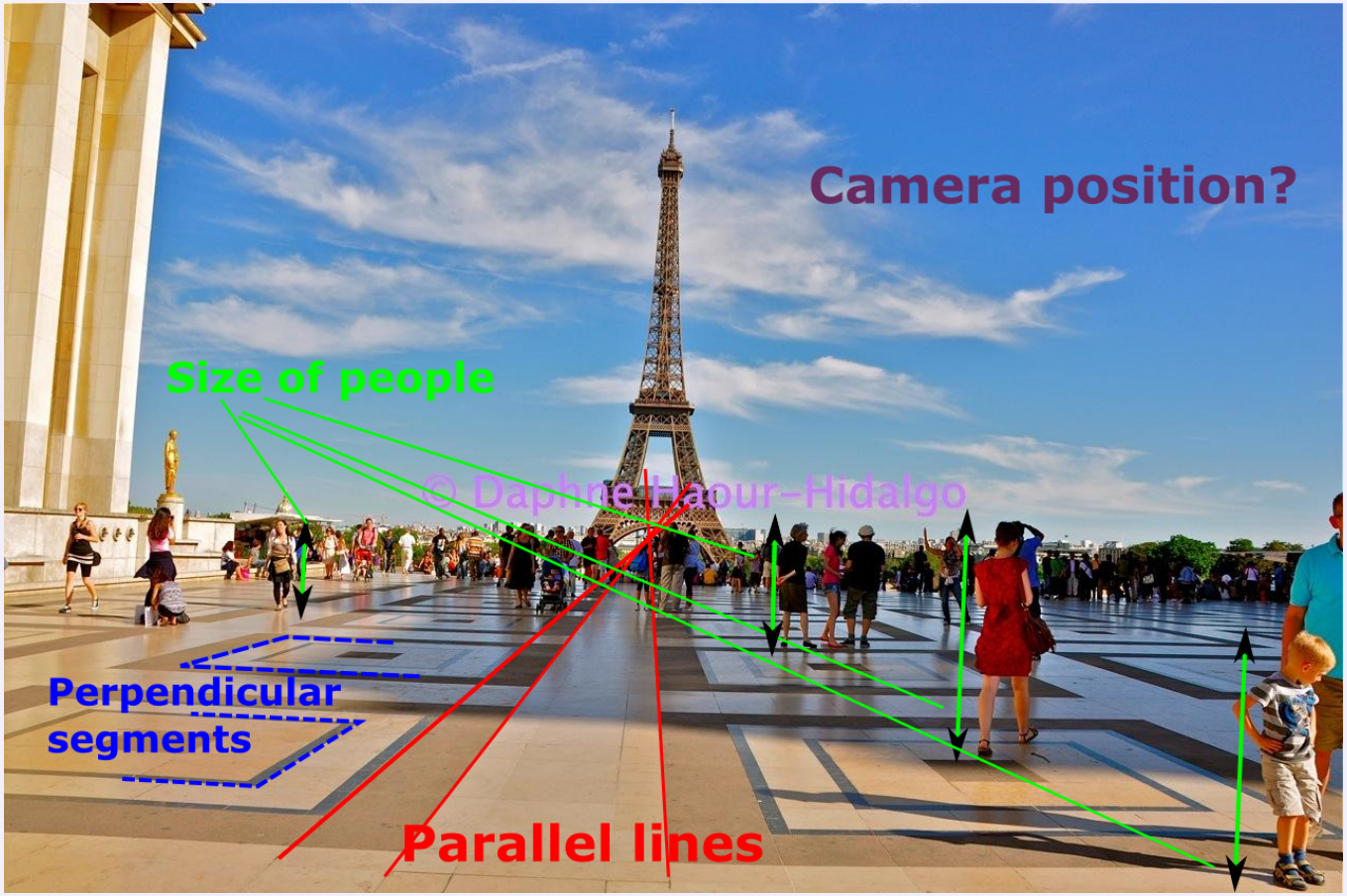
$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}$$

- To get non-negative pixel indexes:

$$x = f\frac{X}{Z} + c_x, \quad y = f\frac{Y}{Z} + c_y$$

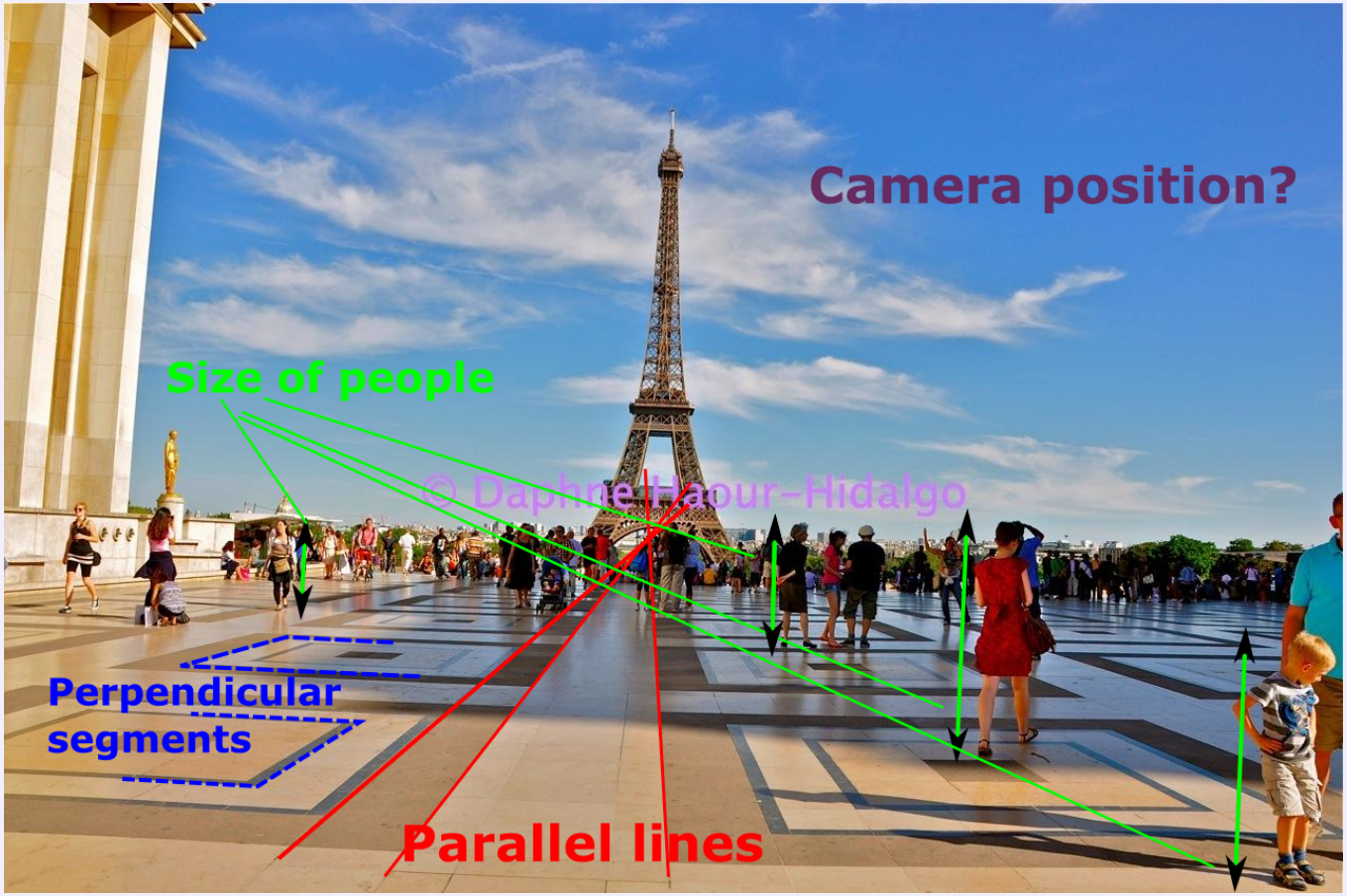
where $(c_x, c_y)$ is called the principal point.

# Lost in projection




Depth, Angles, size, parallelism.

# Preserved by projection

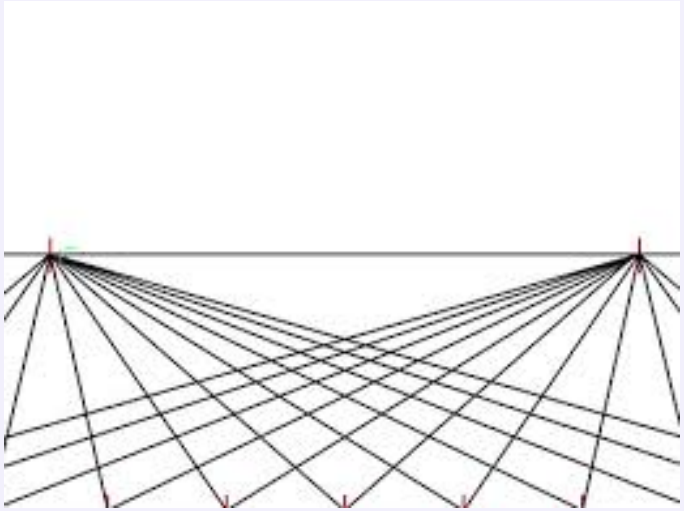


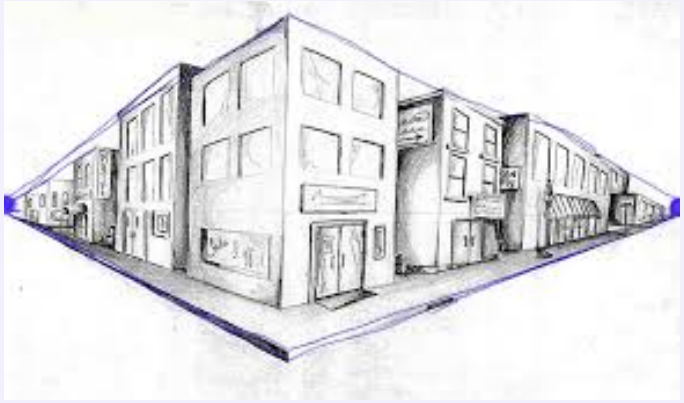

Straight lines, connectivity, neighborhood.

# Vanishing Points

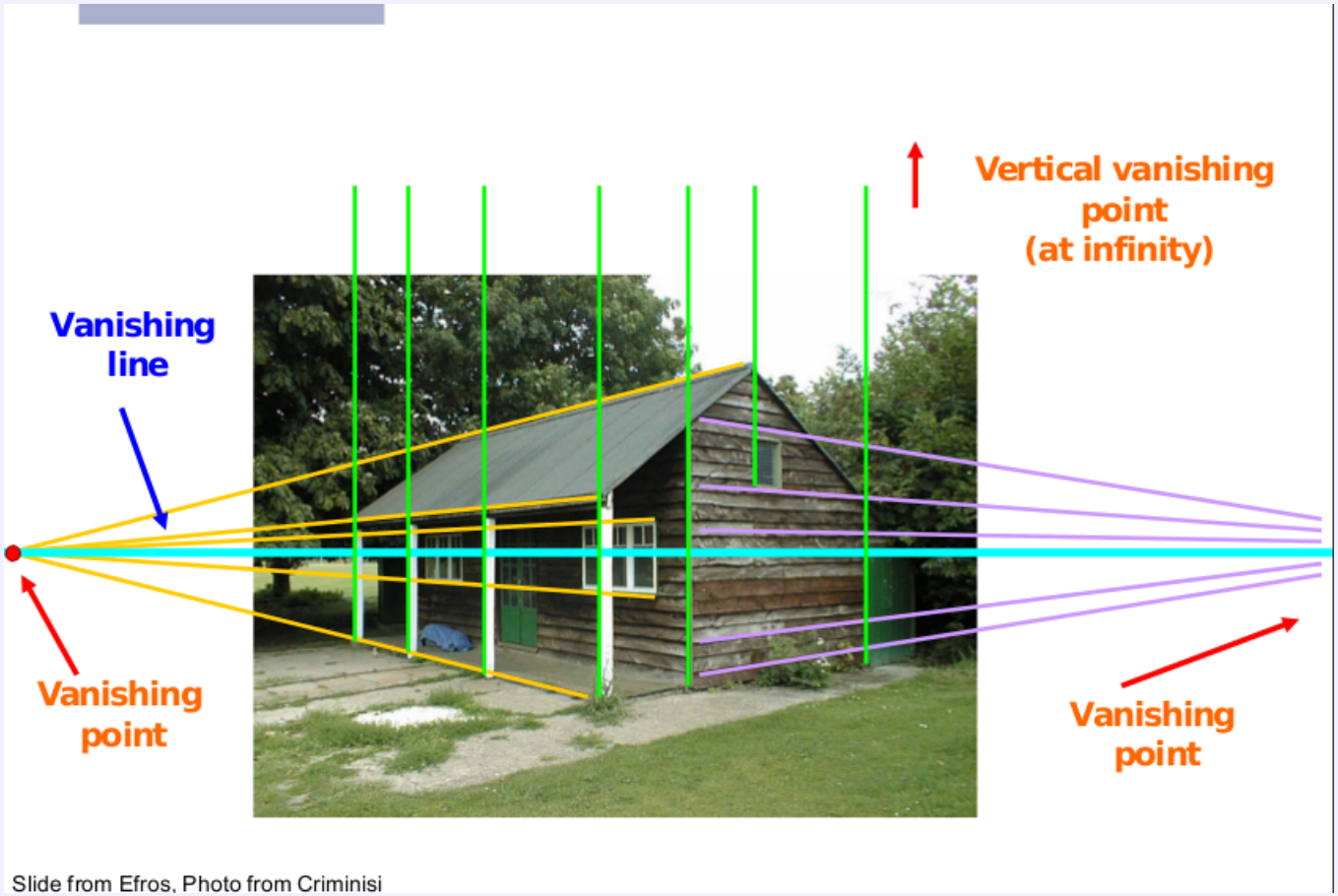

Projections of parallel lines intersect at common points.

# Vanishing line





# Vanishing line



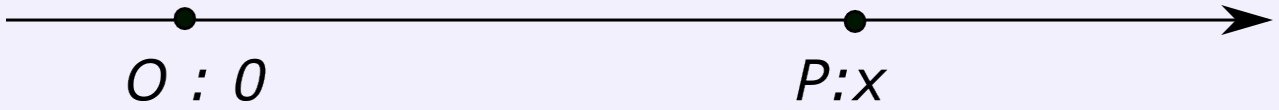

Slide from Efros, Photo from Criminisi

# Shape (VP's) from texture

If we can measure a texture density, then we may estimate the directions of minimal and maximal density change. Theses points at the VPs. thus giving the VL and the 3D surface normal.

# Homogeneous coordinates 1D

- In 1D coordinate is just 1 number.



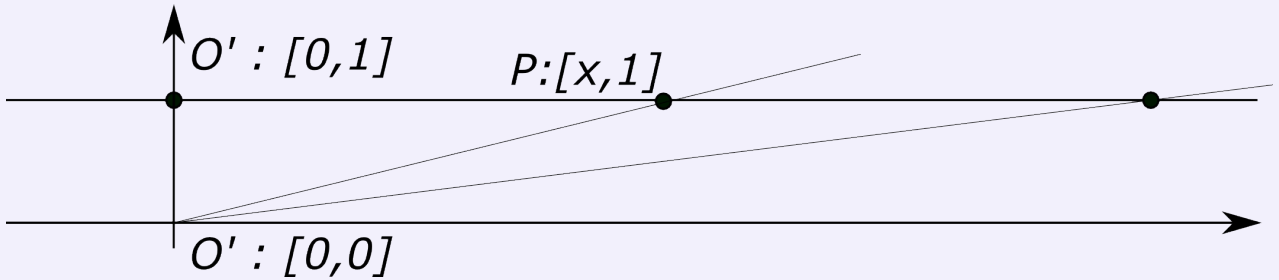

- 1D coordinate to 1D Homogeneous coordinates:

$$x \sim \begin{bmatrix} x \\ 1 \end{bmatrix}$$

- 1D homogeneous coordinate to 1 D coordinate

$$\begin{bmatrix} x \\ w \end{bmatrix} \sim x/w$$

- What can we do with that? we can “tame” infinity!
- A point with homogeneous coordinate $[x, 0]^T$ has “normal” coordinate $x/0 = \infty$ as if we took homogeneous coordinate $[\infty, 1]^T$



# Homogeneous coordinates, 2D

"Natural Coordinates" for projective geometry

- From 2D point coordinate to 2D Homogeneous coordinate

$$(x, y) \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- From 2D homogeneous coordinates to 2D coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Longrightarrow (x/w, y/w)$$

# Homogeneous coordinates, 3D

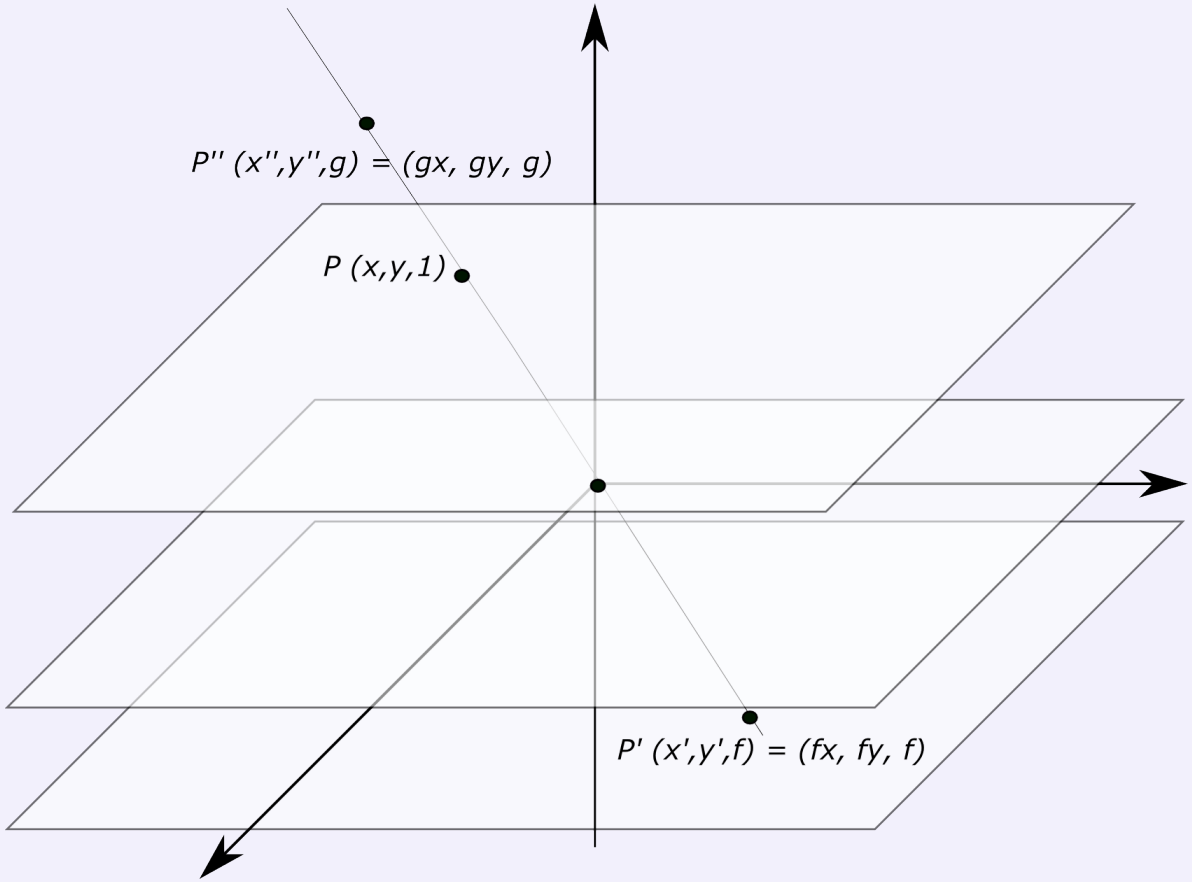
- From 3D point coordinate to 3D Homogeneous coordinate

$$(x, y, z) \Longrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- From 3D homogeneous coordinates to 3D coordinates

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Longrightarrow (x/w, y/w, z/w)$$

# Homogeneous coordinates




Homogeneous coordinates in 2D correspond to points in plane $z = 1$ but also to lines through the origin and this point.

# Cameras and homogeneous coordinates

- Projection to image plane in standard coordinates:

$$P : (x, y, z) \mapsto P' : (f\frac{x}{z}, f\frac{y}{z})$$

- How does the mapping look like in homogeneous coordinates:

$$P : \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \mapsto P' : \begin{bmatrix} fx \\ fy \\ z \end{bmatrix}$$

- Matrix notation

$$\begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{K} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- $K$ is (a simple version of) the Camera Calibration Matrix and contains the intrinsic calibration parameters (here $f$).

# World, Camera and Image Coordinates

In the previous slides we were not precise on the many coordinate systems are implicitly used:

- 3D World Coordinates: Coordinate system of the 3D world.
- Camera Coordinates: 3D coordinate system attached to the camera.
- Image Coordinates: 2D Coordinate system attached to the image plane.
- The coordinate system for the sampled and digitised image

# Intrinsic vs Extrinsic Camera Parameters

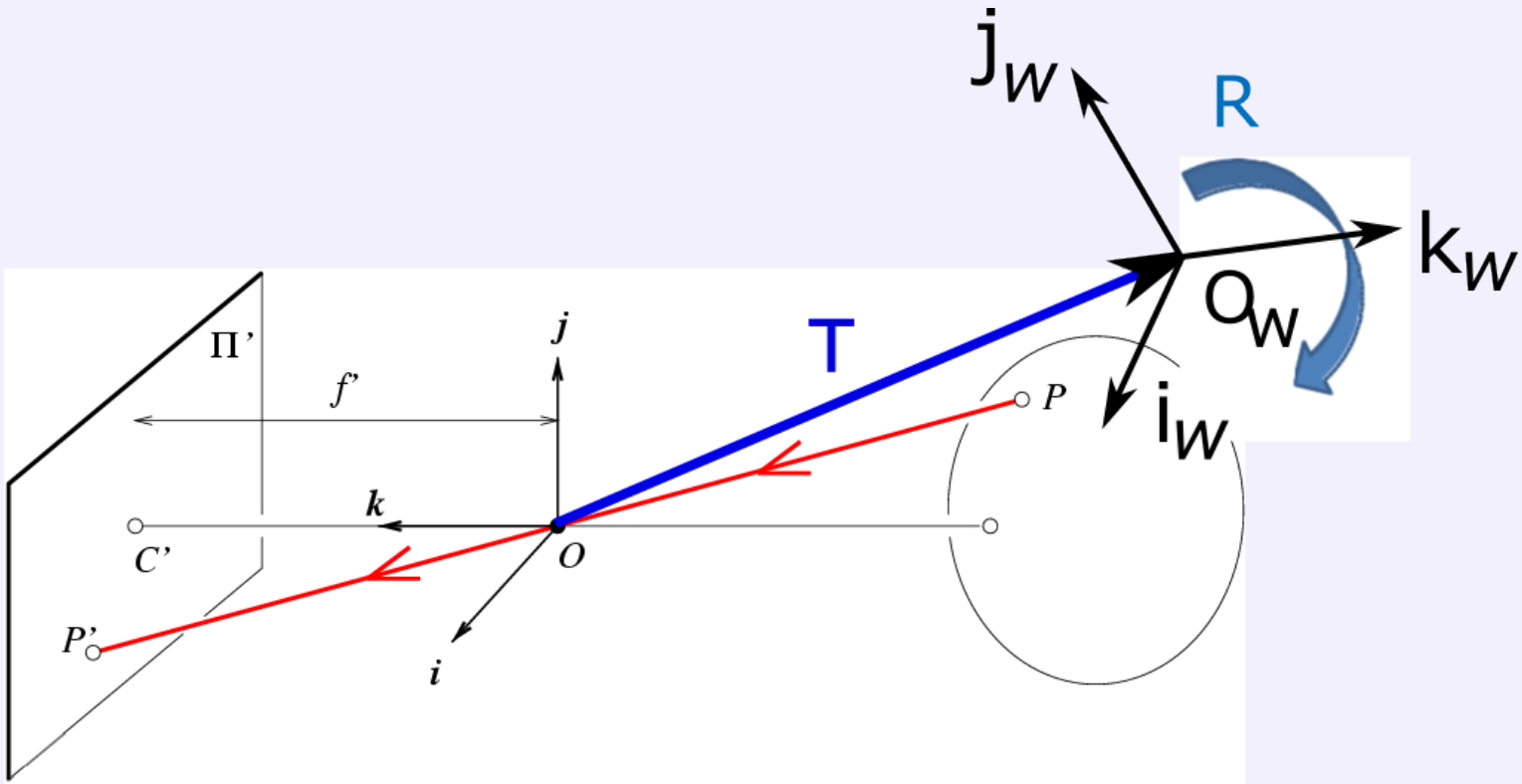
Intrinsic parameters refer to internal parameters:

- The principal point: 2 parameters
- Scale factors (or sampling frequencies) for the pixels sizes in both x and y directions: 2 parameters, multiplied by the focal length, resulting in the effective focal length $(f_x, f_y) = f \cdot (s_x, s_y)$.
- Skewness of pixels: 1 parameter. (Often assumed zero)

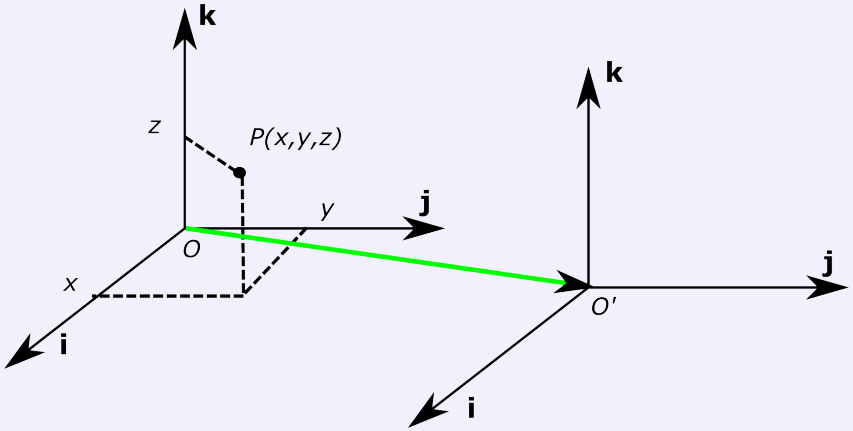
Extrinsic Camera parameters:

- Position of the camera coordinate system in the world coordinates system: translation: 3 parameters,
- Orientation of the camera coordinate system in the world coordinates system: rotation: 3 parameters.

# Oriented and Translated Camera



# Translating Coordinate System



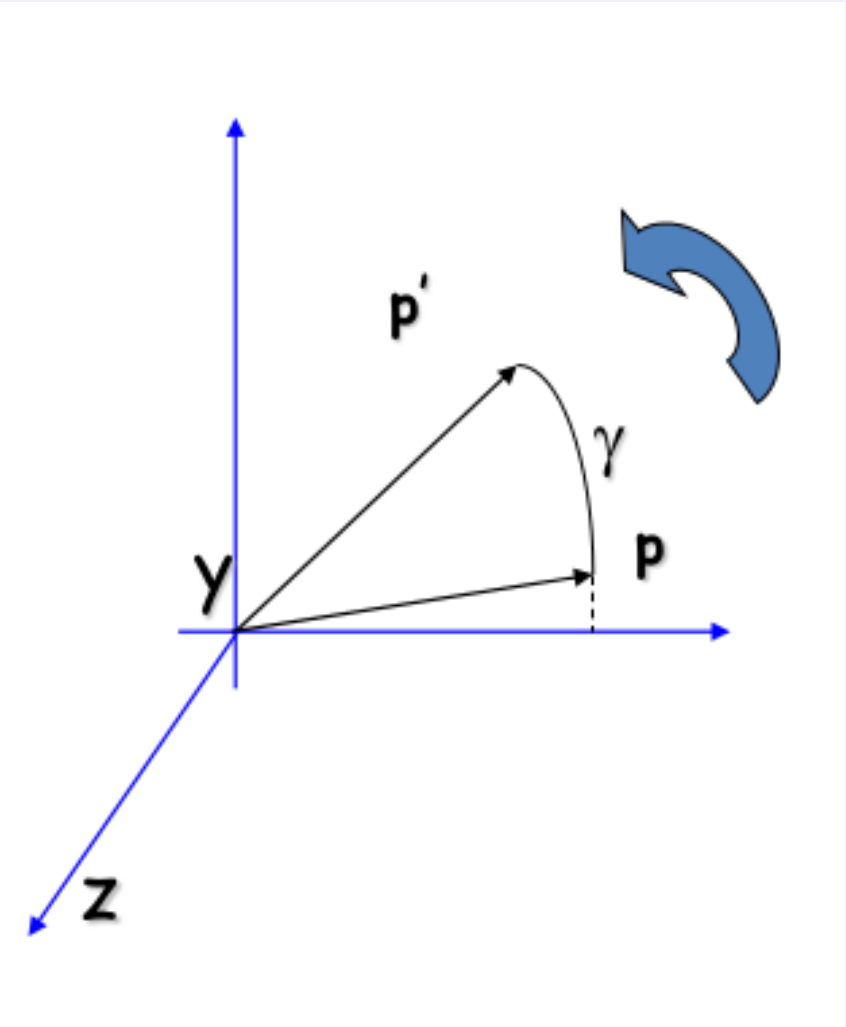

- Subtract the translation vector:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} x_{O'} \\ y_{O'} \\ z_{O'} \end{pmatrix} = \begin{pmatrix} x - x_{O'} \\ y - y_{O'} \\ z - z_{O'} \end{pmatrix}$$

- Transformation in homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & -x_{O'} \\ 0 & 1 & 0 & -y_{O'} \\ 0 & 0 & 1 & -z_{O'} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Rotating Coordinate System




Rotations along coordinate axes

$$R_{x\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

$$R_{y\beta} = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}$$

$$R_{z\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Can also be written in homogeneous coordinates

# 3D rotations

To obtain a 3D rotation apply all 3 single-axis rotations:

$$
\begin{aligned}
R &= R_{x\alpha} R_{y\beta} R_{z\gamma} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

Please notice that the order of the 3 rotation matrices do matter. However, no matter which order $(\alpha, \beta, \gamma)$ may be found such that the resulting 3D rotation is correct.

# Camera Matrix

- Camera calibration matrix, now extended with Image plane transformation (axis scalings, shear, translation)

$$\mathbf{C} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}]$$

- $\mathbf{K}$ $3 \times 3$ matrix encoding the homogeneous transformations inside the camera. $\mathbf{K}$ specifies the Intrinsic parameters.
- $[\mathbf{R}\ \mathbf{t}]$ Concatenation of world coordinates rotation and origin translation to align camera and world coordinates.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \underbrace{\begin{pmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}}_{[\mathbf{R}\ \mathbf{t}]} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

# Exercise

- How many independent parameters are there to estimate in a camera calibration ?

# Exercise

- How many independent parameters are there to estimate in a camera calibration ?

  11: 5 intrinsic and 6 extrinsic (sometimes simplified to 3+6)

- May all calibration parameters be found using Linear algebra ?

# Exercise

- How many independent parameters are there to estimate in a camera calibration ?

  11: 5 intrinsic and 6 extrinsic (sometimes simplified to 3+6)

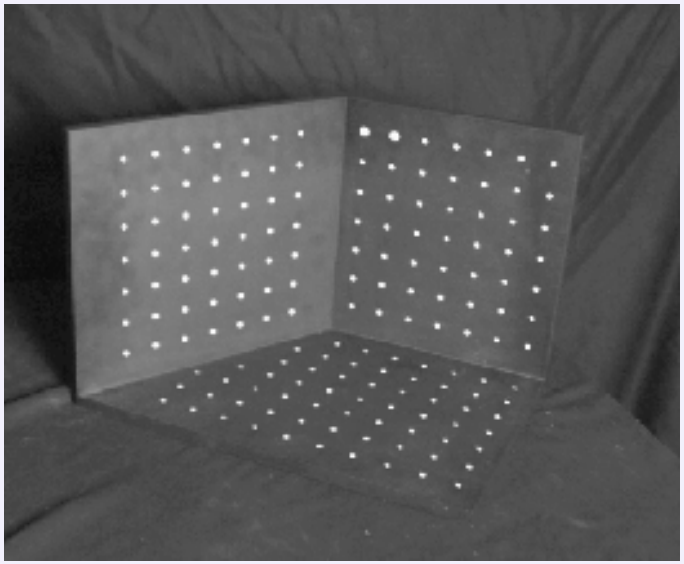
- May all calibration parameters be found using Linear algebra ?

  No, they don't appear in a linear combinations. Some are multiplied together.

# QUESTIONS ?

# Geometric Calibration

- Computing the camera matrix is called geometric calibration.
- Extrinsic parameters: (R,t): Usually easy, but requires metric knowledge on scene features.
- Intrinsic parameters: (**K**): Some parameters ($f$) are easy and some (($u_0, v_0$)) are difficult to estimate correctly.



- Use an object with known geometry
- Use vanishing points / lines
- Use other cues...

# Repetition: The camera matrix

Please recall the definition of the camera matrix:

$$M = K\,[R\ \mathbf{t}]$$

and the projection written out:

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\underbrace{\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}}_{[\mathbf{R}\ \mathbf{t}]}}_{\mathbf{M}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

or

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{pmatrix} \mathbf{m}^1 \\ \mathbf{m}^2 \\ \mathbf{m}^3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{m}^1 U \\ \mathbf{m}^2 U \\ \mathbf{m}^3 U \end{pmatrix}$$

where $\mathbf{m}^i$ is the $i$'th row of the camera matrix.

# Camera calibration

In Camera calibration the task is to recover the 12 unknowns $\mathbf{m}_{ij}$ in the camera calibration matrix. Converting the result before to image coordinates we have:

$$\begin{aligned} x\mathbf{m}_3 U &= \mathbf{m}_1 U \\ y\mathbf{m}_3 U &= \mathbf{m}_2 U \end{aligned}$$

Let $\mathbf{m} = (m_{11}, \ldots, m_{14}, \ldots, m_{34})^\top$ be the vector of the 12 unknowns. Isolating these in the equations above lead to: $A\mathbf{m} = 0$, where:

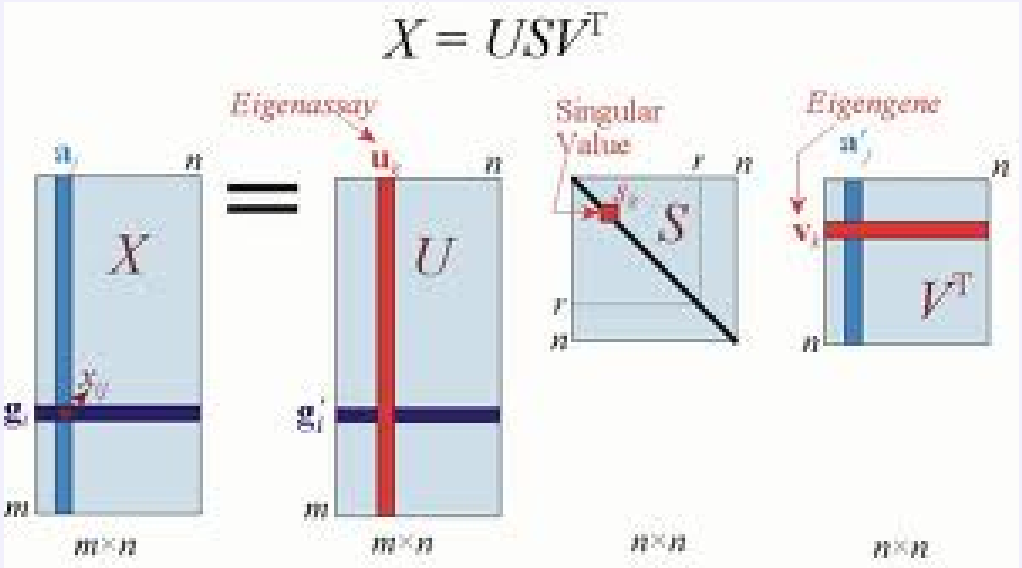
$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 & -x_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 & -y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix}$$

# Solving $Ax = 0$

Many different numerical approaches. Singular Value Decomposition (SVD) is based on decomposition of the matrix $A$ into a product of 3 matrices:

$$A = UDV^\top$$

where $A$ and $U$ is $m \times n$ and $U^\top U = I$,
$D$ is a diagonal $n \times n$ matrix with non-negative singular values,
and $V$ is a $n \times n$ unitary matrix $VV^\top = V^\top V = I$.



# Singular value decomposition (svd)

$$A\,\mathbf{x} \;=\; UDV^{\top}\,\mathbf{x} \;=\; 0$$

We are not interested in the trivial solution $\mathbf{x} = \mathbf{0}$ but rather all solution lying in the Null-space of the matrix $A$.

The singular values in the diagonal matrix $D$ specifies the energy contained in each of the dimensions. Usually, these are sorted in decreasing order.

Often, due to noise all singular values $\sigma_i > 0$. If we know that $A$ should be singular may find the closest such matrix by zeroing out the smallest singular value.

We get the solution to $A\mathbf{x} = \mathbf{0}$ as the last column of $V$. In MATLAB:

```
[U,D,V] = svd(A); x = V(:,end);
```

Notice that $||x|| = 1$

# Camera calibration continued

- Given 6 points $(x, y)$ projected from 6 known 3D points $(X, Y, Z)$ we can solve for the camera matrix elements $m_{ij}$.
- In practise many more than 6 points are preferred to cancel noise.
- From the 12 elements $m_{ij}$ it is possible to recover all 11 parameters $f_x$, $f_y$, $\alpha$, $\beta$, $\gamma$, etc.
- There exist more advanced calibration methods than the shown linear calibration.

# QUESTIONS ?

# Assignment 4 - details

- The assignment consists in implementing a prototypical Content Based Image Retrieval System
- Five parts:
  1. Select and prepare the data you want to work with
  2. Gather descriptors in training and test image data set
  3. Construct codebook using k-means and Bag-Of-Words
  4. Project test images onto codebook, and generate BoW
  5. Retrieve according to similarity measure

# What descriptors

- You are advised to use SIFT features
- Few versions of SIFT include color. You may miss a lot of information.
- You may include color histograms etc. but
- Advise: Keep it simple

# Training and testing

- Download Caltech-101 image database (131 MB) with 101 categories or the newer Caltech-256 base (1.2 GB)
- Split data into two parts: Training data [say 70-80%] and test [say 20-30 %]. Don't look at the test data before testing
- During development you may split the training data into a construction part [say 80%] and an validation part [say 20%].
- Never use the test set for tuning !

# What error ?

- Error on training data versus error on test data
- What if the training error is less than the test error?
- Overfitting is a serious problem. Often caused by using too few training data (or too complicated model)
- We want to generalize in order to retrieve/classify new data

# Cross-validation

- Divide data into say 10 sets (10-fold CV)
- Train on all but one set that is used testing
- Retrain and test on all possible combinations
- Compute the average test error

# How to report

- Amount: 8 pages including everything: Try to be concise, structured etc. Keep text less than 3-4 pages
- Discussion: This is the most important part
- Tables: Remember to tell me what I should notice
- Figures: Remember axis labels etc. What should I see?
- Images: Don't show them too small. What should I see?
- Explanations: Don't expect me to guess what you mean or see yourself
- ERROR ANALYSIS: Explain what you think may cause observed errors or unexpected behaviour

# QUESTIONS ?

# Next time

Next time, our first meeting in 2022, we will continue with stereo analysis, multi view geometry and may be a bit about image stitching.