Lecture 03
Image Formation 2

Prof. Dr. Davide Scaramuzza

sdavide@ifi.uzh.ch

Today afternoon

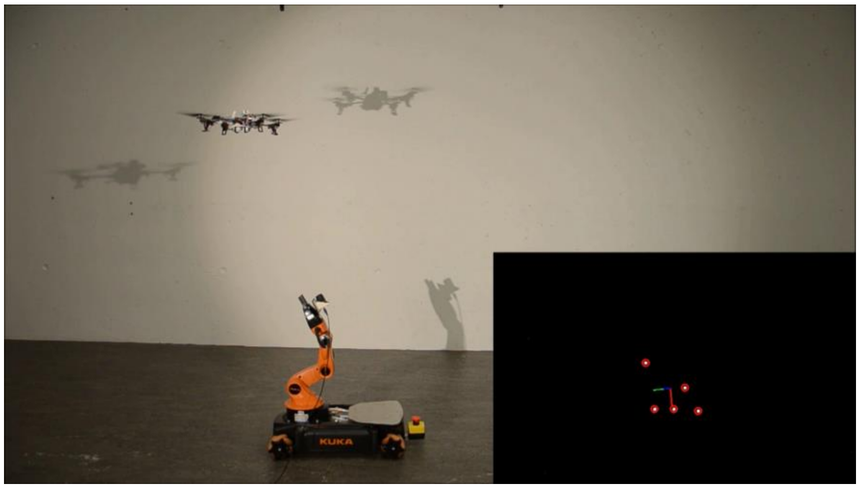
- Room 2.A.01 from 14:15 to 16:00
  - Matlab introduction
  - Filtering

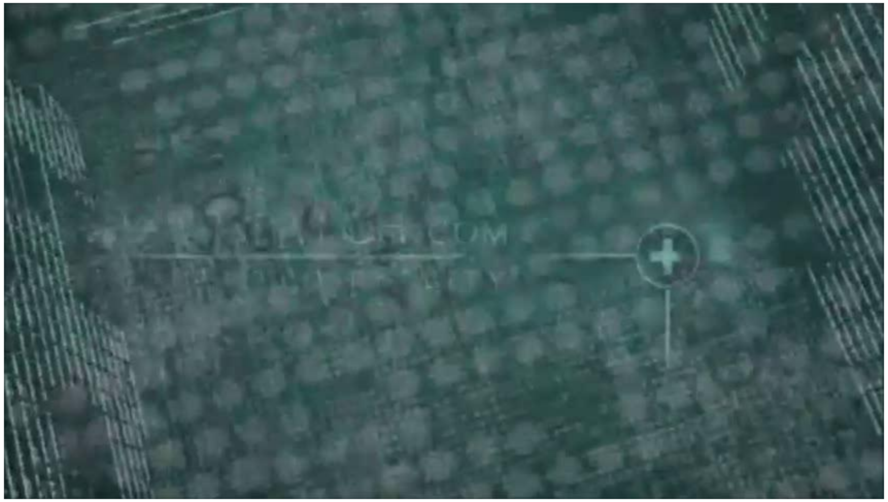
Mini Project List

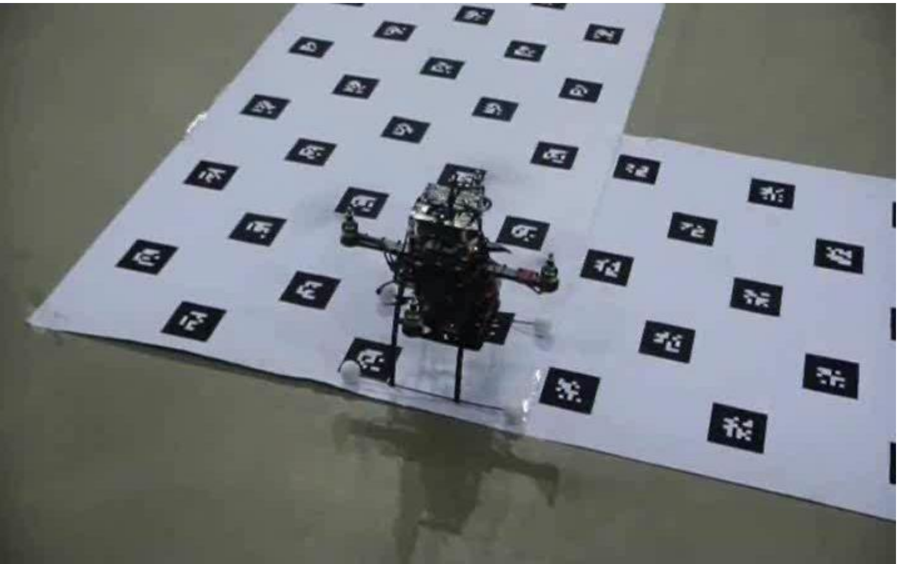
- Deadlines:
  - Discuss and come to an agreement with the teaching assistants before **December 1, 2015**.
  - Hand in your project (code, description, short documentation) by **December 19, 2015**.

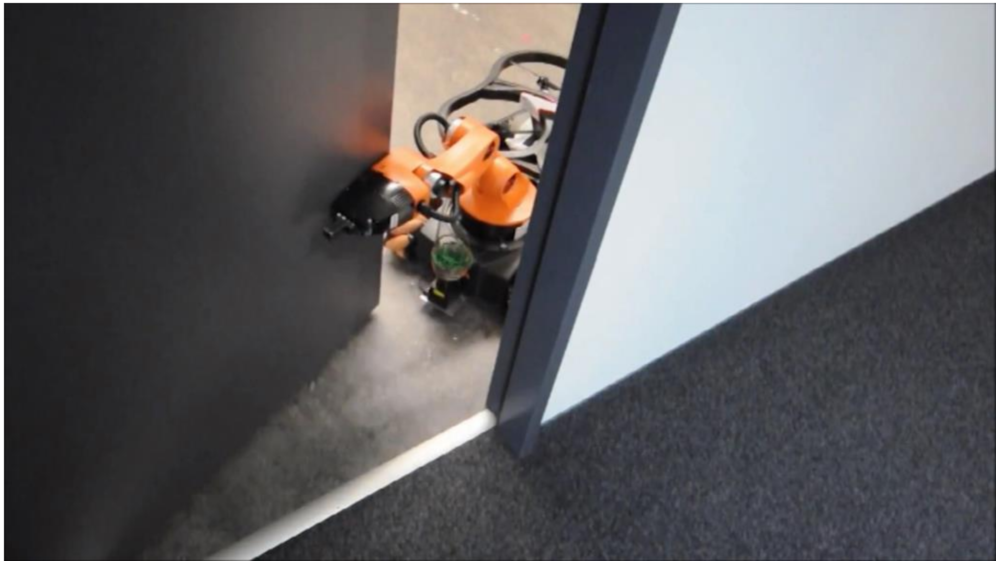
Goal of today's lecure

- Study the algorithms behind robot-position control and augmented reality







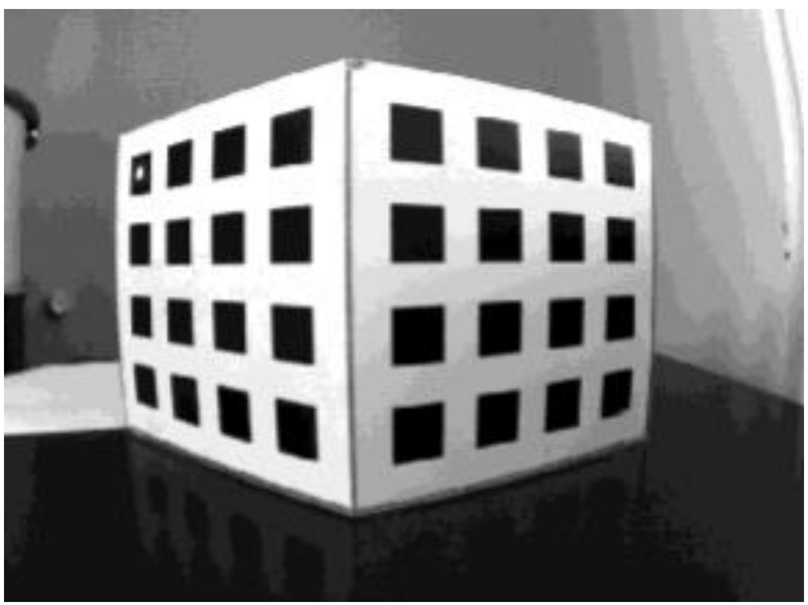


Outline of this lecture

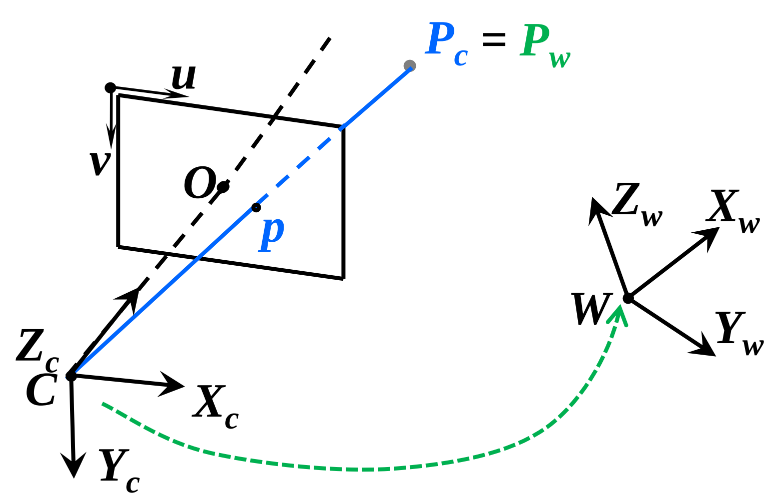
- Camera calibration
  - From 3D objects
  - From planar grids
- Non conventional camera models

Camera calibration

- Calibration is the process to determine the **intrinsic and extrinsic** parameters of the camera model
- A method proposed in 1987 by Tsai consists of measuring the 3D position of $n \geq 6$ control points on a three-dimensional calibration target and the 2D coordinates of their projection in the image. This problem is also called “**Resection**”, or “**Perspective from n Points**”, or “**Camera pose from 3D-to-2D correspondences**”, and is one of the most widely used algorithms in Computer Vision and Robotics
- Solution: The intrinsic and extrinsic parameters are computed directly from the perspective projection equation; let’s see how!



3D position of control points is assigned in a reference frame specified by the user



Camera calibration: Direct Linear Transform (DLT)

Our goal is to compute K, R, and T, that satisfy the perspective projection equation (we neglect the radial distortion)

$$\tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\left[R \mid T\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_1 + u_0 t_3 \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_2 + v_0 t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera calibration: Direct Linear Transform (DLT)

Our goal is to compute K, R, and T, that satisfy the perspective projection equation (we neglect the radial distortion)

$$\tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\left[R \mid T\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera calibration: Direct Linear Transform (DLT)

Our goal is to compute K, R, and T, that satisfy the perspective projection equation (we neglect the radial distortion)

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = M \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

where m_i^T is the i-*th* row of M

Camera calibration: Direct Linear Transform (DLT)

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \rightarrow P$$

Conversion back from homogeneous coordinates to pixel coordinates leads to:

$$u = \frac{\tilde{u}}{\tilde{w}} = \frac{m_1^T \cdot P}{m_3^T \cdot P}$$

$$v = \frac{\tilde{v}}{\tilde{w}} = \frac{m_2^T \cdot P}{m_3^T \cdot P}$$

$$\Rightarrow \quad \begin{aligned} (m_1^T - u_i m_3^T) \cdot P_i &= 0 \\ (m_2^T - v_i m_3^T) \cdot P_i &= 0 \end{aligned}$$

Camera calibration: Direct Linear Transform (DLT)

By re-arranging the terms, we obtain

$$(m_1^T - u_i m_3^T) \cdot P_i = 0$$
$$(m_2^T - v_i m_3^T) \cdot P_i = 0 \Rightarrow \begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

For n points, we can stack all these equations into a big matrix:

$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Camera calibration: Direct Linear Transform (DLT)

By re-arranging the terms, we obtain

$$\begin{matrix} (m_1^T - u_i m_3^T)\cdot P_i = 0 \\ (m_2^T - v_i m_3^T)\cdot P_i = 0 \end{matrix} \Rightarrow \begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

For n points, we can stack all these equations into a big matrix:

$$\underbrace{\begin{pmatrix} X_w^1 & Y_w^1 & Z_w^1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_w^1 & -u_1 Y_w^1 & -u_1 Z_w^1 & -u_1 \\ 0 & 0 & 0 & 0 & X_w^1 & Y_w^1 & Z_w^1 & 1 & -v_1 X_w^1 & -v_1 Y_w^1 & -v_1 Z_w^1 & -v_1 \\ & & & & & & \cdots & \cdots & \cdots & & & \\ X_w^n & Y_w^n & Z_w^n & 1 & 0 & 0 & 0 & 0 & -u_n X_w^n & -u_n Y_w^n & -u_n Z_w^n & -u_n \\ 0 & 0 & 0 & 0 & X_w^n & Y_w^n & Z_w^n & 1 & -v_n X_w^n & -v_n Y_w^n & -v_n Z_w^n & -v_n \end{pmatrix}}_{\text{Q (this matrix is known)}} \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix}}_{\text{M (this matrix is unknown)}} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow \mathrm{Q}\cdot\mathrm{M} = 0$$

Q (this matrix is **known**)

M (this matrix is **unknown**)

Camera calibration: Direct Linear Transform (DLT)

$$Q \cdot M = 0$$

Minimal solution

- Q has 11 Degrees of Freedom (in fact, Q is valid up to a scale factor, thus, 12-1 = 11)
- Each 3D-to-2D point correspondence provides 2 independent equations
- Thus, $5+\frac{1}{2}$ point correspondences are needed (in practice **6 point** correspondences!)

Over-determined solution

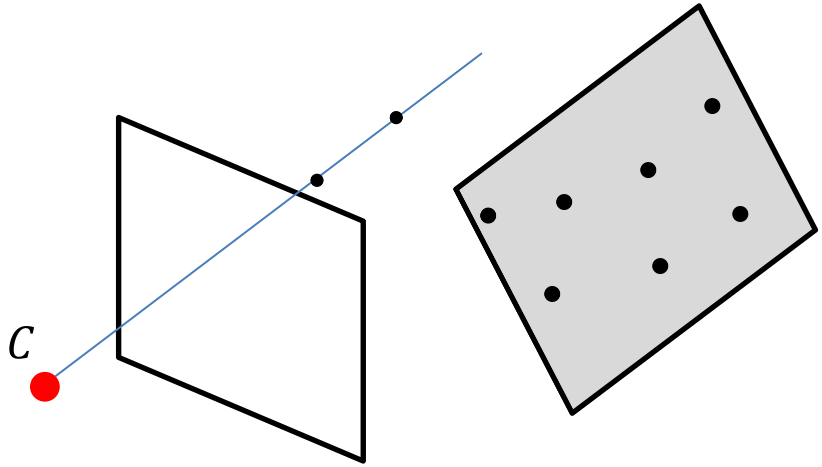
- $n \geq 6$ points
- A solution is to minimize $||QM||$ subject to the constraint $||M||^2 = 1$.
  It can be solved through Singular Value Decomposition (SVD). The solution is the eigenvector corresponding to the smallest eigenvalue of the matrix $Q^T Q$ (because it is the unit vector x that minimizes $x^T Q^T Q x$).

Camera calibration: Direct Linear Transform (DLT)

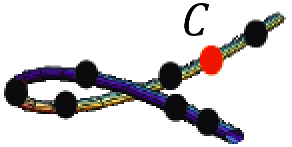
$$\mathrm{Q} \cdot \mathrm{M} = 0$$

Degenerate configurations

1. Points lying on a **plane** and/or along a single **line** passing through the **projection center**



2. Camera and points on a twisted cubic (i.e., smooth curve in 3D space of degree 3)



Camera calibration: Direct Linear Transform (DLT)

- Once we have the M matrix, we can recover the intrinsic and extrinsic parameters by remembering that

$$\mathrm{M} = \mathrm{K}(\mathrm{R} \mid \mathrm{T})$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_1 \\ m_{21} & m_{22} & m_{23} & m_2 \\ m_{31} & m_{32} & m_{33} & m_3 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

Camera calibration: Direct Linear Transform (DLT)

- Once we have the M matrix, we can recover the intrinsic and extrinsic parameters by remembering that

$$\mathrm{M} = \mathrm{K}(\mathrm{R} \mid \mathrm{T})$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_1 \\ m_{21} & m_{22} & m_{23} & m_2 \\ m_{31} & m_{32} & m_{33} & m_3 \end{bmatrix} = \begin{bmatrix} \alpha r_{11} + u_0 r_{31} & \alpha r_{12} + u_0 r_{32} & \alpha r_{13} + u_0 r_{33} & \alpha t_1 + u_0 t_3 \\ \alpha r_{21} + v_0 r_{31} & \alpha r_{22} + v_0 r_{32} & \alpha r_{23} + v_0 r_{33} & \alpha t_2 + v_0 t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

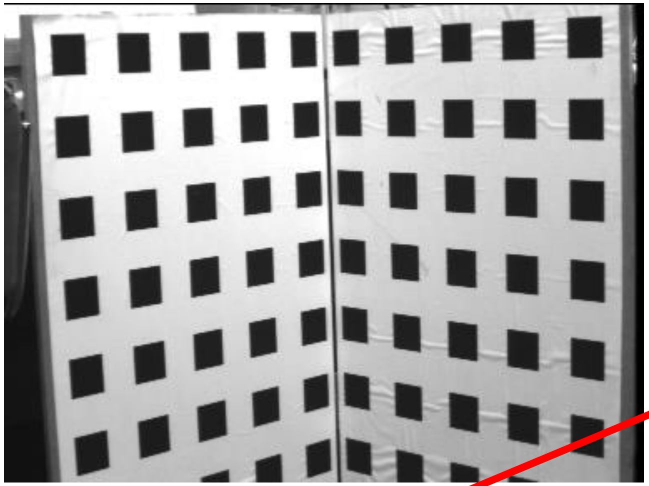
- However, notice that we are not enforcing the constraint that R is orthonormal, i.e., $R \cdot R^T = I$

- To do this, we can use the so-called QR factorization of M, which decomposes M into a R (orthonormal), T, and an upper triangular matrix (i.e., K)

Tsai’s (1987) Calibration example

1. Edge detection
2. Straight line fitting to the detected edges
3. Intersecting the lines to obtain the images corners (corner accuracy <0.1 pixels!)
4. Use >6 points

Why is this ratio not 1?

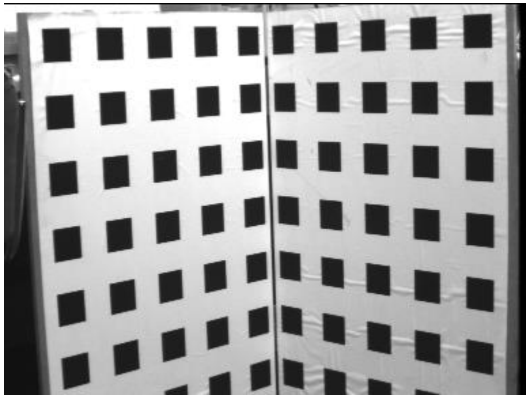


What are the «skew» and «residuals»?

f_y	f_x/f_y	skew	x_0	y_0	residual
1673.3	1.0063	1.39	379.96	305.78	0.365

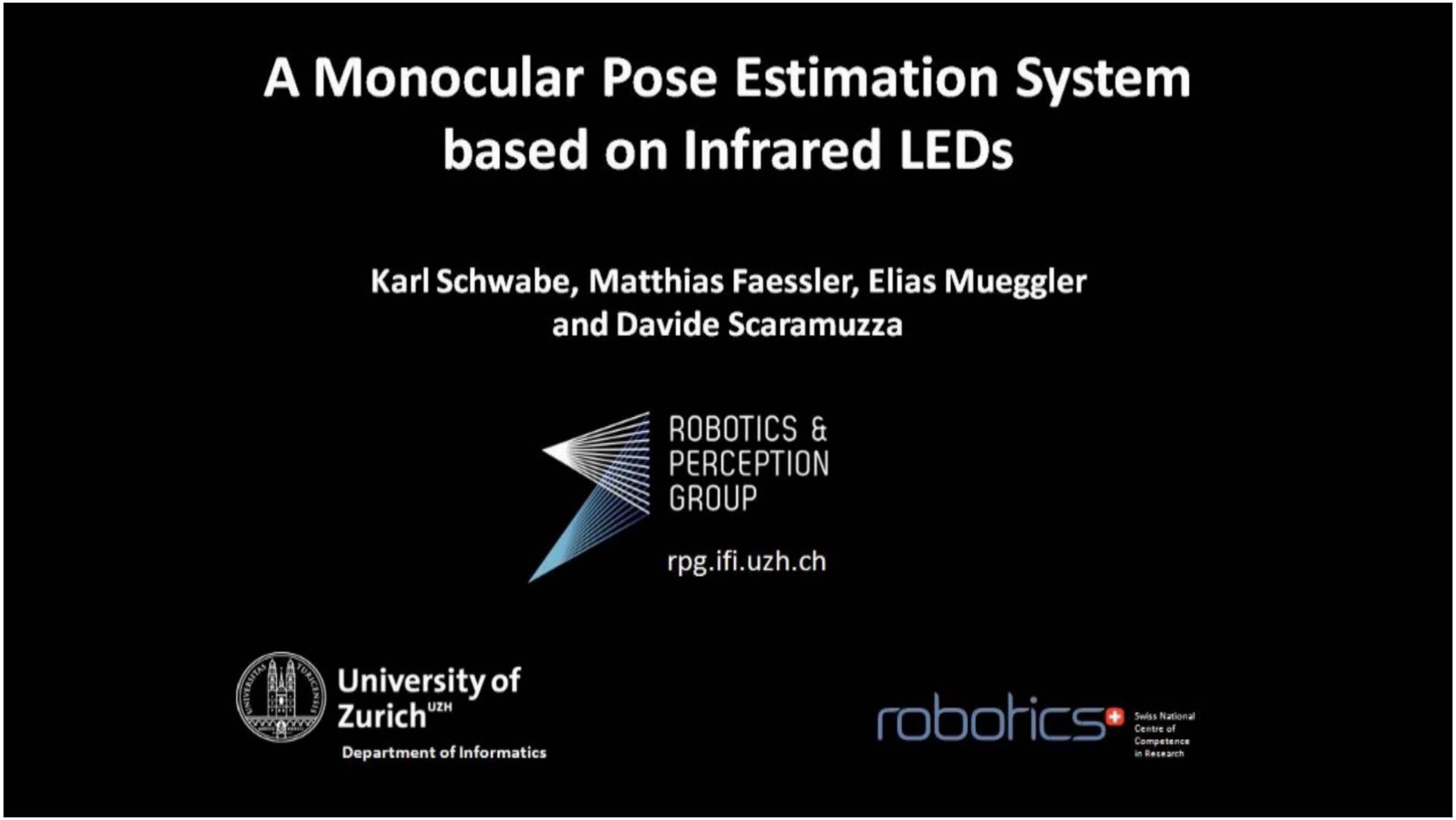
Tsai’s (1987) Calibration example

- The original Tsai calibration (1987) used to consider two different focal lengths α_u, α_v (which means that the pixels are not squared) and a skew factor ($K_{12} \neq 0$, which means the pixes are parallelograms instead of rectangles). This relaxation was used to account for possible misalignments between CCD and lens
- Most of today’s camera are well manufactured, thus, we can assume $\frac{\alpha_u}{\alpha_v} = 1$ and $K_{12} = 0$
- What is the residual? The residual is the *average* “reprojection error”. The reprojection error is computed as the distance (in pixels) between the observed pixel point and the camera-reprojected 3D point. The reprojection error gives as a quantitative measure of the accuracy of the calibration (ideally it should be zero).



f_y	f_x/f_y	skew	x_0	y_0	residual
1673.3	1.0063	1.39	379.96	305.78	0.365

DLT algorithm applied to mutual robot localization



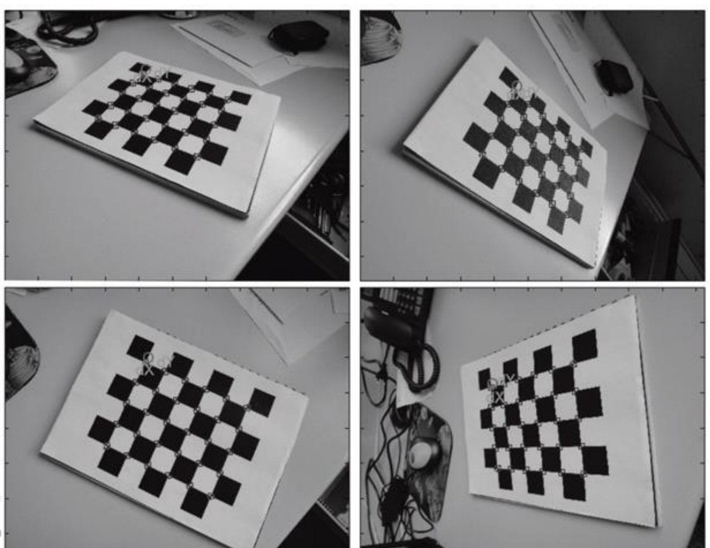
In this case, the camera has been pre-calibrated (i.e., K is known). Can you think of how the DLT algorithm could be modified so that only R and T need to determined and not K?

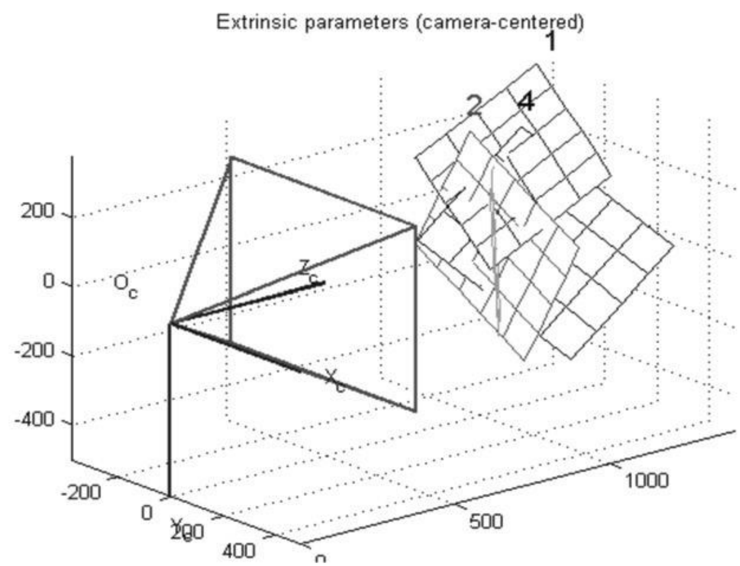
Outline of this lecture

- Camera calibration
  - From 3D objects
  - From planar grids
- Non conventional camera models

Camera calibration from planar grids: homographies

- Tsai calibration is based on DLT algorithm, which requires points not to lie on the same plane
- An alternative method (today's standar camera calibration method) consists of using a planar grid (e.g., a chessboard) and a few images of this shown at different orientations
- This method was invented by Zhang (1999)





Camera calibration from planar grids: homographies

- Our goal is to compute K, R, and T, that satisfy the perspective projection equation (we neglect the radial distortion)
- Since the points lie on a plane, we have $Z_w = 0$

$$\tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid T] \cdot \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} \Rightarrow$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

Camera calibration from planar grids: homographies

- Our goal is to compute K, R, and T, that satisfy the perspective projection equation (we neglect the radial distortion)
- Since the points lie on a plane, we have $Z_w = 0$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = H \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

This matrix is called Homography

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

where h_i^T is the i-*th* row of H

Camera calibration from planar grids: homographies

$$\Rightarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

Conversion back from homogeneous coordinates to pixel coordinates leads to:

$$u = \frac{\tilde{u}}{\tilde{w}} = \frac{h_1^T \cdot P}{h_3^T \cdot P} \qquad v = \frac{\tilde{v}}{\tilde{w}} = \frac{h_2^T \cdot P}{h_3^T \cdot P} \quad \Rightarrow \quad \begin{array}{l} (h_1^T - u_i h_3^T) \cdot P_i = 0 \\ (h_2^T - v_i h_3^T) \cdot P_i = 0 \end{array}$$

where P = $(X_w, Y_w, 1)^T$

Camera calibration from planar grids: homographies

By re-arranging the terms, we obtain

$$\begin{matrix} (h_1^T - u_i h_3^T) \cdot P_i = 0 \\ (h_2^T - v_i h_3^T) \cdot P_i = 0 \end{matrix} \Rightarrow \begin{matrix} P_i^T \cdot h_1 + 0 \cdot h_2^T - u_i P_i^T \cdot h_3^T = 0 \\ 0 \cdot h_1^T + P_i^T \cdot h_2^T - v_i P_i^T \cdot h_3^T = 0 \end{matrix} \Rightarrow \begin{pmatrix} P_i^T & 0^T & -u_1 P_i^T \\ 0^T & P_i^T & -v_1 P_i^T \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

For n points, we can stack all these equations into a big matrix:

$$\underbrace{\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix}}_{\text{Q (this matrix is known)}} \underbrace{\begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix}}_{\text{H (this matrix is unknown)}} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow \mathrm{Q} \cdot \mathrm{H} = 0$$

Q (this matrix is **known**) H (this matrix is **unknown**)

Camera calibration from planar grids: homographies

$$\mathrm{Q} \cdot \mathrm{H} = 0$$

Minimal solution

- $Q_{(n\times 9)}$ has 8 Degrees of Freedom (in fact, Q is valid up to a scale factor, thus, 9-1 = 8)
- Each point correspondence provides 2 independent equations
- Thus, a minimum of **4 non-collinear points** is required

Over-determined solution

- *n* > 4 points
- It can be solved through Singular Value Decomposition (SVD)

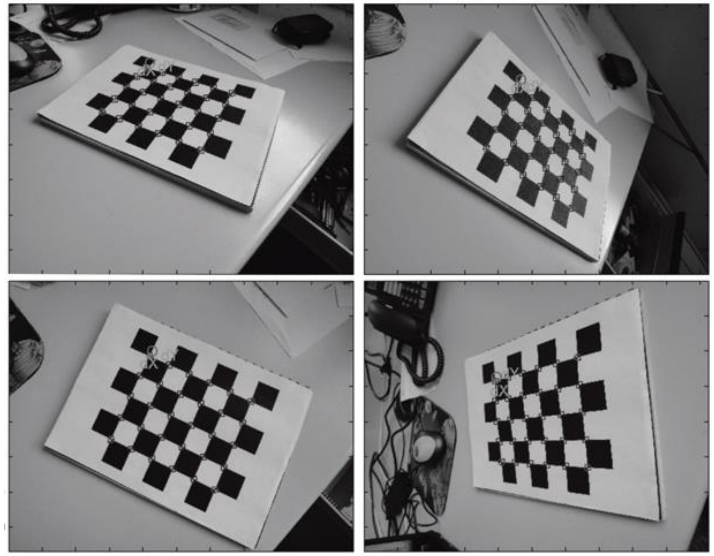
Solving for K, R and T

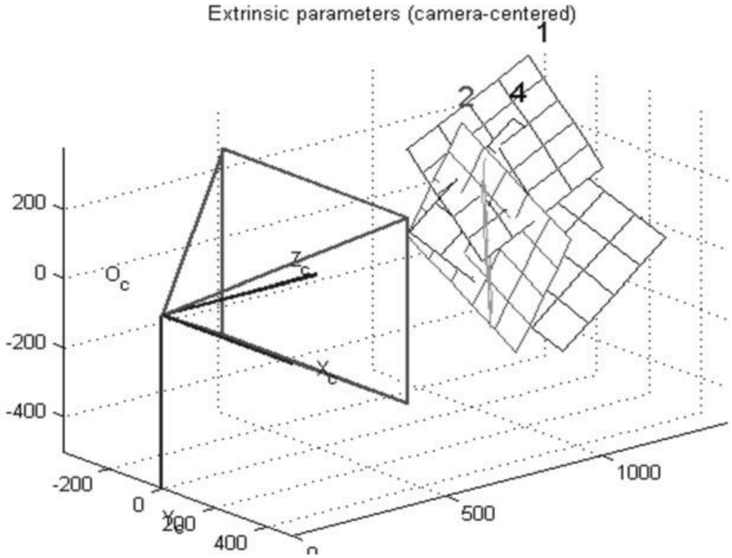
- H can be decomposed by recalling that

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}$$

Camera calibration from planar grids: homographies

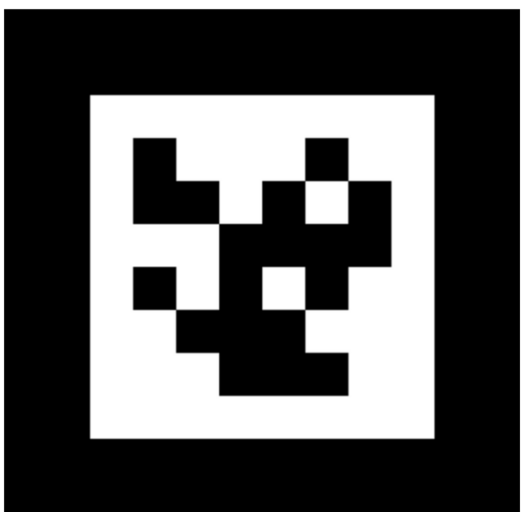
- Demo of Camera Calibration Toolbox for Matlab (world's standard toolbox for calibrating perspective cameras): http://www.vision.caltech.edu/bouguetj/calib_doc/





Application of calibration from planar grids

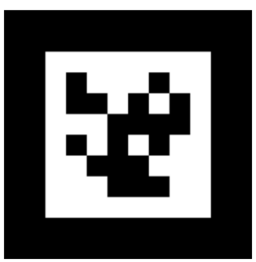
- Today, there thousands of application of this algorithm:
  - Augemented reality

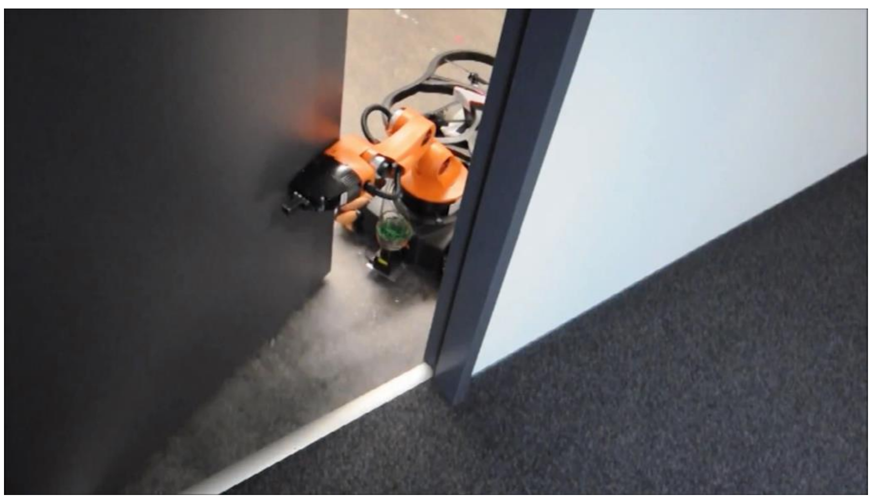


AR Tags: http://april.eecs.umich.edu/wiki/index.php/April_Tags

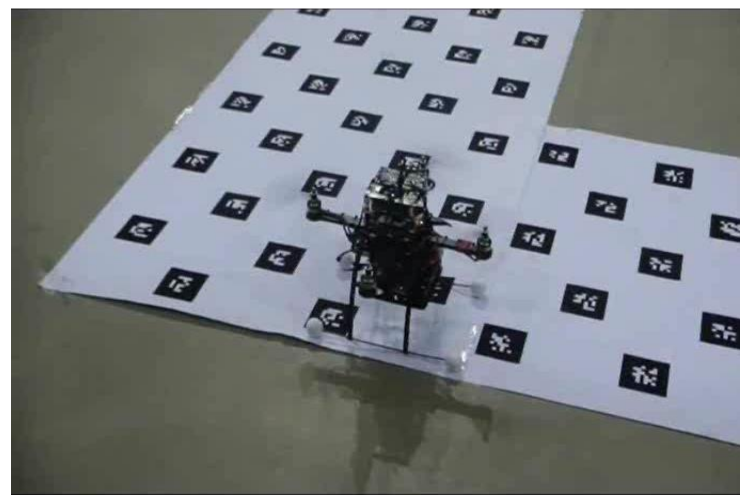
Application of calibration from planar grids

- Today, there are thousands of application of this algorithm:
  - Augemented reality
  - Robotics (beacon-based localization)
- Do we need to know the metric size of the tag?
  - For Augmented Reality?
  - For Robotics?





RPG (us) 2013



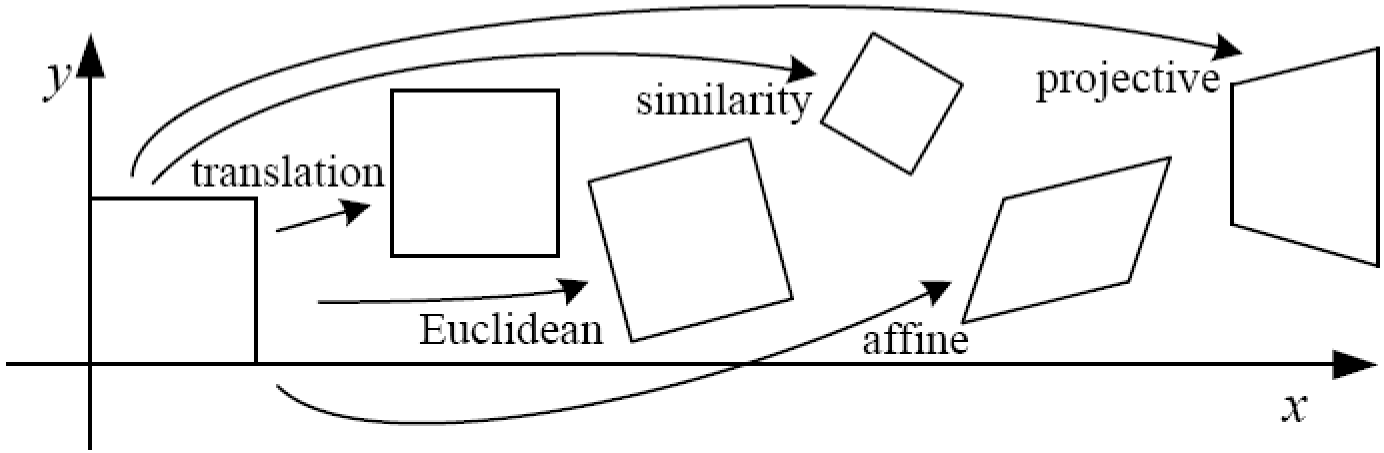
ETH, Pollefeys group, 2010

AR Tags: http://april.eecs.umich.edu/wiki/index.php/April_Tags

Concepts to remember

- Camera calibration
  - DLT algorithm
  - Calibration from planar grids
- Readings:
  - Chapter 2.1 of Szeliski book (freely downloadable from http://szeliski.org/Book/
  - Chapters 4.1-4.3 of Autonomous Mobile Robots book

Transformations – 2D



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation $+ \cdots$	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths $+ \cdots$	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles $+$	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism $+ \cdots$	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	straight lines	

This transformation is called Homography

Outline of this lecture

- Camera calibration
  - From 3D objects
  - From planar grids
- Non conventional camera models

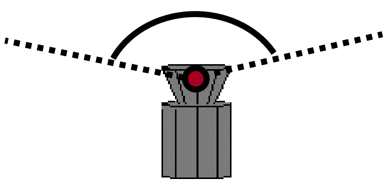
Omnidirectional Cameras

Rome, St. Peter's square

Overview on Omnidirectional Cameras

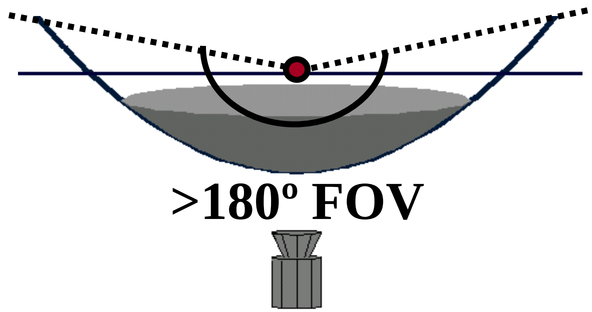
Omnidirectional sensors come in many varieties, but by definition must have a wide field-of-view.

~180º FOV



Wide FOV dioptric cameras (e.g. fisheye)

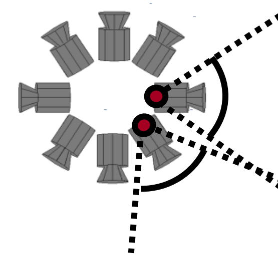
Dioptric



Catadioptric cameras (e.g. cameras and mirror systems)

Catadioptric

~360º FOV

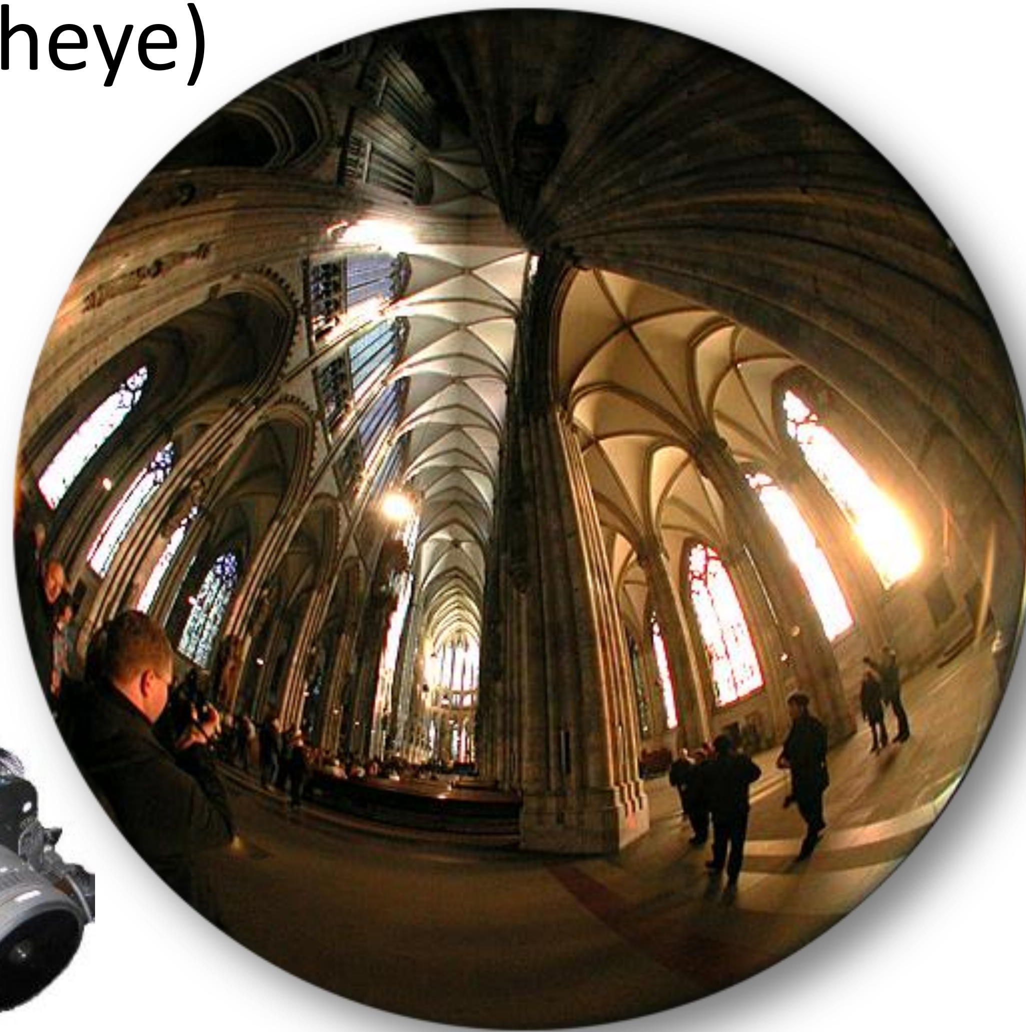


Polydioptric cameras (e.g. multiple overlapping cameras)

Polydioptric

Catadioptric Cameras

Dioptric Cameras (fisheye)



Nikon Coolpix
FC-E9 Lens
360º×183º

Canon EOS-1
Sigma Lens
360º×180º

Applications

Applications

- Daimler, Bosch: for car driving assistance systems

(Courtesy of Daimler)

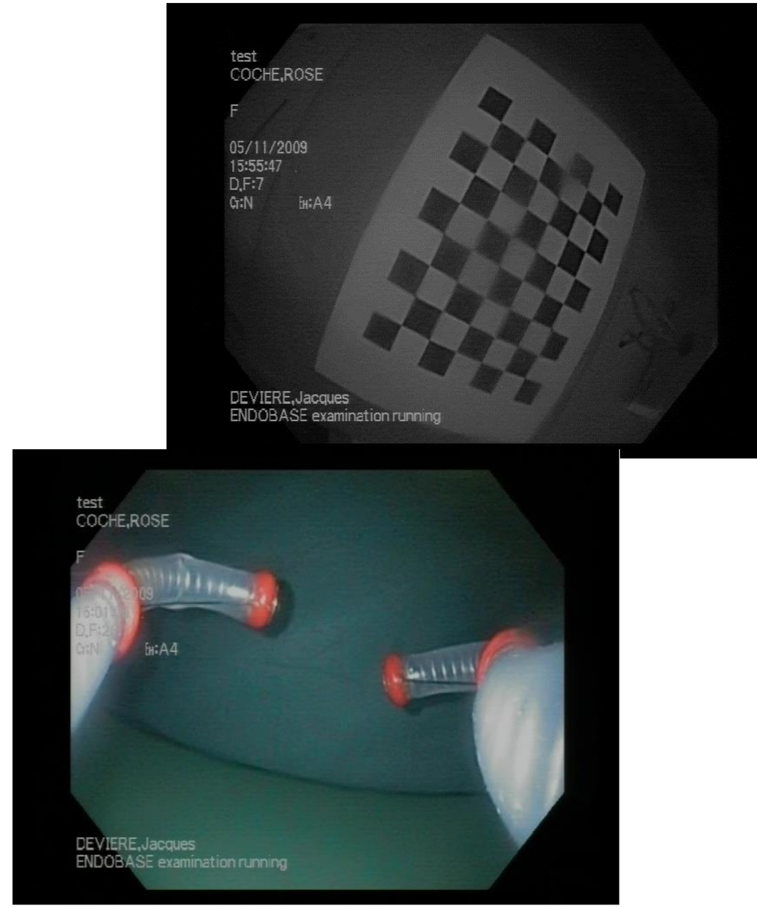
Applications

- Daimler, Bosch: for car driving assistance systems

- Meteorology: for sky observation

Applications

- Daimler, Bosch: for car driving assistance systems
- Meteorology: for sky observation
- Endoscopic Imagery: distortion removal (for the surgeon)



(Courtesy of Endo Tools Therapeutics, Brussels)

Applications

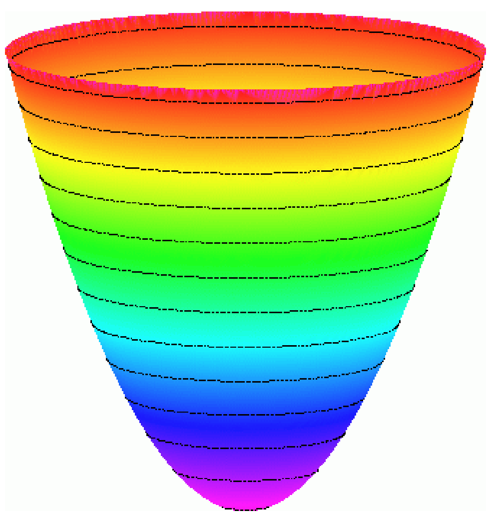
- Daimler, Bosch: for car driving assistance systems
- Meteorology: for sky observation
- Endoscopic Imagery: distortion removal (for the surgeon)
- RoboCup domain

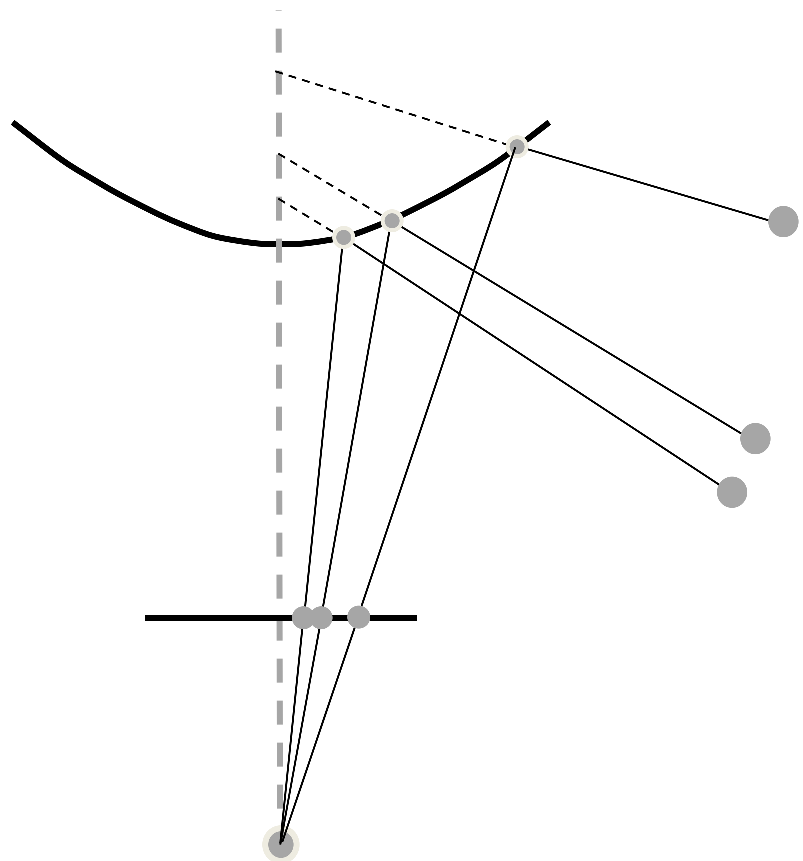
Applications

- Daimler, Bosch: for car driving assistance systems
- Meteorology: for sky observation
- Endoscopic Imagery: distortion removal (for the surgeon)
- RoboCup domain
- Google Street View

Catadioptric cameras

- mirror
- perspective camera

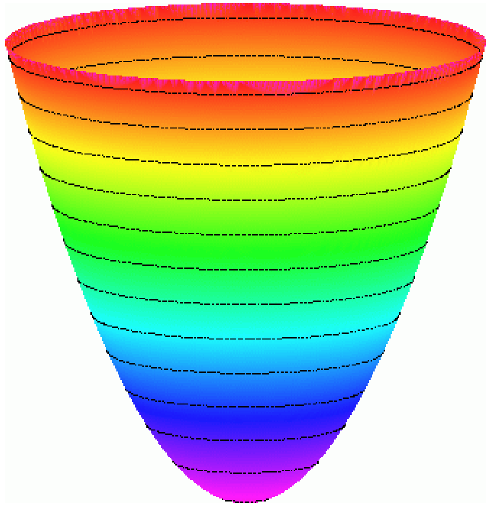


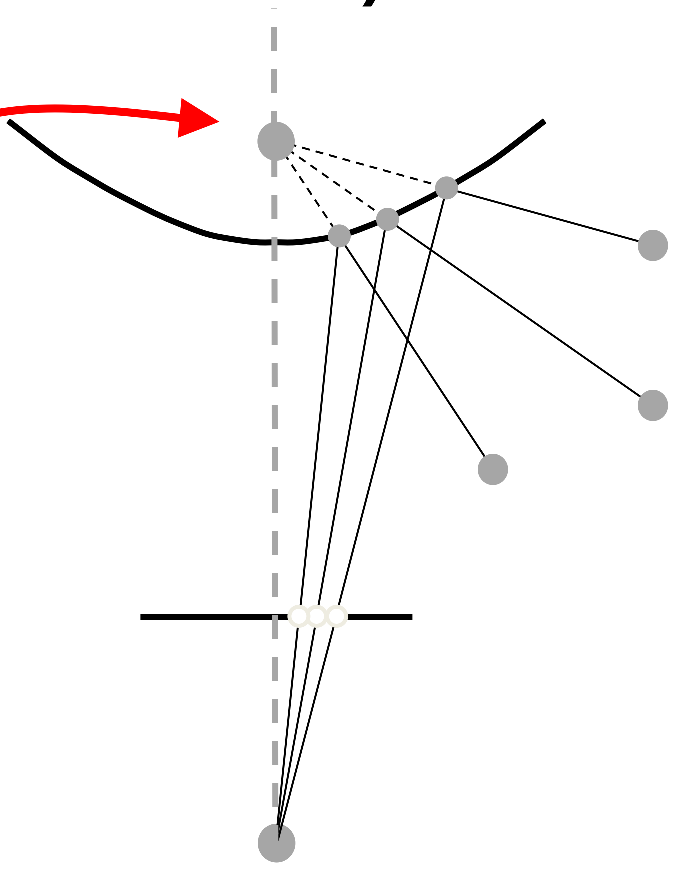


Catadioptric cameras

Central catadioptric cameras

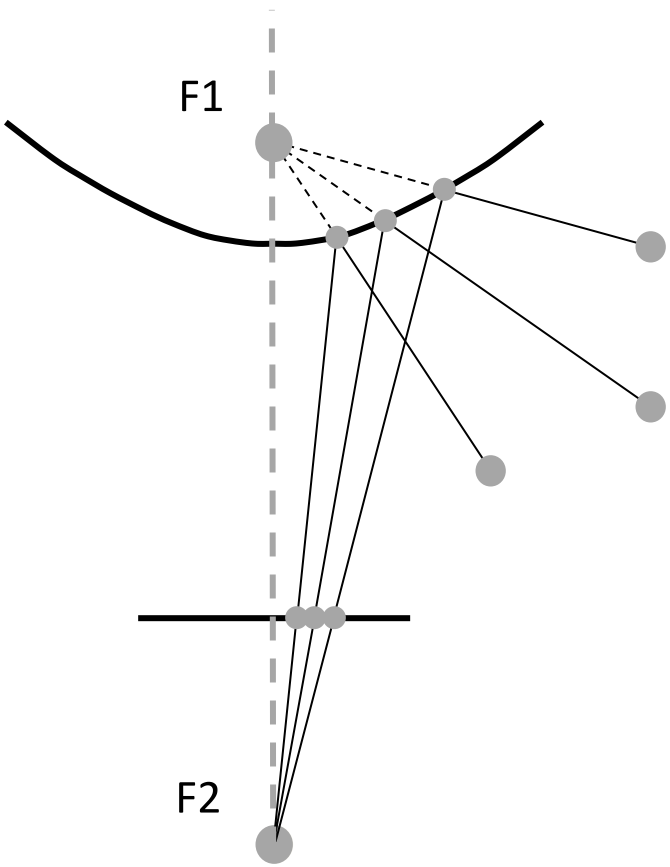
- mirror **(surface of revolution of a conic)**
- camera
- single effective viewpoint

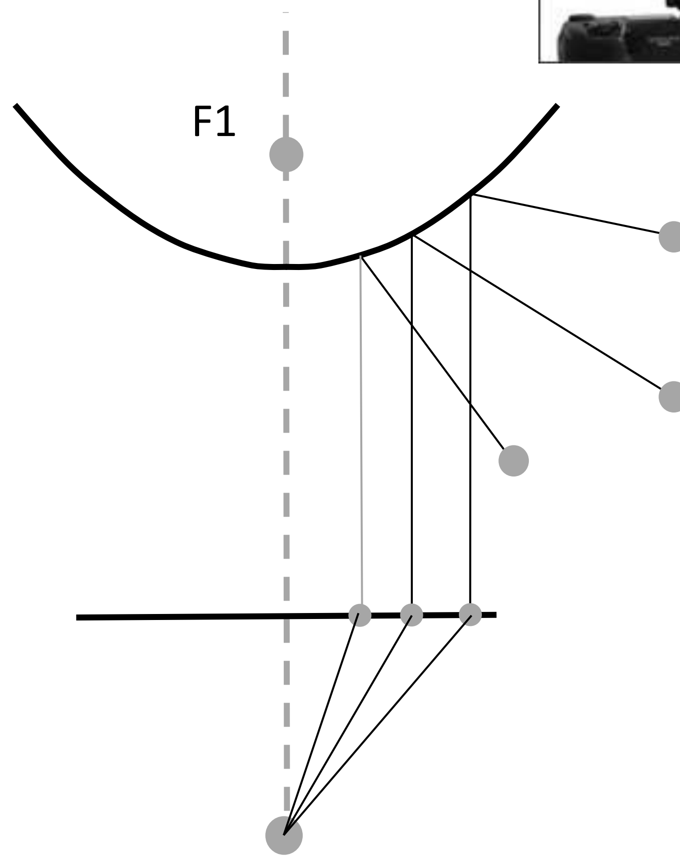




Catadioptric cameras

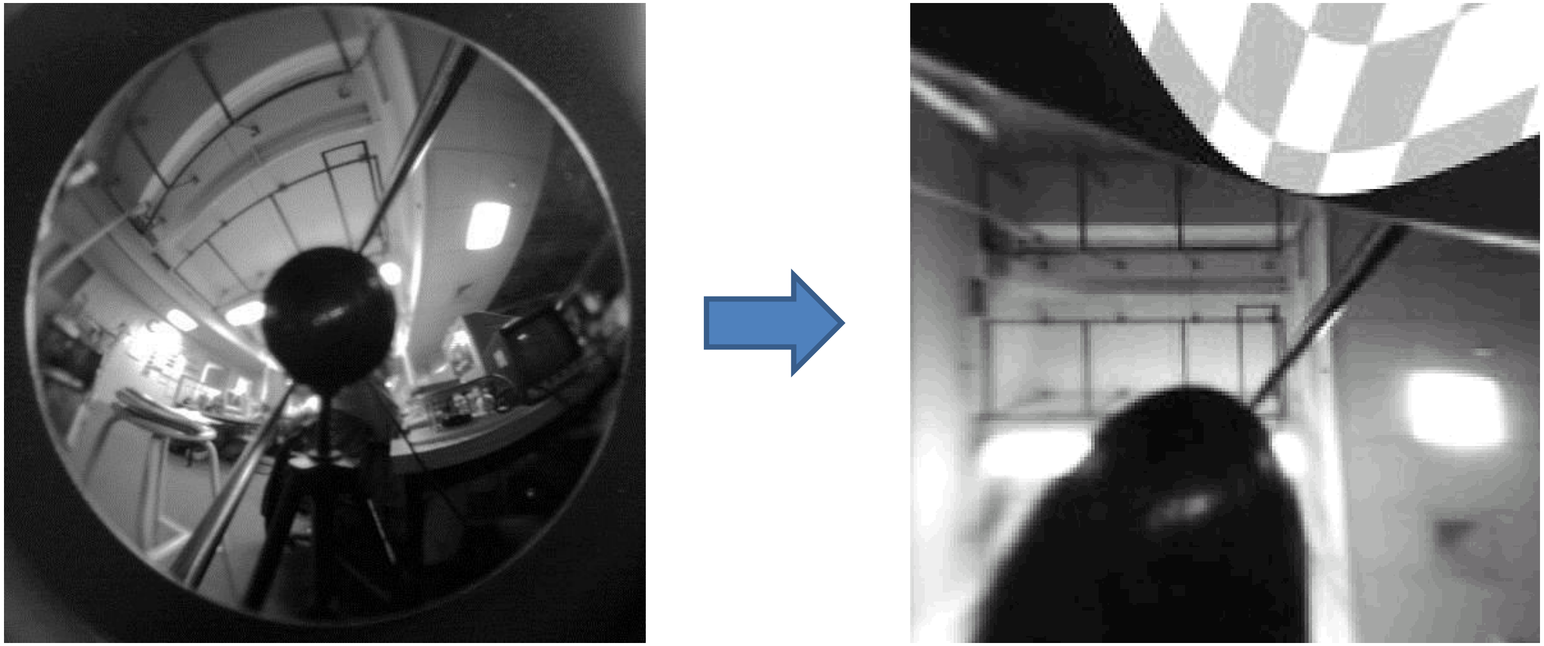
- hyperbola + perspective camera
- parabola + orthographic lens





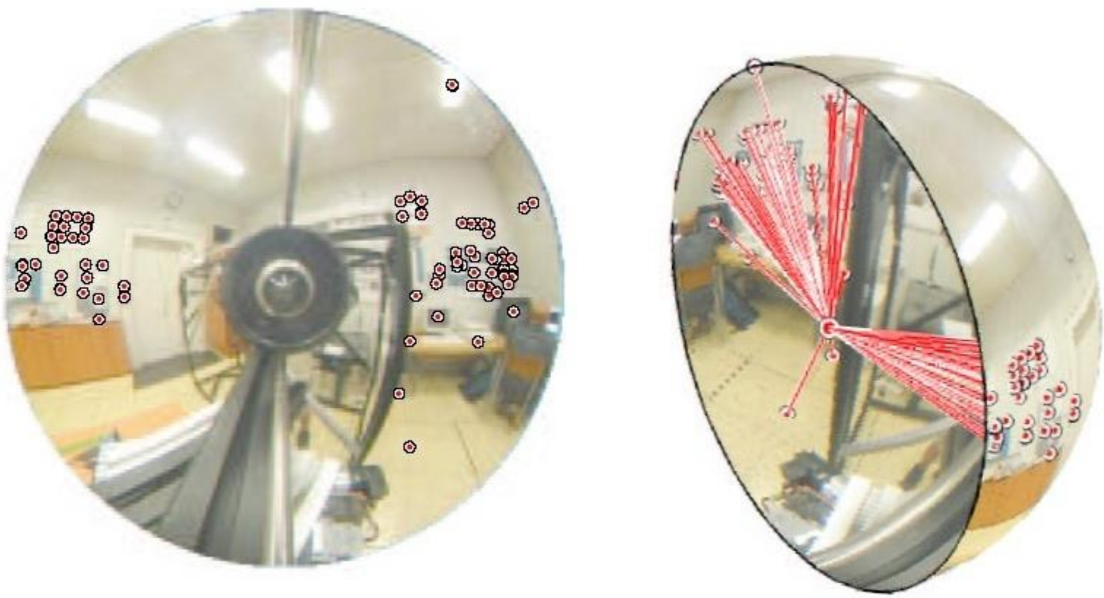
Why is it important that the camera be central (i.e., have a single effective viewpoint)?

- We can unwrap parts or all omnidirectional image into a perspective one



Why is it important that the camera be central (i.e., have a single effective viewpoint)?

- We can unwrap parts or all omnidirectional image into a perspective one
- We can transform image points normalized vectors in the unit sphere
- We can apply standard algorithms valid for perspective geometry.



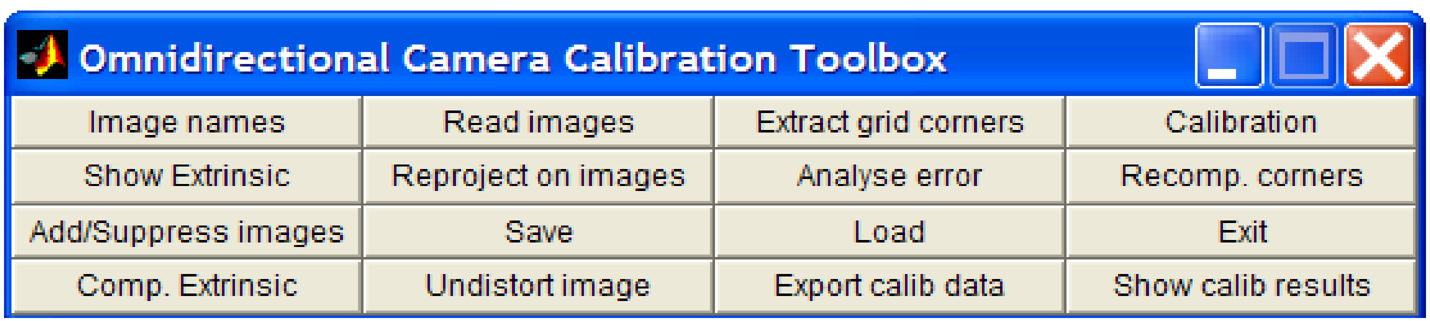
Points   Rays

Omnidirectional camera calibration toolbox for Matlab (Scaramuzza, 2006)

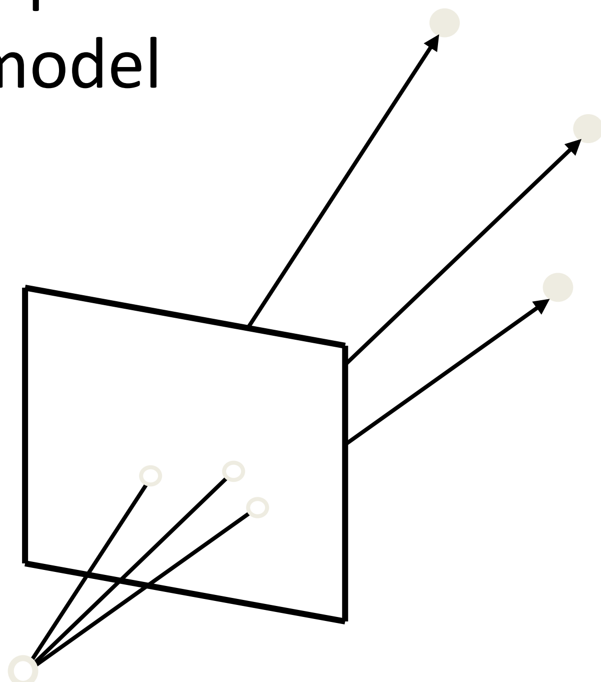
- World's standard toolbox for calibrating omnidirectional cameras (used at NASA, Daimler, IDS, Volkswagen, Audi, VW, Volvo, …)

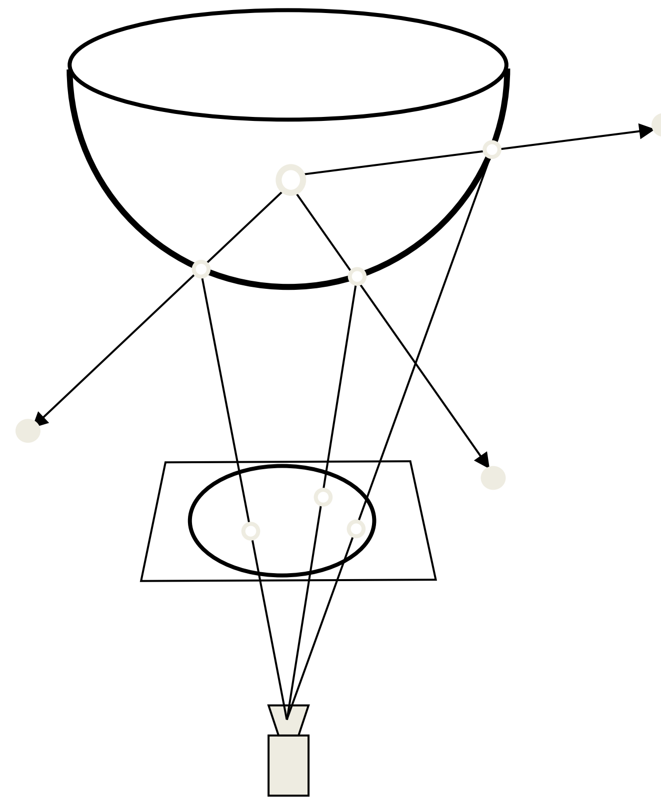
- Main applications are in robotics, endoscopy, video-surveillance, sky observation, automotive (Audi, VW, Volvo, …)

https://sites.google.com/site/scarabotix/ocamcalib-toolbox

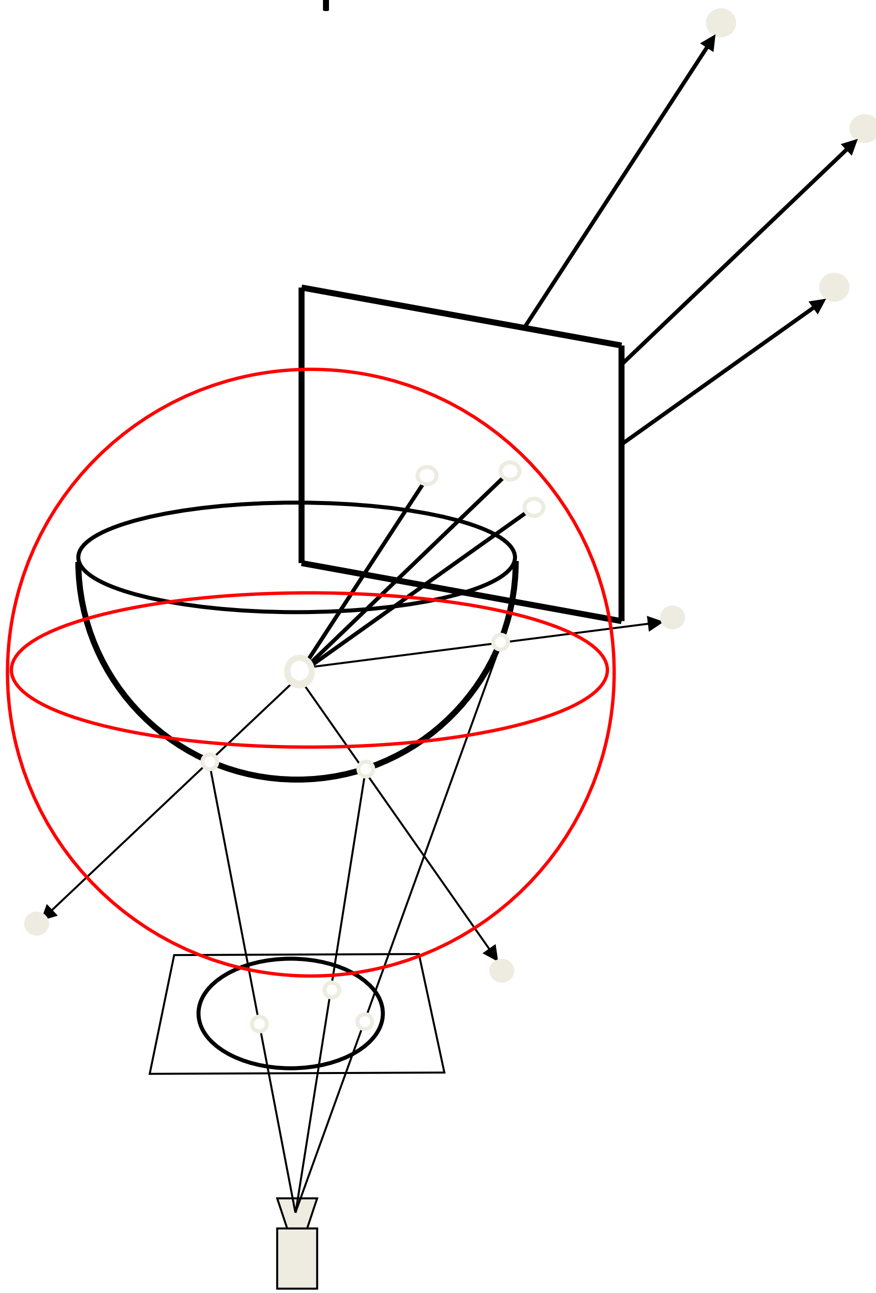


Equivalence between Perspective and Omnidirectional model

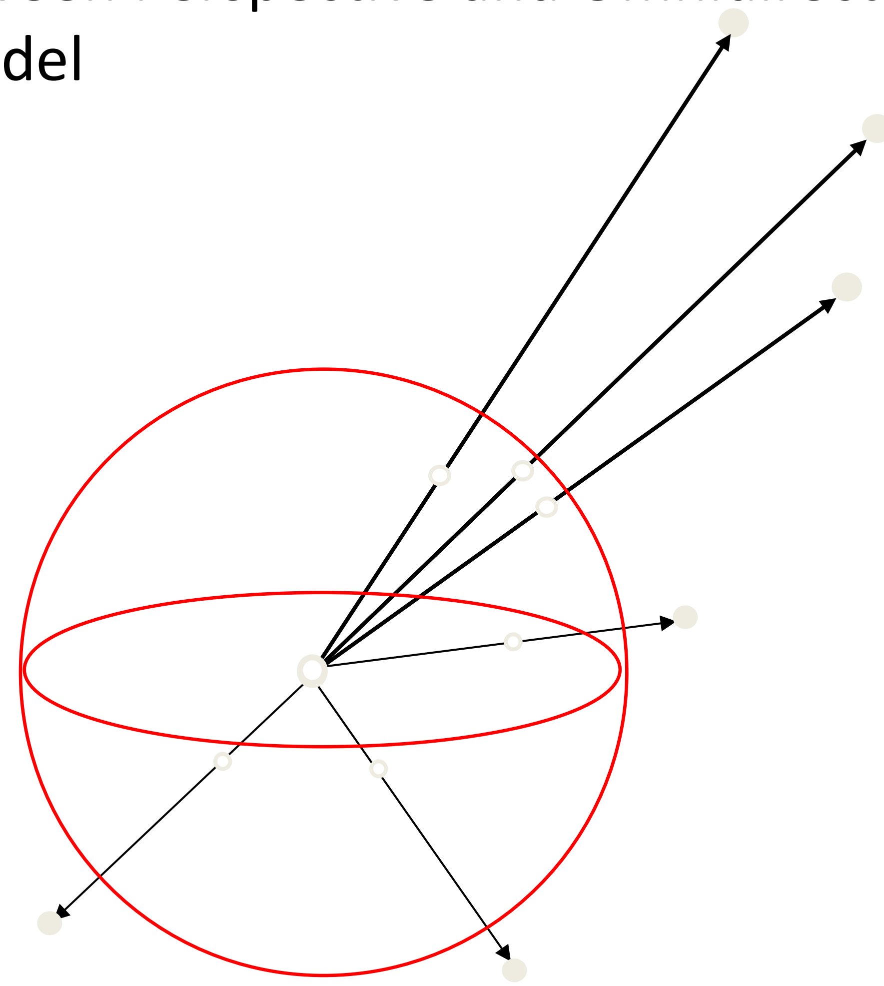




Equivalence between Perspective and Omnidirectional model

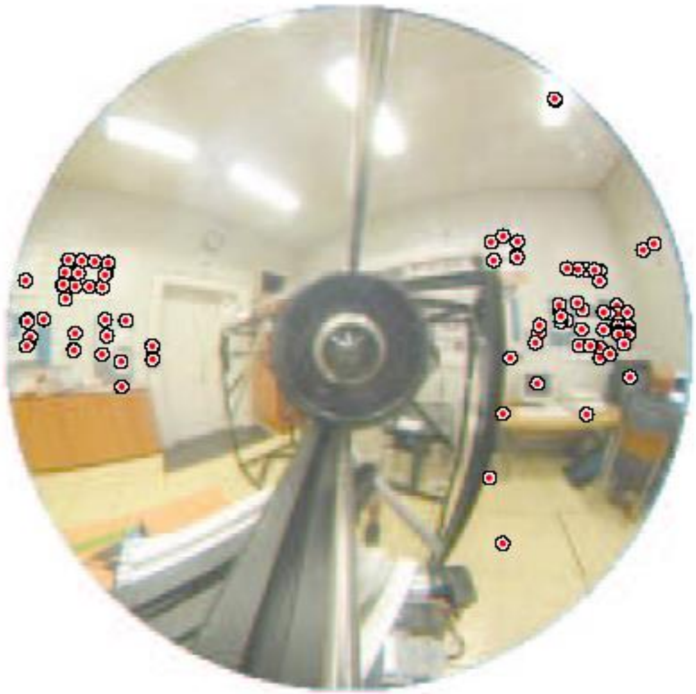


Equivalence between Perspective and Omnidirectional model: the Spherical Model

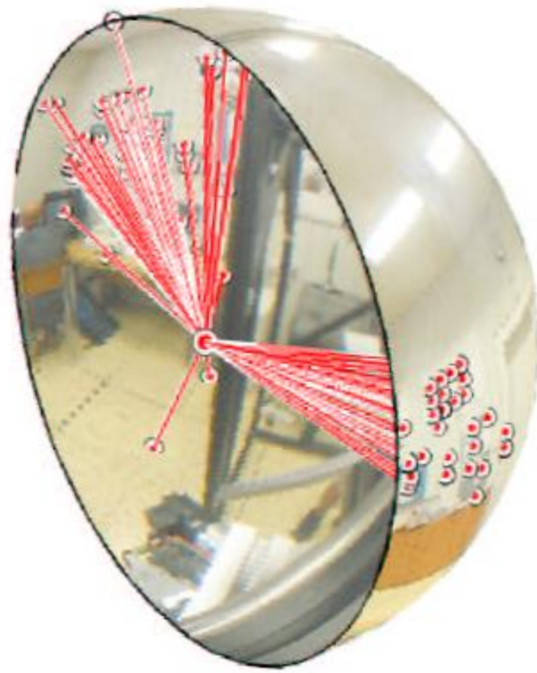


Representation of image points on the unit sphere

Always possible after the camera has been calibrated!



Points



Rays

Summary (things to remember)

- Calibration from planar grid (algorithm)
- Homography
- Orthographic projection
- Omnidirectional cameras
  - Central and non central projection
  - Dioptric
  - Catadioptric (working principle of conic mirrors)
  - Model principle
- Unified (spherical) model for perspective and omnidirectional cameras