Geometry for Graphics and Vision

Luiz Velho
IMPA

Computer Graphics

Geometries

- **Euclidean Geometry**
- **Projective Geometry**

The Euclidean Space

Euclidean Space

Definitions

Properties

- N-Dimensional Vector Space (N = 2, 3)
- Inner Product
- Natural Coordinate System

Tools

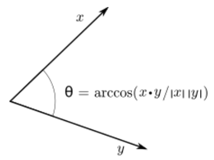
- Linear Algebra

Elements and Operations

- **Vector Type**
  - Constructor / Destructor $x = [x_1, x_2]$
- **Vector Operations**
  - Add $v + w = [v_1 + w_1, v_2 + w_2]$
  - Scale $\lambda x = [\lambda x_1, \lambda x_2]$
- **Null Vector** [0,0]

Metric Properties

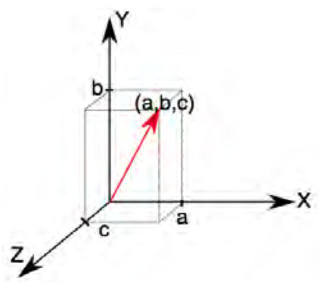
- **Inner Product** $a \cdot b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + \ldots + a_n b_n$
- **Length and Distance** $\| a \| = \sqrt{a^2}$
  $\| a - b \|$
- **Angle** $a \cdot b = \| a \| \| b \| \cos \theta$



- **Unity Vectors** $\hat{b} = b / \| b \|$

Coordinates and Bases

- **Canonical Basis**
- **Coordinate Frame**



Euclidean Transformations

Linear Operators

$$T : \mathbf{R}^n \to \mathbf{R}^n$$

- Definition

$$T(u+v) = T(u) + T(v)$$
$$T(\lambda v) = \lambda T(v)$$

- Invertible
- Linear Invariance
  - Subspaces -> Subspaces
    (lines -> lines)
    (origin -> origin)
- Examples
  - Scaling
  - Rotation
  - Shear

Matrix Representation

- M - Matrix n x n
- Isomorphism
  - Algebra of Linear Operators
  - Algebra of Matrices
- $T \leftrightarrow M$

$$x' = T(x)$$
$$v \to Mv$$

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

Isometries

- Metric Preserving
  $\| T v \| = \| v \|$
  - Invertible
  - Orthonormal Matrix
  - Transformations
    - Rotation
    - Miror
- **OBS: Translation**
  - **Is a Isometry**
  - **But Cannot be Represented by a Matrix**
  - **Not a Linear Transformation**

Affine Transformations

$A(x) = M(x) + t$
Preserve

- Ratios
- Proportions

Transformations

- Linear Transformations
- Translations

OBS: Matrix is not sufficient

Operations: Assessment & Discussion

Properties of Affine Transformations

- Concepts
  - Congruency
  - Similarity
- Transformations
  - Rigid Motions (Isometry)
  - Uniform Scaling (angle)
- Invariant
  - Angles
  - Parallelism

Natural Operations for Modeling

No Unified Representation $v \rightarrow Mv + t$

The Projective Space

Projective Space (2D)

Model of Projective Plane

- Point in Real 2D Projective Space

$$p \in RP^2$$
$$p = (\lambda x_1, \lambda x_2, \lambda x_3) : \lambda \neq 0$$

- Equivalence Relation

$$p = (x_1, x_2, x_3) \equiv \lambda p$$

$$RP^2 := \mathbf{R}^3 - \{(0,0,0)\}$$

<u>3D</u> Projective Transformations

Properties

- Linear Operator in $\mathbf{R}^4$

$$T : \mathbf{R}^4 \to \mathbf{R}^4$$

- T given by M

M, matrix 4×4

- Projective Transformation Induced by T

$$T(p) = M\,p$$

- Note

$$T(p) = \lambda T(p), \lambda \neq 0$$

Anatomy of a Matrix

$$M = \begin{pmatrix} A & T \\ P & S \end{pmatrix}$$

A - Linear Block [3 x 3]
T - Translation Block [3 x 1]
P - Perspective Block [1 x 3]
S - Scaling Block [1 x 1]

Basic Transformations

- **Identity**
- **Scaling**
- **Translation**
- **Perspective**
- **Rotation**

Scaling

- **Uniform / Non-Uniform**

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *OBS: Identity* $a = f = k = 1$

Translation

$$\begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Perspective

- **Viewing Transformation**

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ m & n & p & 1 \end{pmatrix}$$

Rotation

- **Multiple Representations**
  - Orthogonal Matrix with determinant 1 : $M_{3\times 3}$
  - Euler Angles : $R_x(\theta)R_y(\phi)R_z(\gamma)$
  - Rotation Axis / Angle : $(\theta, \hat{x}, \hat{y}, \hat{z})$
  - Quaternion : $q = q_0 + iq_1 + jq_2 + kq_3$
  - Rotor (Geometric Algebra) : $R \in \text{Spin}(V)$
- **Conversion btw Representations**
- **Computation / Derivatives**
  - Exponential Map

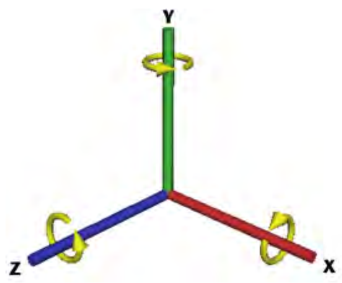
Standard Rotations

- **Euler Angles - Rotation Matrices**

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



OBS: Gimbal Lock

Other Transformations

- Mirror
- Shear
- Skew

Computer Vision

Topics

Cameras → Images

Plenoptic Function → Light Field

- Projective Geometry
- Camera Calibration

Projective Geometry

- Camera
  - Projection
    3D => 2D
- Types of Projection
  - Orthographic (Affine)
  - Perspective
  - Etc..

Vision Problems

- Basic Equation

(*correspondence*) $u \in I \subset \mathbb{R}^2 , \quad x \subset \mathbb{R}^3$

$$u = Px$$

- Camera Calibration

$$u = Px$$

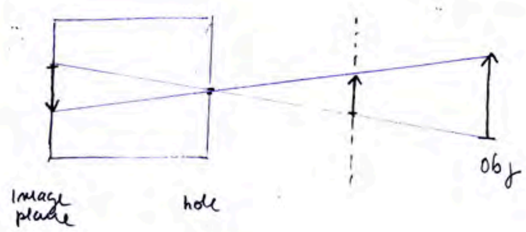
- Shape Reconstruction

$$u = Px$$

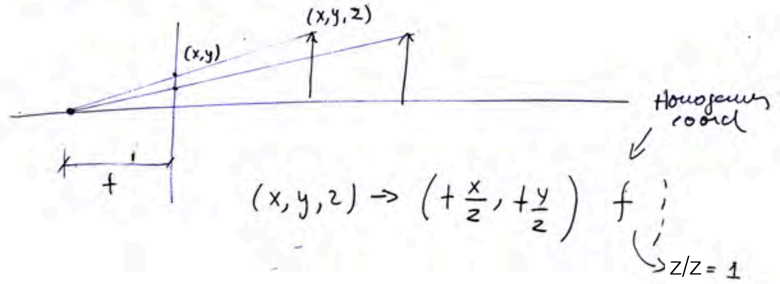
Variants of the Problem

- One Image
- Many Images
- Depth Image..

Pinhole Camera



- perspective projection



Projective / Affine Geometry

- Perspective Transformation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

$$\Rightarrow (u, v)$$

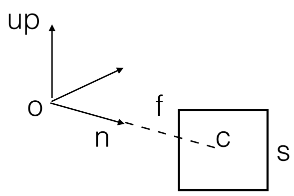
- Weak Perspective : $f \rightarrow \infty$
(*parallel projection*)

$$\begin{bmatrix} x' \\ y' \\ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Affine Geometry

Vision / Graphics

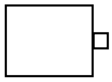
- Camera Model

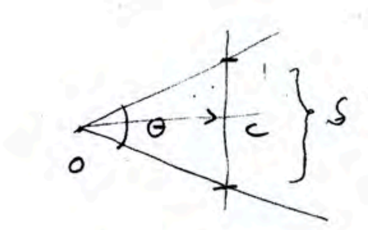


- Camera Parameters
  - Position: o
  - Orientation: up , n
  - Focal Distance: f
  - Image Center: c
  - Image Size: s

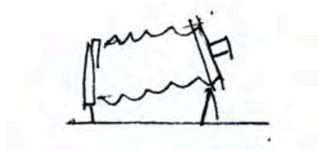
Field of View

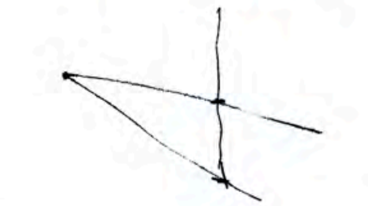
- Pinhole Camera





- View Camera





Camera Transformation

translate, projection, position

division, scale, orientation

$$u = w_d \underbrace{T\,S\,P}\quad \underbrace{R\,T}\quad x$$

pixel, normalized, eye, world

$$u = K \quad \left[R \mid T \right] \quad x$$

intrinsic, extrinsic

focal, center

$$\underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 3\times 3 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ \times \\ 3 \end{bmatrix}}_{3 \times 4}$$

orientation, position

3 x 4

OBS:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & 1 \end{bmatrix}$$

12 parameters
(general)

projective transform
(scale - normalization)

PS: Compare with CG

Fundamental Equation

$$\boxed{u = Px}$$

2D pt

camera
projection

3D pt

$$u = [KA]x$$

intrinsic
(projection)

extrinsic
(affine)

Variants

- Single View
  - Many Points, One Camera

$$u_i = Px_i$$

- Multi View
  - Many Points, Various Cameras (i.e., Images)

$$u_{ik} = P_k x_i$$

Strategy

- Correspondence
  - Key Assumption

I. <u>Single View</u> (*pattern recognition*)

Given 3D object and image,
Find pairs (u_i, x_i)

II. <u>Multi View</u> (*feature matching / tracking*)

Given N images,
Find pairs $(u_i, u_j)_{i \leftrightarrow j}$

Camera Calibration

Calibration, Calibration, Calibration,….

- Inverse Problem

$$u_{ik} = P_k x_i$$

assuming correspondences
compute ***x*** *and/or* ***P***

- OBS: Graphics is Direct Problem

$$Px = u$$

Single View Metrology

$$u_i = Px_i$$

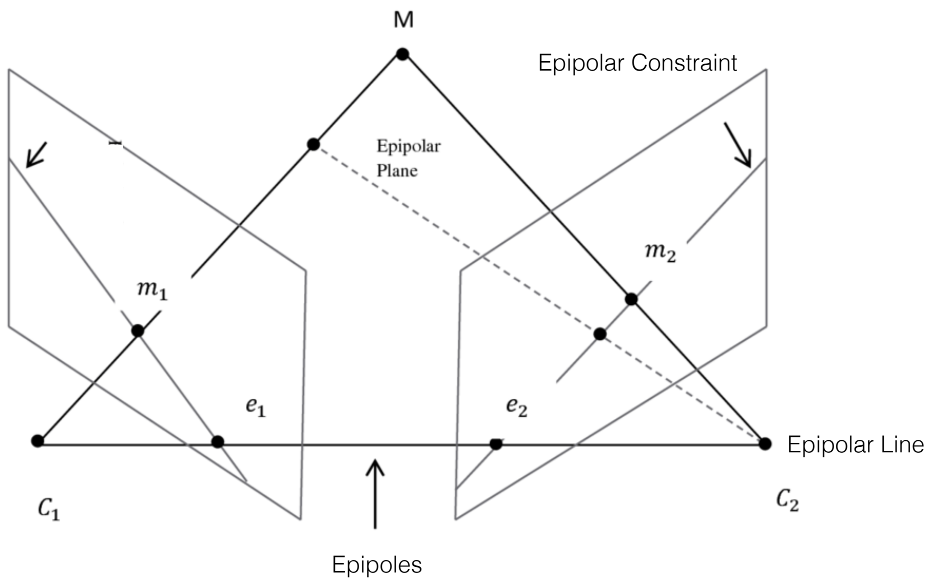
- Problems
  - Camera Calibration:
    Given (u_i, x_i), find P
  - Single View Reconstruction:
    Given $(\bar{u}\bar{v})_i$, find P
- Concepts
  - Absolute Conic

Multi View Metrology

$$u_{ik} = P_k x_i = [K_k A_k] x_i$$

- Problems
  - **Stereo Reconstruction**:
    Given (u_{i0}) and P_0, P_1
    find (u_{i1})
  - **Structure from Motion (SFM)**:
    Given $(u_i, u_j)_{i\leftrightarrow j}$ and K_k
    find $A_k, x_{i\leftrightarrow j}$
  - **Self Calibration**:
    Given $(u_i, u_j)_{i\leftrightarrow j}$,
    find $P_k = [KA]_k$

Epipolar Geometry



Concepts

- Fundamental Matrix

  (*K unknown*)

$$x'^T F x = 0$$

8 pt Algorithm
(self calibration)

- Essential Matrix

  (*K known*)

$$E = \boxed{x'^T F x}$$

metric object

5 pt Algorithm
(SFM)