Eötvös Loránd University

Faculty of Informatics

Department of Algorithms And Their Applications

Normal vector estimation of planar surfaces from vehicle-mounted cameras

Supervisor:
Levente Hajder
associate professor

Author:
Zoltán Zele
ARI MSc student

Budapest, 2020

Contents

Chapter 1

Introduction

As autonomous driving gains popularity these days, research on computer vision algorithms for autonomous systems has become one of the most popular topics in computer vision. A fundamental problem in computer vision is to find the relative pose, i.e. the rotation and translation between two cameras, given the images taken of the same scene by these cameras. For autonomous cars, special constraints can be used to simplify the problem and decrease its degrees-of-freedom (DoF). One of these constraints is the planar motion constraint, which is considered to hold when the optical axes of the cameras are in the same 3D plane and their vertical axes are parallel. Many solvers have been proposed based on this constraint, both minimal and non-minimal.

In this thesis I consider another type of special motion that motorbikes can undergo. Cameras attached to motorbikes can also rotate around their optical axes as one can tilt a bike by taking corners or by changing lanes. Planar motion is a special case of this motorbike motion, when the bike is perfectly straight up. To the best of my knowledge, nobody proposed a solution to this problem before. In this thesis I developed a model for the epipolar geometry describing the motorbike motion when the rotation around the optical axis is known for the first camera and also a non-minimal solver using 6 point correspondences. Also I propose a possible solution when the rotation around the optical axes is known for both cameras by reducing the problem to planar motion. Note that when we consider a series of images taken by a single camera attached to a motorbike, the first assumption means that we only need to know the tilt of the bike at the very first image. If the bike is straight when we start the estimation, we can simply initialize the tilt with 0. The tilt at consecutive images can be estimated from the essential matrix. For the proposed solution when the rotation is known for both images, we need to know the tilt of the bike at every image taken, which can be realized by having a gyroscope or an IMU sensor on the bike.

The second part of this thesis deals with the detection of planar surfaces by robustly fitting point correspondences to special homographies when the planar motion constraint holds. Special homographies can be estimated for

- planes parallel to the camera's XZ-plane; (e.g. ground plane)
- planes parallel to the camera's YZ-plane; (e.g. walls of buildings at the side of the road)
- planes parallel to the camera's XY-plane; (e.g. planes in front of the camera)
- planes perpendicular to the camera's XZ-plane: general vertical planes with a normal vector characterized by a single angle. The two cases above are special cases of this one.

Here the coordinate system is chosen so the Z axis is the optical axis, Y axis is the vertical axis and the X axis is the horizontal axis of the camera.

1.1 Contribution

My contribution in this thesis is twofold. First, I developed a model for the epipolar geometry between two cameras attached to motorbikes and proposed solutions for the essential matrix estimation arising from the model. Based on the idea of rotating the points back around the optical axis, I propose two solvers. First an algorithm based on 6 point correspondences is presented when the rotation around the optical axis is known for the first camera and a method to reduce the problem to planar motion when it is known for both cameras.

In the second part of this thesis, I used special homographies to detect planar surfaces on a stereo image pair. It is based on the work of my supervisor and his colleagues. They decomposed the homographies to retrieve the camera motion, while I used the homographies to detect planes on the images using RANSAC and sequential RANSAC. The method is tested on real images.

Chapter 2

Overview of related literature

In the first part of this thesis I investigate the epipolar geometry in the case of motorbike motion. I do so by building on well developed and widely used methods for epipolar geometry estimation. The epipolar geometry between a stereo image pair is described by a 3×3 matrix called the fundamental matrix. If the intrinsic parameters of the cameras are known, the fundamental matrix can be transformed into the essential matrix. A standard way to estimate the fundamental matrix by point correspondences is the 8-point method described in [1]. Its results can be used as a reference for testing the methods for the constrained cases. In the proposed solution to the essential matrix estimation when the motorbike motion constraint holds and the rotation around the optical axis is known for both cameras, I reduced the problem back to planar motion. As planar motion is a popular constraint in autonomous driving, many estimators have been proposed, both for minimal number of correspondences and non-minimal number of correspondences. A method proposed by Choi et al. [2] uses a minimal of 2 point correspondences to calculate the essential matrix as the intersection between a zero centered circle and a zero centered ellipse as well as the intersection between a line and a zero centered circle. Levente Hajder and Daniel Barath [3] proposed a method for the over-determined case, having more than a minimal number of correspondences that gives an optimal solution in the least-squares sense. They also proposed a method for the estimation of relative planar motion by using only a single affine correspondence [4]. As of now, I am not aware of any works trying to solve the essential matrix estimation for cameras mounted on a motorcycle.

In the second part of this thesis I worked on a plane detector which is based on the work made by my supervisor and his colleagues. They estimated the relative pose of the cameras under planar motion by decomposing special homographies. They were able to construct these homographies using only a single affine correspondence.

Chapter 3

Preliminaries

The material presented here is based on the Computer Vision Course presentations of ELTE [5].

3.1 Perspective camera and homogeneous coordinates

Our goal is to model how real world images are created, in other words, how the world points are projected onto the cameras' image planes. In computer vision we usually use the perspective camera model to model real world cameras. A perspective camera is a simple model of thin optics, but it is a very good geometric approximation [5].

3.1.1 Notations

$\mathbf{X} = [X, Y, Z]^T$: world coordinates

$\mathbf{X} = [X, Y, Z, 1]^T$: homogeneous world coordinates

$\mathbf{X}_c = [X_c, Y_c, Z_c]^T$: camera coordinates

$\mathbf{X}_c = [X_c, Y_c, Z_c, 1]^T$: homogeneous camera coordinates

$\mathbf{u} = [u, v]^T$: image plane coordinates

$\mathbf{u} = [u, v, 1]^T$: homogeneous image plane coordinates

$\mathbf{R}$: 3×3 rotation matrix

$\mathbf{t}$: 3×1 translation vector

$\mathbf{C}$: focal point, center of projection

ϕ: image plane

f: focal length, the distance between the focal point and the image plane

Optical ray: connects a world point and the focal point

Optical axis: contains the focal point and perpendicular to the image plane

$\mathbf{u}_0 = [u_0, v_0]^T$: principal point, the point in the image plane where the optical axis intersects the image plane

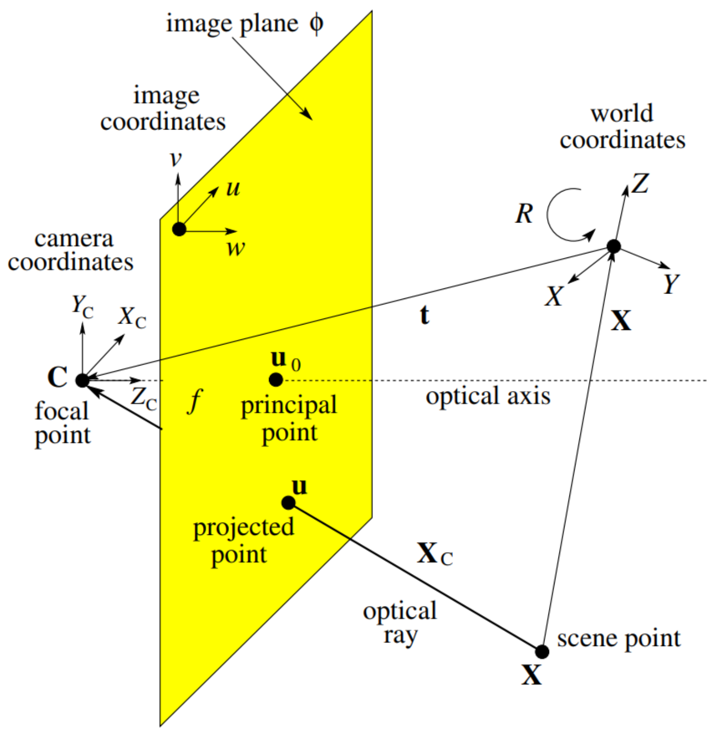

Figure 3.1: Perspective camera model. The image is from the Computer Vision Course presentations [5].

3.1.2 World to camera coordinate system transformation

Before projecting onto the image plane, we need to know the coordinates of the world points with respect to the camera coordinate system. If we know the rotation $\mathbf{R}$ and translation $\mathbf{t}$ of the camera in the world coordinate system, we can calculate the camera coordinates $\mathbf{X}_c$ the following way:

$$\mathbf{X}_c = \mathbf{R}\mathbf{X} + \mathbf{t}.$$

If homogeneous coordinates are used:

$$\mathbf{X}_c = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right] \mathbf{X},$$

where $\left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right]$ is a 3×4 matrix consisting of $\mathbf{R}$ appended by $\mathbf{t}$. $\mathbf{R}$ and $\mathbf{t}$ are called the camera's extrinsic parameters.

3.1.3 Projection onto the image plane

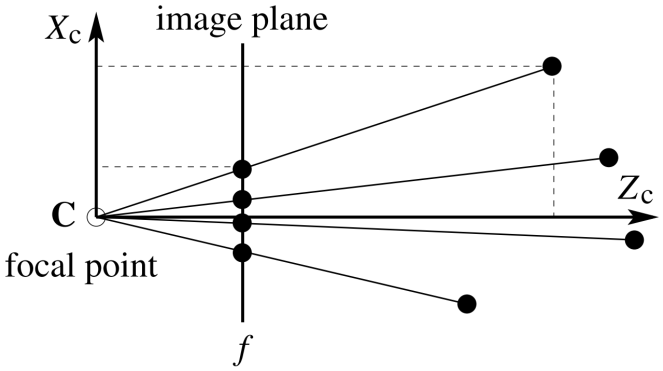

Figure 3.2: Projection onto the image plane. The image is from the Computer Vision Course presentations [5].

By similar triangles one can easily calculate the image coordinates $\mathbf{u} = [u, v]$ from the camera coordinates, if the focal length f, horizontal pixel size k_u, vertical pixel size k_v and the principal point $\mathbf{u}_0 = [u_0, v_0]$ is known:

$$u = \frac{f k_u}{Z_c} X_c + u_0,$$

$$v = \frac{f k_v}{Z_c} Y_c + v_0.$$

Usually $k_u = k_v = k$. The parameters fk_u, fk_v, u_0, v_0 are called the camera's intrinsic parameters. Matrix $\mathbf{K}$ consisting of these parameters is called the camera's calibration matrix:

$$\mathbf{K} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Projection using $\mathbf{K}$:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u X_c + Z_c u_0 \\ fk_v Y_c + Z_c v_0 \\ Z_c \end{bmatrix} = \mathbf{K}\mathbf{X}_c.$$

Homogeneous division (division by the last coordinate Z_c) gives the image coordinates $\mathbf{u}$ in homogeneous coordinates. Note that homogeneous division causes scale ambiguity as for $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$, the projection of the vectors

$$\alpha \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, \beta \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

will result in the same image plane coordinates after homogeneous division. That is because we lose the depth information after projecting to the image plane, so we will say that these vectors are similar up to an unknown scale. The notation $\sim$ is used to indicate the similarity and that homogeneous division should be carried out for the equality to hold.

$$\mathbf{u} \sim \mathbf{K}\mathbf{X}_c.$$

3.1.4 Projection matrix P

If homogeneous coordinates are used, we can determine the image plane coordinates from the world coordinates as follows:

$$\mathbf{u} \sim \mathbf{P}\mathbf{X},$$

where the matrix $\mathbf{P}$ is called the camera's projection matrix.

$$\mathbf{P} = \mathbf{K} \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right].$$

3.2 Epipolar geometry

Epipolar geometry describes the geometry between two views. There are two cameras denoted by their centers (focal points) $\mathbf{C}_1$ and $\mathbf{C}_2$. The line connecting the camera centers is called the baseline. The points where the baseline intersects the image planes are called the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$. The projection of the world point $\mathbf{X}$ has the coordinates $\mathbf{u}_1$ and $\mathbf{u}_2$ on the first and second camera's image plane respectively.

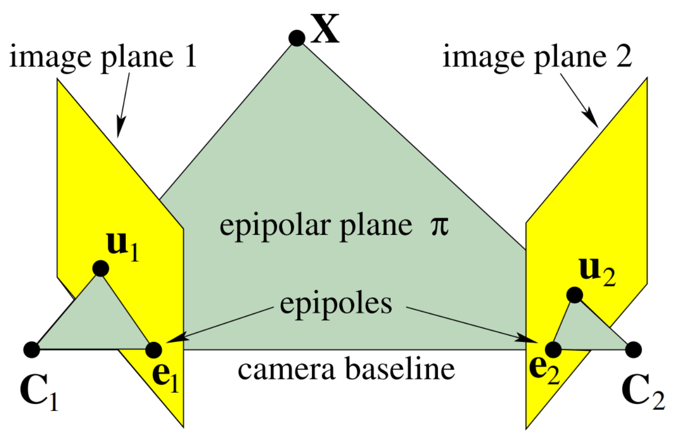

Figure 3.3: Geometry between two views. The image is from the Computer Vision Course presentations [5].

The plane described by the points $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{X}$ is called the epipolar plane π. The epipolar plane intersects the image planes at the epipolar lines $\mathbf{l}_1$ and $\mathbf{l}_2$.

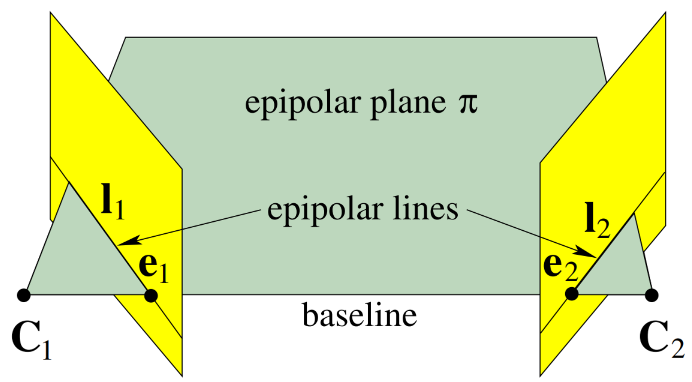

Figure 3.4: Epipolar lines. The image is from the Computer Vision Course presentations [5].

3.2.1 Epipolar constraints

The image point $\mathbf{u}_1$ back-projected into the 3D-space defines a ray. The ray lies in the epipolar plane π, so its projection on the second image is the epipolar line $\mathbf{l}_2$. As the ray contains the world point $\mathbf{X}$, the projection $\mathbf{u}_2$ on the second image must lie on the epipolar line $\mathbf{l}_2$. This constraint can be represented as

$$\mathbf{l}_2^T \mathbf{u}_2 = 0. \tag{3.1}$$

The same constraint can be derived for the first image point $\mathbf{u}_1$:

$$\mathbf{l}_1^T \mathbf{u}_1 = 0. \tag{3.2}$$

Also, all epipolar lines contain the epipoles:

$$\mathbf{l}_2^T \mathbf{e}_2 = \mathbf{l}_1^T \mathbf{e}_1 = 0. \tag{3.3}$$

3.2.2 Fundamental matrix

The fundamental matrix $\mathbf{F}$ is 3×3 matrix that encapsulates the epipolar geometry between the two views. If $\mathbf{u}_1$ and $\mathbf{u}_2$ are corresponding image plane points, then the following equation holds:

$$\mathbf{u}_2^T \mathbf{F} \mathbf{u}_1 = 0. \tag{3.4}$$

This is the same constraint described in equation 3.2 and 3.1 with $\mathbf{l}_1 = \mathbf{F}^T \mathbf{u}_2$ and $\mathbf{l}_2 = \mathbf{F}\mathbf{u}_1$ respectively. Also from equation 3.3

$$\mathbf{u}_1^T \mathbf{F}^T \mathbf{e}_2 = \mathbf{u}_2^T \mathbf{F} \mathbf{e}_1 = 0$$

for all $\mathbf{u}_1$ and $\mathbf{u}_2$. Since

$$(\mathbf{u}_2^T \mathbf{F})\mathbf{e}_1 = \mathbf{u}_2^T (\mathbf{F}\mathbf{e}_1) = 0$$

for all $\mathbf{u}_2$, $\mathbf{F}\mathbf{e}_1 = 0$ must be true. Therefore $\mathbf{e}_1$ is the right null-vector of $\mathbf{F}$ [1]. Similarly $\mathbf{e}_2$ is the left null-vector of $\mathbf{F}$.

$\mathbf{F}$ has 9 elements, but we can only estimate it up to an unknown scale, which decreases its degrees-of-freedom (DoF) by 1. Also its null-space contains $\mathbf{e}_1$, so it is singular:

$$\det \mathbf{F} = 0.$$

That means that $\mathbf{F}$ has 7 DoF.

3.2.3 Essential matrix

If we know the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ corresponding to the first and second camera, we can transform the fundamental matrix $\mathbf{F}$ into the essential matrix

$\mathbf{E}$ as follows:

$$\mathbf{E} = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1, \tag{3.5}$$

and $\mathbf{E}$ into $\mathbf{F}$ like:

$$\mathbf{F} = \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1},$$

where $\mathbf{K}_2^{-T} = (\mathbf{K}_2^T)^{-1}$. The epipolar constraint 3.4 using $\mathbf{E}$ becomes:

$$\mathbf{u}_2^T \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \mathbf{u}_1 = 0.$$

The coordinates $\mathbf{K}_2^{-1}\mathbf{u}_2$ and $\mathbf{K}_1^{-1}\mathbf{u}_1$ corresponding to the image plane coordinates $\mathbf{u}_2$ and $\mathbf{u}_1$ are called normalized coordinates. Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be the projection matrices corresponding to the first and second camera. Then $\mathbf{K}_1^{-1}\mathbf{P}_1$ and $\mathbf{K}_2^{-1}\mathbf{P}_2$ are called normalized projection matrices.

Let the world coordinate system be fixed to the first camera and $\mathbf{R}$ and $\mathbf{t}$ be the relative pose (rotation and translation) between the two cameras. Then the normalized projection matrices are $\mathbf{K}_1^{-1}\mathbf{P}_1 = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array}\right]$ and $\mathbf{K}_2^{-1}\mathbf{P}_2 = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right]$ and the essential matrix has the form:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}.$$

The 3×3 matrix $[\mathbf{t}]_\times$ is a skew-symmetrix matrix composed of the elements of $\mathbf{t}$:

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -\mathbf{t}_3 & \mathbf{t}_2 \\ \mathbf{t}_3 & 0 & -\mathbf{t}_1 \\ -\mathbf{t}_2 & \mathbf{t}_1 & 0 \end{bmatrix}.$$

For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $[\mathbf{a}]_\times \mathbf{b}$ has the property:

$$[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}.$$

$\mathbf{E}$ has 5 DoF. 3 in $\mathbf{R}$, 3 in $\mathbf{t}$, but as with the fundamental matrix, we lose 1 because of the scale ambiguity.

3.3 Planar homographies

Let the world points $\mathbf{X}_i, i = 1..n$, lie on the same 3D plane and let $\mathbf{u}_i$ be their projection on the first camera's image plane and $\mathbf{u}'_i$ be their projection on the second camera's image plane. Then there exists a linear transformation $\mathbf{H}$, such that

$$\mathbf{u}'_i \sim \mathbf{H}\mathbf{u}_i, \quad \forall i = 1..n,$$

if homogeneous coordinates are used. The 3×3 matrix $\mathbf{H}$ is called a planar homography.

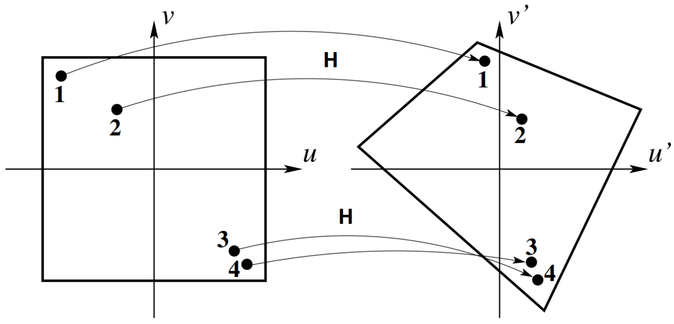

Figure 3.5: Homography. The image is from the Computer Vision Course presentations [5].

Chapter 4

Essential matrix estimation

One of the most fundamental problems in computer vision is the estimation of the relative pose (rotation $\mathbf{R}$ and translation $\mathbf{t}$) between two cameras. If we have n point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n$, on the cameras' image planes, the fundamental matrix can be estimated based on only these point correspondences. If the cameras are calibrated and their calibration matrices are $\mathbf{K}$ and $\mathbf{K}'$, the essential matrix can be calculated by either equation 3.5 or by the normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, where $\hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$. If the world coordinate system is fixed to the first camera's coordinate system and ($\mathbf{R}$,$\mathbf{t}$) is the relative motion between the cameras, the essential matrix has the form:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}.$$

It can be shown that $\mathbf{t}$ and $\mathbf{R}$ can be calculated from $\mathbf{E}$ up to a finite number of possibilities [1].

4.1 General motion

A standard way to estimate the epipolar geometry when we do not assume any kind of special motion between the cameras is the 8-point method [1]. With the 8-point method, we estimate the fundamental matrix from $n \geq 8$ point correspondences. If the cameras are calibrated, the fundamental matrix can be transformed into the essential matrix by equation 3.5 to retrieve the relative pose. The derivation for data normalization and the normalized 8-point method is based on the Computer Vision Course presentations [5].

4.1.1 Data normalization

In order to make the 8-point method numerically stable, the point correspondences have to be in the same order of magnitude. The origin should be moved to the center of gravity and the average of the point distances should be scaled to $\sqrt{2}$. The points in the first image are normalized by the affine transformation $\mathbf{T}$ and the points on the second image are normalized by the affine transformation $\mathbf{T}'$, where

$$\mathbf{T} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -u_c \\ 0 & 1 & -v_c \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{T}' = \begin{bmatrix} s' & 0 & 0 \\ 0 & s' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -u'_c \\ 0 & 1 & -v'_c \\ 0 & 0 & 1 \end{bmatrix}.$$

$\mathbf{u}_c = \begin{bmatrix} u_c & v_c \end{bmatrix}^T$ and $\mathbf{u}'_c = \begin{bmatrix} u'_c & v'_c \end{bmatrix}^T$ are the centers of gravity for the first and second point set:

$$\mathbf{u}_c = \frac{\sum_{i=1}^{n} \mathbf{u}_i}{n}, \mathbf{u}'_c = \frac{\sum_{i=1}^{n} \mathbf{u}'_i}{n}$$

and s, s' are the scaling factors making the average difference between the points and centers of gravity to be $\sqrt{2}$:

$$\frac{\sqrt{2}}{s} = \frac{\sum_{i=1}^{n} ||\mathbf{u}_i - \mathbf{u}_c||_2}{n}, \frac{\sqrt{2}}{s'} = \frac{\sum_{i=1}^{n} ||\mathbf{u}'_i - \mathbf{u}'_c||_2}{n}.$$

The normalized points are then $\hat{\mathbf{u}}_i = \mathbf{T}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{T}'\mathbf{u}'_i$.

4.1.2 Normalized 8-point method

Given point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 8$, the epipolar constraint 3.4 holds:

$$\mathbf{u}'^{T}_i \mathbf{F} \mathbf{u}_i = 0. \tag{4.1}$$

The unknowns are the elements of the fundamental matrix:

$$\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}.$$

For a single correspondence $\begin{bmatrix} u_i & v_i \end{bmatrix}^T \leftrightarrow \begin{bmatrix} u_i' & v_i' \end{bmatrix}^T$, equation 4.1 can be written as:

$$u_i'u_if_1 + u_i'v_if_2 + u_i'f_3 + v_i'u_if_4 + v_i'v_if_5 + v_i'f_6 + u_if_7 + v_if_8 + f_9 = 0. \qquad (4.2)$$

If the notation $\mathbf{f} = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 \end{bmatrix}^T$ is used, equation 4.2 can be written as:

$$\begin{bmatrix} u_i'u_i & u_i'v_i & u_i' & v_i'u_i & v_i'v_i & v_i' & u_i & v_i & 1 \end{bmatrix} \mathbf{f} = 0.$$

Using all points $\mathbf{u}_i, \mathbf{u}_i', i = 1..n, n \geq 8$, the following homogeneous linear system of equations is obtained:

$$\mathbf{Af} = \begin{bmatrix} u_1'u_1 & u_1'v_1 & u_1' & v_1'u_1 & v_1'v_1 & v_1' & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ u_n'u_n & u_n'v_n & u_n' & v_n'u_n & v_n'v_n & v_n' & u_n & v_n & 1 \end{bmatrix} \mathbf{f} = 0.$$

The trivial solution $\mathbf{f} = \mathbf{0}$ should be excluded. This can be done by fixing the norm of $\mathbf{f}$ to be 1: $||\mathbf{f}||_2 = 1$. Now the problem to be solved is to minimize the algebraic error $||\mathbf{Af}||_2$ with respect to $||\mathbf{f}||_2 = 1$. This is a constrained minimization problem and the constraint $||\mathbf{f}||_2 = 1$ can be written algebraically as $\mathbf{f}^T\mathbf{f} - 1 = 0$. Now if Lagrange multipliers are used, the problem becomes:

$$\arg\min_{\mathbf{f}} [\mathbf{f}^T\mathbf{A}^T\mathbf{A}\mathbf{f} - \lambda(\mathbf{f}^T\mathbf{f} - 1)],$$

where λ is the Lagrange multiplier. The optimal solution is the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue.

The solution does not guarantee the singularity constraint $\det \mathbf{F} = 0$, which means that the epipolar lines do not necessarily contain the epipoles. This can lead to less accurate epipolar geometry and should be corrected. The problem now is to find the singular matrix $\mathbf{F}'$ most similar to $\mathbf{F}$:

$$\mathbf{F}' = \arg\min_{\mathbf{F}'} ||\mathbf{F} - \mathbf{F}'||.$$

The optimal solution can be found by singular value decomposition (SVD) if the Frobenius norm is used. The Frobenius norm of a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ is defined as

$$||\mathbf{B}||_{\mathbf{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{B}_{i,j}^2}$$

where $\mathbf{B}_{i,j}$ is the element in the i-th row and j-th column of $\mathbf{B}$. Let

$$\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

be the singular value decomposition of $\mathbf{F}$, where

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

is the matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. Then the closest singular matrix to $\mathbf{F}$ with respect to the Frobenius norm can be found by setting the smallest singular value σ_3 to be 0:

$$\mathbf{F}' = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T.$$

Algorithm 1 Normalized 8-point algorithm

Input: Point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 8$

Output: Fundamental matrix $\mathbf{F}$

1: Data normalization is done for both point point sets, resulting in the normalized points $\hat{\mathbf{u}}_i, \hat{\mathbf{u}}'_i$ and affine transformations $\mathbf{T}, \mathbf{T}'$

2: Build the coefficient matrix $\mathbf{A}$ from the normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$

3: Find $\mathbf{f}$ as the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue

4: Construct $\hat{\mathbf{F}}$ from $\mathbf{f}$

5: Compute the singular value decomposition $\hat{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

6: Force the singularity constraint as

$$\hat{\mathbf{F}}' = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T.$$

7: Denormalization is done by $\mathbf{F} = \mathbf{T}'^T\hat{\mathbf{F}}'\mathbf{T}$

Then if the calibration matrices $\mathbf{K}$ and $\mathbf{K}'$ are known for the cameras, the essential matrix can be calculated by the formula:

$$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K}.$$

4.2 Planar motion

Now we constrain the motion of the cameras to comply with the planar motion constraint. This constraint considered to hold when the optical axis of the cameras are in the same 3D plane and their vertical axis are parallel. A typical example of this kind of motion is when the cameras are attached to a moving car. The car can only move (translate) and steer (rotate) on the plane of the road and cannot make any kind of vertical movement. This means that the cameras' y-coordinate cannot change and they can only rotate around their Y-axis. Let the normalized projection matrices of the cameras be

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array}\right], \mathbf{P}' = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right]$$

with translation and rotation

$$\mathbf{t} = \begin{bmatrix} x \\ 0 \\ z \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}.$$

If polar coordinates are used, the translation $\mathbf{t}$ can be written as

$$\mathbf{t} = \rho \begin{bmatrix} \cos\beta \\ 0 \\ \sin\beta \end{bmatrix}.$$

The description of the problem using the angles α and β is visualized from the top on the figure 4.1.

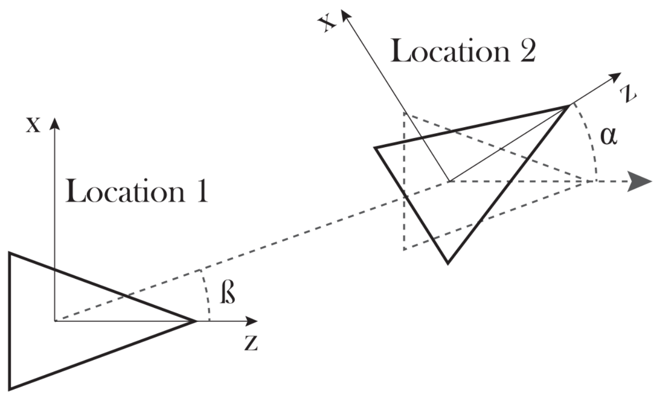

Figure 4.1: Planar motion. The image is from the paper [3].

The essential matrix is then

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

where the cross-product matrix $[\mathbf{t}]_\times$ is

$$[\mathbf{t}]_\times = \rho \begin{bmatrix} 0 & -\sin\beta & 0 \\ \sin\beta & 0 & -\cos\beta \\ 0 & \cos\beta & 0 \end{bmatrix}$$

and the product $[\mathbf{t}]_\times \mathbf{R}$ is

$$\mathbf{E} = \rho \begin{bmatrix} 0 & -\sin\beta & 0 \\ \sin\beta\cos\alpha + \cos\beta\sin\alpha & 0 & \sin\beta\sin\alpha - \cos\beta\cos\alpha \\ 0 & \cos\beta & 0 \end{bmatrix}.$$

Using the trigonometrix identities $\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$ and $\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$:

$$\mathbf{E} = \rho \begin{bmatrix} 0 & -\sin\beta & 0 \\ \sin(\alpha+\beta) & 0 & -\cos(\alpha+\beta) \\ 0 & \cos\beta & 0 \end{bmatrix}.$$

4.2.1 Estimation by the intersection of a zero centered circle and ellipse

As a solution to the estimation of the essential matrix, I chose the circle-ellipse method described in the paper by Choi et al. [2]. $\mathbf{E}$ has 2 DoF (α and β) since the scale ρ cannot be estimated by point correspondences only. It is a minimal solver using $n \geq 2$ point correspondences. Let the correspondences be $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 2$. The epipolar constraint holds for the normalized correspondeces $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i, \hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i, \hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$:

$$\hat{\mathbf{u}}_i'^T \mathbf{E} \hat{\mathbf{u}}_i = 0. \tag{4.3}$$

Let $\hat{\mathbf{u}}_i = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}$ and $\hat{\mathbf{u}}'_i = \begin{bmatrix} u'_i & v'_i & 1 \end{bmatrix}$. Then the epipolar constraint 4.3 above can be written as

$$v_i \cos\beta - u'_i v_i \sin\beta - v'_i \cos(\alpha+\beta) + u_i v'_i \sin(\alpha+\beta) = 0. \tag{4.4}$$

Using the notation $\mathbf{b} = \begin{bmatrix} \cos\beta & \sin\beta & \cos(\alpha+\beta) & \sin(\alpha+\beta) \end{bmatrix}^T$, equation 4.4 becomes

$$\begin{bmatrix} v_i & -u'_i v_i & -v'_i & u_i v'_i \end{bmatrix} \begin{bmatrix} \cos\beta \\ \sin\beta \\ \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix} = 0.$$

For all correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i, i = 1..n, n \geq 2$:

$$\mathbf{Ab} = \begin{bmatrix} v_1 & -u'_1 v_1 & -v'_1 & u_1 v'_1 \\ \vdots & \vdots & \vdots & \vdots \\ v_n & -u'_n v_n & -v'_n & u_n v'_n \end{bmatrix} \begin{bmatrix} \cos\beta \\ \sin\beta \\ \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix} = 0,$$

where $\mathbf{A}_i$ denotes the i-th row of the coefficient matrix $\mathbf{A}$. This is a homogeneous linear system of equations like we saw at the fundamental matrix estimation. The problem to be solved is

$$\mathbf{b} = \arg\min_{\hat{\mathbf{b}}} ||\mathbf{A}\hat{\mathbf{b}}||_2^2.$$

The solver described in this section uses additional constraints to make the estimation non-iterative and use a minimal number of point correspondences. Constraints can be found in the vector $\mathbf{b}$, namely:

$$b_1^2 + b_2^2 = \cos^2\beta + \sin^2\beta = 1 \tag{4.5}$$

and

$$b_3^2 + b_4^2 = \cos^2(\alpha+\beta) + \sin^2(\alpha+\beta) = 1. \tag{4.6}$$

Another way to represent the epipolar constraint 4.3 is

$$\mathbf{Aa} = \mathbf{Bb},$$

with $\mathbf{A}_i = \begin{bmatrix} v_i & -u_i' v_i \end{bmatrix}$, $\mathbf{B}_i = \begin{bmatrix} v_i' & -u_i v_i' \end{bmatrix}$ and $\mathbf{a} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix}$. If we multiply the equation with pseudo-inverse $\mathbf{B}^\dagger$ of the coefficient matrix $\mathbf{B}$ from the left, we get the equation

$$\mathbf{B}^\dagger \mathbf{Aa} = \mathbf{Ca} = \mathbf{b}.$$

The constraints 4.5 and 4.6 can be represented as

$$\mathbf{a}^T\mathbf{a} = 1 \text{ and } \mathbf{b}^T\mathbf{b} = 1.$$

Since $\mathbf{b} = \mathbf{Ca}$

$$\mathbf{b}^T\mathbf{b} = \mathbf{a}^T\mathbf{C}^T\mathbf{Ca} = 1.$$

Let $\mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T$, then the expression $\mathbf{a}^T\mathbf{a} = 1$ equals to $a_1^2 + a_2^2 = 1$, which is the equation of a zero centered unit circle. Denote the elements of the symmetric matrix $\mathbf{C}^T\mathbf{C}$ as

$$\mathbf{C}^T\mathbf{C} = \begin{bmatrix} d & \frac{e}{2} \\ \frac{e}{2} & f \end{bmatrix}.$$

Then expression $\mathbf{a}^T\mathbf{C}^T\mathbf{Ca} = 1$ equals to $da_1^2 + ea_1a_2 + fa_2^2 = 1$, which is the equation of a zero centered ellipse. So in order to find the vector $\mathbf{a}$, we need to compute the intersections between a zero centered unit circle and a zero centered ellipse. Using the singular value decomposition, it is possible to make two ellipses axis-aligned so that the intersections are easily found [2]. The SVD of the symmetric matrix $\mathbf{C}^T\mathbf{C}$ is

$$\mathbf{C}^T\mathbf{C} = \mathbf{U\Sigma U}^T.$$

$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$ is the diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq 0$. The axis-aligned ellipses are defined using $\mathbf{y} = \mathbf{U}^T\mathbf{a}$ as:

$$\mathbf{y}^T\mathbf{y} = 1 \text{ and } \mathbf{y}^T\mathbf{\Sigma y} = 1. \tag{4.7}$$

The solutions to the intersections of the unit circle and axis-aligned ellipse above are:

$$y_1 = \pm\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \text{ and } y_2 = \pm\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}}.$$

There are four candidate solutions that can be obtained by the combinations of y_1 and y_2, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. There can be degenerate cases when $\sigma_2 > 1$ or $\sigma_1 < 1$. In these cases there is no intersection between the circle and the ellipse, but since they are axis-aligned, we can select nearest points between them. If $\sigma_1 < 1$, then the solution is

$$\mathbf{y} = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix},$$

and when $\sigma_2 > 1$ the solution is

$$\mathbf{y} = \begin{bmatrix} 0 \\ \pm 1 \end{bmatrix}.$$

These solutions break the second part of the constraints 4.7, but they are optimal in following way:

$$\mathbf{y} = \arg\min_{\hat{\mathbf{y}}^T\hat{\mathbf{y}}=1} |\hat{\mathbf{y}}^T\mathbf{\Sigma}\hat{\mathbf{y}} - 1|^2.$$

Finally the solution to the original problem is

$$\mathbf{a} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} = \mathbf{U}\mathbf{y} \text{ and } \mathbf{b} = \begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix} = \mathbf{C}\mathbf{a}.$$

Algorithm 2 Circle-ellipse method

Input: Point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 2$, and calibration matrices $\mathbf{K}$ and $\mathbf{K}'$

Output: Essential matrices $\mathbf{E}_i, i = 1..n$, where $n = 2$ if there is no intersection between the circle and ellipse (degenerate case), $n = 4$ otherwise

Normalized points are calculated for both point sets: $\hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$

Build the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ from the normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$

Calculate $\mathbf{C} = \mathbf{B}^{\dagger}\mathbf{A}$

Compute the SVD of the symmetric matrix $\mathbf{C}^T\mathbf{C}$: $\mathbf{C}^T\mathbf{C} = \mathbf{U\Sigma U}^T$, where $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$

if $\sigma_1 < 1$ **then**

    The two solutions are $\mathbf{y}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $\mathbf{y}_2 = \begin{bmatrix} -1 & 0 \end{bmatrix}^T$

else

    if $\sigma_2 > 1$ **then**

        The two solutions are $\mathbf{y}_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and $\mathbf{y}_2 = \begin{bmatrix} 0 & -1 \end{bmatrix}^T$

    else

        The solutions are:

$$\mathbf{y}_1 = \begin{bmatrix} \sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ \sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} \sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ -\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}, \mathbf{y}_3 = \begin{bmatrix} -\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ \sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}, \mathbf{y}_4 = \begin{bmatrix} -\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ -\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}$$

    end if

end if

Calculate $\mathbf{a}_i = \mathbf{U}\mathbf{y}_i$ and $\mathbf{b}_i = \mathbf{C}\mathbf{a}_i$

Construct the essential matrices $\mathbf{E}_i$ from $\mathbf{a}_i$ and $\mathbf{b}_i$

The method results in 4 solutions (2 if the solutions are recovered from a degenerate case). To find which one is the correct solution, we can test for all available correspondences whether the epipolar constraint 4.3 holds. Of course, it can happen that there are some wrong correspondences and noisy correspondences, so a robust method like the RANSAC algorithm [6] should be used. For the distance measure used in RANSAC, the epipolar distance

$$d(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}'_i) = \hat{\mathbf{u}}'^T_i \mathbf{E} \hat{\mathbf{u}}_i$$

or its symmetric version can be used. I will describe the RANSAC algorithm in detail in the section about planar surface detection, as it is an integral part of the algorithm used there.

4.3 Motorcycle motion

In this section I describe my work done in the essential matrix estimation when the motion of the cameras are constrained by the motorcycle motion constraint. First, I describe the model I used to describe the motion that cameras attached to motorbikes can undergo. Motorbikes move similarly to cars, so it seems logical to build upon the theory of planar motion. The difference between the two motion is that motorbikes can also be tilted by taking corners or by changing lanes. Mathematically this means that the cameras can also rotate around their optical axis, not just around their vertical axis. Also planar motion is a special case of motorcycle motion when the bike is perfectly straight on the ground. We sill see that the model is compatible with planar motion as the essential matrix will become the one described in the beginning of the section about planar motion as the angle of rotation around the optical axis tends to 0. Then, I describe a non-minimal method to solve for the essential matrix from 6 correspondences when the rotation around the optical axis is known for the first camera. For a sequence of images, the assumption of the known rotation around the optical axis for the first camera means that we need to know the angle only at the very first image. If the bike is straight in the fist image of the sequence, we can simply initialize the angle with 0. This angle can be estimated from the essential matrix for the images afterwards. Constraints found in the essential matrix should enable us to solve for it from less than 5 correspondences, but in the lack of time and the complexity of the problem, I was not able to do so far. Lastly if the rotation around the optical axis is known for both cameras, I propose a method to reduce the problem to planar motion and solve it using the circle-ellipse algorithm described in the previous section. The assumption of knowing the rotation around the optical axis for both cameras can be realized by having a gyroscope or an IMU sensor on the bike.

4.3.1 The essential matrix under motorcycle motion

We have discussed before that if the world coordinate system is fixed to the first camera's coordinate system and the rotation and translation between the cameras are $\mathbf{R}$ and $\mathbf{t}$, the essential matrix has the following form:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}.$$

The bike moves on the plane of the road like a car, so translation is the same as with planar motion:

$$\mathbf{t} = \rho \begin{bmatrix} \cos\gamma \\ 0 \\ \sin\gamma \end{bmatrix}.$$

However, with motorcycle motion, it can happen that the bike is tilted at the first camera, so the camera's XZ-plane is not parallel to the plane of the road. That means that if we want to use the same translation as we used with planar motion, the world coordinate system has to fixed to the position of the first camera, but its XZ-plane has to be parallel to the ground plane. Let $\mathbf{P} = \mathbf{K}\left[\begin{array}{c|c} \mathbf{R} & \mathbf{0} \end{array}\right]$ and $\mathbf{P}' = \mathbf{K}'\left[\begin{array}{c|c} \mathbf{R}' & \mathbf{t}' \end{array}\right]$ be the projection matrices of the first and second camera.

$$\mathbf{R} = \mathbf{R}_z = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the rotation matrix corresponding to the rotation around the optical axis (Z-axis) of the first camera. $\mathbf{R}' = \mathbf{R}'_z\mathbf{R}_y$ is the rotation of the second camera, were

$$\mathbf{R}'_z = \begin{bmatrix} \cos\beta' & -\sin\beta' & 0 \\ \sin\beta' & \cos\beta' & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{R}_y = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

is the rotation around the optical and vertical axis of the second camera and $\mathbf{t}' = \mathbf{R}'\mathbf{t}$, where

$$\mathbf{t} = \rho \begin{bmatrix} \cos\gamma \\ 0 \\ \sin\gamma \end{bmatrix}.$$

It can be shown that with these cameras, the essential matrix has the form:

$$\mathbf{E} = [\mathbf{t}']_\times \mathbf{R}'\mathbf{R}^T.$$

Given normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, $\hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i$, $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$, $i = 1..n$, the epipolar constraint is the following:

$$\hat{\mathbf{u}}_i'^T [\mathbf{t}']_\times \mathbf{R}'\mathbf{R}^T \hat{\mathbf{u}}_i = 0. \tag{4.8}$$

Note that by using the same translation $\mathbf{t}$ that we used for planar motion, we are assuming that the y-coordinate of the cameras do not change. This holds if the camera is at the center of rotation. When motorbikes are tilted, the center of rotation is at the wheels, so in order to not violate the translation constraint, the camera should be placed as close as possible to the ground. If it is above the center of rotation, translation $\mathbf{t}$ is only an approximation of the translation. Without this approximation, the problem becomes even more complex, so we will not consider it.

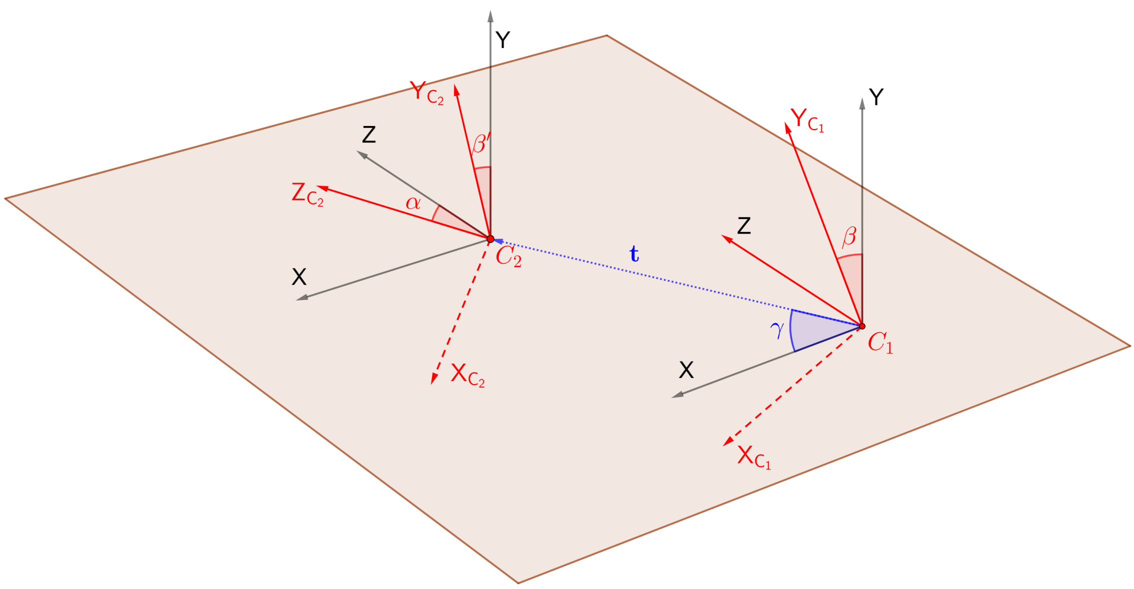

Figure 4.2: Motorcycle motion

4.3.2 Known angle β

From now on we will assume that β, the angle of rotation around the optical axis for the first camera, is known. This is equivalent to knowing what $\mathbf{R}^T$ is. The epipolar constraint 4.8 is equivalent to

$$\hat{\mathbf{u}}_i'^T[\mathbf{t}']_\times\mathbf{R}'(\mathbf{R}^T\hat{\mathbf{u}}_i) = 0.$$

$\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}^T\hat{\mathbf{u}}_i$ can be thought of as the normalized points rotated back, as if the first camera had not been rotated around its optical axis. Let $\mathbf{E}' = [\mathbf{t}']_\times\mathbf{R}'$, then the epipolar constraint 4.8 is the same as

$$\hat{\mathbf{u}}_i'^T\mathbf{E}'\hat{\mathbf{u}}_{\mathbf{R}_i} = 0.$$

Indeed $\mathbf{E}'$ is the essential matrix corresponding to the normalized projection matrices $\hat{\mathbf{P}}_\mathbf{R} = \mathbf{R}^T\left[\begin{array}{c|c}\mathbf{R} & \mathbf{0}\end{array}\right] = \left[\begin{array}{c|c}\mathbf{I} & \mathbf{0}\end{array}\right]$ and $\hat{\mathbf{P}}' = \left[\begin{array}{c|c}\mathbf{R}' & \mathbf{t}'\end{array}\right]$, where $\hat{\mathbf{P}}_\mathbf{R}$ can be thought of as the normalized projection matrix corresponding to first camera rotated back. The new problem is to estimate the essential matrix $\mathbf{E}'$ by the point correspondences $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}_i', i = 1..n$.

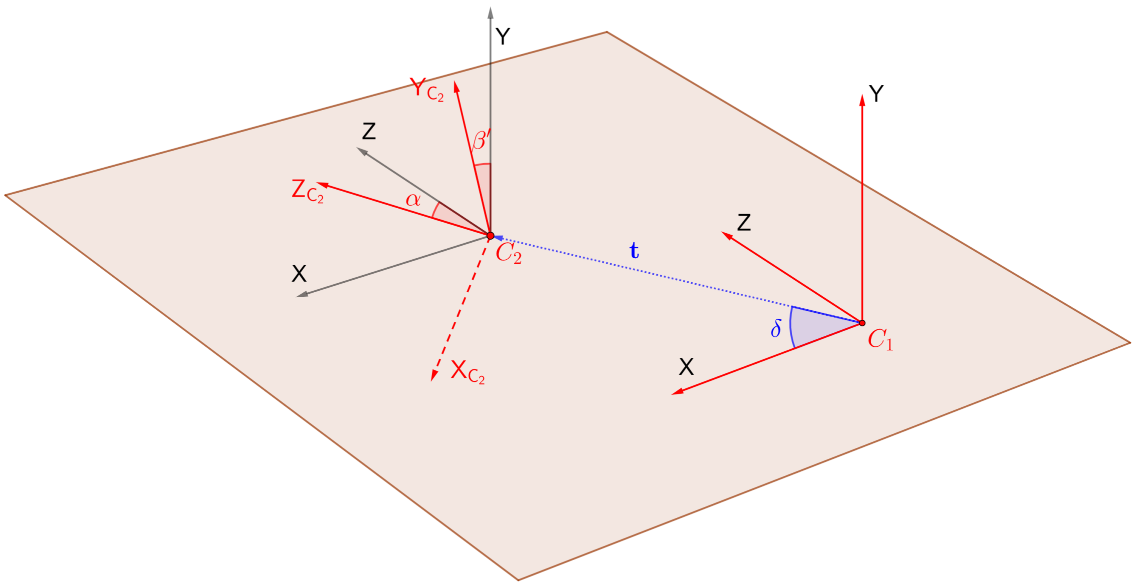

Figure 4.3: Motorcycle motion with the first camera rotated back

Now I will go into the details of how the essential matrix $\mathbf{E}'$ is composed of.

$$\mathbf{R}' = \mathbf{R}'_z\mathbf{R}_y = \begin{bmatrix} \cos\beta' & -\sin\beta' & 0 \\ \sin\beta' & \cos\beta' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\beta'\cos\alpha & -\sin\beta' & \cos\beta'\sin\alpha \\ \sin\beta'\cos\alpha & \cos\beta' & \sin\beta'\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

is the rotation of the second camera, and

$$\mathbf{t}' = \mathbf{R}'\mathbf{t} = \rho \begin{bmatrix} \cos\beta'\cos\alpha & -\sin\beta' & \cos\beta'\sin\alpha \\ \sin\beta'\cos\alpha & \cos\beta' & \sin\beta'\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\gamma \\ 0 \\ \sin\gamma \end{bmatrix} =$$

$$= \rho \begin{bmatrix} \cos\beta'\cos\alpha\cos\gamma + \cos\beta'\sin\alpha\sin\gamma \\ \sin\beta'\cos\alpha\cos\gamma + \sin\beta'\sin\alpha\sin\gamma \\ -\sin\alpha\cos\gamma + \cos\alpha\sin\gamma \end{bmatrix} = \rho \begin{bmatrix} \cos\beta'(\cos\alpha\cos\gamma + \sin\alpha\sin\gamma) \\ \sin\beta'(\cos\alpha\cos\gamma + \sin\alpha\sin\gamma) \\ -\sin\alpha\cos\gamma + \cos\alpha\sin\gamma \end{bmatrix} =$$

$$= \rho \begin{bmatrix} \cos\beta'\cos(\gamma-\alpha) \\ \sin\beta'\cos(\gamma-\alpha) \\ \sin(\gamma-\alpha) \end{bmatrix}$$

is the translation of the second camera. The cross-product matrix $[\mathbf{t}']_\times$ composed of the elements of $\mathbf{t}'$ is

$$[\mathbf{t}']_\times = \rho \begin{bmatrix} 0 & -\sin(\gamma-\alpha) & \sin\beta'\cos(\gamma-\alpha) \\ \sin(\gamma-\alpha) & 0 & -\cos\beta'\cos(\gamma-\alpha) \\ -\sin\beta'\cos(\gamma-\alpha) & \cos\beta'\cos(\gamma-\alpha) & 0 \end{bmatrix}.$$

Finally the essential matrix $\mathbf{E}'$ is

$$\mathbf{E}' = [\mathbf{t}']_\times\mathbf{R}' = \rho \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix},$$

where

$$\begin{aligned}
e_1 &= -\sin(\gamma-\alpha)\sin\beta'\cos\alpha - \sin\beta'\cos(\gamma-\alpha)\sin\alpha\\
&= -\sin\beta'[\sin(\gamma-\alpha)\cos\alpha + \cos(\gamma-\alpha)\sin\alpha]\\
&= -\sin\beta'\sin(\gamma-\alpha+\alpha) = -\sin\beta'\sin\gamma;\\
e_2 &= -\sin(\gamma-\alpha)\cos\beta';\\
e_3 &= -\sin(\gamma-\alpha)\sin\beta'\sin\alpha + \sin\beta'\cos(\gamma-\alpha)\cos\alpha\\
&= \sin\beta'[\cos(\gamma-\alpha)\cos\alpha - \sin(\gamma-\alpha)\sin\alpha]\\
&= \sin\beta'\cos(\gamma-\alpha+\alpha) = \sin\beta'\cos\gamma;\\
e_4 &= \sin(\gamma-\alpha)\cos\beta'\cos\alpha + \cos\beta'\cos(\gamma-\alpha)\sin\alpha\\
&= \cos\beta'[\sin(\gamma-\alpha)\cos\alpha + \cos(\gamma-\alpha)\sin\alpha]\\
&= \cos\beta'\sin(\gamma-\alpha+\alpha) = \cos\beta'\sin\gamma;
\end{aligned}$$

$$\begin{aligned}
e_5 &= -\sin(\gamma-\alpha)\sin\beta';\\
e_6 &= \sin(\gamma-\alpha)\cos\beta'\sin\alpha - \cos\beta'\cos(\gamma-\alpha)\cos\alpha\\
&= -\cos\beta'[\cos(\gamma-\alpha)\cos\alpha - \sin(\gamma-\alpha)\sin\alpha]\\
&= -\cos\beta'\cos(\gamma-\alpha+\alpha) = -\cos\beta'\cos\gamma;\\
e_7 &= -\sin\beta'\cos(\gamma-\alpha)\cos\beta'\cos\alpha\\
&\quad + \cos\beta'\cos(\gamma-\alpha)\sin\beta'\cos\alpha = 0;\\
e_8 &= \sin\beta'\cos(\gamma-\alpha)\sin\beta' + \cos\beta'\cos(\gamma-\alpha)\cos\beta'\\
&= \sin^2\beta'\cos(\gamma-\alpha) + \cos^2\beta'\cos(\gamma-\alpha)\\
&= (\sin^2\beta' + \cos^2\beta')\cos(\gamma-\alpha) = \cos(\gamma-\alpha);\\
e_9 &= -\sin\beta'\cos(\gamma-\alpha)\cos\beta'\sin\alpha\\
&\quad + \cos\beta'\cos(\gamma-\alpha)\sin\beta'\sin\alpha = 0.
\end{aligned}$$

To summarize, after the simplifications, the essential matrix turns out to be

$$\mathbf{E}' = \rho \begin{bmatrix} -\sin\beta'\sin\gamma & -\sin(\gamma-\alpha)\cos\beta' & \sin\beta'\cos\gamma \\ \cos\beta'\sin\gamma & -\sin(\gamma-\alpha)\sin\beta' & -\cos\beta'\cos\gamma \\ 0 & \cos(\gamma-\alpha) & 0 \end{bmatrix}. \tag{4.9}$$

If the motorbike is not tilted in the second image, then $\beta' = 0$, $\cos\beta' = 1$, $\sin\beta' = 0$

and the essential matrix 4.9 becomes

$$\mathbf{E}' = \rho \begin{bmatrix} 0 & -\sin(\gamma-\alpha) & 0 \\ \sin\gamma & 0 & -\cos\gamma \\ 0 & \cos(\gamma-\alpha) & 0 \end{bmatrix},$$

which is the essential matrix corresponding to the relative motion constrained to planar motion with cameras $\mathbf{K}\left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array}\right]$ and $\mathbf{K}'\left[\begin{array}{c|c} \mathbf{R}_y & \mathbf{R}_y\mathbf{t} \end{array}\right]$.

Certain constraints can be derived for the elements of essential matrix 4.9. Denote the elements of 4.9 as

$$\mathbf{E}' = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ 0 & e_7 & 0 \end{bmatrix}.$$

Then

$$\frac{e_1}{e_4} = \frac{-\rho\sin\beta'\sin\gamma}{\rho\cos\beta'\sin\gamma} = -\tan\beta' \implies e_1 = -\tan\beta' e_4,$$
$$\frac{e_3}{e_6} = \frac{\rho\sin\beta'\cos\gamma}{-\rho\cos\beta'\cos\gamma} = -\tan\beta' \implies e_3 = -\tan\beta' e_6,$$

and

$$\frac{e_5}{e_2} = \frac{-\rho\sin(\gamma-\alpha)\sin\beta'}{-\rho\sin(\gamma-\alpha)\cos\beta'} = \tan\beta' \implies e_5 = \tan\beta' e_2.$$

If the notation $\delta = -\tan\beta'$ is used, then $e_1 = \delta e_4$, $e_3 = \delta e_6$ and $e_5 = -\delta e_2$ and the essential matrix 4.9 can be written as

$$\mathbf{E}' = \begin{bmatrix} \delta e_4 & e_2 & \delta e_6 \\ e_4 & -\delta e_2 & e_6 \\ 0 & e_7 & 0 \end{bmatrix},$$

where the unknowns are e_2, e_4, e_6, e_7 and δ. Another constraint can be derived from the squared sum of the elements. Since

$$e_1^2 + e_4^2 = \rho^2\sin^2\beta'\sin^2\gamma + \rho^2\cos^2\beta'\sin^2\gamma = \rho^2\sin^2\gamma(\sin^2\beta' + \cos^2\beta') = \rho^2\sin^2\gamma$$

and

$$e_3^2 + e_6^2 = \rho^2\sin^2\beta'\cos^2\gamma + \rho^2\cos^2\beta'\cos^2\gamma = \rho^2\cos^2\gamma(\sin^2\beta' + \cos^2\beta') = \rho^2\cos^2\gamma,$$

implies that

$$e_1^2 + e_4^2 + e_3^2 + e_6^2 = \rho^2\sin^2\gamma + \rho^2\cos^2\gamma = \rho^2(\sin^2\gamma + \cos^2\gamma) = \rho^2.$$

Also

$$e_2^2 + e_5^2 = \rho^2 \sin^2(\gamma - \alpha)\cos^2\beta' + \rho^2 \sin^2(\gamma - \alpha)\sin^2\beta' =$$

$$= \rho^2 \sin^2(\gamma - \alpha)(\sin^2\beta' + \cos^2\beta') = \rho^2 \sin^2(\gamma - \alpha),$$

so

$$e_2^2 + e_5^2 + e_7^2 = \rho^2 \sin^2(\gamma - \alpha) + \rho^2 \cos^2(\gamma - \alpha) = \rho^2(\sin^2(\gamma - \alpha) + \cos^2(\gamma - \alpha)) = \rho^2.$$

The constraint derived is then

$$e_1^2 + e_4^2 + e_3^2 + e_6^2 = e_2^2 + e_5^2 + e_7^2.$$

So far, I was not able to make use the constraints derived here to estimate $\mathbf{E}'$ from less than 6 correspondences. For 6 correspondences a solver can be derived based on the same idea as the 8-point algorithm.

6-point algorithm

Let the point correspondences be $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$, $i = 1..n$, $n \geq 6$ and the the calibration matrices be $\mathbf{K}$ and $\mathbf{K}'$. The angle of rotation around the optical axis β is assumed to be known for the first camera. Then the rotation matrix $\mathbf{R}_z$ can be constructed. Let normalized correspondences be $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_i$, where $\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}_z^T \mathbf{K}^{-1} \mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1} \mathbf{u}'_i$. The base equation is

$$\hat{\mathbf{u}}_i'^T \mathbf{E}' \hat{\mathbf{u}}_{\mathbf{R}_i} = 0. \tag{4.10}$$

Denote the vectors $\hat{\mathbf{u}}_{\mathbf{R}_i}$ and $\hat{\mathbf{u}}'_i$ as $\hat{\mathbf{u}}_{\mathbf{R}_i} = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$ and $\hat{\mathbf{u}}'_i = \begin{bmatrix} u'_i & v'_i & 1 \end{bmatrix}^T$ and the elements of $\mathbf{E}'$ as

$$\mathbf{E}' = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ 0 & e_7 & 0 \end{bmatrix}.$$

Then equation 4.10 can be written as

$$u'_i u_i e_1 + u'_i v_i e_2 + u'_i e_3 + v'_i u_i e_4 + v'_i v_i e_5 + v'_i e_6 + v_i e_7 = 0.$$

If the notation $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{bmatrix}^T$ is used, then the equation above can be written as

$$\mathbf{A}_i \mathbf{e} = \begin{bmatrix} u'_i u_i & u'_i v_i & u'_i & v'_i u_i & v'_i v_i & v'_i & v_i \end{bmatrix} \mathbf{e} = \mathbf{0}.$$

$\mathbf{E}'$ has 7 elements, but because of the scale ambiguity, 6 correspondences are enough to estimate $\mathbf{e}$. The coefficient matrix $\mathbf{A}$ is then constructed as $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \dots & \mathbf{A}_n \end{bmatrix}^T$, $n \geq 6$. If we constrain $\mathbf{e}$ to be $\mathbf{e}^T\mathbf{e} = 1$, then the optimal solution is the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to its smallest eigenvalue.

Algorithm 3 6-point algorithm

Input: Point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 6$, calibration matrices $\mathbf{K}$ and $\mathbf{K}'$ and the angle of rotation around the optical axis for the first camera β

Output: Essential matrix $\mathbf{E}'$

1: Construct $\mathbf{R}_z$ from β

2: Normalized correspondences are calculated $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_i$, where $\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}_z^T\mathbf{K}^{-1}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$

3: Build the coefficient matrix $\mathbf{A}$ from the normalized point correspondences $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_i$

4: Find $\mathbf{e}$ as the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue

5: Construct $\mathbf{E}'$ from $\mathbf{e}$

4.3.3 Known angles β and β'

I propose another solution for the case when the rotation around the optical axis can be retrieved for both cameras. This assumption cannot be realized with an initialization as the last one. A sensor is needed that can give us the tilt of the bike at every image, like a gyroscope or IMU sensor. The idea is that if we know the the rotations around the optical axis for both cameras, then the points can be rotated back on both pictures to give us a setup corresponding to planar motion.

Let the projection matrices of the cameras be $\mathbf{P} = \mathbf{K}\left[\begin{array}{c|c} \mathbf{R} & \mathbf{0} \end{array}\right]$ and $\mathbf{P}' = \mathbf{K}'\left[\begin{array}{c|c} \mathbf{R}' & \mathbf{t}' \end{array}\right]$, where $\mathbf{R} = \mathbf{R}_z$, $\mathbf{R}' = \mathbf{R}'_z\mathbf{R}_y$ and $\mathbf{t}' = \mathbf{R}'_z\mathbf{R}_y\mathbf{t}$. Let $\mathbf{X}$ be a world point. The coordinates of $\mathbf{X}$ in the first camera's coordinate system is $\mathbf{X}_1 = \mathbf{R}_z\mathbf{X}$, so $\mathbf{R}_z^T\mathbf{X}_1 = \mathbf{X}$. The coordinates of $\mathbf{X}$ in the second camera's coordinate system is $\mathbf{X}_2 = \mathbf{R}'_z\mathbf{R}_y\mathbf{X} + \mathbf{R}'_z\mathbf{R}_y\mathbf{t}$. Then by $\mathbf{X}_1 = \mathbf{R}_z\mathbf{X}$,

$$\mathbf{X}_2 = \mathbf{R}'_z\mathbf{R}_y\mathbf{R}_z^T\mathbf{X}_1 + \mathbf{R}'_z\mathbf{R}_y\mathbf{t}$$

gives the relation between $\mathbf{X}_1$ and $\mathbf{X}_2$. Since the angles β and β' are known, $\mathbf{R}_z$ and

$\mathbf{R}'_z$ are known. Multiplying the above equation with $\mathbf{R}'^T_z$ from the left results in

$$\mathbf{R}'^T_z \mathbf{X}_2 = \mathbf{R}_y \mathbf{R}^T_z \mathbf{X}_1 + \mathbf{R}_y \mathbf{t}.$$

Let $\hat{\mathbf{X}}_1 = \mathbf{R}^T_z \mathbf{X}_1$ and $\hat{\mathbf{X}}_2 = \mathbf{R}'^T_z \mathbf{X}_2$, then the equation above is

$$\hat{\mathbf{X}}_2 = \mathbf{R}_y \hat{\mathbf{X}}_1 + \mathbf{R}_y \mathbf{t}.$$

$\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ is the world point $\mathbf{X}$ in the coordinate system of the normalized cameras $\hat{\mathbf{P}}_\mathbf{R} = \mathbf{R}^T_z \left[\begin{array}{c|c} \mathbf{R_z} & \mathbf{0} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array}\right]$ and $\hat{\mathbf{P}}'_\mathbf{R} = \mathbf{R}'^T_z \left[\begin{array}{c|c} \mathbf{R}'_z \mathbf{R}_y & \mathbf{R}'_z \mathbf{R}_y \mathbf{t} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{R}_y & \mathbf{R}_y \mathbf{t} \end{array}\right]$.

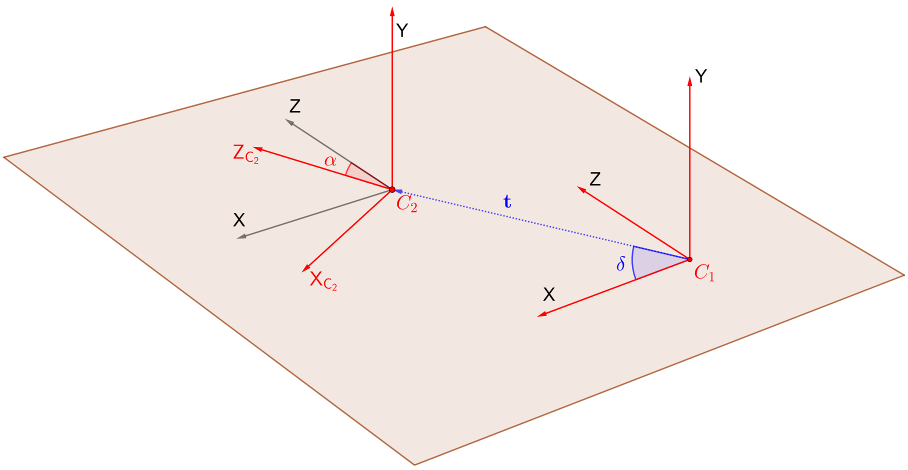

Figure 4.4: Both cameras rotated back results in planar motion

The relative motion between these cameras ($\mathbf{R}_y$, $\mathbf{R}_y \mathbf{t}$) comply with the planar motion constraint and the essential matrix $\mathbf{E}$ is

$$\mathbf{E} = [\mathbf{R}_y \mathbf{t}]_\times \mathbf{R}_y = \rho \begin{bmatrix} 0 & -\sin(\gamma - \alpha) & 0 \\ \sin\gamma & 0 & -\cos\gamma \\ 0 & \cos(\gamma - \alpha) & 0 \end{bmatrix}. \tag{4.11}$$

Let $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$, $i = 1..n$ be the point correspondences between the original cameras $\mathbf{P} = \mathbf{K} \left[\begin{array}{c|c} \mathbf{R} & \mathbf{0} \end{array}\right]$ and $\mathbf{P}' = \mathbf{K}' \left[\begin{array}{c|c} \mathbf{R}' & \mathbf{t}' \end{array}\right]$. Let the normalized and rotated points be $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_{\mathbf{R}_i}$, where $\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}^T_z \mathbf{K}^{-1} \mathbf{u}_i$ and $\hat{\mathbf{u}}'_{\mathbf{R}_i} = \mathbf{R}'^T_z \mathbf{K}'^{-1} \mathbf{u}'_i$. Then the following holds:

$$\hat{\mathbf{u}}'^T_{\mathbf{R}_i} \mathbf{E} \hat{\mathbf{u}}_{\mathbf{R}_i} = 0,$$

where $\mathbf{E}$ is the essential matrix defined in equation 4.11. Solving for $\mathbf{E}$ can be done by any algorithm used for the essential matrix estimation under planar motion, including the circle-ellipse method.

Algorithm 4 Motorcycle motion for known rotations around the optical axis

Input: Point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i, i = 1..n, n \geq 2$, calibration matrices $\mathbf{K}$ and $\mathbf{K}'$ and the angles of rotation around the optical axis for the first and second camera β, β'

Output: Unknown parameters α, γ

1: Construct $\mathbf{R}_z$ from β and $\mathbf{R}'_z$ from β'

2: Normalized correspondences are calculated $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_{\mathbf{R}_i}$, where $\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}_z^T \mathbf{K}^{-1} \mathbf{u}_i$ and $\hat{\mathbf{u}}'_{\mathbf{R_i}} = \mathbf{R'}_z^T \mathbf{K'}^{-1} \mathbf{u}'_i$

3: Calculate $\mathbf{E} = \begin{bmatrix} 0 & e_1 & 0 \\ e_2 & 0 & e_3 \\ 0 & e_4 & 0 \end{bmatrix}$ with the circle ellipse method with correspondences $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_{\mathbf{R}_i}$

4: γ can be calculated as $\mathrm{atan2}(e_2, -e_3)$

5: α can be calculated as $-\mathrm{atan2}(-e_1, e_4) + \gamma$

4.4 Results

Synthetic testing was done to check the essential matrix properties and to compare the 6-point algorithm with the 8-point algorithm. The algorithms were implemented in Matlab and random points were generated inside a unit sphere and two virtual cameras are generated with projection matrices $\mathbf{P} = \mathbf{K} \left[\begin{array}{c|c} \mathbf{R}_z & \mathbf{0} \end{array} \right]$ and $\mathbf{P}' = \mathbf{K}' \left[\begin{array}{c|c} \mathbf{R}'_z \mathbf{R}_y & \mathbf{R}'_z \mathbf{R}_y \mathbf{t} \end{array} \right]$. The relative pose between the cameras was done so both of them face the sphere and the first camera is rotated around the optical axis by 15 degrees and second camera is rotated around the optical axis by 30 degrees. Point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$ are obtained by projecting the points using $\mathbf{P}$ and $\mathbf{P}'$. Gaussian noise is then added to the image plane coordinates with varying variance. The first virtual camera's points are normalized and rotated back: $\hat{\mathbf{u}}_{\mathbf{R}_i} = \mathbf{R}_z^T \mathbf{K}^{-1} \mathbf{u}_i$ and the second camera's points are normalized: $\hat{\mathbf{u}}'_i = \mathbf{K'}^{-1} \mathbf{u}'_i$. Then the essential matrix $\mathbf{E}'$ is estimated from the correspondences $\hat{\mathbf{u}}_{\mathbf{R}_i} \leftrightarrow \hat{\mathbf{u}}'_i$ with the 6- and 8-point algorithms. The resulted essential matrices are then decomposed into rotation and translation and compared to the ground truth rotation and translations. Angular difference is measured between the rotation matrices the following way:

$$\epsilon_{\mathbf{R}} = \cos^{-1}\left((trace(\mathbf{R}\hat{\mathbf{R}}^T) - 1)/2\right),$$

where $\mathbf{R}$ is the ground truth rotation and $\hat{\mathbf{R}}$ is the estimated rotation.

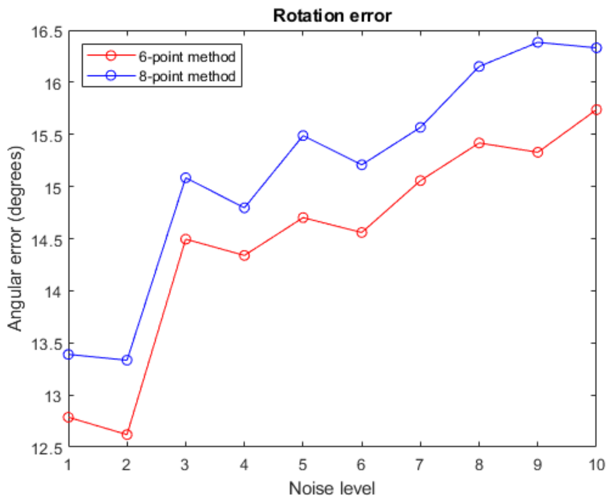

Figure 4.5: Rotation error

For the translations, directional similarity is measured the following way:

$$\epsilon_{\mathbf{t}} = \cos^{-1}\left((\mathbf{t}^T\hat{\mathbf{t}})/(||\mathbf{t}||_2||\hat{\mathbf{t}}||_2)\right),$$

where $\mathbf{t}$ is the ground truth translation and $\hat{\mathbf{t}}$ is the estimated translation. The variance of the noise is varied from 1 to 10 and each data point represents the average of 10000 measurements with that noise level.

Figure 4.6: Translation error

Chapter 5

Planar surface detection

In this chapter, I describe the plane detection algorithm I worked on. The algorithm detects planar surfaces from point correspondences obtained from a stereo image pair. Let the projection matrices of the cameras be $\mathbf{P} = \mathbf{K}\left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array}\right]$ and $\mathbf{P}' = \mathbf{K}'\left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right]$. The cameras are assumed to be calibrated and their calibration matrices are $\mathbf{K}$ and $\mathbf{K}'$. The relative pose (rotation and translation) between the cameras is $\mathbf{R}$ and $\mathbf{t}$. The cameras are attached to a car, so planar motion is assumed to hold:

$$\mathbf{R} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \rho \begin{bmatrix} \cos\beta \\ 0 \\ \sin\beta \end{bmatrix}.$$

The algorithm works on point correspondences between the two images. Let the correspondences be $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$, $i = 1..n$. For calibrated cameras, the normalized points can be calculated as $\hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$. If the points $\mathbf{X}_j$, $j = 1..k$, lie on the same 3D plane, and $\mathbf{u}_j$, $\mathbf{u}'_j$ are the projections of these points onto the first and second cameras' image plane, then there exist a homography $\mathbf{H} \in \mathbb{R}^{3\times3}$, such that

$$\mathbf{u}'_j \sim \mathbf{H}\mathbf{u}_j, \; j = 1..k.$$

The algorithm has two parts:

- Estimate $\mathbf{H}$ from point correspondences.
- Find all points $\mathbf{u}_j$, for which $\mathbf{u}'_j \sim \mathbf{H}\mathbf{u}_j$.

The imaged scene can contain multiple planar surfaces, so finding the homography corresponding to the surface we are interested in is also a part of the problem. Any planar surface can be represented by the normal vector of the surface and a point on the surface. We cannot know which points lie on the plane we are looking for in advance, so we can only estimate a plane from a family of planes characterized by their normal vector. Also, for a fixed normal vector a family of homographies can be obtained the following way [1]:

$$\mathbf{H} \sim \mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{n}^T, \tag{5.1}$$

where $(\mathbf{R}, \mathbf{t})$ is the relative pose between the cameras, d is the distance of the first camera center from the closest point on the plane and $\mathbf{n}$ is the normal vector of the plane. The distance d is unknown, as well as the scale of the translation ρ, and cannot be estimated from point correspondences only. There are 4 kind of families of planes considered here:

- Planes with a normal vector $\mathbf{n} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$, i.e. planes parallel to the ground plane.
- Planes with a normal vector $\mathbf{n} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$, i.e. planes parallel to the first camera's YZ-plane. A typical example of this kind of plane is the plane of the wall of the houses at the side of the road.
- Planes with a normal vector $\mathbf{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, i.e. planes in the front and facing the camera.
- Planes with a normal vector $\mathbf{n} = \begin{bmatrix} \cos\delta & 0 & \sin\delta \end{bmatrix}^T$, i.e. a general vertical plane with a normal characterized by an angle δ. The two cases above are special cases of this one, with $\delta = 0$ and $\delta = \pi/2$.

In the first three cases we can decide the normal of the plane we want to find and in the last case we only constrain the normal to be horizontal. The concrete normal of the found plane can be obtained by decomposing the homography.

5.1 Homography estimation

This general method for homography estimation is based on the Computer Vision Course presentations [5]. Given k point correspondences $\mathbf{u}_j \leftrightarrow \mathbf{u}'_j$, $j = 1..k$, the base equation is

$$\mathbf{u}'_j \sim \mathbf{H}\mathbf{u}_j.$$

Let the coordinates for a correspondence be $\mathbf{u}_j = \begin{bmatrix} u_j & v_j & 1 \end{bmatrix}^T$ and $\mathbf{u}'_j = \begin{bmatrix} u'_j & v'_j & 1 \end{bmatrix}^T$. Then based on the equation above, the following holds:

$$\begin{bmatrix} u'_j \\ v'_j \\ 1 \end{bmatrix} \sim \begin{bmatrix} \alpha u'_j \\ \alpha v'_j \\ \alpha \end{bmatrix} = \begin{bmatrix} h_1 u_j + h_2 v_j + h_3 \\ h_4 u_j + h_5 v_j + h_6 \\ h_7 u_j + h_8 v_j + h_9 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix},$$

where α is the unknown scale. Homogeneous division (division by α) yields the image plane coordinates $\mathbf{u}'_j$. Since $\alpha = h_7 u_j + h_8 v_j + h_9$, after homogeneous division

$$u'_j = \frac{h_1 u_j + h_2 v_j + h_3}{h_7 u_j + h_8 v_j + h_9}$$

and

$$v'_j = \frac{h_4 u_j + h_5 v_j + h_6}{h_7 u_j + h_8 v_j + h_9}.$$

After multiplying by the denominator and subtracting the left side, we get:

$$h_1 u_j + h_2 v_j + h_3 - h_7 u_j u'_j - h_8 v_j u'_j - h_9 u'_j = 0 \tag{5.2}$$

and

$$h_4 u_j + h_5 v_j + h_6 - h_7 u_j v'_j - h_8 v_j v'_j - h_9 v'_j = 0. \tag{5.3}$$

If the notation $\mathbf{h} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{bmatrix}^T$ is used, for a single correspondence $\mathbf{u}_j \leftrightarrow \mathbf{u}'_j$, we get the following homogeneous equation system:

$$\mathbf{A}_j \mathbf{h} = \begin{bmatrix} u_j & v_j & 1 & 0 & 0 & 0 & -u_j u'_j & -v_j u'_j & -u'_j \\ 0 & 0 & 0 & u_j & v_j & 1 & -u_j v'_j & -v_j v'_j & -v'_j \end{bmatrix} \mathbf{h} = 0.$$

$\mathbf{H}$ has 9 elements, but because of the scale ambiguity the rank of $\mathbf{H}$ is 8. A single correspondence gives two equations, so at least 4 points are required for the estimation. The solution is the same as the one for the fundamental matrix estimation. The eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$ gives the optimal solution. The image plane coordinates are not normalized, so to make the estimation reliable, data normalization should be carried out. The affine transformations $\mathbf{T}$ and $\mathbf{T}'$ normalize the points and the homography $\mathbf{H}'$ is estimated from the normalized points. The homography corresponding to the original correspondences can be recovered as

$$\mathbf{H} = \mathbf{T}'^{-1}\mathbf{H}'\mathbf{T}.$$

5.2 Special homographies induced by planar motion

Now we will look at how the homographies are composed of and how to solve the estimation optimally if the form of the rotation $\mathbf{R}$, translation $\mathbf{t}$ and the normal vector of the plane we are interested in $\mathbf{n}$ is known. The equation 5.1 for homographies is valid if normalized coordinates $\hat{\mathbf{u}}_j$, $\hat{\mathbf{u}}'_j$ are used:

$$\hat{\mathbf{u}}'_j \sim (\mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{n}^T)\hat{\mathbf{u}}_j. \tag{5.4}$$

Planar motion is assumed, so

$$\mathbf{R} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \rho \begin{bmatrix} \cos\beta \\ 0 \\ \sin\beta \end{bmatrix}$$

and we are interested in four cases for the normal vector $\mathbf{n}$:

$$\mathbf{n}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{n}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{n}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{n}_4 = \begin{bmatrix} \cos\delta \\ 0 \\ \sin\delta \end{bmatrix}.$$

The substitution into equation 5.4 is discussed for each normal vector $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$, $\mathbf{n}_4$ seperately.

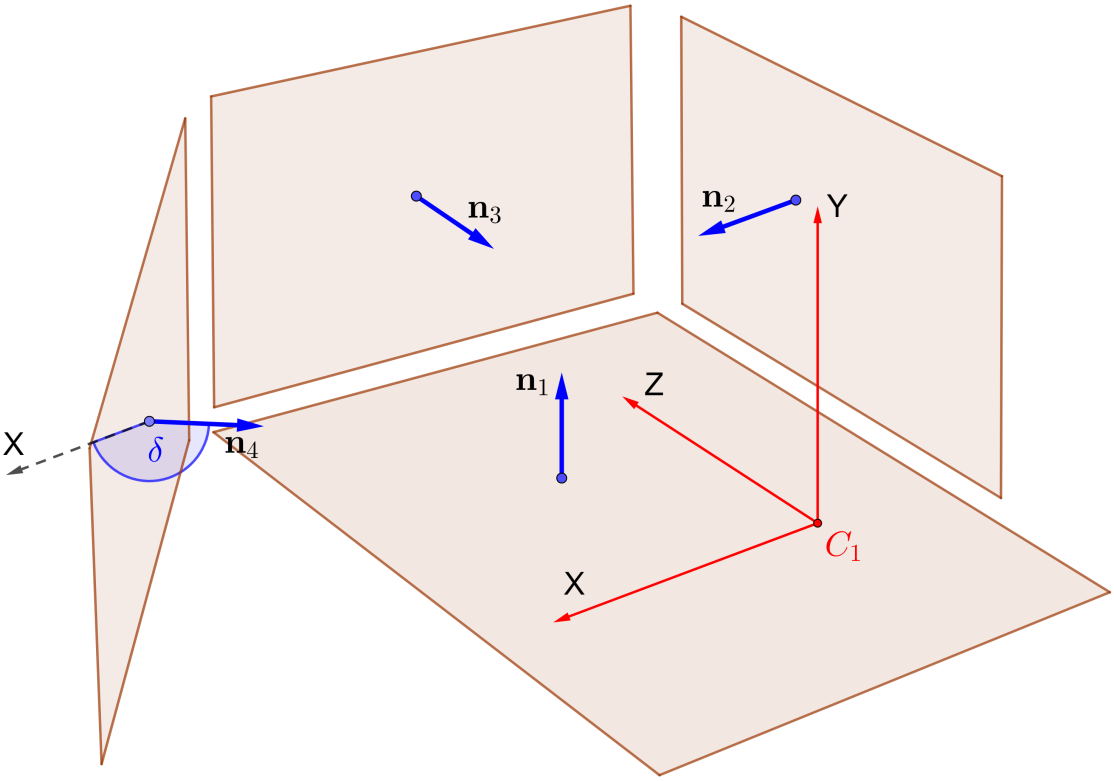

Figure 5.1: The four kind of planes considered

5.2.1 Homography for the vertical normal $\mathbf{n}_1$

Substituting into the equation 5.4 with the normal $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ gives:

$$\mathbf{H}_1 \sim \mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{n}_1^T = \begin{bmatrix}\cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha\end{bmatrix} - \frac{\rho}{d}\begin{bmatrix}\cos\beta \\ 0 \\ \sin\beta\end{bmatrix}\begin{bmatrix}0 & 1 & 0\end{bmatrix} =$$

$$= \begin{bmatrix}\cos\alpha & -\frac{\rho}{d}\cos\beta & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & -\frac{\rho}{d}\sin\beta & \cos\alpha\end{bmatrix}.$$

Let $p = -\frac{\rho}{d}\cos\beta$ and $q = -\frac{\rho}{d}\sin\beta$, then substituting into equation 5.2 with $h_1 = \cos\alpha$, $h_2 = p$, $h_3 = \sin\alpha$, $h_4 = 0$, $h_5 = 1$, $h_6 = 0$, $h_7 = -\sin\alpha$, $h_8 = q$, $h_9 = \cos\alpha$ results in:

$$(u_j - u'_j)\cos\alpha + v_j p + (1 + u'_j u_j)\sin\alpha - u'_j v_j q = 0$$

and into 5.3 results in:

$$v_j + v'_j u_j \sin\alpha - v'_j v_j q - v'_j \cos\alpha = 0.$$

In the latter equation the first term v_j is not multiplied by any of the unknowns, so we can rearrange the equation by moving all the other terms to the other side:

$$v_j = -v'_j u_j \sin\alpha + v'_j v_j q + v'_j \cos\alpha.$$

Let $\mathbf{x} = \begin{bmatrix}\cos\alpha & \sin\alpha & p & q\end{bmatrix}^T$, then for a single correspondence $\hat{\mathbf{u}}_j \leftrightarrow \hat{\mathbf{u}}'_j$, the following inhomogeneous linear system is obtained:

$$\mathbf{A}_j\mathbf{x} = \begin{bmatrix} u_j - u'_j & 1 + u'_j u_j & v_j & -u'_j v_j \\ v'_j & -v'_j u_j & 0 & v'_j v_j \end{bmatrix} \begin{bmatrix}\cos\alpha \\ \sin\alpha \\ p \\ q\end{bmatrix} = \begin{bmatrix} 0 \\ v_j \end{bmatrix} = \mathbf{b}$$

$\mathbf{H}_1$ has 3 free parameters: α, p and q. A correspondence gives 2 equations, so at least 2 correspondences needed for the estimation. Let $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \dots & \mathbf{A}_k\end{bmatrix}^T$ be the matrix composed of the coefficient matrices $\mathbf{A}_j$ corresponding to the point correspondence $\hat{\mathbf{u}}_j \leftrightarrow \hat{\mathbf{u}}'_j$. A constraint can be derived from the first two elements of $\mathbf{x}$:

$$x_1^2 + x_2^2 = \cos^2\alpha + \sin^2\alpha = 1.$$

The problem is to solve the inhomogeneous equation system $\mathbf{Ax} = \mathbf{b}$, with the constraint $x_1^2 + x_2^2 = 1$. My supervisor, Levente Hajder was able to prove that this problem can be solved optimally in the least squares sense by calculating the intersection between two conics. I want to thank him for giving me the source code of the solver and that he allowed me to use it in my thesis.

5.2.2 Homography for the sideways horizontal normal $\mathbf{n}_2$

Substituting into the equation 5.1 with the normal $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$ results in

$$\mathbf{H}_2 \sim \begin{bmatrix} \cos\alpha - p & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha - q & 0 & \cos\alpha \end{bmatrix},$$

where $p = \frac{\varrho}{d}\cos\beta$ and $q = \frac{\varrho}{d}\sin\beta$. Analogous to the previous case, substituting the elements of $\mathbf{H}_2$ into the equation 5.2 results in the equation

$$(u_j - u'_j)\cos\alpha + (1 + u_j u'_j)\sin\alpha - u_j p + u'_j u_j q = 0$$

and into the equation 5.3 results in the equation

$$v_j = v'_j \cos\alpha - v'_j u_j \sin\alpha - v'_j u_j q.$$

Using the notation $\mathbf{x} = \begin{bmatrix}\cos\alpha & \sin\alpha & p & q\end{bmatrix}^T$, the following linear system of equations is obtained:

$$\mathbf{A}_j\mathbf{x} = \begin{bmatrix} u_j - u'_j & 1 + u_j u'_j & -u_j & u'_j u_j \\ v'_j & -v'_j u_j & 0 & -v'_j u_j \end{bmatrix} \begin{bmatrix}\cos\alpha \\ \sin\alpha \\ p \\ q\end{bmatrix} = \begin{bmatrix} 0 \\ v_j \end{bmatrix} = \mathbf{b}.$$

The homography $\mathbf{H}_2$ is different than $\mathbf{H}_1$, but the problem is the same in the sense that the unknowns are α, p and q and the vectors $\mathbf{x}$ and $\mathbf{b}$ are the same as before. We have to solve the inhomogeneous linear system $\mathbf{Ax} = \mathbf{b}$, with the constraint $x_1^2 + x_2^2 = 1$. The optimal solution is calculated by the intersection between two conics.

5.2.3 Homography for the frontal horizontal normal $\mathbf{n}_3$

Substituting into the equation 5.1 with the normal $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ results in

$$\mathbf{H}_3 \sim \begin{bmatrix} \cos\alpha & 0 & \sin\alpha - p \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha - q \end{bmatrix},$$

where $p = \frac{\varrho}{d}\cos\beta$ and $q = \frac{\varrho}{d}\sin\beta$. Analogous to the first two cases, substituting the elements of $\mathbf{H}_3$ into the equation 5.2 results in the equation

$$(u_j - u'_j)\cos\alpha + (1 + u_j u'_j)\sin\alpha - p + u'_j q = 0$$

and into the equation 5.3 results in the equation

$$v_j = v'_j \cos\alpha - v'_j u_j \sin\alpha - v'_j q.$$

Using the notation $\mathbf{x} = \begin{bmatrix} \cos\alpha & \sin\alpha & p & q \end{bmatrix}^T$, the following linear system of equations is obtained:

$$\mathbf{A}_j\mathbf{x} = \begin{bmatrix} u_j - u'_j & 1 + u_j u'_j & -1 & u'_j \\ v'_j & -v'_j u_j & 0 & -v'_j \end{bmatrix} \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ p \\ q \end{bmatrix} = \begin{bmatrix} 0 \\ v_j \end{bmatrix} = \mathbf{b}.$$

The solution is derived in the same exact way as for the previous two cases. The solution to the inhomogeneous linear system $\mathbf{Ax} = \mathbf{b}$ with the constraint $x_1^2 + x_2^2 = 1$ is calculated as the intersection between two conics.

The solution for the homography estimation for the normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$ is the same, the only thing different is the coefficient matrix $\mathbf{A}$.

Algorithm 5 Homography estimation for normal vector $\mathbf{n}$

Input: Normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, $i = 1..n$, $n \geq 2$ and mode $\in \{1, 2, 3\}$

Output: Homography $\mathbf{H}$

```
if mode == 1 then
    Construct the coefficient matrix A from the normalized correspondences û_i ↔ û'_i corresponding to the normal vector n_1 = [0 1 0]^T
end if
if mode == 2 then
    Construct the coefficient matrix A from the normalized correspondences û_i ↔ û'_i corresponding to the normal vector n_2 = [1 0 0]^T
end if
if mode == 3 then
    Construct the coefficient matrix A from the normalized correspondences û_i ↔ û'_i corresponding to the normal vector n_3 = [0 0 1]^T
end if
Construct vector b from the normalized correspondences û_i ↔ û'_i
Find the solution vector x of the inhomogeneous system Ax = b as the intersection between two conics
Construct H_mode from vector x
```

5.2.4 Homography for the general horizontal normal $\mathbf{n}_4$

Substituting into the equation 5.1 with the normal $\begin{bmatrix}\cos\delta & 0 & \sin\delta\end{bmatrix}^T$ results in

$$\mathbf{H}_4 \sim \begin{bmatrix}\cos\alpha - p\cos\delta & 0 & \sin\alpha - p\sin\delta \\ 0 & 1 & 0 \\ -\sin\alpha - q\cos\delta & 0 & \cos\alpha - q\sin\delta\end{bmatrix} = \begin{bmatrix}h_1 & 0 & h_2 \\ 0 & h_3 & 0 \\ h_4 & 0 & h_5\end{bmatrix},$$

with $h_1 = \cos\alpha - p\cos\delta$, $h_2 = \sin\alpha - p\sin\delta$, $h_3 = 1$, $h_4 = -\sin\alpha - q\cos\delta$ and $h_5 = \cos\alpha - q\sin\delta$. With elements of $\mathbf{H}_4$, equations 5.2 and 5.3 become:

$$u_j h_1 + h_2 - u_j u'_j h_4 - u'_j h_5 = 0$$

and

$$v_j h_3 - u_j v'_j h_4 - v'_j h_5 = 0.$$

Let $\mathbf{x} = \begin{bmatrix}h_1 & h_2 & h_3 & h_4 & h_5\end{bmatrix}^T$, then for a single correspondence $\hat{\mathbf{u}}_j \leftrightarrow \hat{\mathbf{u}}'_j$ the following linear system of equations is obtained:

$$\mathbf{A}_j\mathbf{x} = \begin{bmatrix}u_j & 1 & 0 & -u_j u'_j & -u'_j \\ 0 & 0 & v_j & -u_j v'_j & -v'_j\end{bmatrix}\begin{bmatrix}h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix}.$$

There are four unknowns: α, p, q and the angle of the normal vector δ encoded in the elements of $\mathbf{h}$. One correspondence gives two equations, so at least two correspondences needed for the estimation. Let $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \dots & \mathbf{A}_k\end{bmatrix}^T$, $k \geq 2$, then the problem is to solve the homogeneous linear system $\mathbf{Ax} = \mathbf{0}$. If $k = 2$, then the null vector of $\mathbf{A}$ gives the solution. If $k > 2$, then the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue gives the optimal solution if $||\mathbf{x}||_2 = 1$ is assumed. The solution does not guarantee that $h_3 = 1$. Dividing $\mathbf{x}$ by h_3 gives the final result.

We solved for the homography $\mathbf{H}_4$, but unlike the other cases, the angle δ of the normal vector $\mathbf{n}_4$ is unknown as well. In order to retrieve it, we have to decompose the homography. The homography $\mathbf{H}_4$ has the form:

$$\mathbf{H}_4 \sim \begin{bmatrix}h_1 & 0 & h_2 \\ 0 & h_3 & 0 \\ h_4 & 0 & h_5\end{bmatrix}.$$

In order to eliminate the scale ambiguity, and have $h_3 = 1$, the calculated homography has to be divided by h_3. Denote the scaled homography by $\hat{\mathbf{H}}_4$. Then

$$\hat{\mathbf{H}}_4 = \begin{bmatrix} h_1 & 0 & h_2 \\ 0 & 1 & 0 \\ h_4 & 0 & h_5 \end{bmatrix},$$

where $h_1 = \cos\alpha - p\cos\delta$, $h_2 = \sin\alpha - p\sin\delta$, $h_4 = -\sin\alpha - q\cos\delta$ and $h_5 = \cos\alpha - q\sin\delta$. p can be retrieved from h_1 the following way:

$$p = \frac{\cos\alpha - h_1}{\cos\delta}$$

and q from h_4 as

$$q = -\frac{h_4 + \sin\alpha}{\cos\delta}.$$

Then p is substituted back to h_2:

$$h_2 = \sin\alpha - \frac{\cos\alpha - h_1}{\cos\delta}\ sin\delta$$

and q back to h_5:

$$h_5 = \cos\alpha + \frac{h_4 + \sin\alpha}{\cos\delta}\sin\delta.$$

Multiplying both equations by $\cos\delta$ results in:

$$h_2\cos\delta = \sin\alpha\cos\delta - \cos\alpha\sin\delta + h_1\sin\delta = \sin(\alpha - \delta) + h_1\sin\delta$$

and

$$h_5\cos\delta = \cos\alpha\cos\delta + \sin\alpha\sin\delta + h_4\sin\delta = \cos(\alpha - \delta) + h_1\sin\delta.$$

This can be written the following way:

$$\mathbf{Bc} = \mathbf{d},$$

where

$$\mathbf{B} = \begin{bmatrix} h_5 & -h_4 \\ h_2 & -h_1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \cos\delta \\ \sin\delta \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \cos(\alpha - \delta) \\ \sin(\alpha - \delta) \end{bmatrix}.$$

Constraints can be derived for the vectors $\mathbf{c}$ and $\mathbf{d}$:

$$\mathbf{c}^T\mathbf{c} = \cos^2\delta + \sin^2\delta = 1$$

and

$$\mathbf{d}^T\mathbf{d} = \cos^2(\alpha - \delta) + \sin^2(\alpha - \delta) = 1.$$

$\mathbf{c}^T\mathbf{c} = 1$ defines a unit circle and since $\mathbf{d} = \mathbf{Bc}$, $\mathbf{d}^T\mathbf{d} = \mathbf{c}^T\mathbf{B}^T\mathbf{Bc} = 1$ defines a zero centered ellipse. The solution is the intersection between a unit circle and a zero centeed ellipse. The solutions can be calculated the same way as in the section about the essential matrix estimation under planar motion. The singular value decomposition of the symmetric matrix $\mathbf{B}^T\mathbf{B}$ is:

$$\mathbf{B}^T\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T,$$

where $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq 0$. Let $\mathbf{y} = \mathbf{U}^T\mathbf{c}$, then the axis-aligned circle and ellipse are:

$$\mathbf{y}^T\mathbf{y} = 1 \quad \text{and} \quad \mathbf{y}^T\mathbf{\Sigma}\mathbf{y} = 1.$$

Four solutions $\mathbf{y}_i$, $i = 1..4$, can be obtained by the combinations of

$$y_1 = \pm\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \text{ and } y_2 = \pm\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}}.$$

Then the four solution for $\mathbf{c}$ and $\mathbf{d}$ are

$$\mathbf{c}_i = \mathbf{U}\mathbf{y}_i \quad \text{and} \quad \mathbf{d}_i = \mathbf{B}\mathbf{c}_i, \; i = 1..4.$$

The four solutions for the normal $\mathbf{n}_4$ are

$$\mathbf{n}_{4_i} = \begin{bmatrix} c_{i_1} \\ 0 \\ c_{i_2} \end{bmatrix}, \; i = 1..4,$$

where c_{i_1} and c_{i_2} denote the first and second element of the vector $\mathbf{c}_i$ respectively. Two solutions out of the four are correct because if $\mathbf{n}_4$ is a normal vector of a 3D plane, then $-\mathbf{n}_4$ is also a normal vector of the same plane.

Algorithm 6 Homography estimation for general horizontal normal vector $\mathbf{n}$

Input: Normalized point correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, $i = 1..n$, $n \geq 2$

Output: Homography $\mathbf{H}$ and normal vector $\mathbf{n}$

1: Construct the coefficient matrix $\mathbf{A}$ from the normalized correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, $i = 1..n$

2: Calculate $\mathbf{h}$ as the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to its smallest eigenvalue

3: Set $\mathbf{h} = \frac{1}{h_3}\mathbf{h}$

4: Construct $\mathbf{H} = \begin{bmatrix} h_1 & 0 & h_2 \\ 0 & 1 & 0 \\ h_4 & 0 & h_5 \end{bmatrix}$ from $\mathbf{h}$

5: Construct matrix $\mathbf{B} = \begin{bmatrix} h_5 & -h_4 \\ h_2 & -h_1 \end{bmatrix}$

6: Compute the SVD of the symmetric matrix $\mathbf{B}^T\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, where $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$ is the matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq 0$

7: Four solution obtained as the intersection between a unit circle and a zero centered ellipse:

8: $\mathbf{y}_1 = \begin{bmatrix} \sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ \sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}$, $\mathbf{y}_2 = \begin{bmatrix} \sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ -\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}$, $\mathbf{y}_3 = \begin{bmatrix} -\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ \sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}$, $\mathbf{y}_4 = \begin{bmatrix} -\sqrt{\frac{1-\sigma_2}{\sigma_1-\sigma_2}} \\ -\sqrt{\frac{\sigma_1-1}{\sigma_1-\sigma_2}} \end{bmatrix}$

9: Calculate $\mathbf{c}_i = \mathbf{U}\mathbf{y}_i$

10: Four normals are obtained as $\mathbf{n}_i = \begin{bmatrix} \mathbf{c}_{i_1} & 0 & \mathbf{c}_{i_2} \end{bmatrix}^T$, and two of those are correct

5.3 Plane detection by special homographies

Our goal is to find planar surfaces on a stereo image pair. Let the point correspondences between two image be $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$, $i = 1..n$. The cameras are assumed to be calibrated and their calibration matrices are $\mathbf{K}$ and $\mathbf{K}'$. Then normalized points can be calculated: $\hat{\mathbf{u}}_i = \mathbf{K}^{-1}\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = \mathbf{K}'^{-1}\mathbf{u}'_i$. These are all the points that we were able to find a correspondence between. A correspondence may be from a world point that lie on a 3D plane we are looking for or not. Let the correspondences $\mathbf{u}_j \leftrightarrow \mathbf{u}'_j$, $j = k..l$, $1 \leq k \leq l \leq n$, be the projections of co-planar world points and let $\mathbf{n}$ be the normal of the plane they lie on. Let's say we are looking for a planar surface with the normal $\mathbf{n}$. The points are co-planar so there exists a homography $\mathbf{H}$ so that

$$\hat{\mathbf{u}}'_j \sim \mathbf{H}\hat{\mathbf{u}}_j, \quad \forall j = k..l. \tag{5.5}$$

We also know that $\mathbf{H}$ has the form

$$\mathbf{H} \sim \mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{n}^T.$$

If the planar motion constraint holds and $\mathbf{n}$ is one of the 4 cases discussed before, then it has an explicit form of one of the cases derived in the previous section. Given we know two points on the plane, we can estimate the homography $\mathbf{H}$, and check for all correspondences if

$$\hat{\mathbf{u}}'_i \sim \mathbf{H}\hat{\mathbf{u}}_i, \quad i = 1..n \tag{5.6}$$

holds. Ideally the points for which this will hold are the points $\hat{\mathbf{u}}_j \leftrightarrow \hat{\mathbf{u}}'_j$, $j = k..l$. The most ideal case would be if $k = 1$ and $l = n$, so all correspondences are from points lying on the plain we are interested in and there is no noise. In this case we can select any two point correspondences to estimate $\mathbf{H}$ and confirm that indeed all points are co-planar. If there is some noise, then the equation 5.6 will not give equality (after homogeneous division) even for points lying on the plane we are looking for. Denote the vector after the homogeneous division of $\mathbf{H}\hat{\mathbf{u}}_j$ as $\hat{\mathbf{u}}_{h_j}$. $\hat{\mathbf{u}}_{h_j}$ will be close to but not exactly equal to $\hat{\mathbf{u}}'_j$. To know if the transformed point $\hat{\mathbf{u}}_{h_j}$ is "good enough", we need to define a distance measure. The standard euclidean norm is one way to measure distance:

$$d(\hat{\mathbf{u}}_j, \hat{\mathbf{u}}'_j, \mathbf{H}) = ||\hat{\mathbf{u}}'_j - \hat{\mathbf{u}}_{h_j}||_2. \tag{5.7}$$

Any valid homography is non-singular and has an inverse. Multiplying the equation 5.5 with the inverse of $\mathbf{H}$ gives:

$$\hat{\mathbf{u}}_j \sim \mathbf{H}^{-1}\hat{\mathbf{u}}'_j, \quad \forall j = k..l.$$

If $\mathbf{H}$ is a homography between the first and second image, then $\mathbf{H}^{-1}$ is a homography between the second and first image. It is useful in defining the symmetric version of the distance above:

$$d(\hat{\mathbf{u}}_j, \hat{\mathbf{u}}'_j, \mathbf{H}) = ||\hat{\mathbf{u}}'_j - \hat{\mathbf{u}}_{h_j}||_2 + ||\hat{\mathbf{u}}_j - \hat{\mathbf{u}}'_{h_j}||_2, \tag{5.8}$$

where $\hat{\mathbf{u}}'_{h_j}$ is the vector resulted from the homogeneous division of $\mathbf{H}^{-1}\hat{\mathbf{u}}'_j$. We will say that the correspondence $\hat{\mathbf{u}}_j \leftrightarrow \hat{\mathbf{u}}'_j$ fits the homography $\mathbf{H}$ if $d(\hat{\mathbf{u}}_j, \hat{\mathbf{u}}'_j, \mathbf{H}) < \epsilon$ for some threshold $\epsilon \in \mathbb{R}^+$.

In most images, not all correspondences correspond to co-planar world points. In order to estimate $\mathbf{H}$, we need to find at least two correspondences from world points that lie on a plane with the normal we are looking for. Those points that are not like this will be called outliers and those that are like this will be called inliers. RANSAC [6] is a famous model fitting algorithm, that works well even in cases where the outliers are outnumbering the inliers.

5.3.1 Homography fitting by RANSAC

Let $D = \{\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i | i = 1..n\}$ be the set of all normalized point correspondences. The basic idea behind RANSAC is that we randomly select a minimal sample from D to construct a model (in this case a homography $\mathbf{H}$) and check for all elements in D if it fits the model ($d(\hat{\mathbf{u}}_j, \hat{\mathbf{u}}'_j, \mathbf{H}) < \epsilon$). If the data fits the model, we add it to the set of inliers. We save the set of all inliers and sample again to construct another model. If the new model has more inliers we save it as the best model and so on. After several iterations the model with the most amount of inliers will be chosen as the model we are looking for. In our case the best model will be $\mathbf{H}$ corresponding to the world plane with the normal we are looking for and the inliers that fit it are the points on that plane.

Algorithm 7 Homography fitting by RANSAC

Input: Normalized point correspondences $D = \{\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i | i = 1..n\}$, $n \geq 2$ and distance threshold ϵ

Output: Homography $\mathbf{H}$ and the set of inliers I

Initialize the set of inliers as $I = \{\}$
Set iteration = 1
while iteration < max iterations **do**
    Select a minimal sample s from D
    Estimate the homography $\hat{\mathbf{H}}$ from s
    Initialize the inliers of the model $\hat{\mathbf{H}}$ as $\hat{I} = \{\}$
    for i=1..n **do**
        if $d(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}'_i, \hat{\mathbf{H}}) < \epsilon$ **then**
            Set $\hat{I} = \hat{I} \cup \{\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i\}$
        end if
    end for
    if $|\hat{I}| > |I|$ **then**
        Set $\mathbf{H} = \hat{\mathbf{H}}$ and $I = \hat{I}$
    end if
    Set iteration = iteration + 1
end while

5.3.2 Multi homography fitting by sequential RANSAC

The imaged scene can contain multiple planar surfaces with the normal we are interested in and RANSAC will select the homography corresponding to the plane that has the most number of corresponding points. If we want to find multiple planes we can use the sequential RANSAC algorithm. Sequential RANSAC first finds the model with the most number of correspondences and then removes the inliers from the set D. RANSAC is run on the set without the inliers $D \setminus I$ and finds the model with the second most number of inliers and so on. After we reach a specific number of iterations or the inlier number not reaches a specified threshold, the algorithm stops.

Algorithm 8 Multi homography fitting by sequential RANSAC

Input: Normalized point correspondences $D = \{\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i | i = 1..n\}$, $n \geq 2$, distance threshold ϵ, and the number of desired homographies m

Output: Set of homographies $H = \{\mathbf{H}_i | i = 1..m\}$ and the set of inliers $I = \{I_i | i = 1..m\}$

Initialize the set of inliers as $I = \{\}$ and models as $H = \{\}$
for j = 1..m **do**
  Initialize $I_j = \{\}$
  for iteration = 1..max iterations **do**
    Select a minimal sample s from D
    Estimate the homography $\hat{\mathbf{H}}$ from s
    Initialize the inliers of the model $\hat{\mathbf{H}}$ as $\hat{I} = \{\}$
    for i=1..n **do**
      if $d(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}'_i, \hat{\mathbf{H}}) < \epsilon$ **then**
        Set $\hat{I} = \hat{I} \cup \{\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i\}$
      end if
    end for
    if $|\hat{I}| > |I_j|$ **then**
      Set $\mathbf{H}_j = \hat{\mathbf{H}}$ and $I_j = \hat{I}$
    end if
  end for
  Set $H = H \cup H_j$, $I = I \cup \{I_j\}$ and $D = D \setminus I_j$
end for

5.4 Results

The algorithms for the homography estimation are implemented in Matlab and tested for synthetically generated correspondences and for images taken from the Málaga Urban Dataset [7].

5.4.1 Synthetic tests

For the synthetic tests, two cameras are generated as $\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ and $\mathbf{P}' = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$, where $\mathbf{K}$ is the common calibration matrix and $\mathbf{R}$ is a random rotation around the Y-axis and $\mathbf{t} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \end{bmatrix}^T$ is a random rotation with angle β. A distance d and normal $\mathbf{n} \in \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ is chosen and random world points are generated on the plane with the normal $\mathbf{n}$ and distance d and also other planes with the normals corresponding to the other cases. The points are then projected onto the virtual cameras' image planes using $\mathbf{P}$ and $\mathbf{P}'$ and normalized by the calibration matrix $\mathbf{K}$. The ground truth homography is constructed as $\mathbf{H} = \mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{n}^T$. The inliers are the points corresponding to the selected normal $\mathbf{n}$ and the outliers are points corresponding to the other two cases. 100 points are generated as inliers and 1300 and 2000 points generated as outliers corresponding to other planes. With a threshold of $\epsilon = 0.001$ the homography fitting by RANSAC correctly finds the homography $\mathbf{H}$ and the 100 inliers. For the general horizontal normal $\mathbf{n}_4$, plane points are generated and projected the same way without outliers. The homography is estimated and decomposed to retrieve the normal. 4 solutions are obtained and one of them is the normal $\mathbf{n}_4$ we are looking for and one is $-\mathbf{n}_4$. Both of them are correct and the other two are incorrect.

5.4.2 Real world tests

The algorithms are tested on real world images from the Málaga Urban Dataset [7]. Two consecutive images are selected from the dataset and point correspondences are obtained by ASIFT [8]. Red dots will indicate points on the detected plane. The algorithm works with two images, but the found points are shown in the perspective of the first image. Sequential RANSAC was ran on all of the images with a 1000 iterations on a single RANSAC homography fit. Figure 5.2 shows the first example image pair. In the perspective of the first camera there are vertical planes on the side and the plane of the road. First a plane with a normal of $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ is estimated. The threshold was set to $\epsilon = 0.002$. Figure 5.3 show the first detected plane. Because RANSAC will select the model with the most inliers first, there is no guarantee that it will find a good homography, but a homography with the most inliers that also has the form that we constrain it to.

(a) The first image

(b) The second image

Figure 5.2: The first example

When we are computing correspondences, features has to be detected between the image pair first. These features are in most cases edges or corners or curves that the ground plane does not have much of. Consequently there will be less features on the ground than on the side of buildings.

Figure 5.3: First detected plane for the vertical normal

The plane detected first on figure 5.3 is not the ground plane but a plane that is parallel with it. It has more inliers than the second detected plane that is indeed on the plane of the road. Figure 5.4 shows the second detected plane.

Figure 5.4: Second detected plane for the vertical normal

Most of the points are on the on the ground plane, but there are also outliers on the side of the building that somehow fit the homography. It can happen that the algorithm does not give a useful result, as with the third detected plane. The fourth detected plane is also containing the ground plane and some outliers from the side of the building. Figure 5.5 shows the third and fourth detected planes. Unfortunately the algorithm can give bad results as shown on the third detected plane, so a homography rejection or a point selection method should be developed in order to reduce the number of bad results.

Next, vertical planes on the side of the road are estimated with a normal of $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. The threshold was set to $\epsilon = 0.004$. There are a lot of correspondences on the side of the left building so the first detected plane is the one we would expect. Figure 5.6 shows the first detected plane.

(a) The third detected plane

(b) The fourth detected plane

Figure 5.5: Other detected planes for the vertical normal

The other detected vertical planes are also this plane, but slightly different. Figure 5.7 shows the second and third detected plane. There is no frontal plane that can be seen on both images, so frontal planes are not estimated here. For the general vertical planes, the threshold was set to $\epsilon = 0.004$ again. The vertical plane detected on figure 5.6 dominates the scene and it is the first that is detected for a general vertical plane. Figure 5.8 shows the first detected general vertical plane.

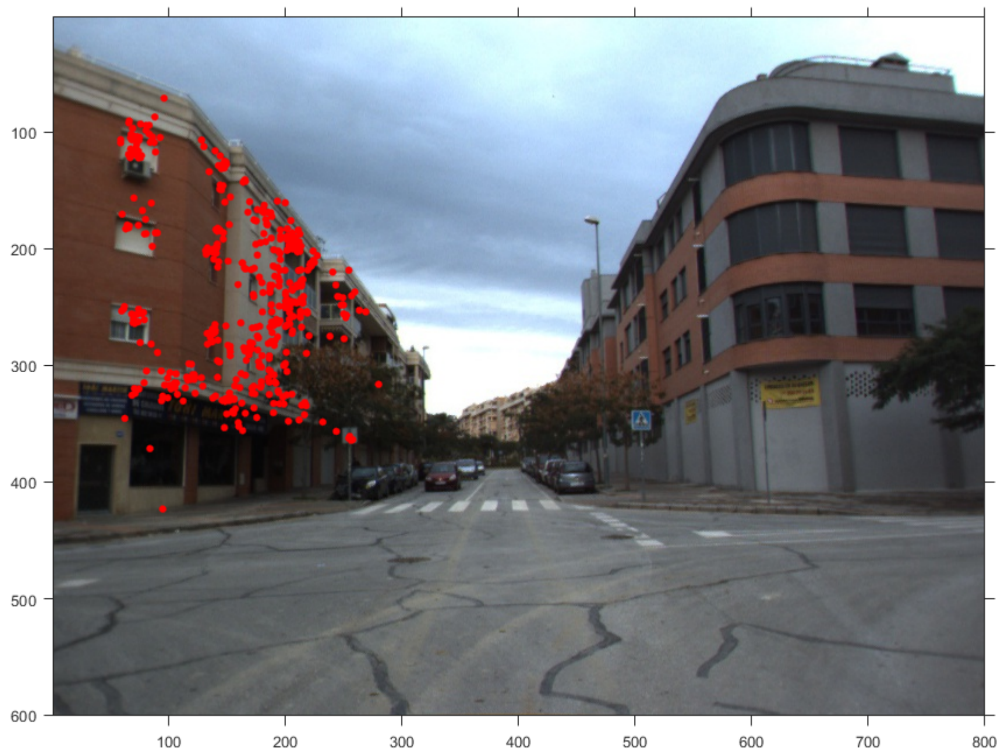

Figure 5.6: First detected plane for the horizontal sideways normal

(a) The second detected plane

(b) The third detected plane

Figure 5.7: Other detected planes for the horizontal sideways normal

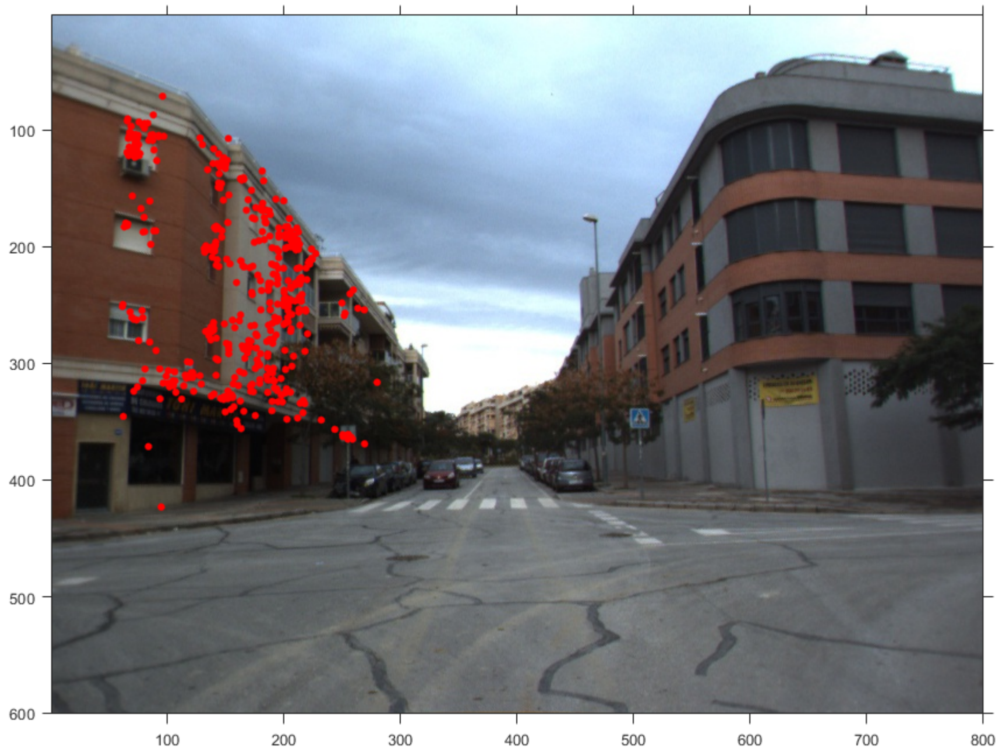

Figure 5.8: First detected plane for the general horizontal normal

The plane on figure 5.8 has almost the same inliers as on figure 5.6 because the detected plane is the same. I am not sure if the second detected plane is a valid one, but it definitely not fits the scene structure. The third detected plane is almost the same as the first. The second and third detected planes are shown on figure 5.9.

(a) The second detected plane (b) The third detected plane

Figure 5.9: Other detected planes for the general horizontal normal

The results for other images are summarized more concisely. The next example image pair is shown on figure 5.10. Figure 5.11 shows the first 4 planes detected with a vertical normal $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ and threshold $\epsilon = 0.002$. The first and third detected planes are containing the ground plane with outliers on the side of the building on the right. The second and fourth planes are not good results. Figure 5.12 shows the first two results for the planes with a sideways horizontal normal $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, with the threshold $\epsilon = 0.004$. The results are what we would expect, the planes of the two buildings on the side of the road. The last example image pair can be seen on figure 5.13. It is selected to test for planes with a frontal horizontal normal $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. The threshold was set to $\epsilon = 0.002$. Figure 5.14 shows the first 4 planes detected for the vertical normal. The results are not good as none of them contain the ground plane we are looking for.

(a) The first image (b) The second image

Figure 5.10: The second example image pair

(a) The first detected plane

(b) The second detected plane

(c) The third detected plane

(d) The fourth detected plane

Figure 5.11: Detected planes for the vertical normal

(a) The first detected plane

(b) The second detected plane

Figure 5.12: The detected planes for the horizontal sideways normal

(a) The first image

(b) The second image

Figure 5.13: The last example image pair

(a) The first detected plane

(b) The second detected plane

(c) The third detected plane

(d) The fourth detected plane

Figure 5.14: Detected planes for the vertical normal

(a) The first detected plane

(b) The second detected plane

(c) The third detected plane

(d) The fourth detected plane

Figure 5.15: Detected planes for the frontal horizontal normal

Figure 5.15 shows the results for the frontal vertical planes. Unfortunately, the first 4 planes do not contain the wall directly facing us. This can happen if there are not enough correspondences found on the wall. The first four planes detected for the general vertical plane can be seen on figure 5.16. The planes are almost identical to the planes detected for the frontal horizontal normal.

(a) The first detected plane

(b) The second detected plane

(c) The third detected plane

(d) The fourth detected plane

Figure 5.16: Detected planes for the general horizontal normal

Chapter 6

Conclusion

In the first part of this thesis, epipolar geometry was investigated in the case when the cameras are attached to motorbikes. Based on the theory of planar motion, the model of motorcycle motion is developed. The difference between planar and motorcycle motion is that bikes can also be tilted. If the bike is tilted, then the camera attached to it rotates around the center of rotation of the bike. The developed model is only an approximation as you cannot place the camera on the center of rotation (where the wheels touch the road). The camera should be placed as close to the ground as possible to not violate the constraint. I proposed a 6-point algorithm to solve for the essential matrix when the rotation around the optical axis is known for the first camera. It is based on the idea of the 8-point algorithm and compared to it with varying noise levels. Using 6 points instead of 8 resulted in reduced error in both rotation and translation. For the case when the rotation around the optical axis is known for both cameras, I proposed a method to reduce the problem to planar motion by rotating back the corresponding points around the optical axis. The essential matrix corresponding the rotated points can be estimated by any algorithm developed for planar motion, including the presented circle-ellipse method. For a sequence of images, the rotation around the optical axis for the first camera is needed at only the first image, as the rotation for the consecutive images can be calculated from the essential matrix.

In the second part of this thesis, a planar surface detector is proposed. Special homographies corresponding to planes with different normal vectors were robustly fitted to point correspondences between an image pair. The inliers of the homographies were the points on the plane. The methods were tested on real world images with mixed results. Planes corresponding to the walls of buildings were correctly detected in most cases. The plane of the road contains a lot less features than the walls of buildings, so not many correspondences were found on the road. Consequently the plane of the road is not detected in many cases. Also, the homography with the most amount of inliers is not always the best model for the scene. To find the correct planes, additional homography rejection or point selection methods should be developed.

Bibliography

[1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. USA: Cambridge University Press, 2003. ISBN: 0521540518.

[2] Sunglok Choi and Jong-Hwan Kim. "Fast and reliable minimal relative pose estimation under planar motion". In: *Image and Vision Computing* 69 (2018), pp. 103 –112. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2017.08.007. URL: http://www.sciencedirect.com/science/article/pii/S0262885617301233.

[3] Levente Hajder and Dániel Baráth. "Least-squares Optimal Relative Planar Motion for Vehicle-mounted Cameras". In: *ArXiv* abs/1912.06464 (2019).

[4] Levente Hajder and Dániel Baráth. "Relative planar motion for vehicle-mounted cameras from a single affine correspondence". In: *ArXiv* abs/1912.06465 (2019).

[5] *Computer Vision Course Homepage*. http://cg.elte.hu/index.php/computer-vision/. Accessed: 2020-05-13.

[6] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[7] José-Luis Blanco, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. "The Málaga Urban Dataset: High-rate Stereo and Lidars in a realistic urban scenario". In: *International Journal of Robotics Research* 33.2 (2014), pp. 207–214. DOI: 10.1177/0278364913507326. URL: http://www.mrpt.org/MalagaUrbanDataset.

[8] Guoshen Yu and Jean-Michel Morel. "ASIFT: An Algorithm for Fully Affine Invariant Comparison". In: *Image Processing On Line* 1 (2011), pp. 11–38. DOI: `10.5201/ipol.2011.my-asift`.

List of Figures