Robot Vision

TTK4255

Lecture 01 – Introduction

Annette Stahl

(Annette.Stahl@ntnu.no)

Department of Engineering Cybernetics – ITK

NTNU, Trondheim

Spring Semester

06. January 2020

Welcome to TTK4255 - Robot Vision

Annette Stahl (Annette.Stahl@ntnu.no)

Simen Haugo (Simen.Haugo@ntnu.no)

Outline of the first lecture:

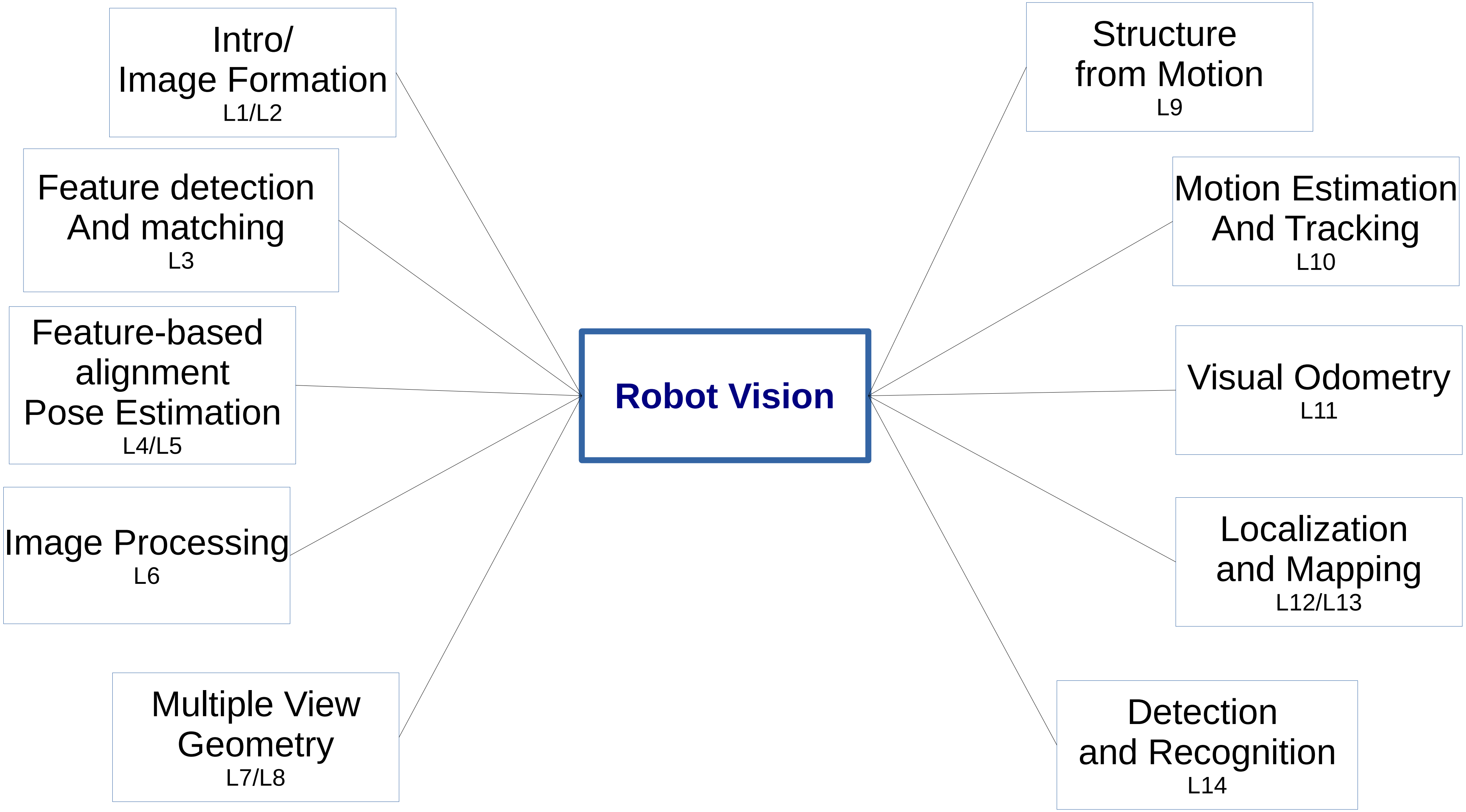
- Course Motivation
- Topics to be covered in TTK4255
- Challenges of Robot/Computer Vision

- Basic Course Administration

  15 min break

- Projective Projection
- Rigid Body Motion

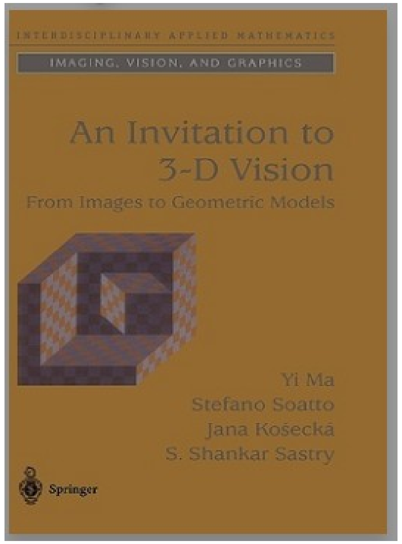
Robotic Vision – Course Content

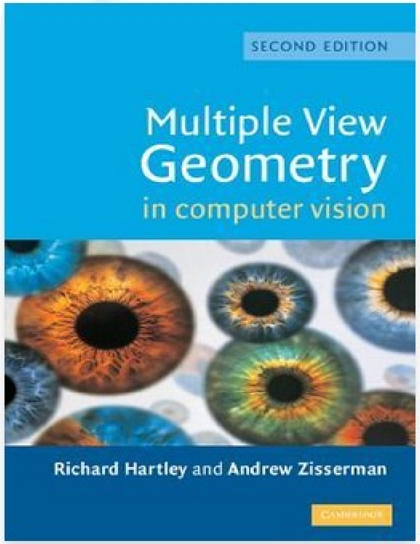


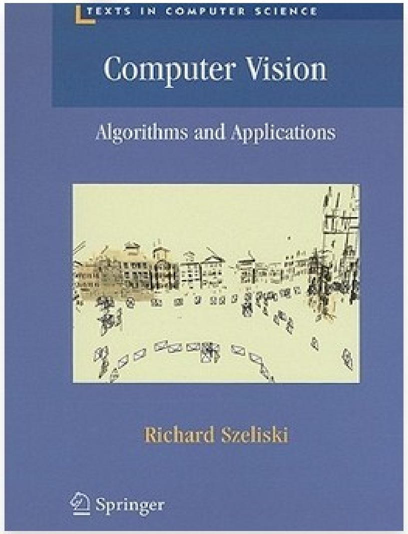
Robotic Vision – References

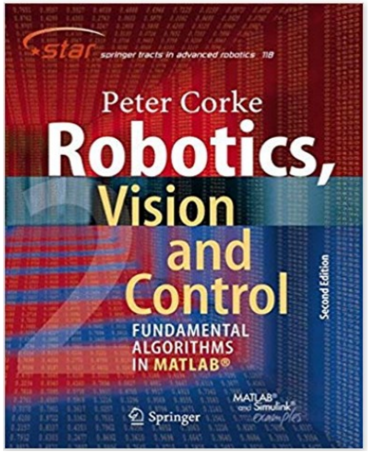
References will be provided during the course duration.

Main Resources:

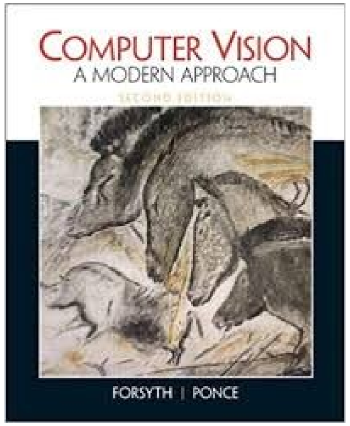


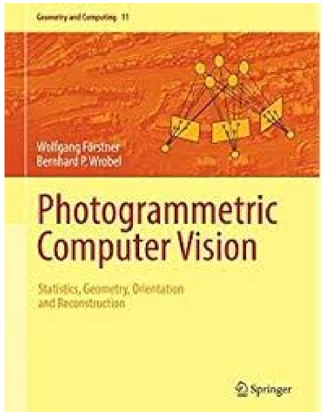


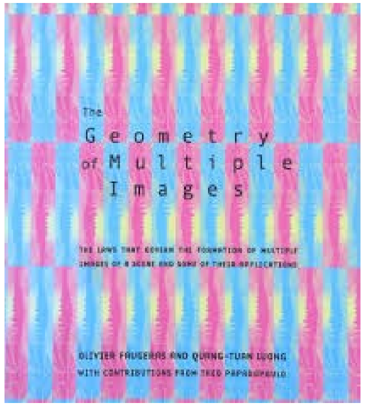


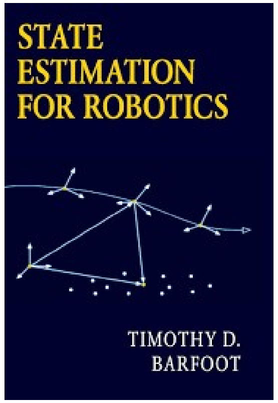


Additional Literature:









Robot Vision TTK4255 - Administration

Recommended previous knowledge

TMA 4245 Statistikk
TTK 4115 Linear System Theory
TTT 4275 Estimering, deteksjon of klassifisering
TTT 4120 Digitalsignalbehandling
TDT 4195 Grunnleggende visuell databehandling
TTK 4130 Modelling and Simulation

Relevant Courses

4. Semester	TMA4245 - Statistikk
5. Semester	TTK4115 - Linear System Theory TTT4120 - Digititalsignalbehandling TDT4195 - Grunnleggende visuell databehandling
6. Semester	TTK4145 - Sanntidsprogrammering TTT4275 - Estimering, deteksjon og klassifisering
7. Semester	TTK- SF - Sensorfusjon TTK4190 - Fartøystyring TTK4230 - Control Systems
8. Semester	TDT4265 - Datasyn og dyp læring TDT4171 - Artificial Intelligence Methods Big data cybernetics?
9. Semester	TDT4173 - Maskinlæring og case-basert resonnering TTK4101 - Instrumentering og måleteknikk TTK4235 - Tilpassede datasystemer TTK23 Introduction to Autonomous Robotics Systems for Industry 4.0 TTK19 Strukturer og sammenhenger i komplekse systemer TTK14 Kybernetiske metoder i fiskeri og havbruk TTK20 Hyperspectral remote sensing TTK21 Visual localization and Mapping

Robot Vision TTK4255 - Administration

Main Study Profiles

The subject is relevant for the following “Hovedprofiler”:

- Navigasjon og fartøystyring
- Robotsystemer
- Autonome systemer
- Medisinsk billeddannelse
- Fiskeri- og havbrukskybernettikk

Robot Vision TTK4255 - Administration

Teaching Activities and Evaluation

Lectures on **Monday 10:15 – 12:00** in **S8**

- Weekly Lectures
- Weekly Tutorials
- Project work
- Exercises (“fail/pass”) of exercises and project work
- Written Exam

- **No lectures: week (8), 15, 16, (17)**

→ Check Blackboard / Piazza for up-to-date information!!!

Exercise and Reference Group

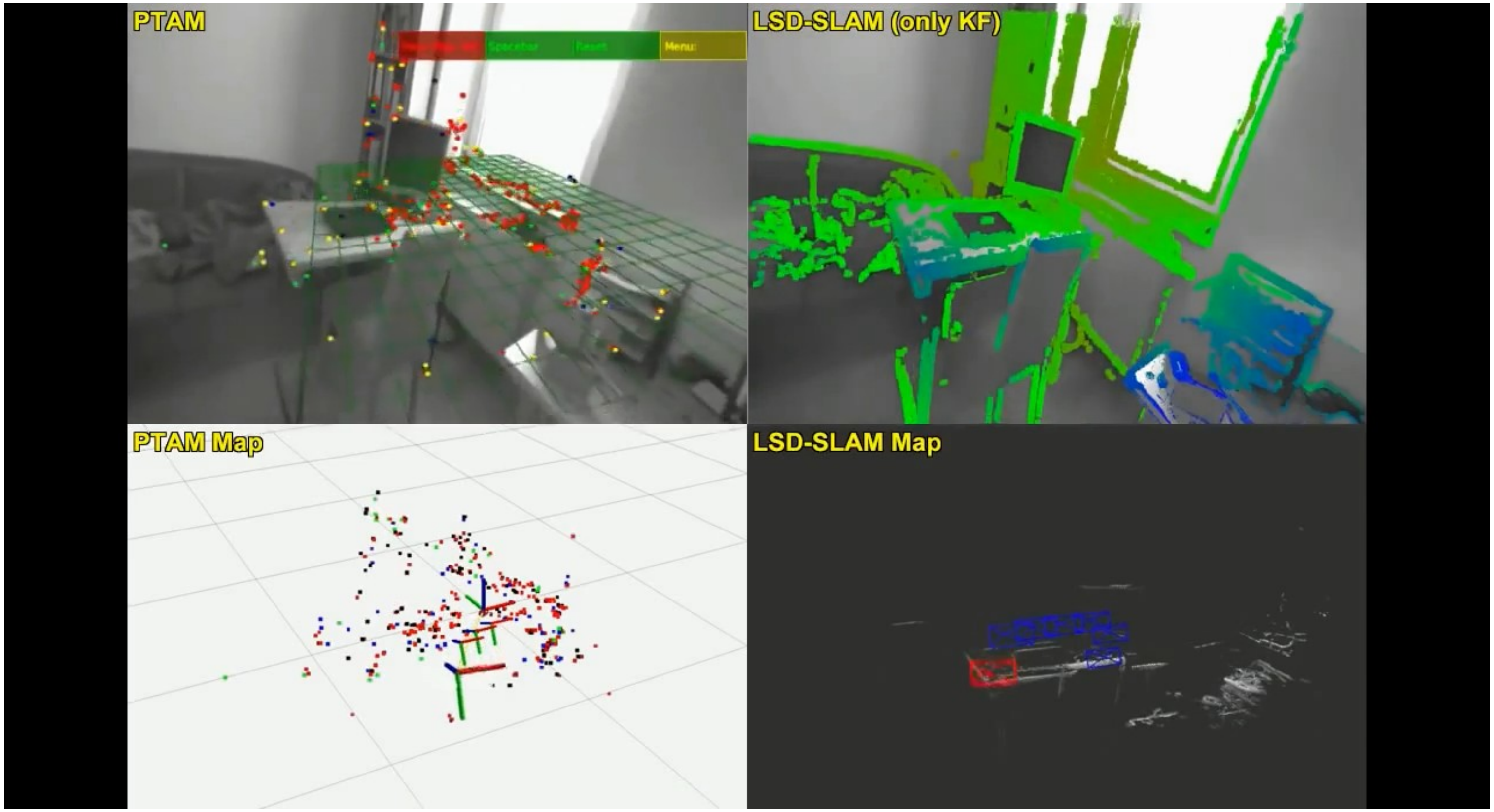
Simen

Break – 15 Minutes

Model Representation

An appropriate representation of the environment is important as this enables the computational analysis/interpretation of a scene "seen" by a robot/computer.

Visual Localization and Mapping approaches usually result in 3D point clouds of the scene helping a robot to navigate.



In addition, labeled objects that are recognized help "to understand" a scene.

➡ We need a good understanding what a camera "sees" of a 3D scene.

Open-Source Code Available: http://vision.in.tum.de/lsdslam
Publication: LSD-SLAM: Large-Scale Direct Monocular SLAM (J. Engel, T. Schöps, D. Cremers), ECCV 2014. Authors: Jakob Engel Thomas Schöps Daniel Cremers

3D Scenes recorded by "2D cameras"

We are familiar with the concept and measurements of Euclidean geometry

→ it is a good approximation to the properties of a general physical space

When we are considering the imaging process of a camera, the Euclidean geometry becomes insufficient

→ parallelism, lengths, and angles are no longer preserved in images

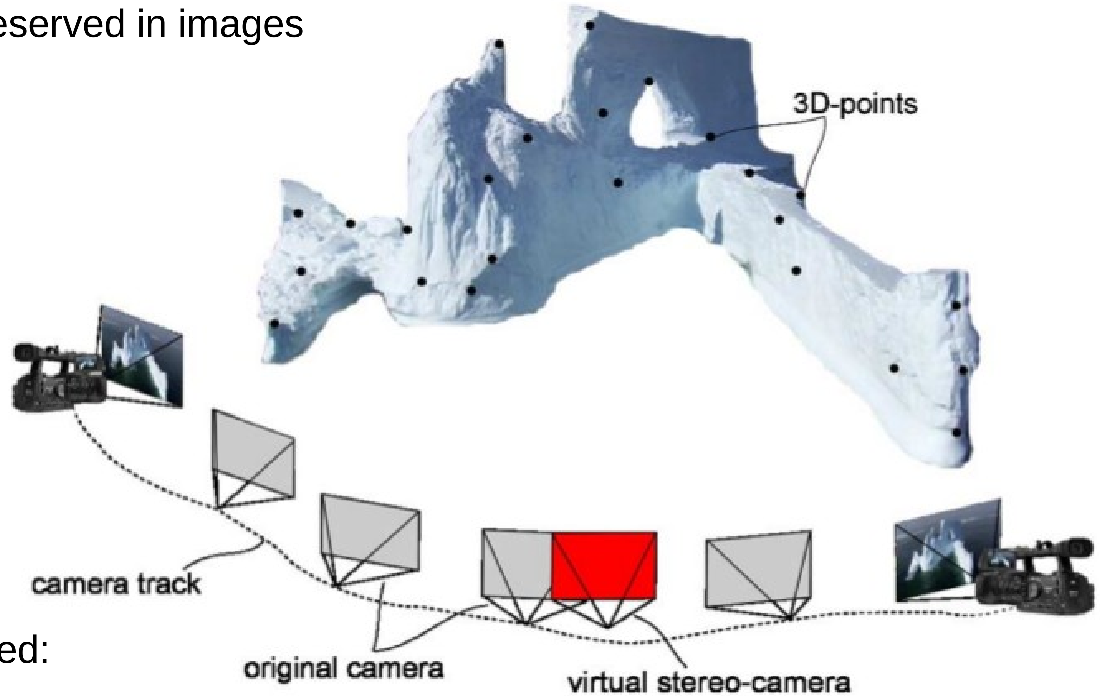


Image courtesy: Zhang, 3D TV Content Creation, 2011

Geometric relationship between

- a three-dimensional (3D) scene and
- its two-dimensional (2D) images

taken from a moving camera.

To understand and model this relationship,

two well studied **sets of transformations** are needed:

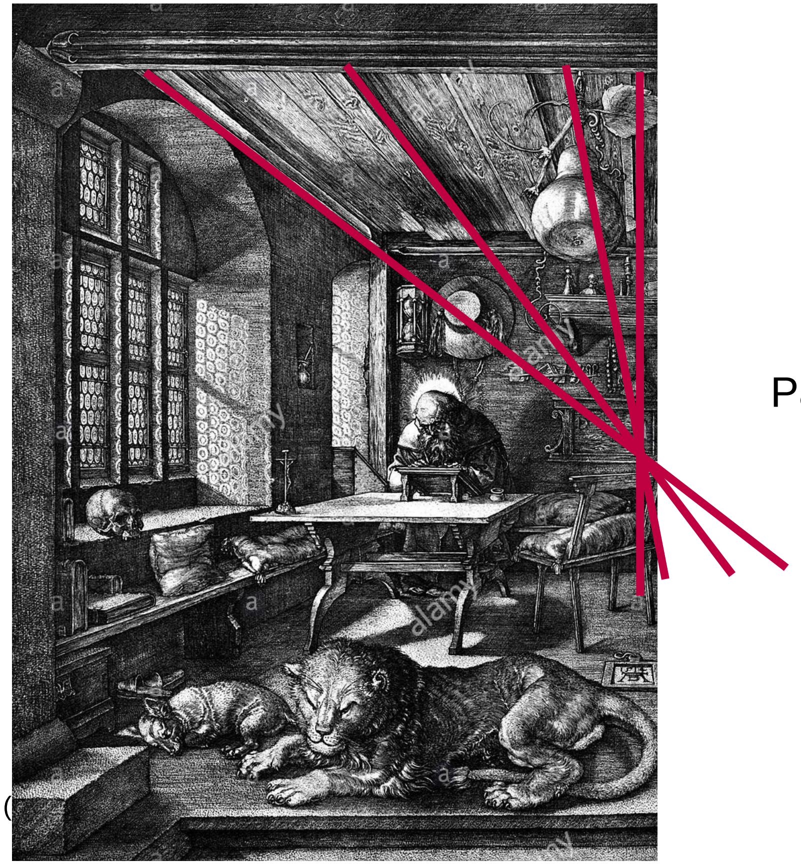
- **Perspective projection**: Describes the image formation process
  (projection of 3D scene to a 2D image)
- **Euclidean motion – rigid-body motion**: Models how the camera moves (translation, rotation)

Perspective Projection

Fundamental question:

How does the three-dimensional world appear in a two-dimensional image?

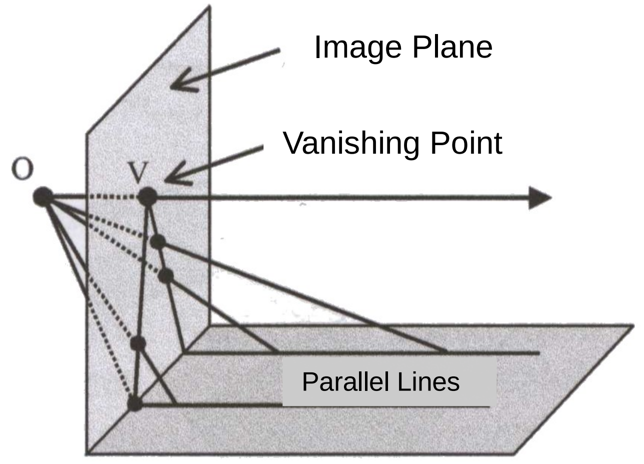
Early examples of the perspective projection date back to the Middle Ages, where painters (Masaccio 1427, Albrecht Dürer 1514) discovered basic principles of this projection.

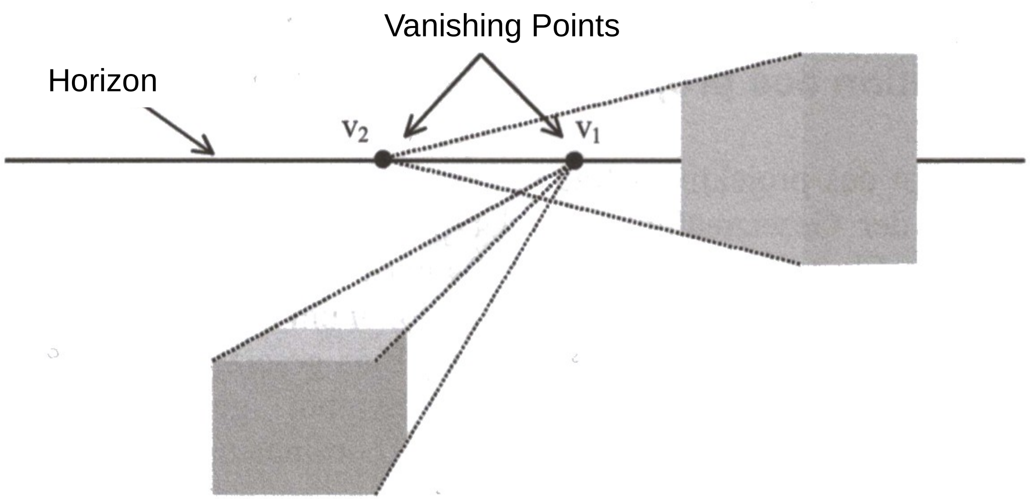


Parallel lines in the 3D world meet in the vanishing points.

Perspective Projection

Principles/Basics of the perspective projection





Courtesy: O. Schreer. Stereoanalyse und Bildsynthese. Springer, 2005

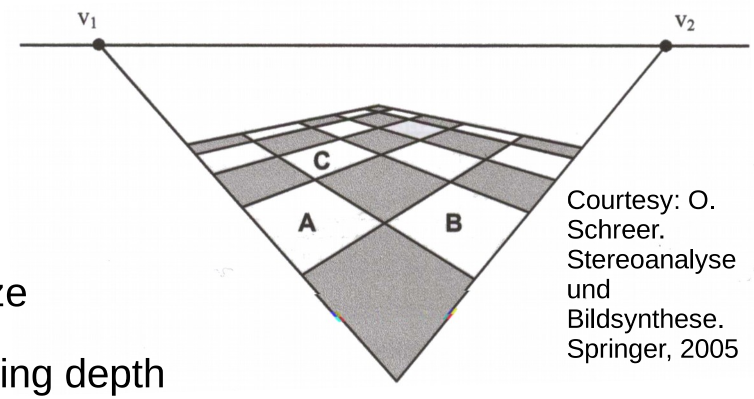
Center of projection: point *O* (optical center)

Projection: Parallel lines in 3D space projected onto the image plane, do converge in the image plane

Vanishing point V: intersection point of parallel lines in the scene in the image plane

Vanishing Line (Horizon) contains all vanishing points $V_1, V_2, \ldots$

Scale, angles, parallelism is not preserved: Objects of the same size in reality appear smaller in perspective with increasing depth



Courtesy: O. Schreer. Stereoanalyse und Bildsynthese. Springer, 2005

Projective Space

The projective space allows a mathematical description of the perspective projection.

Any **point** $x \in \mathbb{R}^n$ can be identified with a point in $\mathbb{R}^n$ with the coordinates

$$x = [x_1, x_2, ..., x_n]^\top \in \mathbb{R}^n$$

Note, will also use the notation $[x, y, z]^\top$ instead of $[x_1, x_2, x_3]^\top \in \mathbb{R}^3$

Homogeneous Coordinates

Homogeneous Coordinates are introduced to unify the mathematical framework.

$$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \in \mathbb{P}^n$$

2D example homogeneous coordinates $\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$

($\mathbb{P}^2$, 3-vector)

→ A point in projective space $\mathbb{P}^n$ of dimension n is described by a vector with $n+1$ components

→ a point (vector) in the projective space is defined up to scaling factor ($(n+1)th$ - component)

Meaning: - only direction not the length is relevant

- all vectors $\lambda x \ \forall \lambda \neq 0$ describe exact the same point in the projective space $\mathbb{P}^n$

Transforming (computing with) a homogeneous point may result in a vector with last component = 1

Projective Space

One can "recover" the corresponding vector from the homogeneous coordinates by the normalization with the last component

$$\tilde{x} = [x_1, x_2, \ldots, x_n, x_{n+1}]^\top \Rightarrow x = \left[\frac{x_1}{x_{n+1}}, \frac{x_2}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}\right]^\top$$

Example: $\begin{pmatrix} x \\ y \\ w \end{pmatrix} \rightarrow \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \end{pmatrix}$

One may observe that the scale factor x_{n+1} of the point can be zero indicating that the corresponding point is at infinity.

Definition of the Homography:

$$\tilde{x}_2^n = P^{n+1}\, \tilde{x}_1^n, \qquad P^{n+1} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n+1} \\ \vdots & \ddots & \vdots \\ p_{n+1,1} & \cdots & p_{n+1,n+1} \end{bmatrix} := H$$

A perspective projection can be described by a transformation of a point in projective space into another point in projective space with the matrix H. This transformation is known as projectivity or homography.

Relation of Sub-spaces

The linear Transformation of a point m_1 to m_2

$$m_2 = Hm_1$$

Can be expressed by the matrix multiplication with H.

Employing different constraints in H lead to different subspaces (2D example):

Projective Space $\supseteq$ Affine Space $\supseteq$ Euclidean Space

$$H = \begin{bmatrix} h & h & h \\ h & h & h \\ h & h & 1 \end{bmatrix} \qquad H = \begin{bmatrix} a & a & b \\ a & a & b \\ 0 & 0 & 1 \end{bmatrix} \qquad H = \begin{array}{c} \begin{matrix} \quad \boldsymbol{R} \quad & \quad \boldsymbol{T} \end{matrix} \\ \begin{bmatrix} r & r & b \\ r & r & b \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

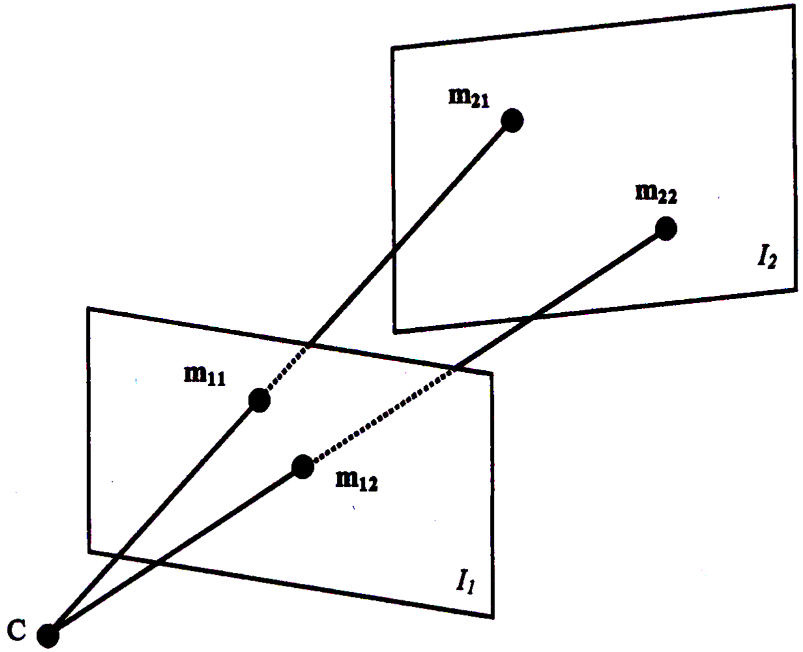
8 degrees of freedom 6 degrees of freedom 3 degrees of freedom

The product of two linear transformations H_2 and H_1 represents a linear transformation H_3

$$H_1 \cdot H_2 = H_3$$

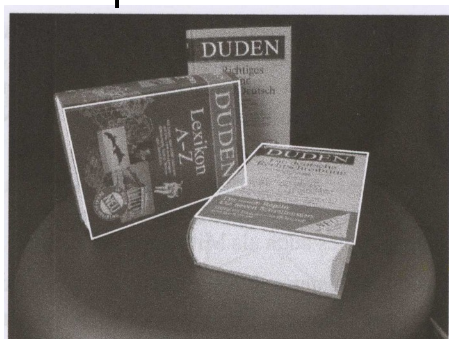
Homography in $\mathbb{P}^2$

Projective Transformation in $\mathbb{P}^2$



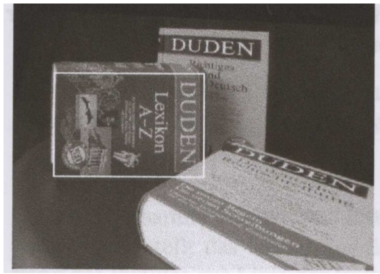
Courtesy: O. Schreer. Stereoanalyse und Bildsynthese. Springer, 2005

Example:



Original Image

Quadrilateral of
one book
is mapped to
a rectangle
(of pre-defined size)



First book

Second Book

Points of plain I_1 are projected onto plain I_2 by a projective transformation.

$$\tilde{m}_{2i} = H\tilde{m}_{1i}$$

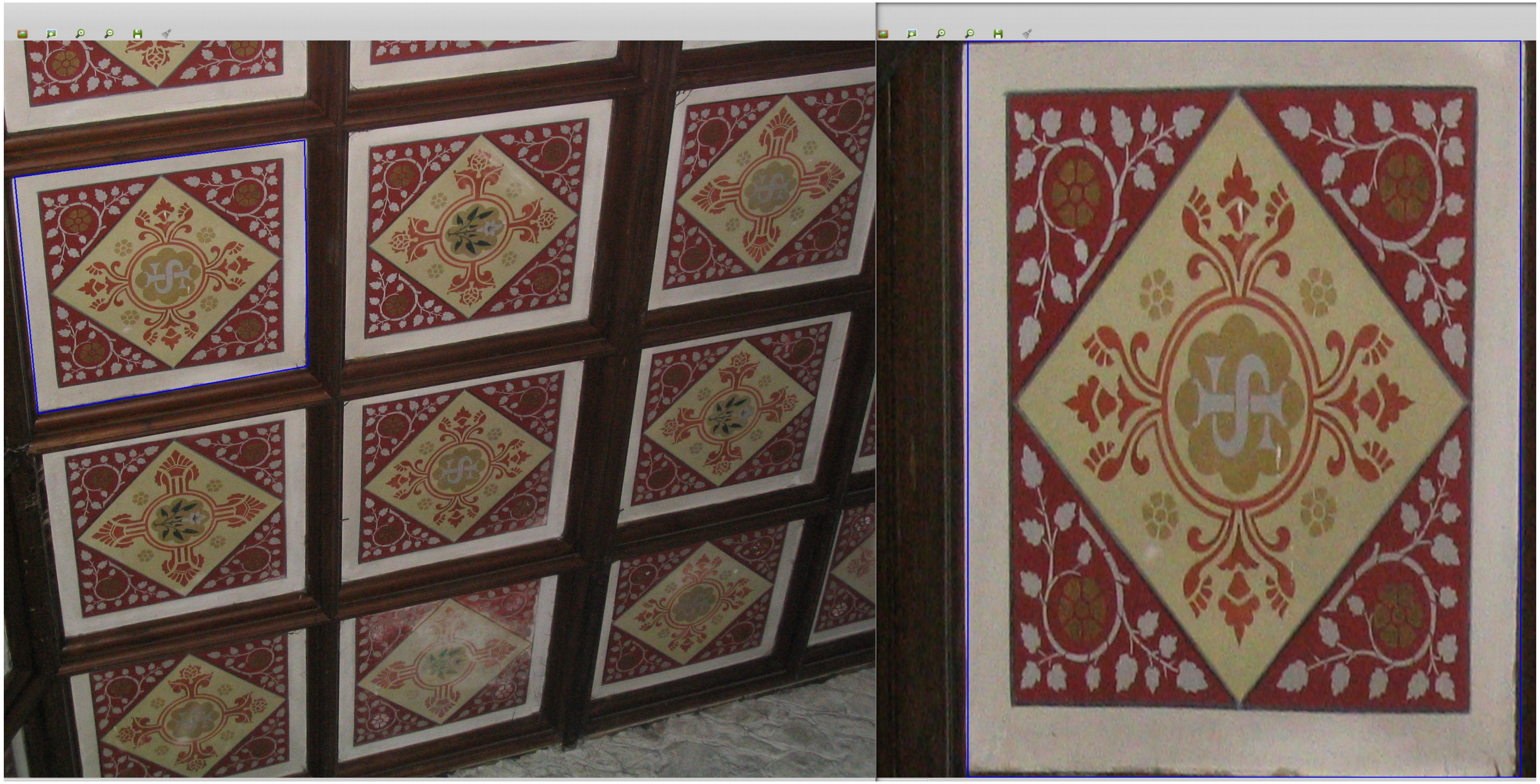
Homography describes the transformation of points of a plain onto another plain by applying a linear transformation.

The Homography matrix H must be non-singular/invertible.

Projective Transformation in $\mathbb{P}^3$: $\tilde{m}_2 = H \cdot \tilde{m}_1$, where H is a 4x4 matrix.

Homography

Demo



Affine Space

The affine space is a sub space of the projective space.

In “normal” coordinates $m_j \in \mathbb{R}^n$ the following transformation defines the affine space:

$$m_2^n = A^n m_1^n + b^n$$

Where A^n is a non-singular matrix. In homogeneous coordinates:

$$\tilde{m}_2^n = \begin{bmatrix} A^n & b^n \\ 0_n^\top & 1 \end{bmatrix} \tilde{m}_1^n$$

→ constrained transformation matrix

→ more invariants (compared to projective transformations):

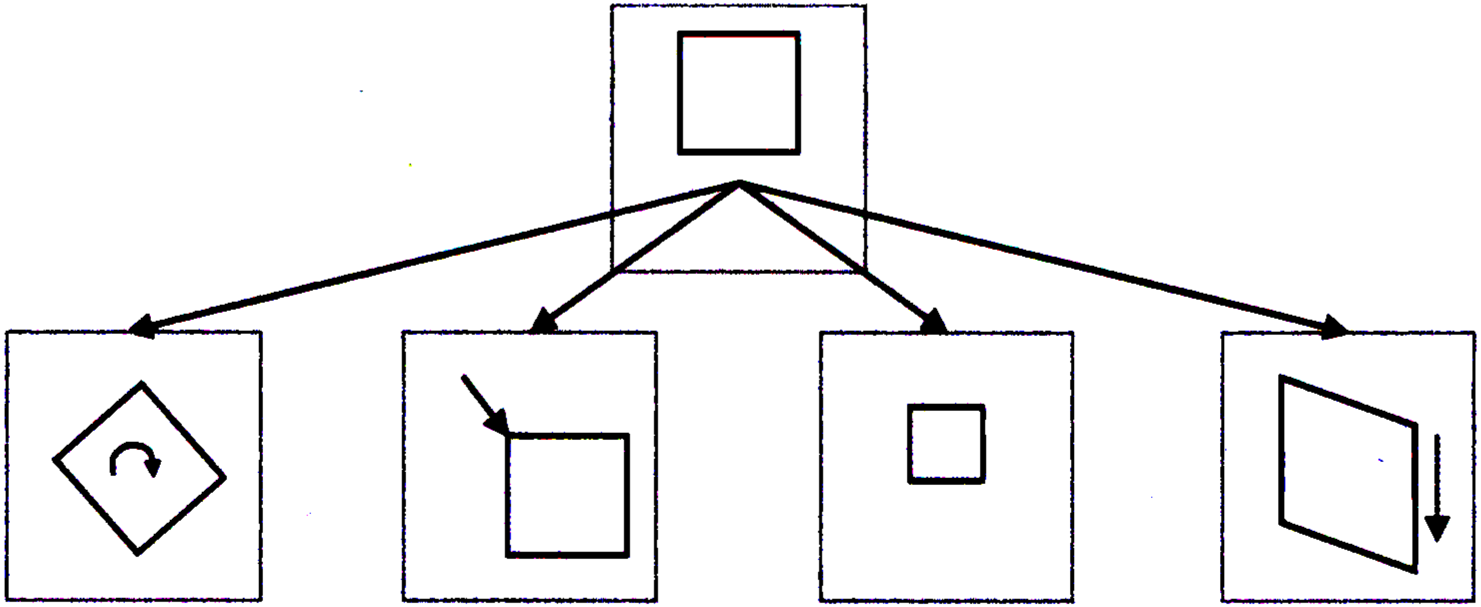
→ parallelism and length relationships are preserved

→ points in infinity transform to points in infinity

$$\begin{pmatrix} x_2 \\ y_2 \\ 0 \end{pmatrix} = \begin{bmatrix} A^n & b^n \\ 0_n^\top & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 0 \end{pmatrix}$$

Affine Space

Affine transformations include:



Rotation **Translation** **Scale** **Screw**

$$
a) \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}
\quad b) \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}
\quad c) \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}
\quad d) \begin{bmatrix} 1 & s_x & 0 \\ s_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

Courtesy: O. Schreer. Stereoanalyse und Bildsynthese. Springer, 2005

Intro Euclidean Geometry

Euclidean geometry is a subset of the projective geometry (is more general and least restrictive in the hierarchy of fundamental geometries)

Euclidean and projective geometry exist in any number of dimensions:

- a line in one-dimensional projective space, denoted as $\mathbb{P}^1$, corresponds to 1D Euclidean space
- the projective plane in $\mathbb{P}^2$ is analogous to 2D Euclidean plane
- the three-dimensional projective space $\mathbb{P}^3$ is related to 3D Euclidean space

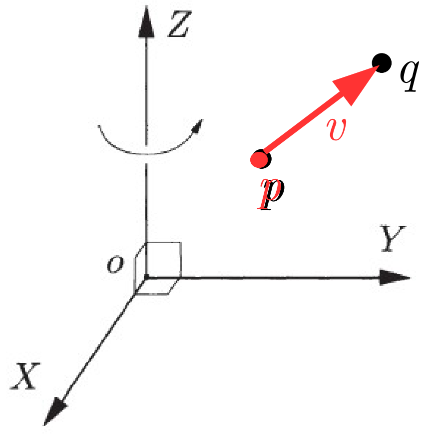
All projective transformations form a group which is called projective linear group.

Specializations or subgroups of the transformation: affine group, Euclidean group, oriented Euclidean group

Different transformations have different geometric invariances and properties:

- Length and area are invariant under Euclidean transformation
- parallelism and line at infinity are invariant under affine transformation
- general projective transformation preserves concurrency, collinearity, and cross ratio

Recall: Euclidean Space



A 3D Euclidean Space $\mathbb{E}^3$ is represented by a Cartesian coordinate frame.

Every **point** p can be identified with a point in $\mathbb{R}^3$

Vector $v = [v_1, v_2, v_3]^\top \in \mathbb{R}^3$ is determined by a pair of points $p, q \in \mathbb{E}^3$ and represented by an arrow connecting both points.

Norm of a vector v is

$$\| v \| = \sqrt{\langle v, v \rangle} = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

(Note: A point can also be seen as vector from the origin to the point)

In order to be able to measure distances, angles, length of curves, or volumes one needs an appropriate metric!

Euclidean metric for $\mathbb{E}^3$ is defined by an **inner product** (**dot product**) → **Euclidean metric**

$$\langle u, v \rangle, \quad \forall u, v \in \mathbb{R}^3$$

Two vectors are **orthogonal**, if

$$\langle u, v \rangle = 0$$

Euclidean space $\mathbb{E}^3$ can be describes as a space that can be identified with $\mathbb{R}^3$ and has a metric (on its vector space) given by the inner product.

Recall: Euclidean Space

Given two vectors, their **cross product** results in a third vector

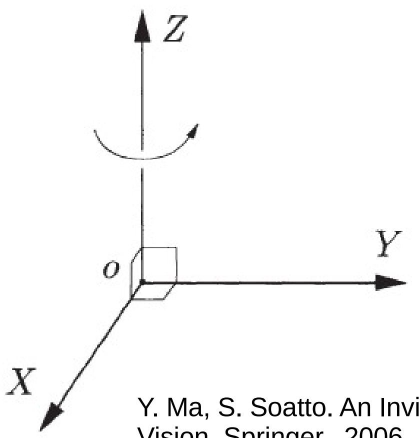
$$u \times v = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} \in \mathbb{R}^3$$

The cross product of two vectors
- is linear in each of its arguments
- is orthogonal to each of its factors

The order of the factors defines an **orientation**
The cross product conforms the **right-hand rule**
The cross product defines a **map** between

-a vector u and

-a 3x3 skew-symmetric matrix



Y. Ma, S. Soatto. An Invitation to 3D Vision, Springer, 2006

The Euclidean space is a sub space of the projective space.

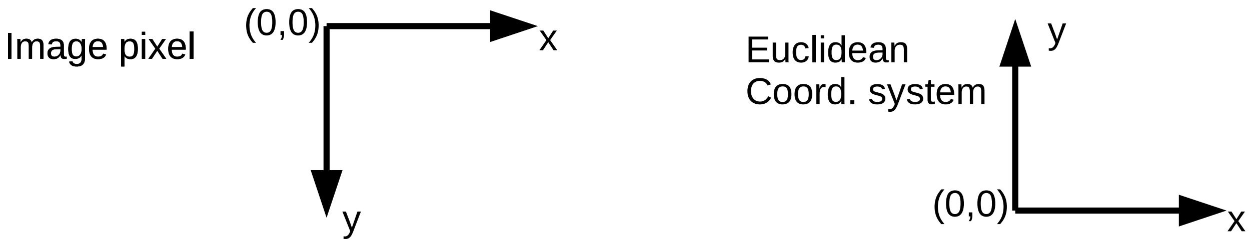
Note: The Euclidean space has more constraints than the affine space.

→ more invariants: Length and angles of geometric objects are not changed after an Euclidean transformation

Recall: Euclidean Space

Practical hint:

(Note: Different coordinate systems definitions and conventions are often a source of confusion!)



→ If an implementation turns out to do something “unexpected” - check the coordinate systems
Inverted

+ access of matrices (row, column) $\leftrightarrow$ (y, x)

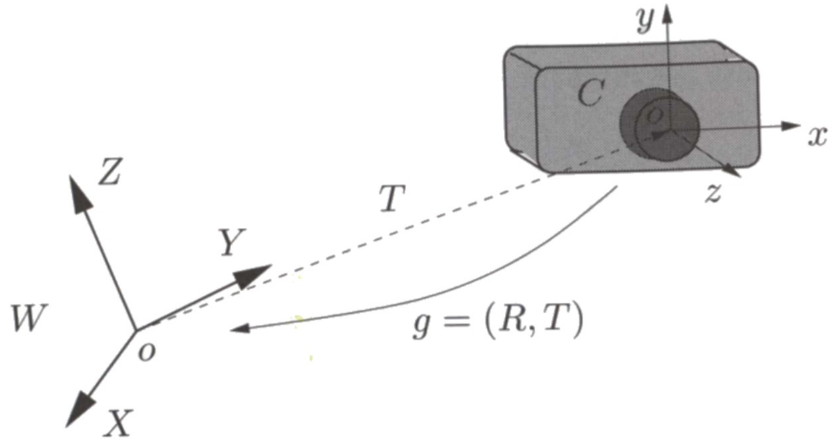
Euclidean Motion - Rigid Body Motion

A map $g : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a **rigid-body motion** or a **special Euclidean transformation** if it preserves the norm and the cross product of any two vectors. The Euclidean space is a sub space of the projective space. (Note: rigid body - the distance between any two points on the object does not change over time as the body (object) moves.)

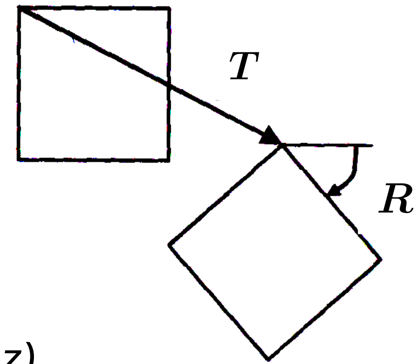
That means

- distances (norm of vectors) and
- orientations (cross product of vectors) and
- angles between vectors (inner product(norm))

are preserved!



When the camera moves the camera frame also moves along with the camera. The configuration of the camera is then determined by two components:



1. **Translational part T:** the vector between the origin of the world frame and that of the camera frame

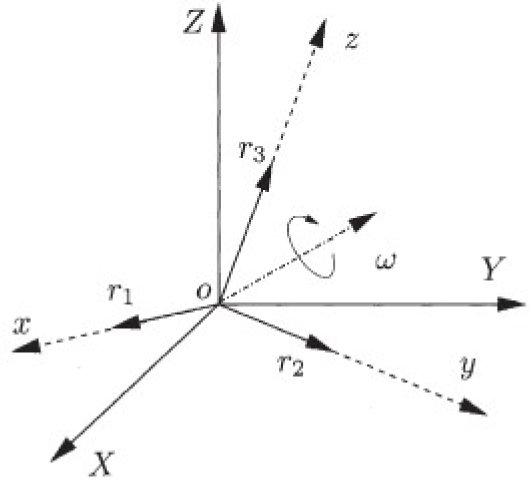
2. **Rotational part R:** the relative orientation of the camera frame *c* with coordinate axes *(x,y,z)* relative to the fixed world frame w with coordinate axes *(x,y,z)*

The collection of all such motions or transformations is denoted by $SE(3)$

Rotational Motion Representation

A rigid object (camera) is rotating about a fixed point *o*

How to describe its orientation relative to the world coordinate frame *w*?



Y. Ma, S. Soatto. An Invitation to 3D Vision, Springer, 2006

Motivation: The orientation of a camera in space is important
→ We need a good understanding if we wish to rotate around a certain point *w*.

Define center of rotation *o* as the center of *w*

Attach another coordinate frame *c* to the rotating object (camera) with its origin at *o*

The orientation of the frame *c* relative to the frame w are define by three orthonormal vectors along the three principal axes *x, y, z* of the frame *c*
The orientation of the camera is then determine by the 3x3 matrix

$$R_{wc} = [r_1, r_2, r_3] \in \mathbb{R}^{3\times 3}$$

Since this is an orthogonal matrix it follows that

$$R_{wc}^\top R_{wc} = R_{wc} R_{wc}^\top = I; \quad R_{wc}^{-1} = R_{wc}^\top; \quad \det R_{wc} = 1$$

This matrix is orientation preserving → special orthogonal matrix = rotation matrix

Rotational Motion Representation

Recall: The Space of all such special orthogonal matrices is denoted by

$$SO(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = I, \det R = +1\}$$

Special orthogonal group of $\mathbb{R}^3$ or **rotation group**
(rotations preserve both inner and cross product of vectors).

The configuration (orientation) of a continuously rotating object can be then described as a trajectory over time

$$R(t) : t \longmapsto SO(3)$$

The matrix R_{wc} represents the coord. Transformation from frame c to frame w

$$x_c(t) = R_{cw}(t) x_w$$

Since an inverse transformation of a rotiation is also a rotation it yields that

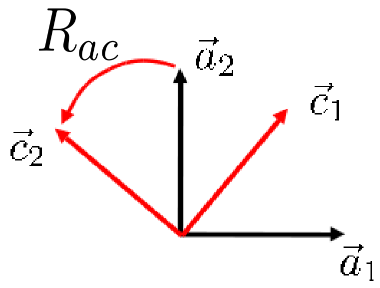
$$R_{cw} = R_{wc}^{-1} = R_{wc}^\top$$

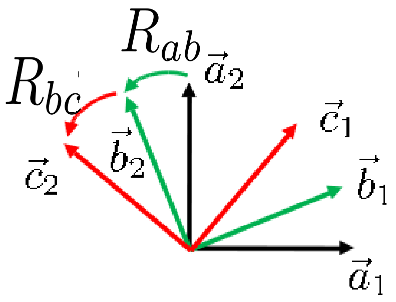
Invertibility: We can move a rigid object for example a camera from one place to another and put it back to its origin position – reverting the action

Rotational Motion Representation

$$x_c(t) = R_{cw}(t)x_w$$

Composition: We can combine several of such motions to generate a new one:





$$x_a = R_{ac}x_c$$

$$x_b = R_{bc}x_c$$

$$x_a = R_{ab}x_b = R_{ab}R_{bc}x_c$$

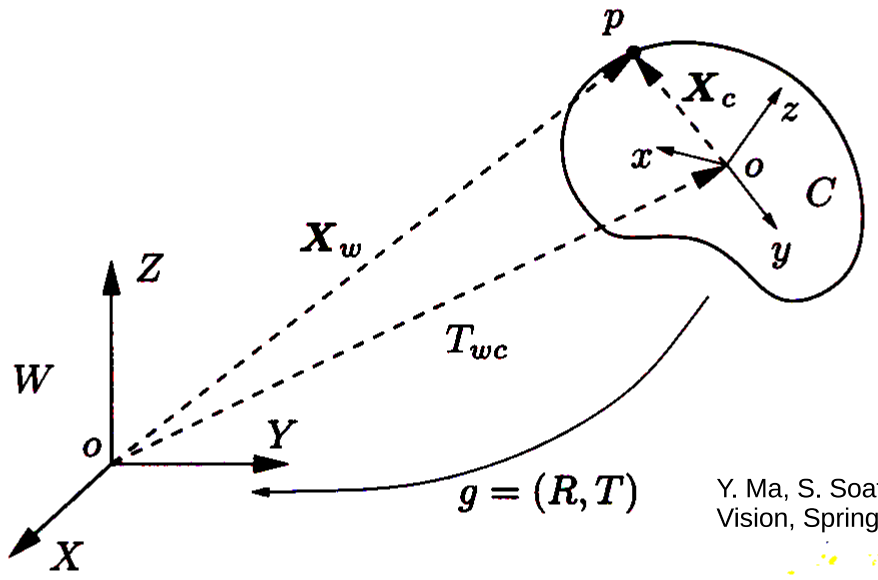
$$R_{ac} = R_{ab}R_{bc}$$

$$R_{ad} = R_{ab}R_{bc}R_{cd}, \quad \text{etc.}$$

→ Property of invertibiltiy and composition - notion of a group

Recap from TTK4130 - Modellering og simulering

Rigid Body Motion



Y. Ma, S. Soatto. An Invitation to 3D Vision, Springer, 2006

The world coordinates of a rigid body motion between a moving frame *c* and a world frame *w*

$$x_w = R_{wc} x_c + T_{wc} \qquad (m_2 = R m_1 + T)$$

The rigid-body motion is usually described by $g = (R, T)$. Then g represents a transformation of coordinates between two frames.

The set of all possible configurations of a rigid body can be described by the space of **Rigid body motions** or **special Euclidean transformations**

$$SE(3) = \{g = (R, T) \mid R \in SO(3), T \in \mathbb{R}^3\}$$

Demo: Camera Motion

https://www.youtube.com/watch?v=ufvPS5wJAx0

Rigid Body Motion – Homogeneous Rep.

A rigid-body motion in a linear form

$$\tilde{x}_w = \begin{bmatrix} x_w \\ 1 \end{bmatrix} = \begin{bmatrix} R_{wc} & T_{wc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ 1 \end{bmatrix} = \tilde{g}_{wc}\tilde{x}_c$$

Where $\tilde{g}_{wc} \in \mathbb{R}^{4x4}$ is the homogeneous representation of the rigid-body motion $g_{wc} = (R_{wc}T_{wc}) \in SE(3)$

The homogeneous representation of g leads to a matrix representation of the $SE(3)$

$$SE(3) = \left\{ \tilde{g} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \mid R \in SO(3), T \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4x4}$$

.

Coordinate Transformations

- Euclidean Transformation is a coordinate transformation from one orthogonal coordinate system into another orthogonal coordinate system.
- Note: We often exploit the Euclidean Space (transformations to describe coordinate systems, transformations and positions of cameras).

Coordinate transformations

The time $t \in \mathbb{R}$ denotes the index of the camera position and the corresponding image.

Relative displacement between some fixed world frame w and the camera frame c at time t

$$g = \begin{bmatrix} R(t) & T(t) \\ 0 & 1 \end{bmatrix} \in SE(3) \qquad g(t) = (R(t), T(t)) \in SE(3)$$

With $g(0) = I$, where the camera frame coincides with the world frame

Coordinates of a point $P \in \mathbb{E}^3$ relative to the world frame are $x_0 = x(0)$ its coordinates

relative to the camera at time t are given by

$$x(t) = R(t)x_0 + T(t) \qquad x(t) = g(t)x_0$$

Coordinate Transformations

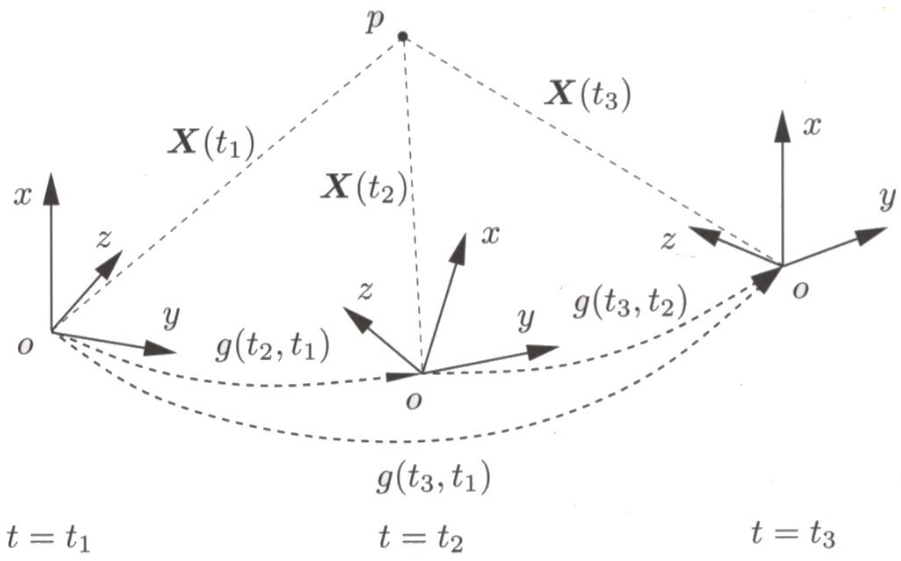
If the camera is at locations $g(t_1), g(t_2), \ldots g(t_m)$ at times $t_1, t_2, \ldots t_m$ respectively, then the coordinates of the point p are given as $x(t_i) = g(t_i)x_0, i = 1, 2, \ldots, m$

If the starting time not $t=0$, the relative motion between the camera at time t_1 and time t_2 will be denoted by $g(t_2, t_1) \in SE(3)$ and the relation ship between the coordinates of the same point p at different times

$$x(t_2) = g(t_2, t_1)x(t_1)$$

If we consider a third position of the camera at $t = t_3$ we have the following relationship

$$x(t_3) = g(t_3, t_2)x(t_2) = g(t_3, t_2)g(t_2, t_1)x(t_1)$$



Y. Ma, S. Soatto. An Invitation to 3D Vision, Springer, 2006

Coordinate Transformations

Compare this with the direct relation ship between t_3 and t_1

$$x(t_3) = g(t_3, t_1)x(t_1)$$

We see that the composition rule for consecutive motions must hold

$$g(t_3, t_1) = g(t_3, t_2)g(t_2, t_1)$$

It describes the coordinates *x* of the point p relative to any camera position if they are known with respect to a particular one.

The same implies for the rule of inverse

$$g^{-1}(t_2, t_1) = g(t_1, t_2)$$

	Rotation $SO(3)$	Rigid-body motion $SE(3)$
Matrix representation	$R: \begin{cases} R^T R = I \\ \det(R) = 1 \end{cases}$	$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$
Coordinates (3-D)	$\boldsymbol{X} = R\boldsymbol{X}_0$	$\boldsymbol{X} = R\boldsymbol{X}_0 + T$
Inverse	$R^{-1} = R^T$	$g^{-1} = \begin{bmatrix} R^T & -R^T T \\ 0 & 1 \end{bmatrix}$
Composition	$R_{ik} = R_{ij}R_{jk}$	$g_{ik} = g_{ij}g_{jk}$

Y. Ma, S. Soatto. An Invitation to 3D Vision, Springer, 2006

Summary

Transformation	Matrix	DoF 2D	DoF 3D
Euclidean	$\begin{bmatrix} R & T \\ 0^\top & 1 \end{bmatrix}$	3	6
Affine	$\begin{bmatrix} A & b \\ 0^\top & 1 \end{bmatrix}$	6	12
Projective	$\begin{bmatrix} A & b \\ v^\top & 1 \end{bmatrix}$	8	15

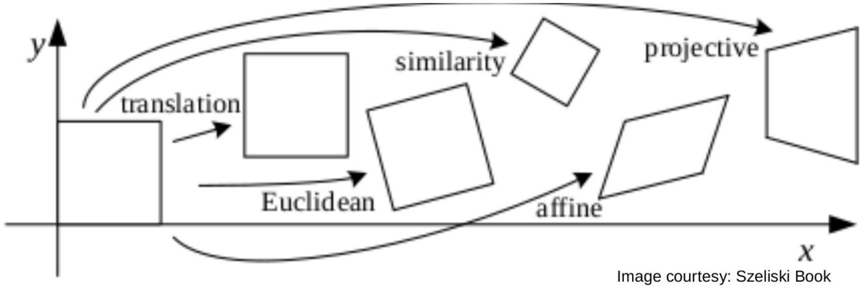
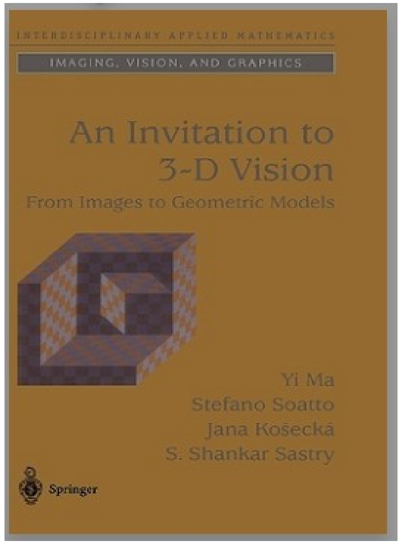


Image courtesy: Szeliski Book

Geometry	Euclidean	Affine	Projective
# components	n	n	n+1
Up to a scale factor	no	no	yes
Transformations			
Rotation, Translation	x	x	x
Scaling, Screw		x	x
Perspective Projection			x
Invariants			
Length, Angle	x		
Relation, Parallelism	x	x	
concurrency, collinearity, cross ratio	x	x	x

Literature

Todays Lecture mainly based on Chapter 2 of Ma and Soatto's book:



Chapter 6 of Barfoot's book: