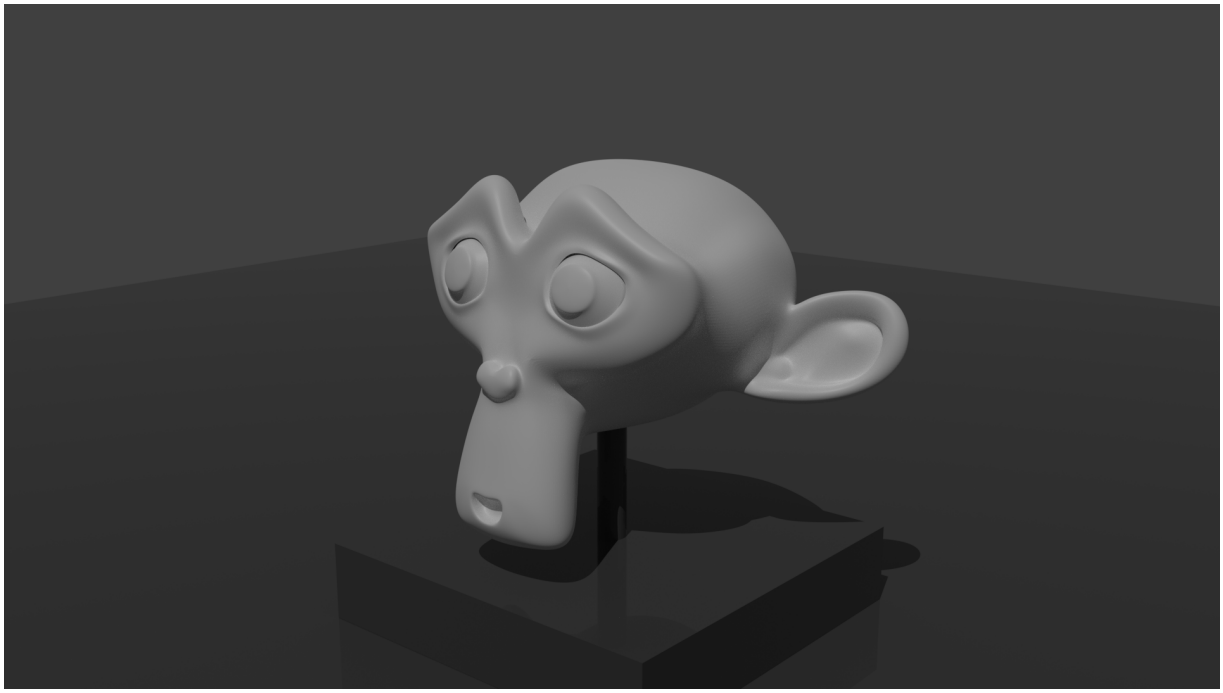
Labs 3 & 4 - Stereovision

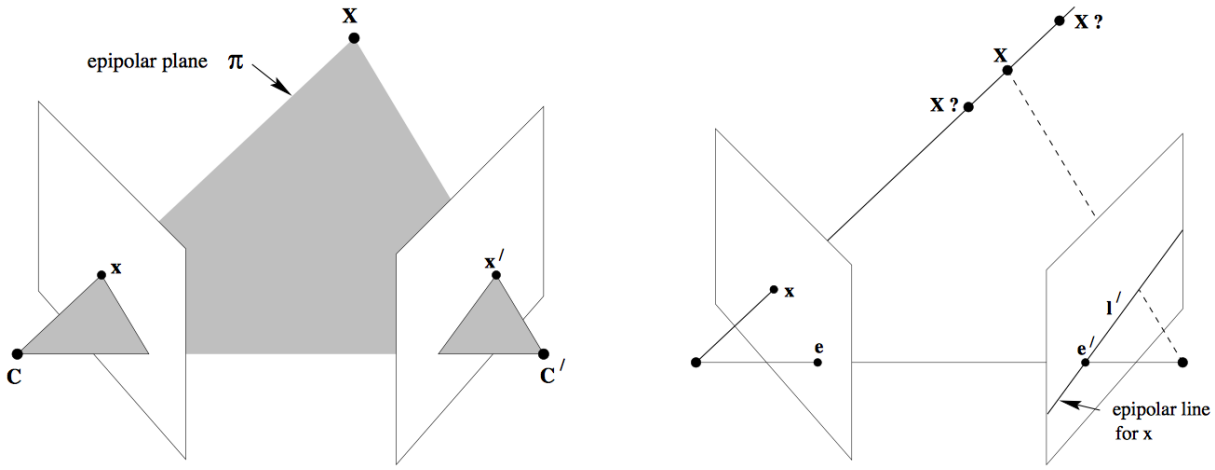


Stereovision is a discipline that deals with the reconstruction of 3D information from images. For the reconstruction of a point, several images of this point are needed. These images must be taken from different points of view. The key step of the reconstruction, which is often problematic, is to identify the images of the point to be reconstructed in each view.

1 Epipolar Geometry

Epipolar geometry involves two cameras. The epipolar geometry describes the geometric properties between two views of the same scene and depends only on the intrinsic parameters of the cameras and their relative positions. It provides, in particular, the epipolar constraint, which will be very useful to produce the matches between views.

2 The Fondamental Matrix



Let us imagine that we have two images, right and left, of the world space. Let's take a point $\vec{x}$ in the right image space. The point $\vec{X}$ of the world space, of which $\vec{x}$ is the image, can be anywhere on the line passing through $\vec{x}$ and the optical center of the right camera. We will call this line the back-projected ray of $\vec{x}$. Let us note $\vec{x'}$ the image of $\vec{X}$ in the left image space. The locus of $\vec{x'}$ is therefore the image line of the back-projected ray of $\vec{x}$. This line is called the epipolar line and is denoted $\vec{l'}$. The epipolar line passes through the epipole $\vec{e'}$, image of the optical center of the right camera.

In 2D projective geometry, a line with equation $ax + by + c = 0$ is represented by a vector with three components $(a, b, c)^T$ defined to within one factor. Thus, we have the following relationship:

The point $\vec{x}$ belongs to the line $\vec{l}$ if and only if $x^T\vec{l} = 0$.

Moreover, in 2D projective geometry, the following remarkable relations are valid:

— The intersection of two lines l and l' is given by $x = l \times l'$,

— The line passing through two points x and x' is given by $l = x \times x'$.

Note that the vector product can be written as a product of matrix $x \times y = [x]_\times y$ where

$$[x]_\times = \begin{pmatrix} 0 & -x3 & x2 \\ x3 & 0 & -x1 \\ -x2 & x1 & 0 \end{pmatrix}$$

To find the equation of the epipolar line in the left image space, we just need to find the coordinates of two points of this line. The first is the image $P'\vec{C}$ of the optical center $\vec{C}$ of the right camera where P' is the projection matrix of the left camera. The

second is $P'P^+\vec{x}$ where P^+ is the pseudo inverse of the projection matrix P of the right camera. The epipolar line thus has the equation $l' = [P'\vec{C}]_\times P'P^+\vec{x} = F\vec{x}$ with $F = [P'\vec{C}]_\times P'P^+$. F is called fundamental matrix.

Since the epipolar line $\vec{l'} = F\vec{x}$ is the locus of $\vec{x'}$, $\vec{x'}$ therefore belongs to $\vec{l'}$ which leads to the epipolar constraint :

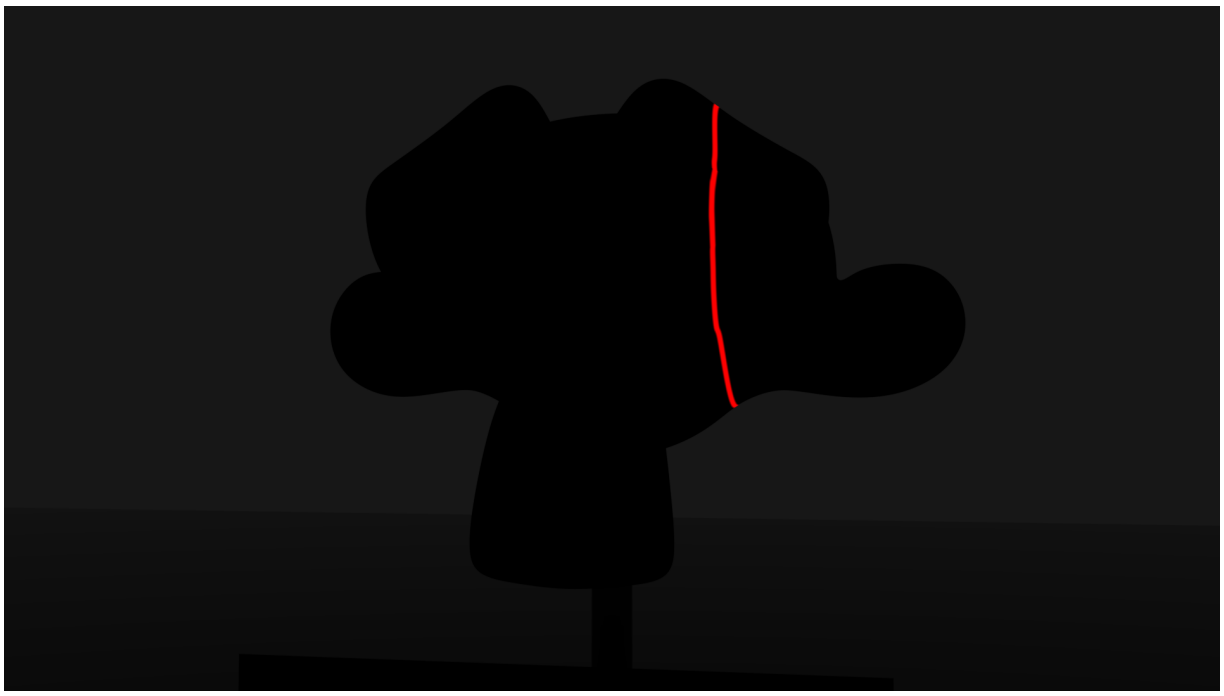
The fundamental matrix is such that for any pair of points corresponding $\vec{x} \leftrightarrow \vec{x'}$ in the two images, we have $\vec{x'}^T F\vec{x} = 0$.

3 Computation of the fundamental matrix

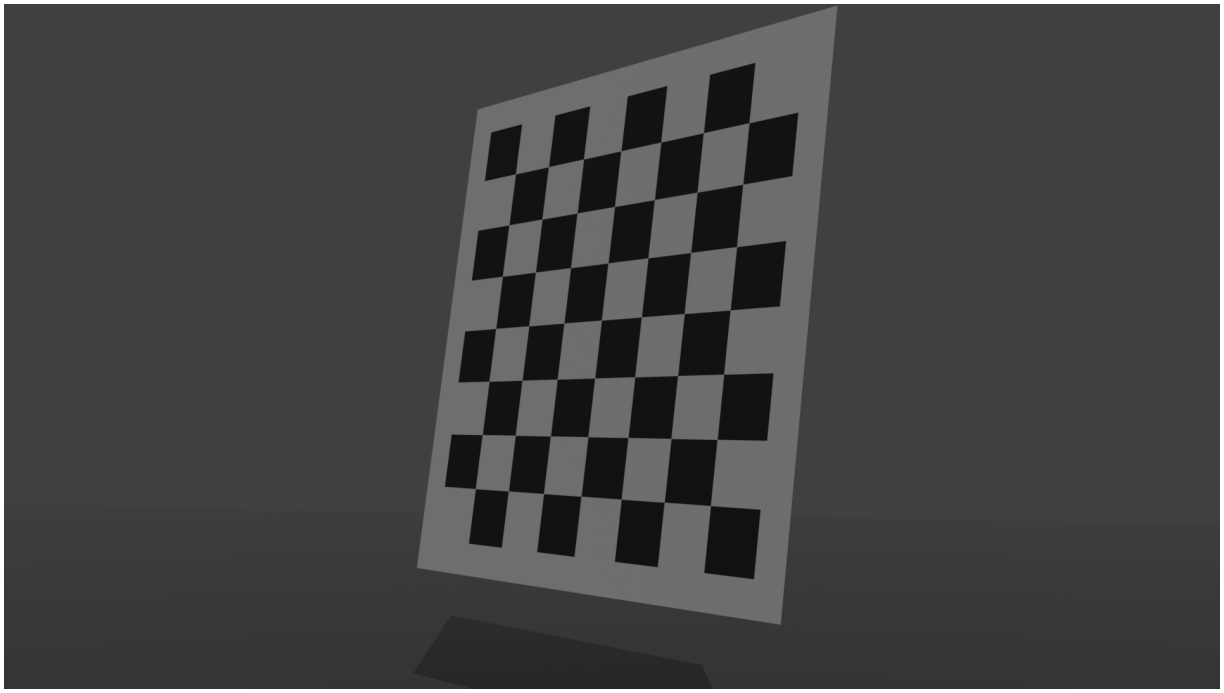
The fundamental matrix F has seven degrees of freedom. It has nine components but these are defined to within one scale factor, which removes one degree of freedom. Moreover, the matrix F is a singular matrix ($det(F) = 0$) which gives us seven degrees of freedom. So we need at least seven correspondences to compute F. The equation $x_i'^T F x_i = 0$ and the seven correspondences allow us to write a system of equations of the form $Af = 0$, where f is the vector which contains the components of the matrix F. Let us assume that A is a 7×9 matrix of rank 7. The general solution of $Af = 0$ can be written $\alpha f_1 + (1-\alpha) f_2$ where f_1 and f_2 are two particular independent solutions of $Af = 0$. We then use the singularity constraint $det(\alpha F_1 + (1-\alpha)F_2) = 0$ to determine α. Since the singularity constraint gives rise to a third degree equation, we may have one or three solutions for F.

4 Goal

In the zip of the statement you will find two sequences of images taken by two cameras during the scanning of an object by a laser plane.



You will also find shots of a checkerboard in different positions that will help you calibrate your cameras.



The goal is to reconstruct the scanned object in 3D.

5 OpenCV

In practice you will use the OpenCV library. In python, you have access to its functions through the `cv2` module.

OpenCV modelize the camera projection like this:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

donc

$$P = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$$

$$P_{int} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$P_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} = (R \quad T)$$

$$P = P_{int} P_{ext}$$

if you compare this with the camera geometry we have seen in the first lab, it comes that

$$T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = - \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = -RC$$

This relation will be usefull later.

`cv2.findChessboardCorners()`: This functiom will find image coordinates of chessboard corners.

`cv2.cornerSubPix()`: This function will improve precision of corner's positions.

`cv2.drawChessboardCorners()`: This function will draw found corners for visual validation purpose.

`cv2.calibrateCamera()`: This function will compute P_{int}, rotation vectors, and T,

based on corners found on multiple chessboards.

`cv2.Rodrigues()`: This function will compute $\boldsymbol{R}$ based on rotation vector.

At this point you have $\boldsymbol{P_{int}}$, $\boldsymbol{R}$ and $\boldsymbol{T}$. You can compute $\boldsymbol{P}$ and $\boldsymbol{C}$ which let you compute $\boldsymbol{F}$.

You can also compute $\boldsymbol{F}$ with `cv2.findFundamentalMat()`.

With $\boldsymbol{F}$ and what you've learn in the second lab you should be able to find matching pairs in left/right images.

`cv2.triangulatePoints()` will compute 3D points from projection matrices and matching pairs.

You can find help with the calibration and reconstruction functions on the site https://docs.opencv.org/4.6.0/d9/d0c/group__calib3d.html

6 Files

Here are the files needed for this lab: Files

7 Evaluation

You will present your results during the oral exam in January.

7.1 Grid

- Projection matrices of the cameras are computed (3 points)
- Fundamental matrix is computed (2 points)
- Red lines are sampled for the pairing process (1 point)
- Epipolar lines are computed (2 points)
- Matching pairs are found (3 points)
- 3D points are triangulated (3 points)
- The final 3D points cloud is displayed (1 point)
- The student can explain what's going on (5 points)