Eötvös Loránd University

Faculty of Informatics

Dept. of Software Technology and Methodology

3D reconstruction from a vehicle mounted camera

Supervisor:
Hajder Levente
Associate Professor

Author:
Baigalmaa Bayarsaikhan
Computer Science for Autonomous Systems MSc

Budapest, 2024

Contents

Chapter 1

Abstract

Precise vehicle motion estimation become the main focus of 3D reconstruction. Visual odometry, a field in computer vision, localizes vehicle ego motion by fusing sensor data attached to the vehicle. This thesis work worked on a solution of visual odometry using consecutive stereo images and proposed two different methods on rectified and original frame pairs, which follow separate computation in vehicle pose estimation. First, find feature correspondences one pair based on three dimensional points produced by disparity map, the rectified stereo pair, and the other pair tracked over consequent frame using optical flow. The process yields three pairs of feature matches, which are corresponding image points and spatial coordinates. Hence, outlier removal step considered with a pose estimation under the planar motion algorithm adopted with RANSAC. Afterwards, vehicle pose estimation employed a solution to Perspective-3-Point problem. Another solution for unrectified stereo pairs starts with SIFT feature matching technique, followed by corresponding image point triangulation for three dimensional points and optical flow tracks to find consecutive frame matches. Therefore, three-point feature matching has become available to track image sequences. Outlier removal, in this case, utilized idea of monocular visual odometry. Hence, one point under the circular planar motion algorithm was implemented to get fewer contaminated feature points. Vehicle ego-motion is produced by a five-point pose estimation algorithm. Both methods evaluated on KITTI benchmark and ELTE car dataset. Experiments has done separately in real vehicle environments; however, the results included an evaluation of the combination of the methods.

Chapter 2

Introduction

3D reconstruction is one of the main tasks for machine perception, virtual reality, 3D modeling, and robotics. In this thesis, 3D outdoor scene reconstruction is aimed at achieving this by using stereo camera images mounted on vehicles. Before digging into 3D scene reconstruction, I would like to elaborate on what 3D reconstruction is and how machines work to reconstruct spatial objects.

First, 3D reconstruction is the process of building 3D objects from a source or sources in which enough information about an object has been duplicated or presented. The sources can usually be laser scanners or images, or they could be both. From two-dimensional information like images with depth and perspective data, one can easily build world structure, which helps robots grasp spatial concepts and define how to react to them. Second, 3D reconstruction in robotics and computer vision imitates the human vision system. Since, as mankind, we train computers to do what we want to do and simulate human things like sight, action, movement, thinking, perception, and even the ability to make predictions, cameras are considered eyes, and many researchers are conducting human observation to accelerate autonomous vehicles, robots, smart homes, medical, and other industrial technologies.

The camera plays a main role in image processing and video processing. The main part of the field is to develop machines with optimal sight, expanding with an understanding of their environment. There is much research devised to process camera images for 3D reconstruction, also called structure [1], [2], and [3]. Research uses more than one camera image to get a precise result. For example, a pair of cameras from the so-called stereo camera commonly used in computer vision and robotics due to their eyes-like camera setup could make it easier to imitate the concept of the

human eye and apply physical findings to computers. This stereo image processing is broadening research possibilities in computer vision.

Stereovision has been an important source of information in this work, especially in finding correspondence between images and 3D world coordinates. There are two different methods conducted based on the way to generate spatial points. One uses the advantage of stereo block matching, in which rectified images are utilized, and the other is designed for stereo-like image pairs, which are from two monocular camera installations and do not need to be a stereo camera produced by the factories or attached to a vehicle in a parallel direction. In the second case, rectification can be necessary; however, during the process, projective transformation is applied to 2D image pairs [4], which can modify the image shapes and sizes. Therefore, instead of rectifying images, traditional feature matching and triangulation techniques are used as alternatives.

Moreover, the outlier removal process was introduced by one-point [4] and two point [5] motion estimations under the planar motion constraint. Since feature correlations have the possibility of comparing both monocular and 3D to 2D feature correspondences, outlier removal methods employ consecutive frame correspondences in a one-point algorithm. Also, spatial points and consecutively matched features were derived in the case of the two-point pose estimation algorithm. The one-point algorithm proposed in the [4] paper is designated for circular planar motion and estimates the vehicle rotation angle. On the other hand, two 3D and 2D point correspondences are estimated to get the projection matrix under planar motion in [5] paper by imposing rotation around the y-axis and translation in the x and z-axes of the vehicle camera coordinate system. After all, pose estimation became available on filtered feature matches in the previous step. Perspective-3-Point [6] and five-point algorithm [7] adopted in the generation of vehicle trajectory consist of each location accumulated at a time. The Perspective-N-Point problem determines the localization of an object utilizing a tetrahedron constructed by control points and the camera center, a notion widely used in Simultaneous Localization and Mapping (SLAM) and Visual Odometry motion estimation. In contrast to Perspective-N-Point problem, the five-point relative pose estimation algorithm commonly used in monocular ego-motion computation also produces valuable results between corresponding image features.

The structure of the thesis work starts with related research on SLAM and vi-

sual odometry in different possible concepts around the topic. The later sections consist of theoretical concepts to give a mathematical explanation of the solution, elaboration of methods, and a real vehicle environment experiment and its results, followed by a conclusion.

Chapter 3

Related Works

Most of the SLAM pipelines are designed for feature matching, pose estimation, and 3D reconstruction, such as [8], [1], and [2]. The localized and accumulated pose estimation over time is called visual odometry. Since September 1980, when the first Visual Odometry [9] research was conducted, there have been lots of different solutions devised and evolving with expanded Visual Odometry that simultaneously produce both structure and motion at the same time, called Simultaneous Localization and Mapping (SLAM). In contrast to Visual Odometry, the SLAM has an advantage over the loop-closure drift reduction concept [10], which corrects accumulated drift when known features are encountered again. However, visual odometry is regarded as loop-closure-free visual SLAM in robotics or bundle adjustment-free SLAM in computer vision [11].

Visual Odometry, SLAM, and related solutions have been elaborated in subsections 3.2 and 3.3. Moreover, wheeled vehicle relative motion is discussed with an example in subsection 3.4. Correct feature matching between two consecutive frames influences a lot in optimal SLAM and visual odometry because noisy matches drastically worsen the outcome [11]. Hence, the feature matching technique can be either appearance-based, intensity-based, or a combination of both. In the 3.1 sub-section, focus more on related research on feature matching.

3.1 Large Scene Image Matching

[12] proposed a method to accurately match large-view scene images. Optimal matches are computed by a two-point pose estimation similar to the one-point algo-

rithm [4]; instead, rotation and moving angles are represented. Scenarios for estimation motion and homography matching over RANSAC iteration are proposed in the algorithm. Motion derives similar to [4]; however, additional computation is required by finding the intersection of two ellipsoids formed by the epipolar constraint's linear equation solution. After, multiple homography estimations are done over feature matching to remove outliers due to real-world features that mostly belong to plane structures such as walls, windows, or buildings. 4 points in the same plane can describe homography [13]. Finding at least four inlier points for homography estimation is a challenging task in the real world. Hence, the homography matrix was computed by exhaustive searching and voting on previously estimated two-point motion, two image points, and world coordinate correspondences. The best solution is found by computing the homography transformation reprojection error on normalized feature points. They analyzed the stability of the two-point algorithm under 2D motion and known camera intrinsic assumptions in terms of non-2D motion cases and concluded the proposed method maintains a good result under 8.55 degrees of vertical translation in roll and pitch angles and 15 percent horizontal translation. The number of initial matchings and plane searching resolutions is proportional to computational time. They conducted an experiment on visual loop detection under the assumption of stability and compared the results with 4 point homography [13] and 7 point epipolar RANSAC. In the case of small overlapping feature matching, the proposed two-point approach finds true matches and removes false matches well. To conclude, Under the assumption that motion is approximately 2D planar and known camera intrinsic, the proposed two-point large-scale scene image matching produces good feature matching with rejected outliers, even in small overlapping situations.

[14] proposed large-scale stereo matching, considered a generative probabilistic model for stereo matching. They proposed a Bayesian stereo feature matching approach to accurately produce disparity without global optimization in real time. The motivation of the research is to explore robust stereo matching possibilities in stereo high-ambiguity correspondences. Three main parts are taken into account: support points, stereo matching, and disparity estimation. The authors proposed the notion of support points, robustly matched pixels by their uniqueness and texture. Vertical and horizontal Sobel filters are applied to support point matching. Moreover, consistency with the same left-to-right and right-to-left correspondences imposed robustness. Ambiguity matches are removed by the threshold ratio between the best

and second-best matches. Also, mismatches are eliminated in cases where different disparities are represented throughout the neighborhood. A stereo-matching probabilistic generative model is used to draw image samples on one image by support point and observation in the other image. In other words, the support point and observation in the left image are given, and two steps can be taken to find a sample from the corresponding right image observation. First, disparity computation needs to be done by support point and left image observation. Then, a right-image sample was obtained by left-image observation and disparity, as found previously. Disparity map computation relies on maximum a-posteriori (MAP) estimation on both left and right images. The experiments have done accuracy and running time comparisons with different approaches. [15] [16] [17] [18] on Middlebury dataset and non-occluded pixel computed in error evaluation. As a result, the proposed matching entropy disambiguates the problem, performs lower than uniform prior, and the disparity computation time for left and right images is 0.6 seconds. Overall disparity evaluation compared to [15] and baselines, the proposed method performs well in differentiating correct disparity and textured objects.

3.2 Monocular SLAM

Large-Scale Direct Monocular SLAM [19] is one of the famous research projects that proposed a feature-less direct monocular SLAM algorithm. The research team's proposed algorithm is subdivided into three different parts, which are tracking, depth map estimation, and map optimization. In contrast to traditional feature tracking methods [20], [21], [22], key frames, including three dimensional point, depth value, and depth variance information, have been used to track each image of the monocular sequences of images. Then depth map estimation takes place by selecting a suitable key frame and refining it with tracked non-key frames. To align images, scale ambiguity, loop closure, and convergence radius estimation for tracking have come up as constraints to be found. In the last step, pose graph optimization [23] techniques are utilized. In this LSD SLAM, monocular images processed with scale ambiguity are corrected with a proposed alignment called direct, scale-drift aware image alignment. Experiments are shown in challenging environments such as large scale change, multiple rotational routes, and hand-held trajectories. As a result, LSD SLAM represents the absolute trajectory error on a real-life dataset, which was 3

times lower than semi-dense visual odometry for a monocular [24] and more than 20 times lower than parallel tracking and mapping in a small AR workspace [25].

Another one of the famous SLAM methods, based on ORB (Oriented FAST[26] and Rotated BRIEF[27]) feature matching, is called ORB SLAM [8]. The main idea in this work was utilizing features in all steps: tracking, mapping, relocalization, and loop closure. Tracking is the decision-maker for new key frame entry and localization of the camera in each frame. Then local mapping performs reconstruction around the camera pose. In this stage, the reconstruction is optimized by local bundle adjustment [28] and triangulation [29] new points by searching for new key frame correspondences along connected key frames by covisibility graph [30].Then, loop closure drift detection was optimized by a sparser covisibility subgraph called the Essential Graph. In the end, whole optimization has been estimated with the Levenberg-Marquardt algorithm implemented in g2o [31], which is C++ framework designed for optimizing graph-based non-linear error functions. This method was tested against three different datasets. As a result, indoor accuracy is lower than 1 cm and outdoor accuracy, in cases of scale aligned with ground truth, is a few meters.

3.3 Stereo Visual Odometry

Depth estimation is the challenge in the monocular approach. It requires a checkerboard, markers, a known object in the image, or additional sensor fusing to tackle the problem. On the contrary, stereo pairs allow us to define depth by disparity. [2] is an example of Stereo Visual Odometry in which three-dimensional real points are produced by the Stereo Semi Global Matching technique [32], hence planar detection in spatial space described by 2x2x2cm regular grid voxel creation over the three dimensional points is possible. The proposed method uses stereo pair images and their transposed bird eye view images separately for three-dimensional points and road detection, respectively. Virtual camera images are defined with the same camera intrinsic matrix, a 3x3 matrix containing the camera focal length and principal points of the image plane, as in stereo cameras. The virtual camera images are rotated and transformed into a bird's-eye view, where features of the road are able to be detected. Then, SIFT-generated features of bird-eye view images are classified into inliers or outliers depending on the point belonging to the ground and

the height value. In this case, the author mentioned that detecting features in a bird's-eye view is more advantageous than a stereo pair. Camera poses are produced in two different ways: one with three and two-dimensional point correspondence motion estimation called the Perspective-N-Point algorithm [33] and the other with 3D to 3D point estimation. Feature matching, image pair (stereo or bird eye view), and pose estimation build four different alternatives to visual odometry estimation. The combinations are SIFT feature matching with bird eye view images, SIFT feature matching with stereo left image, SURF feature matching with bird eye view images, and 3D to 3D pose estimation. Root mean square error (RMSE) was computed to assess angular and linear speed for each testing variation. The result shows all of the combinations perform slightly differently except for 3D to 3D pose estimation. In the end, the four combinations were compared in terms of real trajectory errors. Conclusion made on SIFT with bird-eye view side due to good performance in angular error and longitudinal displacement error compared to others. To conclude [2], featuring ground plane estimation on bird eye view images combined with traditional matching and pose estimation methods shows a possible opt for visual stereo simultaneous localization and mapping problem.

There is another proposed Visual Odometry [10] using stereo disparity and optical flow on the KITTI Benchmark dataset. Authors proposed visual odometry state-of-the-art solution, introduced accumulation of tracked features point to get refined camera pose. Therefore, those refined estimations, combined with optimization, reduce the drift scale overall. First, rigid body motion, which describes transformation between consecutive frames, follows the approach called A Head-Wearable Short-Baseline Stereo System for Simultaneous Estimation of Structure and Motion [34]. Let me call this Short-Baseline Stereo SLAM for this thesis writing, where feature points consist of pixel points (u, v) and disparity (d) for each pixel (u, v, d). Short-Baseline Stereo SLAM also tracked features by Kanade-Lucas-Tomasi (KLT) [35] and matched by optical flow algorithm. To correct the accumulated pose estimation error, the authors proposed a method that tracked feature history, including the reduction in ego-motion drift explained by the augmented feature set. The set is composed of all previous positions of each feature converted into a combined current frame called an integrated feature. With this augmented feature set, new feature correspondences are derived, and optimization is updated accordingly. Camera pose objective function computed by minimizing new feature correspondence and

[34] feature correspondences. Also, for each step, the propagation process involved transforms both measures and integrated features. As the author mentioned, the computation time for this proposed method is O(n) for n features. As stated in the research paper, the proposed method's KITTI benchmark dataset result performs better than the original by up to 12 percent in translation and 20 percent in rotation.

3.4 Relative Motion under Planar Constraint

[4] shows how the motion estimation problem is reduced to one or two degrees of freedom under known environment constraints. The notion of circular motion in a wheeled vehicle, also known as the Instantaneous Center of Rotation (ICR), ensures the Ackermann steering principle [36]. Therefore, relative motion between two consecutive frames is estimated by one feature correspondence and epipolar constraint [37]. Then, the refinement process has been done with the proposed One Point Ransac algorithm, a joint effort of the One Point Algorithm and RANdom SAmple Consensus (RANSAC)[38]. [11] is proposed upon [4] and tackled scale estimation using feature correspondences. The scale limits motion estimation due to scale ambiguity in monocular view. The proposed Monocular Visual Odometry utilized the KLT [35] feature tracker followed by optical flow to find the corresponding flow on the new image. Then, the flow outliers were excluded by One Point RANSAC as [4] but the proposed method introduced a new error function geometrically and algebraically to avoid the trigonometric computation used initially. After removing outliers, relative motion estimation has included inliers (good feature correspondences) and employed the normalized one-point algorithm, which produces more accuracy than the original due to the minimization of geometric error. Moreover, the two-point algorithm, computation of angle and translation, is estimated through the linear equation [39] and Newton's iteration method [40]. Final motion selected by either one-point or two-point algorithms results under the firewall condition, where motion rotation angle and translation differ by less than 10 degrees. To solve the scale problem, the authors estimated planar homography such that the ground is flat and vehicles move at a constant interval on the ground. This is the condition without scale ambiguity; hence, the one-point algorithm is used to compute the scale of relative motion based on the planar homography of the ground plane, refined by One-Point RANSAC. After these steps, the final relative pose accumulated in the

conventional coordinate system. The experiment with this proposed method has taken both synthetic and real datasets. As a consequence, the normalized One Point algorithm results in less drift error, improved by 25 to 45 percent compared to the original [4] due to the proposed geometric error estimation. Moreover, scale estimation errors range from 0.1 to 0.2 per frame for 10 to 20-meter-long trajectories.

[41] proposed non-iterative minimal relative pose estimation under planar motion. Epipolar constraint [37] under planar motion, rotation around the vertical axis and translation around the horizontal and forward axes of a moving vehicle on the road, produces relative pose estimation. Moreover, the author mentioned iterative two-point [40], linear three-point [42] and intersection of two ellipses [43] algorithms as a wrap-up of alternatives to planar motion estimation over an epipolar constraint and further result comparison. The proposed two-point algorithm proposes unit circle and line intersection transformations. Firstly, the epipolar constraint's linear equation is rearranged into a vector representation of a unit circle and ellipse. The intersection between a unit circle and an ellipse leads to an optimal solution in degeneracy. Lastly, they proposed the intersection of a line and a unit circle solution, which simplified the previous solution by transforming an ellipse into a line by its trigonometric identities. The experiment has been compared with different planar and general algorithms in the synthetic and KITTI Visual Odometry datasets. 5x5 and 11x11 kernel feature extraction using sum-of-absolute-difference (SAD), followed by a normalized 8-point algorithm [44] with adoptive RANSAC [45], used both outlier removal and pose estimation on inliers. Estimated scale by ground plane fitting and interval scaling with asymmetric kernel fitting [46]. Then, two-step RANSAC, similar to [11] [4] [47] with large and small thresholds, respectively, refined final motion accumulation. As a consequence, high-speed conditions over 50 km/h result in a worse 8-point algorithm result. As a result, planar motion algorithms were able to differentiate planar from non-planar motion noise up to 6 degrees. Planar motion algorithm with two steps: RANSAC reduced computing time in outlier rejection by 30 times faster than the 5-point algorithm [48] and 8 times faster than the 8-point algorithm. Computing time overall is 28 percent faster than the 5-point algorithm with single-step RANSAC. They concluded that the that the proposed algorithm can be applied to visual place recognition and 3D reconstruction applications.

3.5 3D Scene Reconstruction

[1] proposed dense 3D reconstruction in real-time. They proposed a method to reconstruct a 3D environment from stereo image sequences. The proposed 3D reconstruction pipeline consists of four steps: sparse feature matching, visual odometry, dense stereo matching, and 3D reconstruction. Also, a calibrated and rectified stereo setup is assumed. Firstly, they circularly matched features over two consecutive stereo images. Subset feature matching is based on the idea of stereo matching [14] to reach more time-efficient circular feature matching. The proposed robust visual odometry starts with producing 3D points by matched feature triangulation [29] and motion estimated by Gauss-Newton optimization over minimizing reprojection error. Then, motion estimation refinement has been done with RANSAC [38] 50 iterations, 3 randomly selected correspondences, and a Kalman filter applied lastly to ensure constant acceleration. Efficient Large Scale Stereo Matching research [14], effective matching over high resolution images, is used in dense stereo matching. They proposed a greedy 3D reconstruction approach over Bundle Adjustment [28] which is not computationally efficient in real time problems. Greedy 3D reconstruction is designed to solve the 3D association problem, duplicated 3D points are not necessary to reconstruct. The proposed solution is the reprojection of a 3D reconstruction of the previous frame to the image plane of the current frame. If all the projected points match with valid disparity, the 3D points are combined by their mean, which reduces the number of points to be stored and improves accuracy by averaging noise over frames. The experiment was performed on a real image sequence dataset with GPS/IMU ground truth and comparison with [3]. As a consequence, circular feature matching runs in less than 0.5 seconds over 15000 features, and the proposed robust visual odometry runs in 4.3 milliseconds for 200 feature matching, which is around 2 times faster than the compared approach. Also, a quality-based evaluation was presented as a result of the proposed greedy 3D reconstruction.

Chapter 4

Theoretical Framework

Most of the theoretical notions adopted from "Multiple View Geometry by Richard Hartley and Andrew Zisserman"[42] are related to the concepts cited in each section below.

4.1 Camera Model and Projection

Cameras are the main perception tool for robotics and autonomous vehicles, like eyes for humans. It maps 3D world points into 2D image points, also referred to as Euclidean 3-space to 2-space. The map between different dimensional spaces is commonly called *Projection*, one dimension projected into another. Projection of points helps machines perceive surroundings and process them accordingly under given instructions.

As mentioned in [49], a camera model can be divided into two specialized models: one with a finite center and one with an infinite center. The most simple and well-known camera model is the pinhole, which belongs to the finite camera model followed by central projection. The pinhole camera was thought of as a central projection of world points into an image plane (also called a view plane). Generally, center of projection means world points mapped into the view plane through the intersection of the image plane and rays emitted from the center of projection (also denoted as *camera centre* or *optical centre*) to world points (X, Y, Z) as depicted in Figure 4.1. The ray perpendicular to the center of the projection is the *principal axis* or *principal ray*, and a point intersecting with the view plane represents *principal point*. This projection is also called *Perspective Projection* in

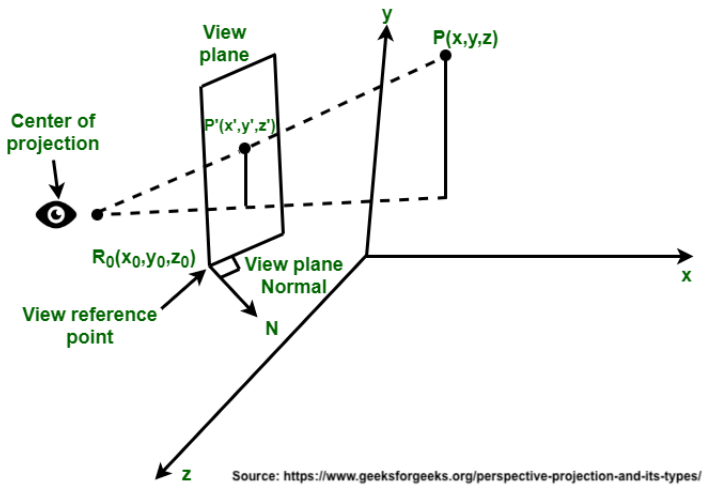


Figure 4.1: Central Projection

most cases. The camera models, represented by an especially characterized matrix, make the model clear for computational devices like computers. Hence, perspective projection in homogeneous coordinates is illustrated by *Perspective Transformation* (P), composed of a camera matrix (3x3 diagonal) and a projection matrix (3x4). This notion is also known for "central projection using homogeneous coordinates" in [49]. At this point, camera coordinates (x, y, z) can be easily defined by the transformation matrix (P) and homogeneous world coordinates (X, Y, Z, 1).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathrm{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ where } \mathrm{P} = \begin{bmatrix} f & 0 & px \\ 0 & f & py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ where f is the focal length}$$

and (px, py) is the principal point. Camera coordinates (x, y, z) are aligned to the principal axis by the camera matrix. Pixel coordinates (u, v) can be derived by z coordinate division in Figure 4.2.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ z/z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} fx & 0 & px \\ 0 & fy & py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Pixel coordinate | Homogeneous division | 3D Camera coordinates in front of camera | Camera Matrix | Projection Matrix | World coordinate

Figure 4.2: World to pixel coordinates projection

4.2 Camera Rotation and translation

World coordinates are expressed in a Euclidean coordinate frame, known as the World Coordinate Frame [49]. The projection matrix (P) is extracted into a rotation (3x3) and translation (3x1) matrix, which depicts the transformation between the world and camera coordinate frames as shown in Figure 4.3.

In Figure 4.3, O is the world coordinate frame center, and C is the camera coordinate

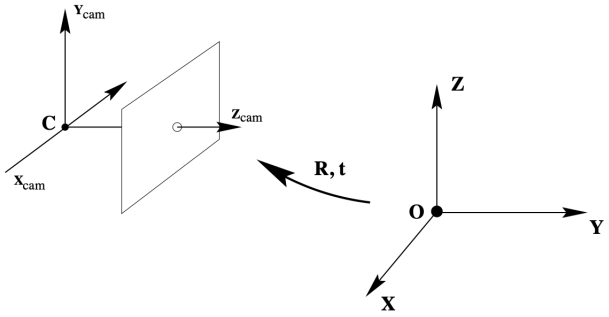


Figure 4.3: The Euclidean transformation between the world and camera coordinate frame [49]

frame center. The X point in the world coordinate frame is able to be transformed into x in the camera coordinate frame by the orientation of the camera coordinate frame (3x3 rotation matrix denoted as R) and the camera centre coordinates C (known as translation) in the world coordinate frame.

x = R(X - C) = RX - RC homogeneous representation of the transformation and corresponding pixel coordinate in Figure 4.4:

A more common and convenient representation without clearly stating the camera centre is as follows: $\mathbf{x = K[RX + t]}$, where K is the camera matrix and t = -RC. The camera matrix K is a matrix for internal camera parameters (f, px, py) with 3 degrees of freedom. The rotation matrix (R), with 3 degrees of freedom, and the camera centre in the world coordinate frame (C), with 3 degrees of freedom, are called external parameters. In total, the pinhole camera projection matrix has 9 degrees of freedom.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ z/z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} fx & s & px \\ 0 & fy & py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Point in view plane

Point in camera coordinate frame

Camera matrix (K)

R- Orientation of camera coordinate frame

C- Camera centre in world coordinate frame

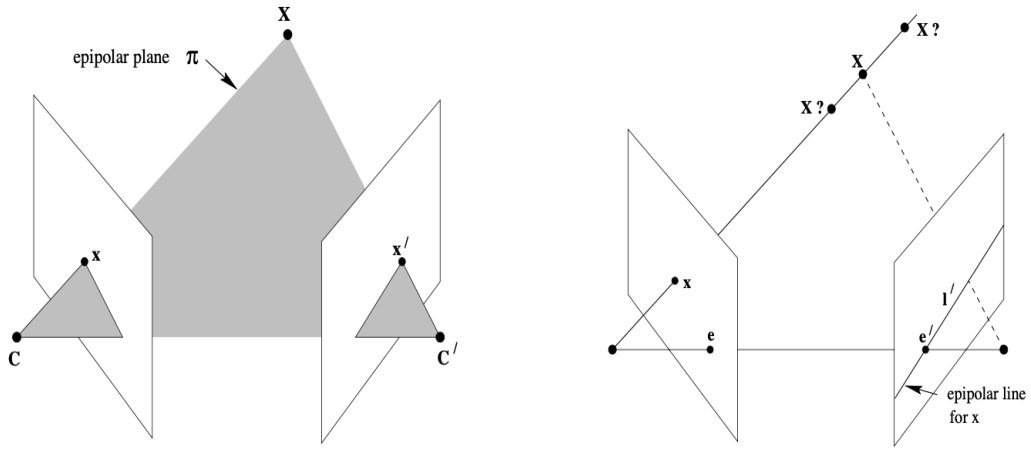
Point in world coordinate frame

Figure 4.4: Homogeneous representation of matrices

4.3 Stereo View and Epipolar geometry

Stereo view is machine vision with two cameras, the same as human eyes. It is also called a stereo rig. As the human brain perceives depth information by looking at an object in space, the stereo view is able to compute disparity and depth. This makes stereovision more advantageous than single-camera viewing. In the stereo setup, there are left and right cameras looking at the same space object from different perspectives. Hence, left and right images feature correspondences, making disparity calculations available. Figure 4.5 shows a rough stereo view setup to give an initial visual understanding of what the stereo view looks like, where point p is observed from different perspectives. In Figure 4.5, O1 and O2 are the left and

Figure 4.5: Stereo View



(a) Epipolar plane (b) Epipolar Line

Figure 4.6: Epipolar Geometry [37]

right camera centres, respectively; x1 and x2 points are the different views of the same world point p and the line connecting two camera centers called *baseline*, a fixed distance known by the installation of the two cameras. *Epipolar geometry* is a geometrical explanation of the above-mentioned stereo setup. Epipolar geometry elucidates the relation between two view planes (the image plane) and the epipolar plane, where camera center locations and points in the world coordinate frame are coplanar. Hence, this notion motivates correspondence searches between view planes. Let's look at an example: suppose there is an x point in the left image and a corresponding x' point in the right image. Also, rays emitting from camera centres through the image points x and x', known as *back projection*, intersect at some world point X in space. This means that given two corresponding points in view images, one can define a world point, which the two cameras are looking at in space. The reverse process is available through transformation in the former section. This is illustrated in Figure 4.6 (a). However, what if only one point on the left image is given, and what would be the corresponding point on the right image?.

Based on epipolar plane geometry, two correspondences lie on the back projection of the rays. Therefore, we can define the epipolar plane by the ray intersecting through the x point and baseline. If we assume that the ray is going through left camera centre C and the point x as a continuous line in space (line in 3D), the projection of each world point into the line of the right image forms a line in 2D (called the *Epipolar line*), and the correspondence can be searched through the line. This is visually explained in Figure 4.6 (b). The opposite order is also true; in the

case of correspondence, search on the left epipolar line when the right image point is given. Also, e and e' points in Figure 4.6 (b) are denoted as *Epipoles*, intersection points of baseline and view planes, left and right images, respectively. At this point, we have seen the basic geometric representation of Epipolar geometry.

However, how do machines perceive this notation? *Fundamental* and *Essential matrices* are algebraic representations, properties of matrices. The fundamental matrix is a specialized representation of the essential matrix and includes a calibrated camera matrix (K). The ray and its map into a two-dimensional image plane are point-to-line projective mappings, represented by the fundamental matrix (F). It assumes camera centers at different locations. Unique seven-degree-of-freedom 3x3 rank-2 homogeneous matrix, which satisfies $\mathbf{x'Fx} = \mathbf{0}$ on the corresponding points x and x', is called the fundamental matrix. Properties of the Fundamental Matrix:

- **Correspondence**: $\mathbf{x'^{T}Fx}=\mathbf{0}$ if x and x' are corresponding image points
- **Epipolar lines** on two view planes defined as $\mathbf{l'} = \mathbf{Fx}$ and $\mathbf{l} = \mathbf{F}^{T}\mathbf{x'}$, l is epipolar line corresponding to x' and l' is epipolar line corresponding to x
- **Epipoles** satisfy $\mathbf{Fe} = \mathbf{0}$ (right null vector of F) and $\mathbf{F}^{T}\mathbf{e'} = \mathbf{0}$ (left null vector of F).
- **Transpose**: If the fundamental matrix pair of the projection matrix (P, P') is F, then F^{T} is the matrix for the opposite pair (P', P). Here P is the projection matrix for the left view and P' for the right view.

The essential part of the fundamental matrix is the formation of relative motion and its camera parameters. In other words, it is able to be extracted into motion matrices and camera matrices, namely rotation (R), translation (t), and camera matrices K (left camera matrix) and K' (right camera matrix). The formulated elaboration of the properties is shown in Figure 4.7. In the book [42], **Essential matrix** is described as a specialization of the fundamental matrix. This means normalized coordinates ($\hat{x}$ and $\hat{x}$') utilized instead of the original x and x' points. So, the normalized coordinates and normalized camera matrix notions are introduced in Figures 4.8 and 4.9. Hence, **Essential matrix** (E) is a fundamental matrix expressed by the normalized camera matrix, and it has the form $E = [t]_{x}\mathrm{R} = R[R^{T}t]_{x}$. Properties of Essential Matrix E:

- $\hat{\mathbf{x}}'^{T}\ \mathbf{E}\ \hat{\mathbf{x}} = \mathbf{0}$, known as **epipolar constraint** if $\hat{x}$ and $\hat{x}$' are corresponding normalized coordinates of x and x'

Computation from camera matrices $\mathtt{P}, \mathtt{P}'$:

- ◇ General cameras,
  $\mathtt{F} = [\mathbf{e}']_\times \mathtt{P}'\mathtt{P}^+$, where $\mathtt{P}^+$ is the pseudo-inverse of $\mathtt{P}$, and $\mathbf{e}' = \mathtt{P}'\mathbf{C}$, with $\mathtt{P}\mathbf{C} = \mathbf{0}$.
- ◇ Canonical cameras, $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$,
  $\mathtt{F} = [\mathbf{e}']_\times \mathtt{M} = \mathtt{M}^{-\mathsf{T}}[\mathbf{e}]_\times$, where $\mathbf{e}' = \mathbf{m}$ and $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.
- ◇ Cameras not at infinity $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}' = \mathtt{K}'[\mathtt{R} \mid \mathbf{t}]$,
  $\mathtt{F} = \mathtt{K}'^{-\mathsf{T}}[\mathbf{t}]_\times \mathtt{R}\mathtt{K}^{-1} = [\mathtt{K}'\mathbf{t}]_\times \mathtt{K}'\mathtt{R}\mathtt{K}^{-1} = \mathtt{K}'^{-\mathsf{T}}\mathtt{R}\mathtt{K}^{\mathsf{T}}[\mathtt{K}\mathtt{R}^{\mathsf{T}}\mathbf{t}]_\times$.

Figure 4.7: Fundamental Matrix Property [37]

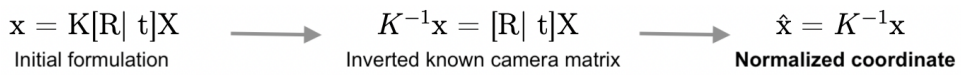


Figure 4.8: Normalization coordinates

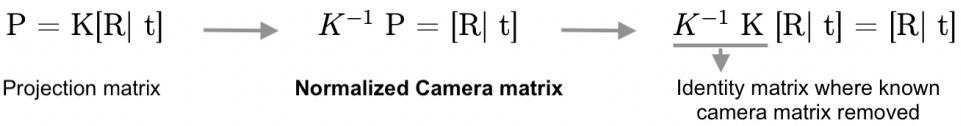


Figure 4.9: Normalized Camera matrix

- **Homogeneous** representation like Fundamental matrix
- E is a 3x3 essential matrix if and only if the E matrix has **Singular Value Decomposition (SVD)** with a diagonal matrix D = (1,1,0).

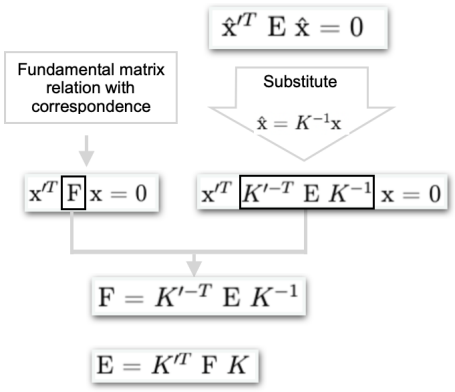


Figure 4.10: Essential and Fundamental matrix

The relationship between fundamental matrix F and essential matrix E is explained in Figure 4.10. Here, Essential and Fundamental matrices encapsulated corresponding feature points x' and x in Fundamental matrix F and normalized coordinates x̂ and x̂' in Essential matrix E. Now, let's see the computation of the fundamental matrix, which is able to produce the essential matrix. Fast and efficient fundamental

matrix estimation algorithms are one of the most interesting fields among computer vision researchers. Therefore, the essential matrix can be extracted with known camera matrices. *Seven-point Fundamental Matrix* [50] estimation algorithm summarized in Figure 4.11. The essential matrix can also be estimated using the same concept as the algorithm. To summarize the algorithm, it elaborates the term x'^{T} F x = 0 (1) by homogeneous coordinates of the corresponding seven image points x and x'. Then, simple matrix multiplication linearly forms Af = 0, where A is a 7x9 matrix and f is a stacked (column matrix) unknown parameter of the fundamental matrix F. The equation (1) takes into account def(F) = 0 and F = $\alpha F1 + (1-\alpha)F2$, α is an unknown scalar constraint. The determinant of such an equation forms a cubic polynomial. A cubic polynomial has one or three solutions. Therefore, the best solution is found by a minimal line and point distance.

4.4 RANSAC

RANdom SAmple Consensus (RANSAC) uses a small possible set as initialization, then consistent data is added to the initial set when possible. For example, in the experiment on [51], Assumed line fitting problem in the paradigm of RANSAC. Given a set of points and a number of random points to be chosen as a subset, we can assume that the number of random points set to two due to the line can be found by connecting two line segments. Then, the task is to find the best-fitting line model along the set. Initially, choose two points from the given set, which will form a line. Next, find the distance between the line and the remaining set of points. If the distance meets the threshold, keep points that satisfy the condition, denoted as inliers. If not, it is called an outlier. This process repeats until it has reached the iteration number computed based on known constraints, the number of random points, the inlier and outlier ratios, and confidence. Algorithm 1 shows pseudocode for the RANSAC algorithm [38]. As an input to the algorithm, it takes the minimum number of data points, the dataset designated for the model, and the maximum number of iterations to find the consensus set calculated by the described formula in Section B of [38]. As an initialization of the algorithm, a randomly chosen subset contains the number of data points and the model of the subset defined in lines 3 to 6. Then, over the maximum number of iterations, the gross error in the set of data points P is calculated in line 8, and inliers are counted among the errors on threshold t.

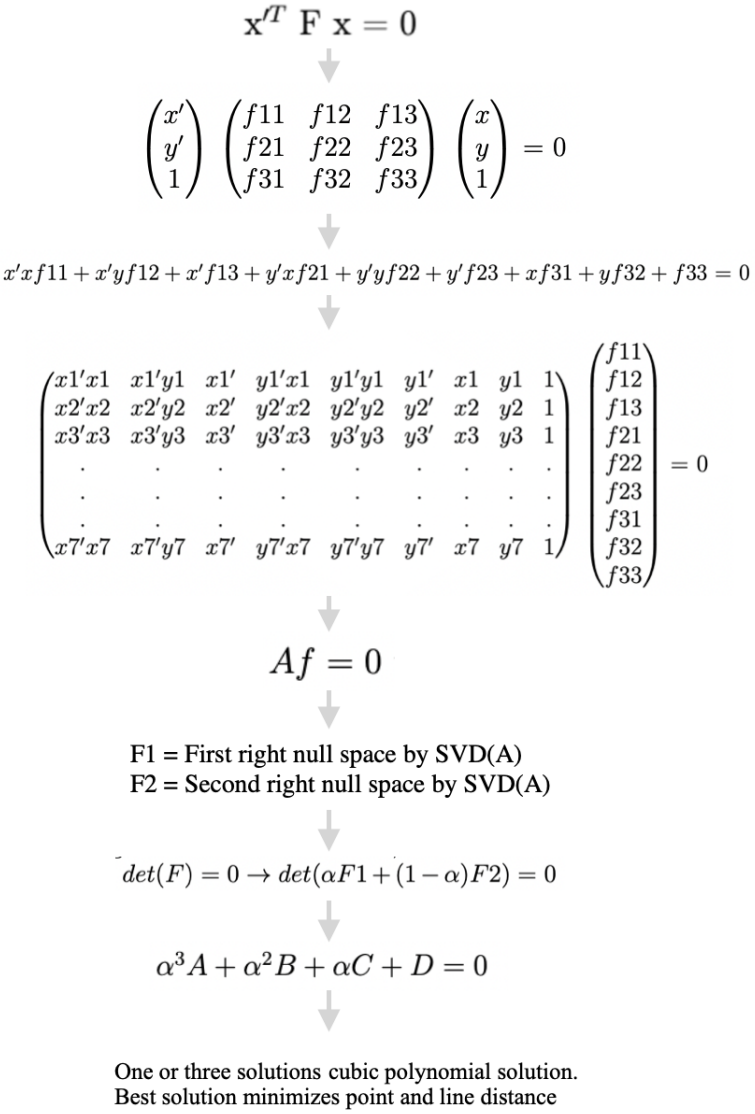


Figure 4.11: Seven-points Algorithm

Also, the highest number of inliers along the subsets becomes the best consensus set. Lastly, the next subset of the data points is updated accordingly in lines 14 to 16.

4.5 Perspective-N-Point Problem

As described in [6], Perspective-N-Point problem is the Local Determination Problem (LDP), which is location determination in space by control points observed in the image. The control points refer to the recognition of a set of common points in images. This problem was solved with a least-squares solution alongside manual setup for corresponding image points and 3D control points [52] [53].

Algorithm 1 RANSAC

Input: n (minimum of data points), P (set of data points), t (threshold), it (number of iteration)
Output: M1* consensus set and Best Inliers
$index \leftarrow$ random(P, n)
$S1 \leftarrow$ P[$index$]
$M1 \leftarrow$ model($S1$)
$Inlier_{best} \leftarrow 0$
for $iteration = 1, \ldots it$ **do**
    $S1^* = gross_{error}(P, M1)$
    $in \leftarrow count_{inlier}(S1^*, t)$
    if $in > Inlier_{best}$ **then**
        $Inlier_{best} \leftarrow in$
        $M1^* = M1$
    end if
    $index \leftarrow$ random(P, n)
    $S1 \leftarrow$ P[$index$]
    $M1 \leftarrow$ model[$S1$]
end forreturn $M1^*$, $Inlier_{best}$

However, nowadays, those feature correspondences are automatically computed and produce errors, which is not possible with the least-squares method. The paper [6] described the Perspective-N-Point problem mathematically as *given n spatial control points and every angle between n control points and an extra point, called the Center of Perspective (CP), find lengths of line connecting the controls and CP.* Here, the solution utilized RANSAC (RANdom SAmple Consensus), which was elaborated in the former section. 1, 2, 3, and 4 are considered as **n** to get a minimal solution. In the Perspective-1-Point and Perspective-2-Point cases, n equals 1, 2, and the paper states the infinity solution. However, Perspective-3-Point problem has three systems of equations with at most four positive solutions. According to [6], there are three steps involved in solving the three systems of equations. First, find the length of connecting lines, CP to control points. Second, place control points related to specified control points, which form a 3D reference frame. Lastly, the calcula-

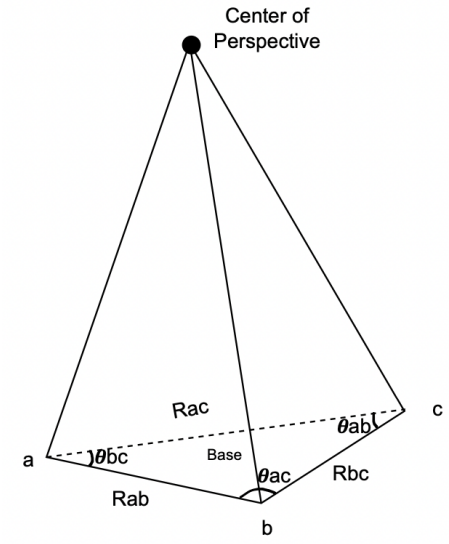


Figure 4.12: Tetrahedron formation

tion of the orientation of the image plane with respect to the reference frame. The length of connecting lines is solved by a polynomial equation, which will be explained in the following sentences. Three control points and the Center of Perspective form a tetrahedron, depicted in Figure 4.12, and given the length of the base (Rab, Rac, Rbc) and trihedral opposing angles (θbc, θac, θab), the problem takes into account finding the remaining sides of the tetrahedron (a, b, c), where each connection of CP is solved by a four-root polynomial equation. The extraction of polynomials is structured in each step in Figure 4.13. The paper [6] suggested finding a polynomial solution with an iterative approach.

4.6 Triangulation

Problems with point correspondence: As described in the former section, point correspondence searches through epipolar lines, x point correspondence looks through l' = Fx line in the other image, and x' searches on $l = x'^T F$. Fundamental matrix estimated with given correspondence's covariance matrix [54] so that the point matches able to lie around the line, not exactly on the line. Therefore, *envelope of epipolar lines* determines the searching region by the covariance matrix of the fundamental matrix to make searching more broad. As a result, it is possible to get several lines for a point at a certain noise level. Moreover, point correspondence instability is problem that does not depend on a specific algorithm. The reasons can be caused by first, given point correspondences when estimating the fundamental matrix, or second, a point close to the epipole. If most of the correspondences are in the background of the image, foreground point epipolar lines are unlikely to produce confident correspondence. This shows that initial corresponding points are crucial for fundamental matrix estimation, on which epipolar line extraction depends. Another instability problem is matching points close to the epipole, which results in an in an unstable epipolar line due to the uncertain location of the epipole.

One way to tackle the above problem is **rectification**. The rectification process resamples pairs of images by inducing epipolar lines to be parallel. In other words, rectification is a projection in which a specific transformation yields a resample where horizontally parallel epipolar lines are depicted so that the disparity between the images is only in the x direction. Visual interpretation is shown in Figure 4.14. Figure 4.14 shows the rectification process, starting from the fundamental matrix

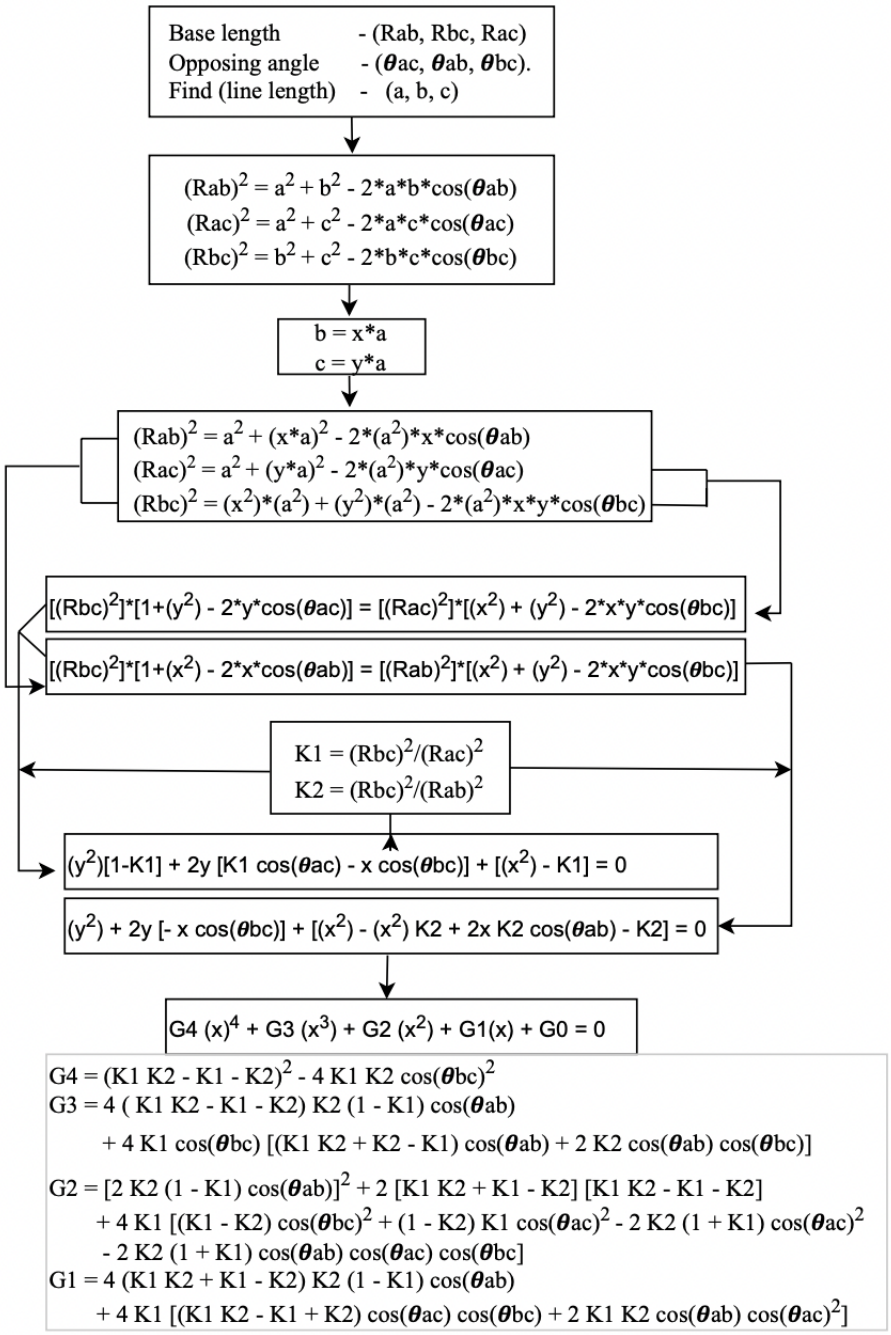


Figure 4.13: Perspective-3-Point problem polynomial equation

representation of two image pairs and then applying a 2D projective transformation to the right image by translation. T is translation from a point to the origin; rotation R rotates epipole e' to a point (f', 0, 1) around the origin. G is a mapping matrix, maps epipole e' to infinity. After transforming the right image by H', the matching transformation H is necessary to be computed under the condition $H = H_0 H_a$ such that the minimization function becomes a linear least-squares parameter minimization problem. The produced H and H' then bring a rectified image pair. Appropriate

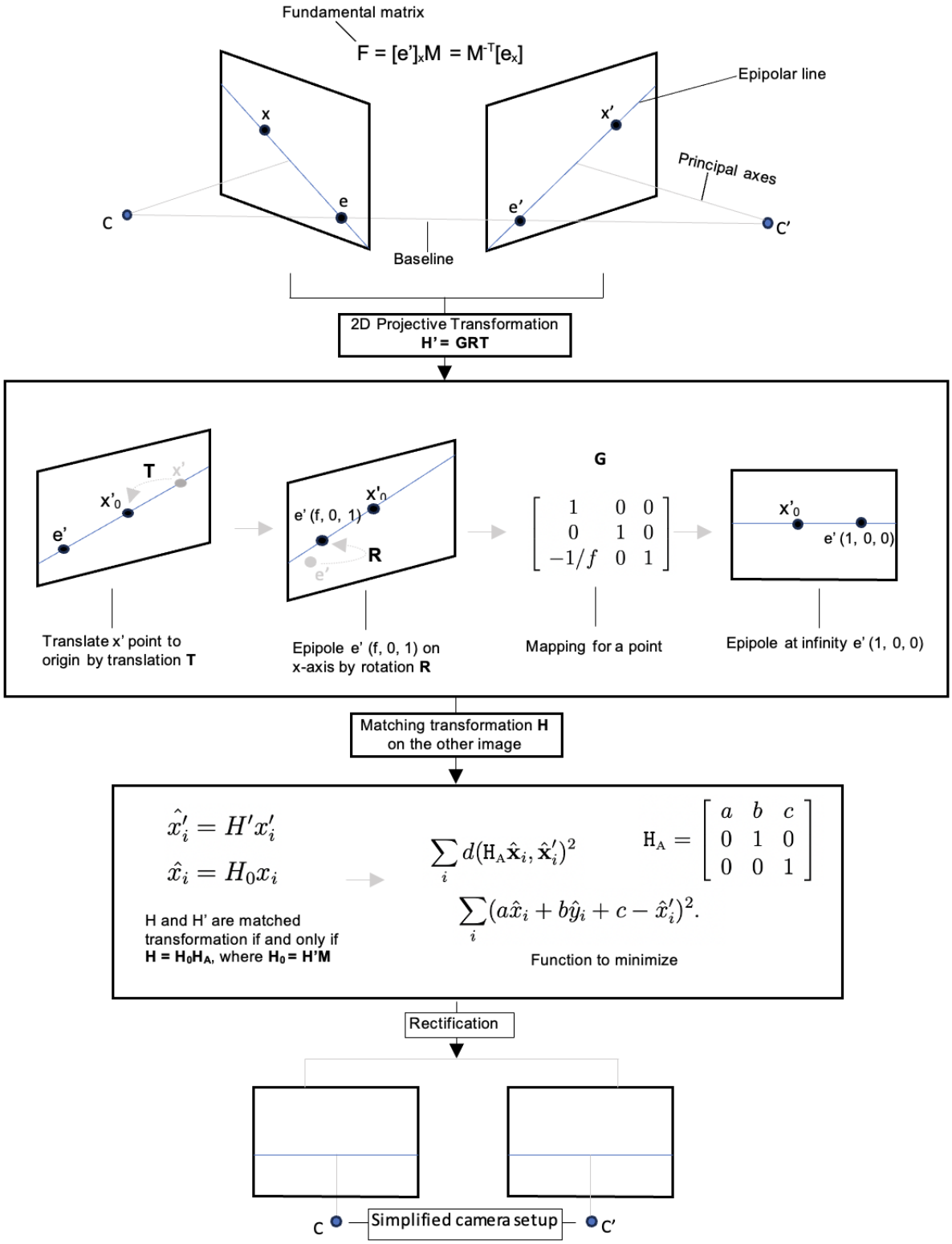


Figure 4.14: Rectification process

2D projective matrix H is essential in order to apply less distorted transformation results, images like the original. Such a rectification problem makes the corresponding problem simpler by searching for correspondences only on the horizontal line.

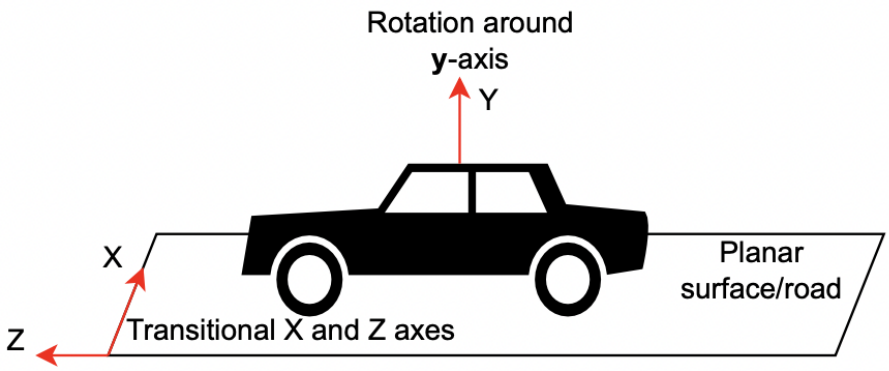


Figure 4.15: Motion on planar surface

4.7 Planar Motion

One part of computer vision deals with the **Planar Motion** (also known as *ground plane motion GPM*) problem, in which how an object moves on a surface described. There is a notion of planar object motion [55]; however, Car or robotic motion estimation under a planar constraint, with a with a ground plane perpendicular to the height vector of the vehicle and parallel to the moving direction, is what this thesis work denotes as planar motion. As the camera is the main observer of the environment, a moving vehicle equipped with a camera or cameras moves with the vehicle. Vehicle forward movement and turns become the camera's movement around horizontal axes and rotation along the vertical axis, respectively [56]. The algorithms mentioned in previous sections, one point [4] and two points [5] pose estimation method, use this notion of planar motion. Considering the car on the road is moving on the planar surface and the plane described by the z and x axes of the camera coordinate system, the car is also rotating around the y axis, as depicted in Figure 4.15.

Chapter 5

Methodology

In this thesis work, spatial points are generated by optical flow-based feature extraction techniques combined with vehicle ego-motion estimation. Accurately estimating the visual vehicle trajectory is the main focus of the reconstruction task. Therefore, two methods were utilized to estimate vehicle pose estimation. One method uses disparity maps for three-dimensional coordinate generation, and another produces world coordinates by triangulation. The reason behind generating world coordinates by different methods is that common benchmarks used in evaluation have clarifications, such as undistorted images and rectification. However, there are datasets that have only the original dataset available. For this type of assessment, the thesis work implemented a separate method to generate world coordinates. With that said, feature matching is available to be extracted by world coordinates and corresponding image points. Afterwards, outlier removal algorithms [4] [5] and Perspective-3-Point [6], 5 points[7] pose estimation algorithms employed to estimate vehicle ego motion, also called visual odometry, as discussed in the Related Work section. The main steps utilized in this work are adopted from [4] [11] [57], which are feature extraction, outlier removal, and pose estimation. Figures 5.1 and 5.2 depict both the different methods called "KLT and Optical Flow" and "SIFT and Optical Flow." This enabled the application of different algorithms used in monocular and stereo images. Moreover, pose estimation is an iterative process that localizes vehicles at each time, such that the depicted steps in Figures 5.1 and 5.2 are computed for all consecutive image pairs.

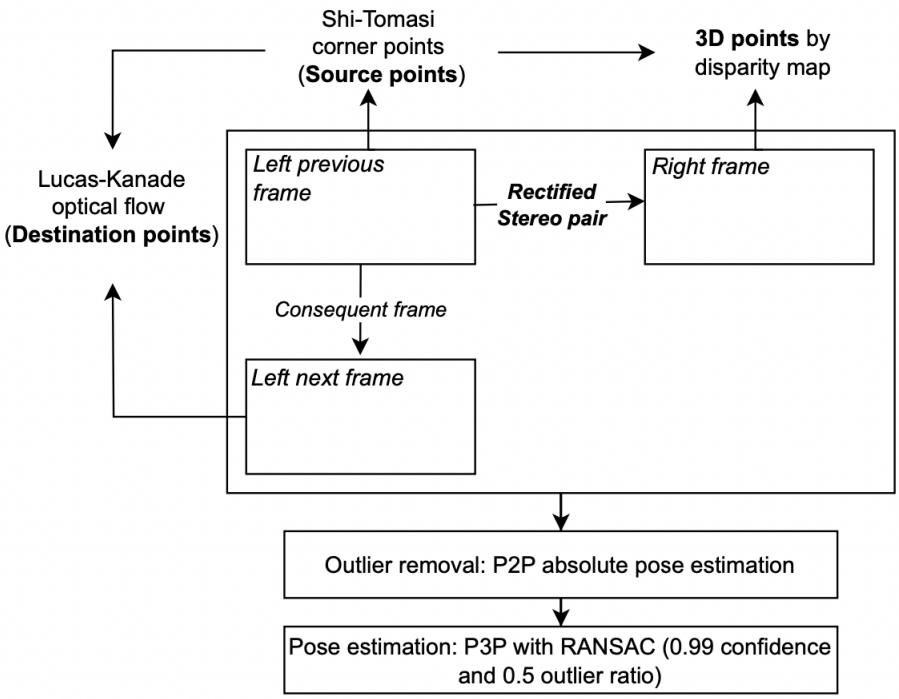


Figure 5.1: KLT and Optical flow

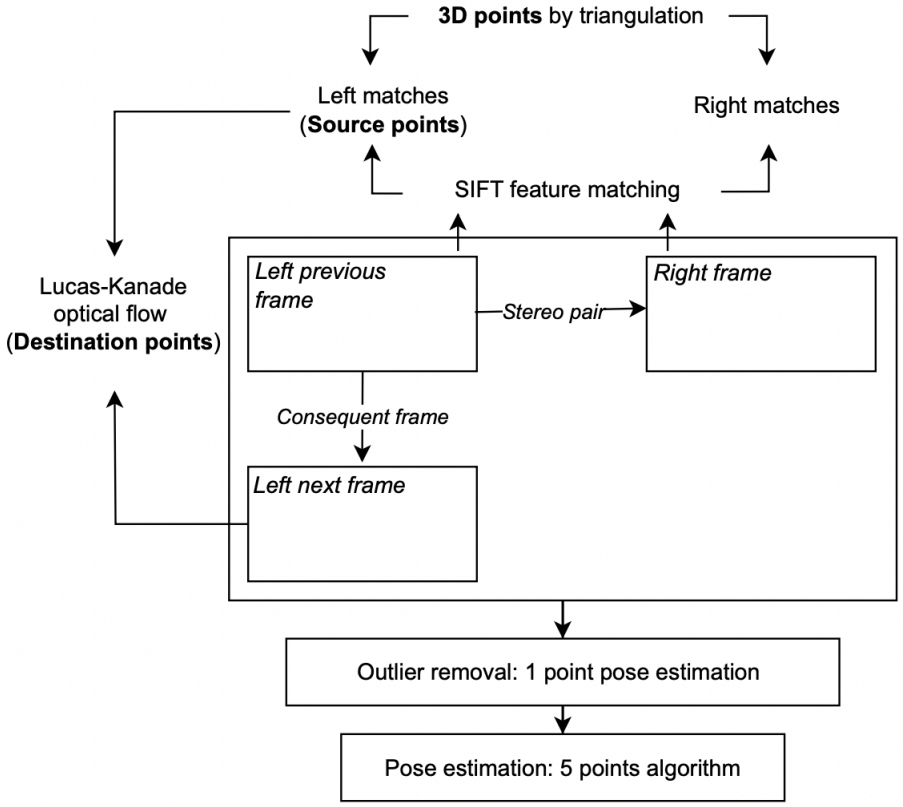


Figure 5.2: SIFT and Optical flow

5.1 KLT and Optical flow

In this way, features are extracted by rectified stereo pairs and consequent frames. The advantage of using a rectified stereo pair is that matched points (a pair of points) can be found on the horizontal axis of the image plane (view plane). Another available property for rectified pairs is acceptable disparity map estimation with block matching technique [58]. Using this knowledge, three sub-steps have been taken to find the corresponding world and image coordinates. First, Shi-Tomasi corner detection [59], a small modified version of Harris corner detection and also called Good Features to Track (GFTT), named after the proposed paper in the OpenCV library, applied the left previous frame as illustrated in Figure 5.1, and it became the first pair of feature matches, called **source points** in the illusion. Then, according to the source points found, **spatial coordinates** were estimated by the disparity map of the rectified stereo setup. Lastly, again with the source points Lucas-Kanade optical flow [60] tracked over the left previous and next frame, this yields **destination points**, in other words, the second pair of the feature match. At this point, feature matching has been done with consequent image corresponding points (source and destination) and previous frame spatial coordinates. After generating feature matches, destination points and 3D points are utilized to estimate pose estimation as follows: First, a two-point absolute pose estimation algorithm under planar motion[5], wrapped up in Figure 5.3, combined with RANSAC, was adopted to remove mismatching points in world (3D) to image (2D) coordinate correspondences. The algorithm assumes In the camera matrix (K), two corresponding 3D and 2D points are given under the planar motion constraint, where the vehicle rotates around the y-axis and moves in the x and z planes. Its representation is as follows:

$$\mathrm{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathrm{R} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad \mathrm{t} = \begin{bmatrix} t_x \\ 0 \\ t_z \end{bmatrix}$$

and $\mathrm{E} = [t]_x \mathrm{R}$, where $[t]_x$ is a skew-symmetric matrix of translation t and R is rotation around the y-axis. At this point, projection matrix M can be rearranged with camera matrix K and essential matrix E:

$$\mathrm{M} = \begin{bmatrix} f_x \cos\theta - c_x \sin\theta & 0 & f_x \sin\theta + c_x \cos\theta & f_x t_x + c_x t_z \\ -c_y \sin\theta & f_y & c_y \cos\theta & c_y t_z \\ -\sin\theta & 0 & \cos\theta & t + z \end{bmatrix}, \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathrm{M} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image coordinates (u, v, w) and homogeneous world coordinates (X, Y, Z, 1) relate to the projection matrix M as shown above. This can form AW = B in Figure 5.3. By taking each column of A, which denotes $\sin\theta$, $\cos\theta$, t_x and t_z respectively, A'W' = B' can be formed, and the solution of the inhomogeneous linear equation produces the rotation angle θ and translation vector. Therefore, pose estimation is derived by substituting the angle and translation into matrix M, which forms the P matrix in the figure. As a last step, pose estimation was solved by the Perspective-3-Point problem solution [6] with RANSAC 0.99 confidence and a 0.5 outlier ratio.

5.2 SIFT and Optical Flow

In contrast to "KLT and Optical Flow," the "SIFT and Optical Flow" method matches features between unrectified stereo pairs. Hence, spatial coordinates can be estimated by corresponding feature matches. The sub steps are feature matching and detection, spatial coordinate estimation, and optical flow tracking. First, SIFT [21] keypoints are generated on both left and right images of the stereo pair; then, FLANN matcher [61] correlates the keypoints with fast nearest neighbor search to get to corresponding pairs, and a 0.7 distance ratio is applied between the selected corresponding pair. Left-matched points from corresponding image matches become **source points** due to those being the left-image feature points. Second, **Spatial coordinates** are estimated by triangulation of left and right correspondent keypoints. To get minimal reprojection error points, the method compared linear, iterative, and optimal triangulation estimation methods elaborated in [29]. Lastly, **Destination points** tracked along with source points and left consequent frame using Lucas-Kanade[60] optical flow implementation. As a result, we got the corresponding image points (source and destination) from the consequent frame and spatial coordinates. In this case, vehicle motion ego-motion is estimated on corresponding feature points from the left previous frame and the left next frame. Such

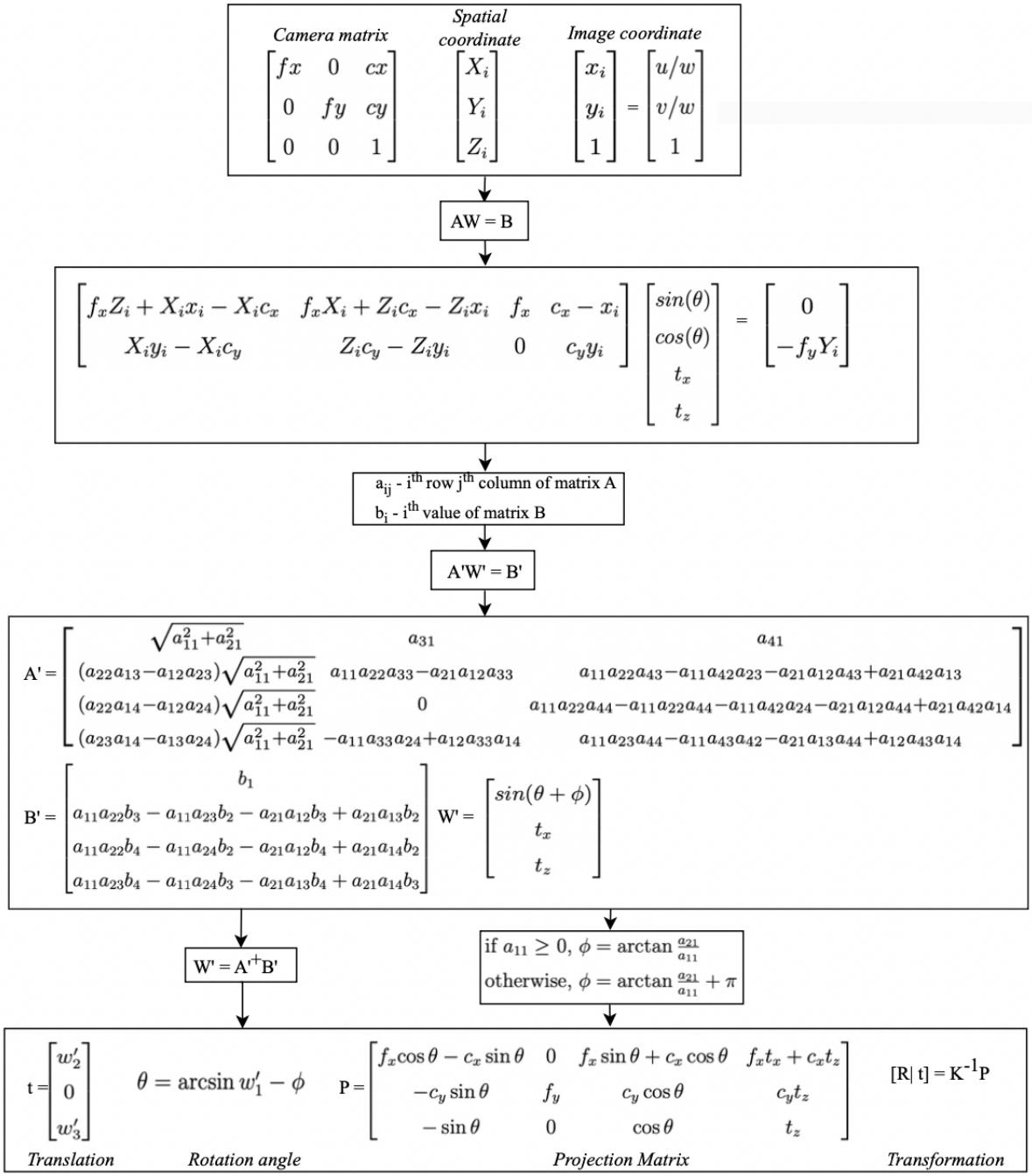


Figure 5.3: Two Point Absolute Pose Estimation under Planar Motion

that, the outlier removal one-point algorithm [4] was applied to get less contaminated feature correspondences. The one-point algorithm is visualized as a scheme in Figure 5.4. The algorithm is designed for a circular planar motion model with only one-point correspondences, which is considered enough to estimate the rotation angle. In this thesis work, the one-point algorithm was utilized to remove outliers by estimating rotation angle; hence, an estimated transformation matrix was used to remove mismatched outliers from all the feature correspondences. Vehicle pose

estimation, computed by the 5-point algorithm [7], estimates vehicle motion under the Essential Matrix (E) Epipolar Constraint and known camera matrix (K) as follows: $\mathbf{K^{-T}[t]_x R K^{-1}} = \mathbf{0}$. Such that the essential matrix, representation of rotation (R) and translation (t), estimated at $\mathbf{p_2^T K^{-T} E K^{-1} p_1} = \mathbf{0}$ by corresponding feature pairs p_1 and p_2, left previous and next image feature correspondences, respectively.

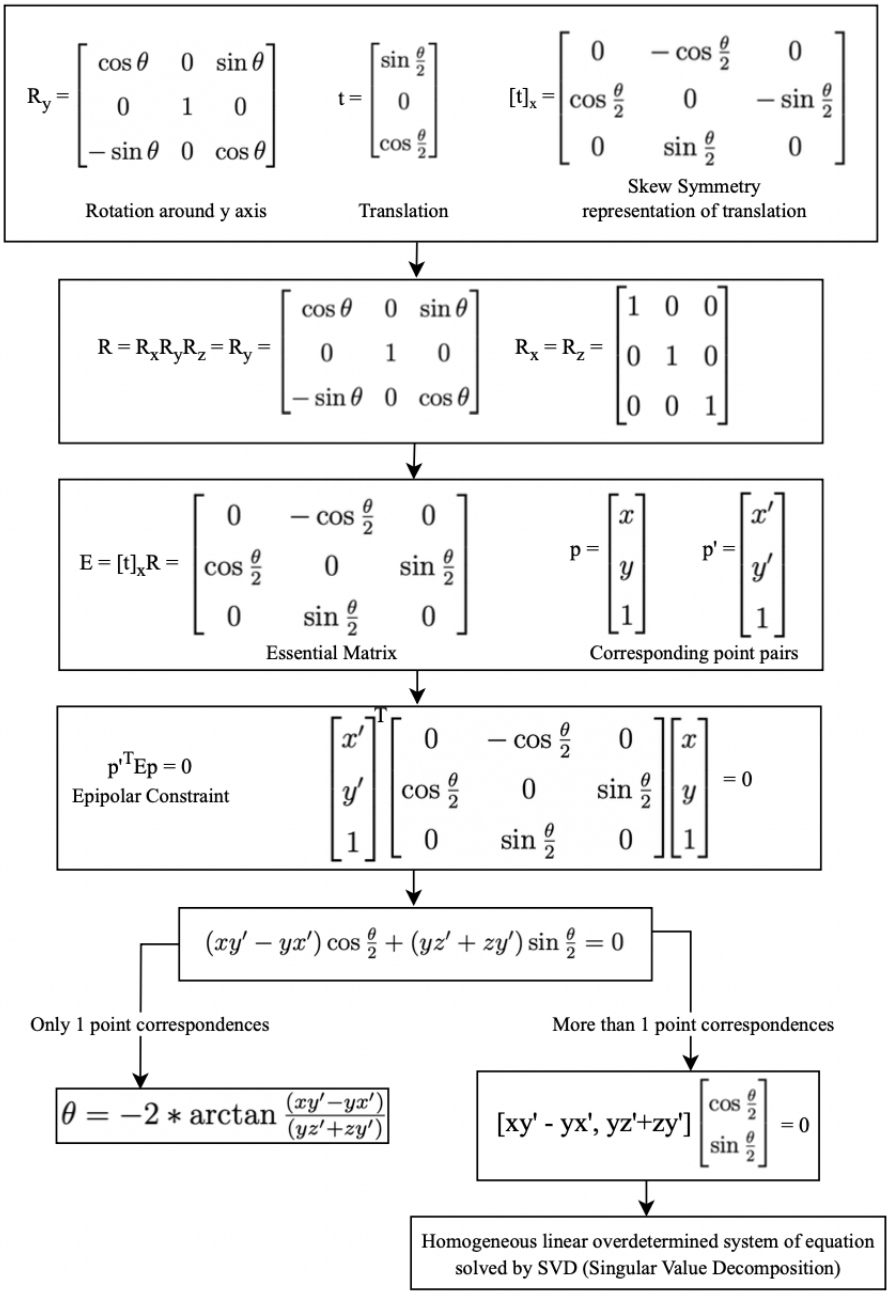


Figure 5.4: One point relative motion estimation under planar motion

Chapter 6

Experiment

The SIFT and KLT-based methods are assessed on the KITTI benchmark and the ELTE car dataset. Both datasets are real vehicle-equipped environments with available stereo pairs, and cameras are mounted on the top of the cars. Moreover, both "KLT and Optical Flow" and "SIFT and Optical Flow" methods were tested separately on the KITTI and ELTE car datasets, respectively. However, there are some evaluation results that have been combined with both methods. As explained in the methodology section, this work solves the visual odometry problem using different methods on rectified and unrectified datasets. The unrectified images are able to be corrected with rectification methods in order to generalize algorithms, but transformation changes in rectified images would worsen 3D world coordinate estimation. In this case, a more precise rectification method could apply in the original consecutive frames. The ELTE car dataset mounted a LIDAR and four HikVision/HikRobot MV-CA020-20GC cameras. The cameras are attached to the top of the car around the front wheel axle and placed relative to the forward axis at a 20 to 60 degree angle, which means each camera takes monocular images at a time. The camera setup

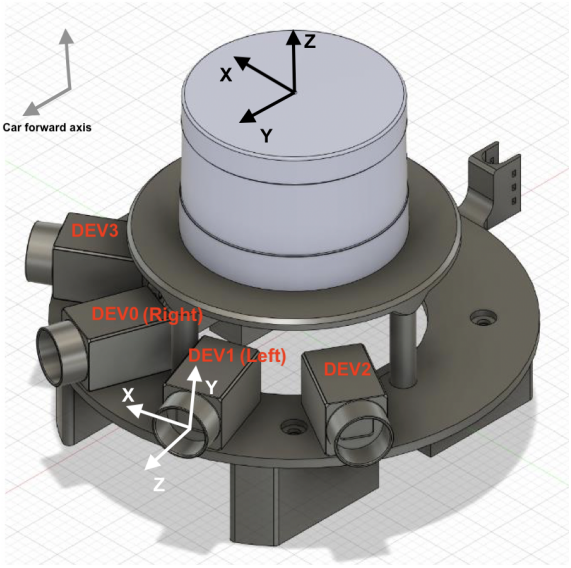


Figure 6.1: ELTE car camera setup [62]

depicted in Figure 6.1 was imitated from sensor pack documentation. Camera and Lidar directional axes are illustrated in white and black. Sensors are measured with respect to the Lidar coordinate frame, depicted in black, and considered as the world or reference coordinate frame. Also, the camera's standard setups are the following: frame rate of 4FPS, 1920x1200pixel resolution, and 4.8μmx4.8μm pixel size. In this experiment, two middle camera images were considered stereo image pairs, as shown in Figure 6.1. The two middle cameras are DEV1, left pair, and DEV0, right pair of stereo. Therefore, those images are suitable for the "SIFT and Optical Flow" method to triangulate corresponding points between stereo pairs. Noise-reduced inlier points are detected using the one-point algorithm under circular planar motion [4] and as proposed in the paper, the optimal rotation angle is found using the no iteration histogram approach; the optimal value of the histogram taken by the median, called histogram voting. Therefore, pose estimation with the five-point [7] algorithm becomes more reliable with noise-reduced feature points. The input of the outlier and pose estimation algorithms are the corresponding feature points of consecutive frames. Evaluation of the method assessed with the vehicle ground truth trajectory produced by the GPS sensor.

KITTI Visual Odometry/SLAM Evaluation 2012 has rectified 22 stereo sequences; however, only 11 sequence ground truths are given to assess the method. Therefore, experiments have been done on the sequences with ground truth. In this set of data, the "KLT and Optical Flow" model is able to be utilized to find valuable 3D points from disparity maps. The world coordinates from the disparity map were calculated as follows: ($\mathrm{Z} = \frac{b*f}{d}$, $\mathrm{X} = \frac{(x-c_x)*Z}{f}$, $\mathrm{Y} = \frac{(y-c_y)*Z}{f}$) where b is baseline, distance between camera centers, f is camera focal length, (c_x, c_y) is principal points, and d denotes the disparity of the image point (x, y). Only spatial coordinates corresponding to source points are estimated using the above formula. Disparity map production uses the stereo semi-global block matching technique [58] which computes on 96 disparities, 16 blocks, and three-sized windows. During the spatial point generation, reprojection error resulted close to 0, which convinced accurate world coordinate calculation to be evaluated later in outlier removal and pose estimation. As an outlier removal, 3D and 2D points from the left next frame, correspondences applied in the two-point absolute pose estimation algorithm, proposed in [5], adopted with RANSAC with 0.99 confidence and a 0.7 outlier ratio. Less contaminated point

correspondences are applied to the Perspective-3-Point solution [6] to find vehicle motion estimation.

The experiments are mainly focused on methods separately, however, it is possible to test the "SIFT and Optical Flow" method in rectified datasets, whereas the "KLT and Optical Flow" method is not that suitable to be tested on original images. datasets containing unrectified images due to less accurate matches are highly possible to find. Moreover, a trajectory has been drawn under pose estimation computation. Hence, results visualized up to scale and errors accumulated in transformation at each time are not corrected with loop closure. In the next section, experiment results are depicted, starting with "KLT and Optical Flow" results compared with "KLT with One Point Outlier Removal and Five Point Pose Estimation," followed by "SIFT and Optical Flow" on the ELTE car and KITTI datasets.

Chapter 7

Results

All the experiment results are listed as figures below. Figures 1 to 11 show the comparison results of "KLT and Optical Flow" and "KLT feature correspondences with a five-point pose estimation algorithm." Then, Figure 12 shows the results from ELTE car images assessed with "SIFT and Optical Flow." In Figures 13 to 17, "SIFT and Optical Flow" are evaluated on the KITTI dataset.

In Figure 7.1, from the comparison, one-point noise reduction with a five-point algorithm corrected the trajectory better than the perspective-N-point solution. However, the perspective-N-point solution scaled the trajectory closer than the five-point algorithm. Even though a small brightness change was observed in the consecutive frames. During the experiment, some sequences with high-brightness

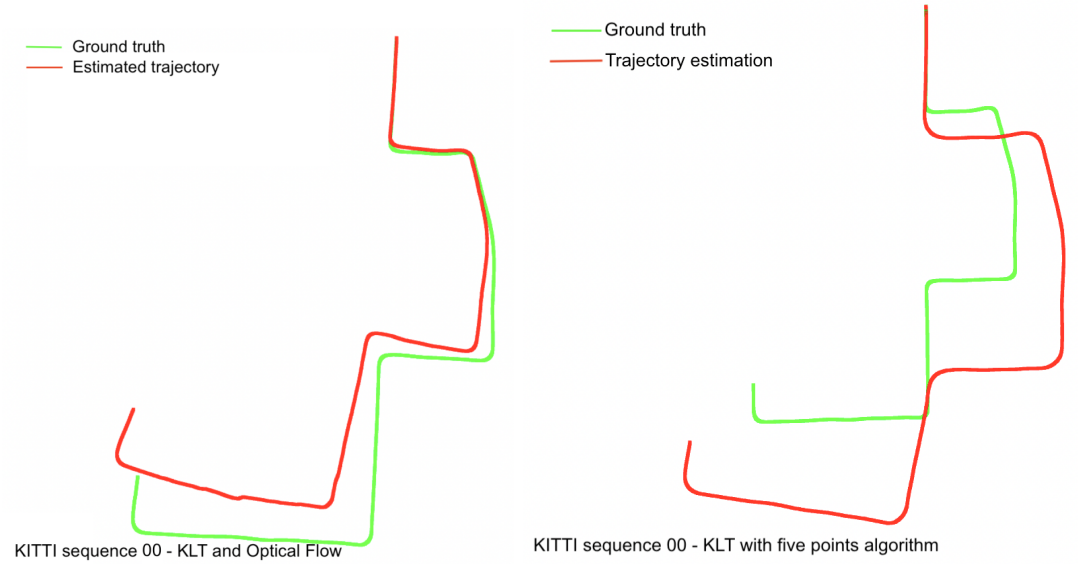


Figure 7.1: KLT results comparison

consecutive pairs affected the result accuracy. Figure 7.2, sequence 01, was one of the examples where, after first rotation, a few frames do have high brightness, and the effect can be seen clearly in the right result of Figure 7.2, where the direction has changed. In Figure 7.3, sequence 02, the luminous environment of the taken images

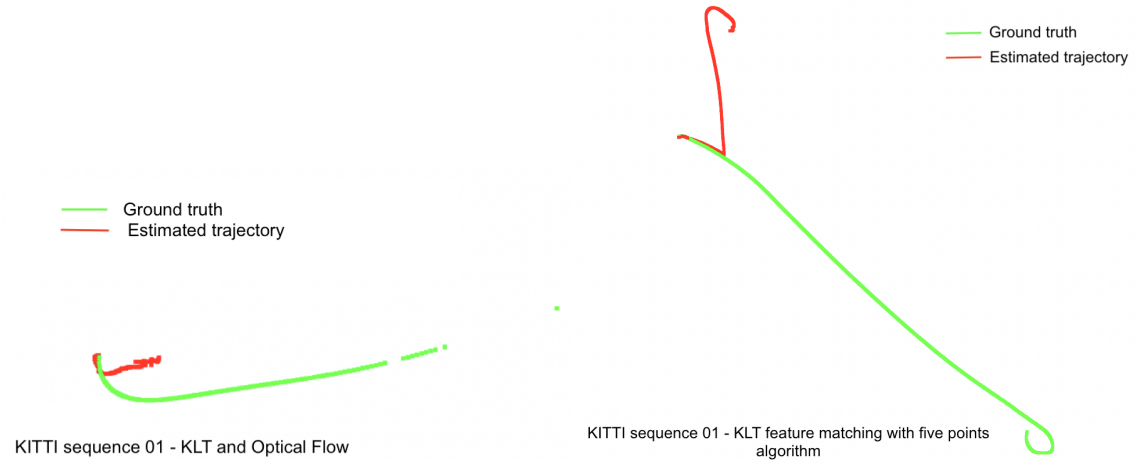


Figure 7.2: KLT results comparison

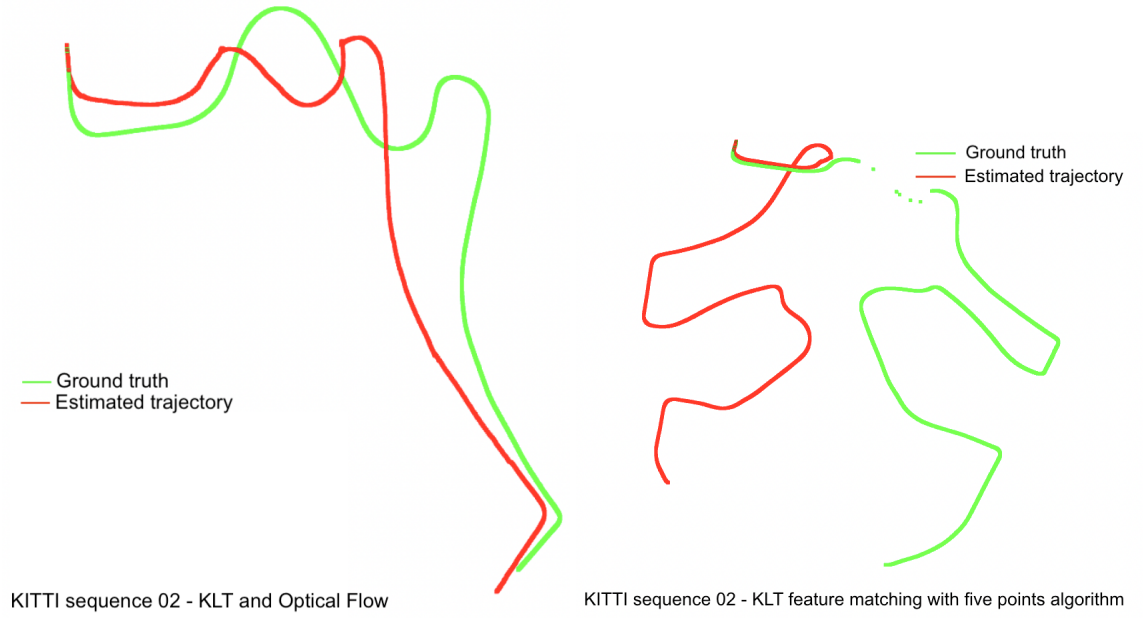


Figure 7.3: KLT results comparison

mostly occurred in frames. KLT with the Perspective-N-Point solution worked well in turns compared to the five-point algorithm in the case of the RANSAC iteration, fixed to enough numbers for both outlier removal and pose estimation steps. The right side of the figure shows the full trajectory of the sequence, where a five-point

algorithm estimates motion in different directions when curvy motion happens.

In the next sequence 03, Figure 7.4, both solutions shaped a trajectory close to

Figure 7.4: KLT results comparison

the ground truth. However, the Perspective-N-Point scale was more accurate than the five-point solution. Moreover, the brightness of the environment occurred and was corrected with enough fixed iteration applied for RANSAC implementation. In the case of pure translation, the five-point algorithm performs better than Perspective-N-Points, as shown in Figure 7.5. Perspective-N-Point solution motion

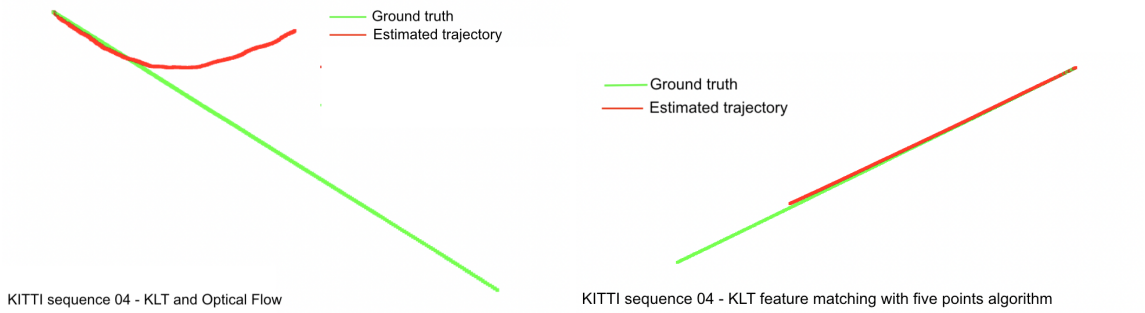


Figure 7.5: KLT results comparison

estimation error accumulated over time in the long straight run. In sequence 05, Figure 7.6, the images were partially filled with bright features. Hence, five points algorithm estimated turns better than Perspective-N-Point solution. Less accurate rotation estimation accumulated and affected the result. However, it is possible to be corrected with a sufficient number of RANSAC iterations. The sequence 06 shown in Figure 7.7 was a challenging path for both solutions in terms of translation and brightness and was not highly disturbed by the consecutive frames. Due to poor

pure translation in long run resulted inaccurate trajectory in Perspective-N-Point solution, whereas, five points solution perform better in pure translation. Here, the closed form of trajectory is not implemented and is not expected to yield an exact closed loop in trajectory. The experiment shows that a sufficient number of RANSAC iterations produces quite the same result as the default setup.

Figure 7.8: Environment images were taken in a shiny situation. Also, ground

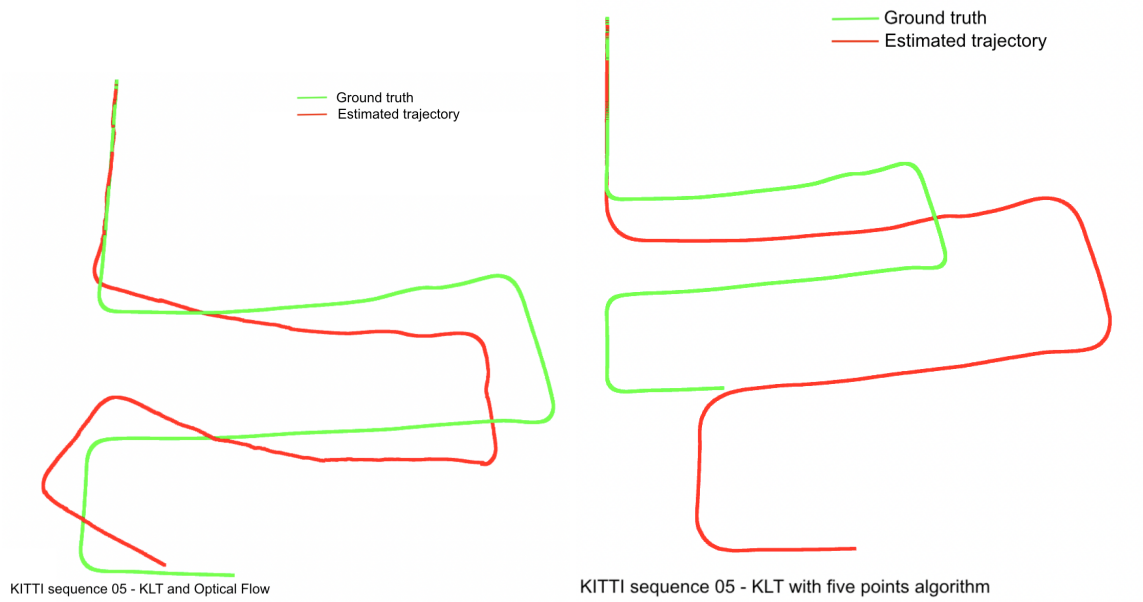


Figure 7.6: KLT results comparison

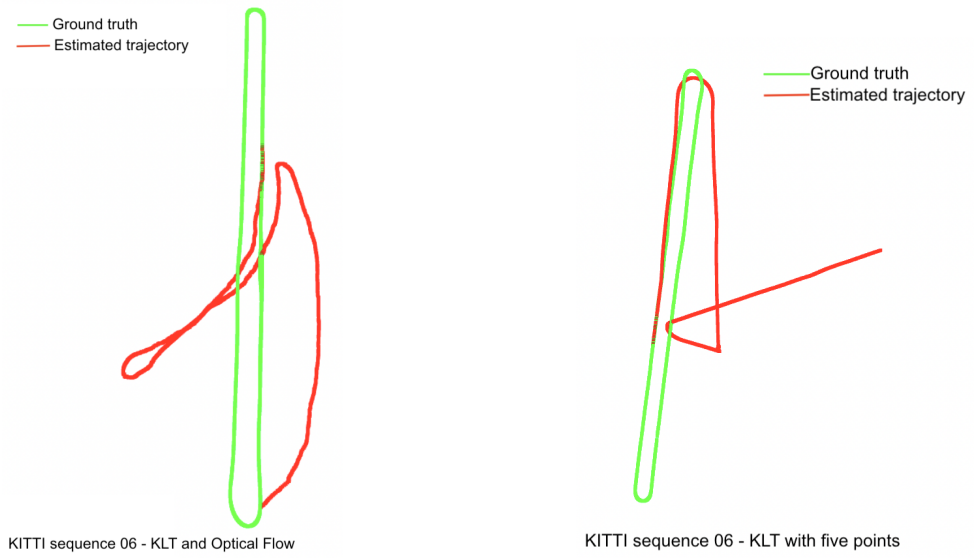


Figure 7.7: KLT results comparison

truth is incomplete in the right image of sequence 07, Figure 7.8, due to the inlier

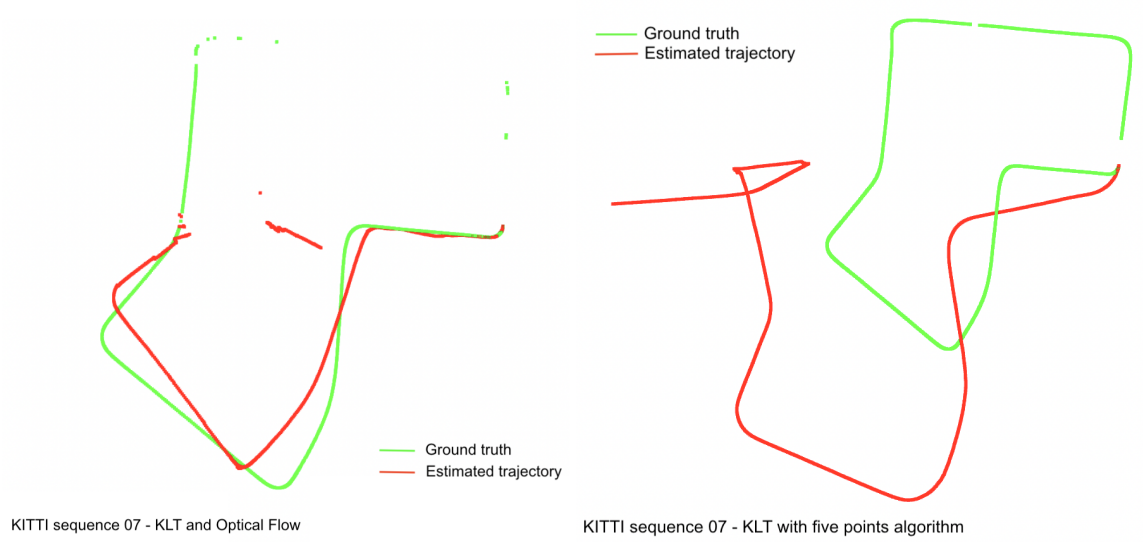


Figure 7.8: KLT results comparison

condition after motion estimation. As an observation, Figures 7.7 and 7.8 are both in closed form, with one in bright condition and the other not. Here, one might conclude that the closed form of the trajectory is necessary for both highly bright and constant situations.

The sequence 08, in Figure 7.8, has observable changes in brightness through

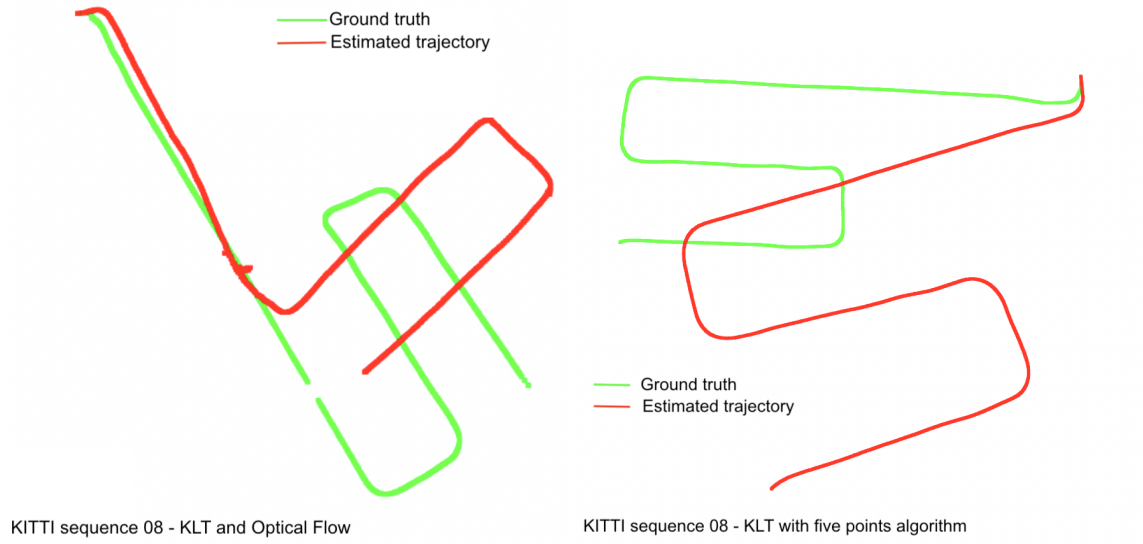


Figure 7.9: KLT results comparison

consecutive frames, and the result in the right with Perspective-N-Point is shown as a sensitive result by turning in the wrong direction, although a five-point solution estimated the trajectory closer to ground truth. Besides, Perspective-N-Point

solution could produce a corrected result with enough RANSAC iteration.

In Figure 7.10, sequence 09 frames are mostly affected by brightness changes,

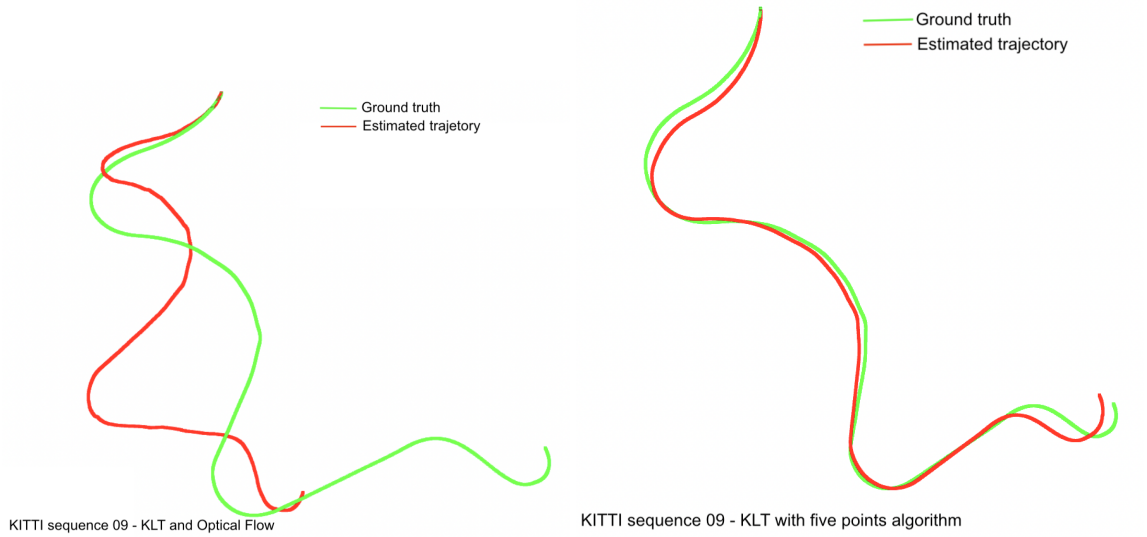


Figure 7.10: KLT results comparison

such that motion estimation errors accumulate in the perspective-n-point solution, while the five-point solution performs well. The resulting trajectory produced a similar shape to the ground truth in the right image. Same as sequence 09, in Figure 7.10, the sequence 10 images captured under a luminous environment, and both solution estimation errors summed more than other sequences; however, the end result shaped the ground truth.

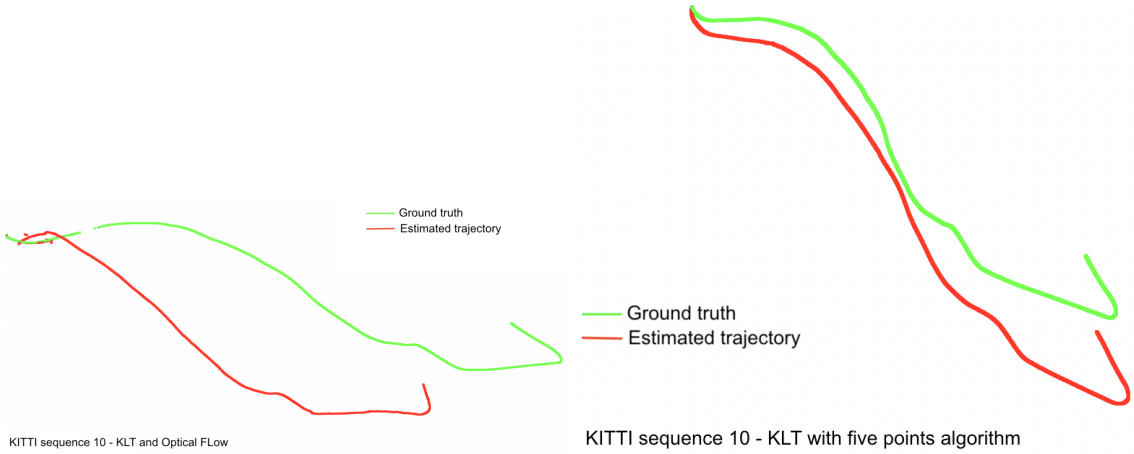


Figure 7.11: KLT results comparison

ELTE car results on two routes are depicted in Figure 7.12. Both starting points show up in different locations due to the frame alignment implemented as the starting point. Overall trajectory shape produced an acceptable result in comparison with ground truth. Moreover, computational time took longer than other presented methods. Out of each frame estimation time, feature matching occupied 70 percent of the time. In addition, the first route images have partially experienced displacement in brightness, in contrast to the second route, which has been highly influenced by a shiny environment.

The next experiment has been done to show the completeness of the method

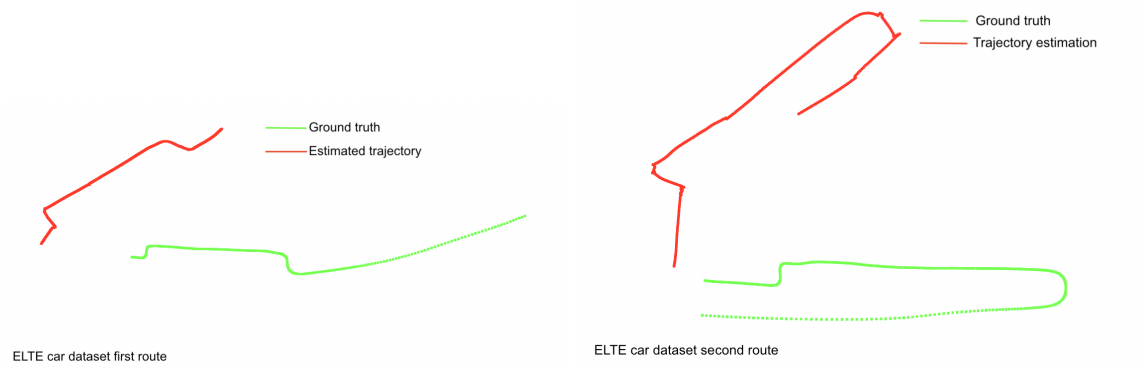


Figure 7.12: SIFT and Optical Flow on ELTE car dataset

in different possible situations around the chosen methods and datasets. Hence, the "SIFT and Optical Flow" method was experimented with on KITTI 00 to 10 sequences and compared with ground truth. Figure 7.13 shows the sequence 00

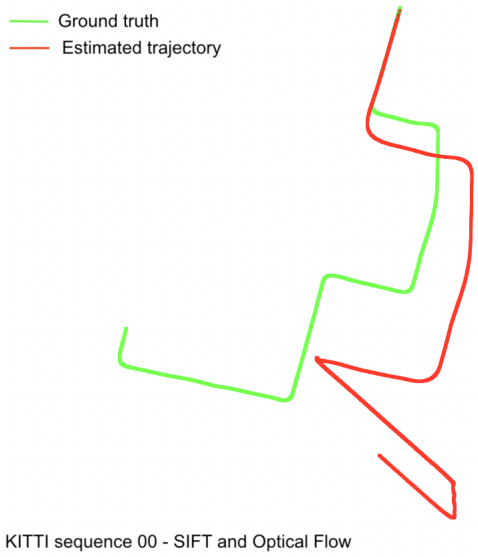


Figure 7.13: KITTI dataset results

result, which occurred with different rotations in the last turns. The brightness displacement was small as an observation of the image frames. As a KLT with a five-point case, the trajectory stretched up to scale.

The sequences 01 and 02 experiments are depicted in Figure 7.14. Sequence 01

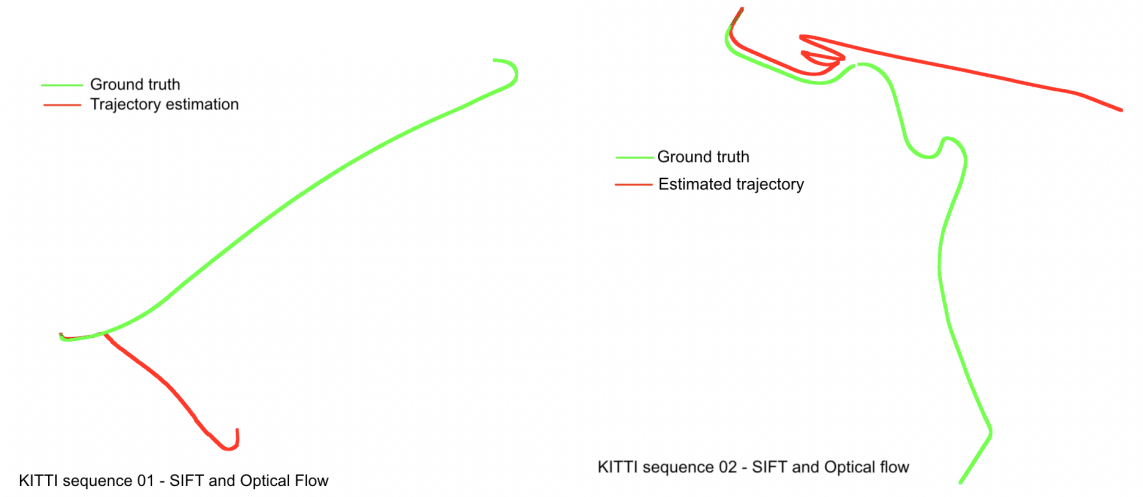


Figure 7.14: KITTI dataset results

produced the same result as the KLT-based five-point method, which has affected the brightness of the images on the straight path. Contrary to the KLT-based five-point method, the sequence 02 curves have shaped the trajectory. Also, the experiment rotational difference in the result can be caused by a high brightness change, therefore, low feature matching for motion estimation. Sequences 03 and 04 are illustrated in Figure 7.15. Sequence 03 ego-motion performed better than KLT

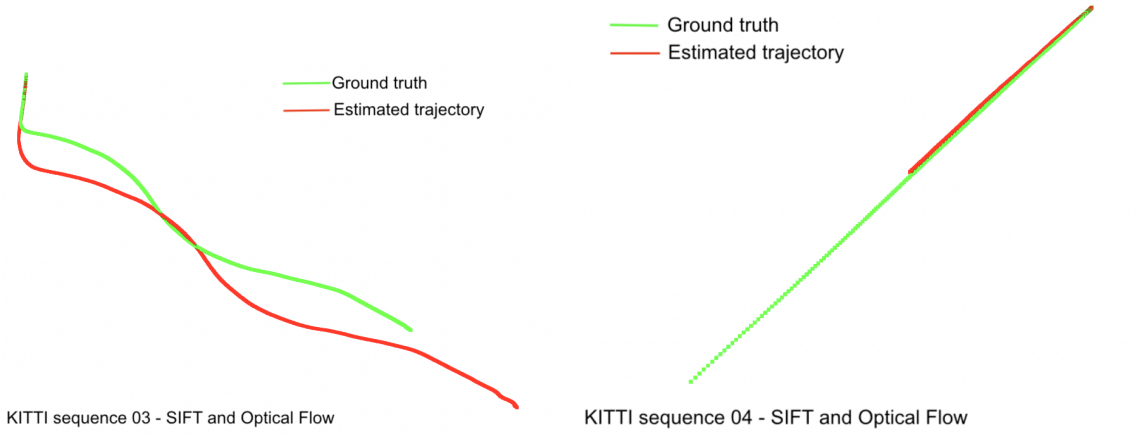


Figure 7.15: KITTI dataset results

and the five-point solution in motion estimation. Whereas, sequence 04 estimation

results are the are the same in terms of the pure translation of the vehicle. In sequences 03 and 04, both luminous images occurred partially in some parts of the consecutive frame. This could be considered a moderate change in brightness. Sequence 05 experiment result was similar to KLT and five-point solution estimated trajectory, in which small changes occurred in the lighting of the images. The result is visualized in Figure 7.16 on the left. On the right, sequence 06 is depicted. The pure translation trajectory with closed form performed better than the two methods tested, "KLT and Optical Flow" and KLT with a five-point algorithm. There was

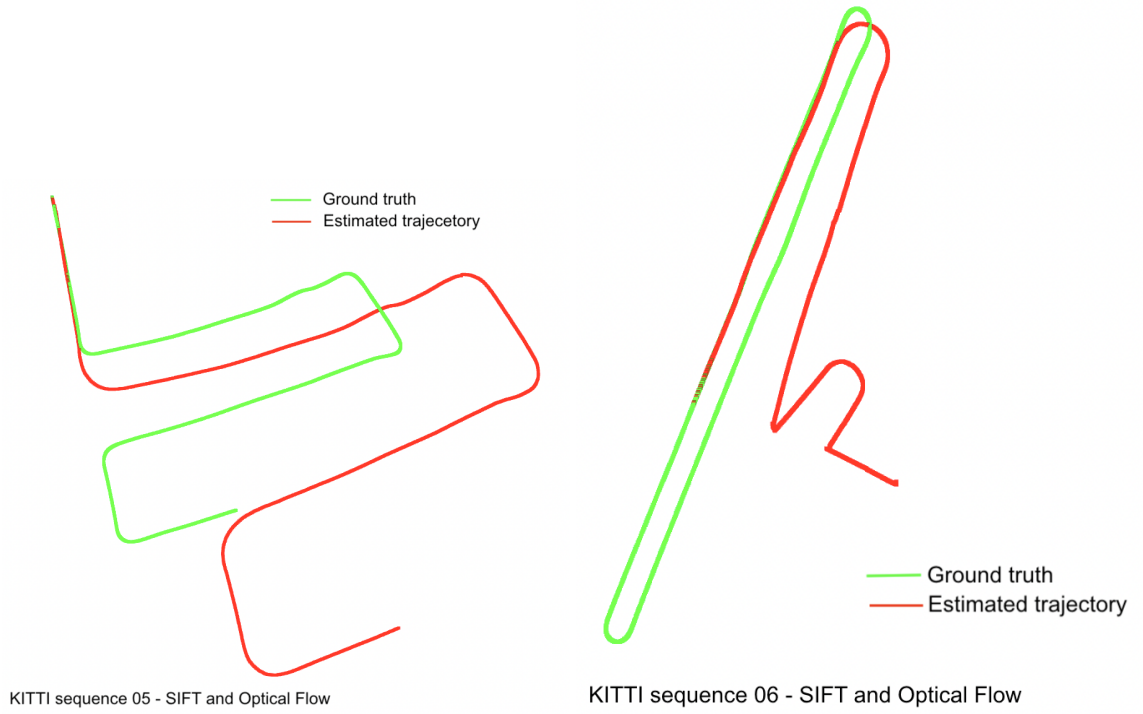


Figure 7.16: KITTI dataset results

another short but closed sequence, like sequence 06 in the KITTI benchmark, but additional high brightness contrast. This is sequence 07; even though the result was not expected to be good in terms of loop closure, the trajectory estimated by KLT with the five-point algorithm resulted better out of the three methods experimented with. The next sequence is 08, which has a small brightness displacement and an estimated trajectory that is kind of a corrected version of KLT with a five-point solution where vehicle motion follows ground truth except for the for the absolute scale of the translation. Sequence 09 in Figure 7.18 has experimented over the full trajectory sequence. KLT with five points and "SIFT and Optical Flow" performed well compared to ground truth in a high-brightness contraction situation. However,

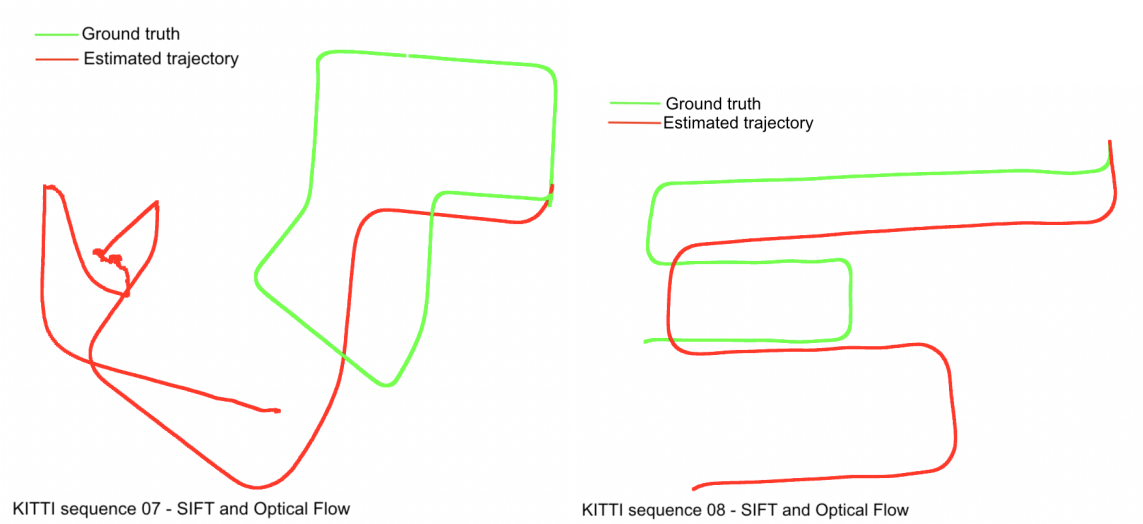


Figure 7.17: KITTI dataset results

sequence 10 was the least optimal solution out of the compared methods. A shining environment was observed when capturing images of the sequence.

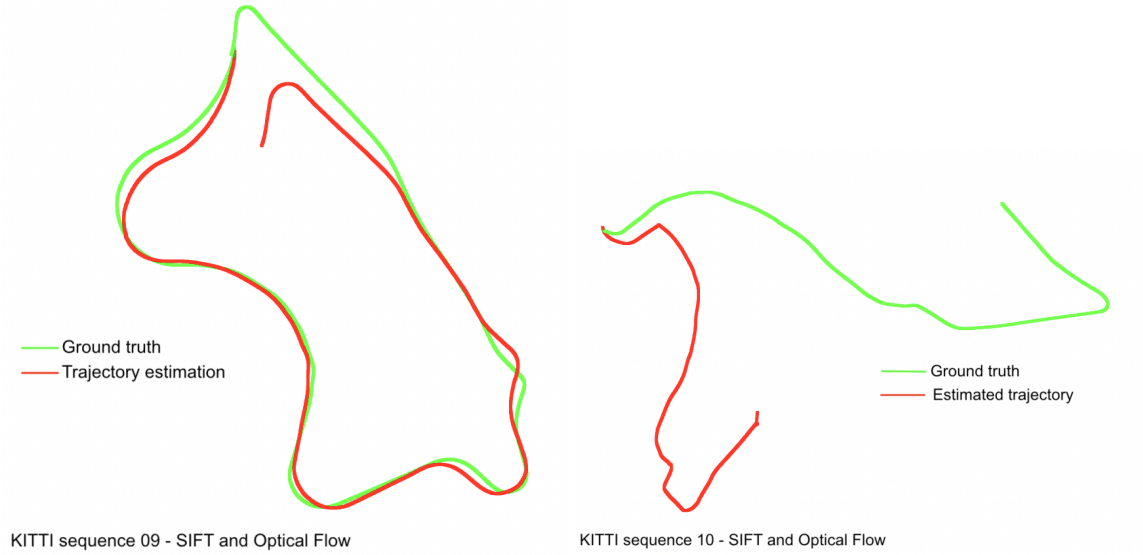


Figure 7.18: KITTI dataset results

Chapter 8

Conclusion

This thesis is devoted to reconstructing spatial points. Accurate Visual odometry estimation produces spatial coordinates by combining all world coordinates used in the computation of the vehicle's ego-motion. For this reason, this thesis mainly focused on computing vehicle trajectory estimation with world coordinates calculated by stereo image views.

There are two different methods implemented based on world coordinate generation, and camera setup depends on the stereo camera or stereo-like view. The first method is "KLT Optical Flow," in which feature correspondences are produced by a stereo camera pair of images and a left camera consequent frame. The correspondences are integrated to estimate pose by employing perspective-N-points problem solving algorithms. The second method matches a stereo-like view by using the SIFT feature matching technique with triangulation, which forms world coordinates. Afterwards, monocular view, five-point, and one point pose estimation methods were utilized in the final result.

According to the experimental result, KLT feature matching results in the in the same shape as ground truth in most cases except for round trajectories, which need loop closure or accumulated error correction for trajectories visited before. Moreover, for each KLT experiment, compared in results, noise removal steps influenced producing good feature matches. Especially one-point noise removal and five points were better at correcting trajectories, for example, sequences 04, 05, 06, 08, and 09, than the perspective-N-points method.

The "SIFT and Optical Flow" experiment produces an acceptable trajectory in half of the KITTI sequences. However, in a round trajectory like KITTI sequence

06, rotation is better than KLT-based methods. Moreover, ELTE car images experimented with in 800 frames in two routes resulted in a closer shape with a true value, but in the case of pure rotation, the method could rotate in a different direction than the true one.

In terms of computation time, KLT with the five-point method was the fastest. Moreover, in a luminous environment, the five-point algorithm is less affected, even though feature matching includes optical flow tracking, whereas RANSAC, with enough iteration, has the ability to perform better in situations with brightness. Lastly, a pure translation case was considered during the experiment, where the five-point method derives a convincing result compared to the perspective-n-point solution.

8.1 Further improvement

From the thesis implementation and experiments, optimality of rotation angle could increase motion estimation of vehicles and decrease accumulated error over time. Moreover, trajectories were corrected with loop closure methods, dealing with errors at every location encountered before. Hence, accurate absolute rotation angle estimation and loop closure are able to yield more precise results in the future.

Acknowledgements

Special thanks to my thesis supervisor, Hajder Levente, who is always ready to lend a hand, and I am so grateful for his time spent since the autumn semester, starting from topic selection to the finishing point of this work. Also, I would like to thank all the amazing professors in the Faculty of Informatics at Eötvös Loránd University. Diligent and proficient assistants, associate professors, and lecturers taught me valuable lessons and gave me a lifetime asset throughout my master studies.

Bibliography

[1] Andreas Geiger, Julius Ziegler, and Christoph Stiller. "StereoScan: Dense 3d reconstruction in real-time". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011, pp. 963–968. DOI: 10.1109/IVS.2011.5940405.

[2] Arturo de la Escalera et al. "Stereo visual odometry in urban environments based on detecting ground features". In: *Robotics and Autonomous Systems* 80 (2016), pp. 1–10. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2016.03.004. URL: https://www.sciencedirect.com/science/article/pii/S0921889015303183.

[3] Bernd Kitt, Andreas Geiger, and Henning Lategahn. "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme". In: *2010 IEEE Intelligent Vehicles Symposium*. 2010, pp. 486–492. DOI: 10.1109/IVS.2010.5548123.

[4] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Y. Siegwart. "Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC". In: *2009 IEEE International Conference on Robotics and Automation* (2009), pp. 4293–4299. URL: https://api.semanticscholar.org/CorpusID:546494.

[5] Sung-In Choi and Soon-Yong Park. "A new 2-point absolute pose estimation algorithm under planar motion". In: *Advanced Robotics* 29.15 (2015), pp. 1005–1013. DOI: 10.1080/01691864.2015.1024285. eprint: https://doi.org/10.1080/01691864.2015.1024285. URL: https://doi.org/10.1080/01691864.2015.1024285.

[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 (1981), 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[7] D. Nister. "An efficient solution to the five-point relative pose problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6 (2004), pp. 756–770. DOI: 10.1109/TPAMI.2004.17.

[8] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), 1147–1163. ISSN: 1941-0468. DOI: 10.1109/tro.2015.2463671. URL: http://dx.doi.org/10.1109/TRO.2015.2463671.

[9] Hans P. Moravec. "Obstacle avoidance and navigation in the real world by a seeing robot rover". In: 1980. URL: https://api.semanticscholar.org/CorpusID:128525458.

[10] Hernán Badino, Akihiro Yamamoto, and Takeo Kanade. "Visual Odometry by Multi-frame Feature Integration". In: *2013 IEEE International Conference on Computer Vision Workshops* (2013), pp. 222–229. URL: https://api.semanticscholar.org/CorpusID:1022760.

[11] Sunglok Choi et al. "What does ground tell us? Monocular visual odometry under planar motion constraint". In: *2011 11th International Conference on Control, Automation and Systems*. 2011, pp. 1480–1485.

[12] Chih Chung Chou and Chieh-Chih Wang. "2-point RANSAC for scene image matching under large viewpoint changes". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3646–3651. DOI: 10.1109/ICRA.2015.7139705.

[13] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 88–93. ISBN: 9780511186189, 9780521540513.

[14] Andreas Geiger, Martin Roser, and Raquel Urtasun. "Efficient Large-Scale Stereo Matching". In: *Computer Vision – ACCV 2010*. Ed. by Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 25–38. ISBN: 978-3-642-19315-6.

[15] Vladimir Kolmogorov and Ramin Zabih. "Computing visual correspondence with occlusions using graph cuts". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* 2 (2001), 508–515 vol.2. URL: https://api.semanticscholar.org/CorpusID:2457778.

[16] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Belief Propagation for Early Vision". In: *International Journal of Computer Vision* 70 (2004), pp. 41–54. URL: https://api.semanticscholar.org/CorpusID:8702465.

[17] Jan Cech and Radim Sára. "Efficient Sampling of Disparity Space for Fast And Accurate Matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition* (2007), pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:16206680.

[18] Jana Kostková and Radim Sára. "Stratified Dense Matching for Stereopsis in Complex Scenes." In: *Bmvc*. Vol. 5. 2003, p. 6.

[19] Jakob Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 834–849. ISBN: 978-3-319-10605-2.

[20] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[21] Tony Lindeberg. *Scale invariant feature transform*. QC 20120524. 2012. DOI: 10.4249/scholarpedia.10491. URL: http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform.

[22] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.

[23] Rainer Kümmerle et al. "G2o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation* (2011), pp. 3607–3613. URL: https://api.semanticscholar.org/CorpusID:206849534.

[24] Jakob Engel, Jürgen Sturm, and Daniel Cremers. "Semi-dense Visual Odometry for a Monocular Camera". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 1449–1456. DOI: 10.1109/ICCV.2013.183.

[25] Georg Klein and David Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852.

[26] Edward Rosten and Tom Drummond. "Machine Learning for High-Speed Corner Detection". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. ISBN: 978-3-540-33833-8.

[27] Michael Calonder et al. "BRIEF: Binary Robust Independent Elementary Features". In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792. ISBN: 978-3-642-15561-1.

[28] Álvaro Parra Bustos et al. "Visual SLAM: Why Bundle Adjust?" In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 2385–2391. DOI: 10.1109/ICRA.2019.8793749.

[29] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, p. 353. ISBN: 9780511186189, 9780521540513.

[30] Klaus H. Strobl. "Loop Closing for Visual Pose Tracking during Close-Range 3-D Modeling". In: *Advances in Visual Computing*. Ed. by George Bebis et al. Cham: Springer International Publishing, 2014, pp. 390–401. ISBN: 978-3-319-14249-4.

[31] Rainer Kümmerle et al. "G2o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3607–3613. DOI: 10.1109/ICRA.2011.5979949.

[32] Heiko Hirschmuller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341. DOI: 10.1109/TPAMI.2007.1166.

[33] Joel A. Hesch and Stergios I. Roumeliotis. "A Direct Least-Squares (DLS) method for PnP". In: *2011 International Conference on Computer Vision*. 2011, pp. 383–390. DOI: 10.1109/ICCV.2011.6126266.

[34] Hernán Badino and Takeo Kanade. "A Head-Wearable Short-Baseline Stereo System for the Simultaneous Estimation of Structure and Motion". In: *IAPR International Workshop on Machine Vision Applications*. 2011. URL: https://api.semanticscholar.org/CorpusID:14493553.

[35] Jianbo Shi and Tomasi. "Good features to track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[36] Amirreza Mirzajani et al. "Ackerman steering mechanism design for front axle steering vehicles". In: May 2023.

[37] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 239–259. ISBN: 9780511186189, 9780521540513.

[38] 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 (1981), pp. 381–395. URL: https://api.semanticscholar.org/CorpusID:972888.

[39] Sunglok Choi et al. "Numerical Solutions to Relative Pose Problem under Planar Motion". In: 2010. URL: https://api.semanticscholar.org/CorpusID:86867888.

[40] Diego Ortin and José M. M. Montiel. "Indoor robot motion based on monocular images". In: *Robotica* 19 (2001), pp. 331 –342. URL: https://api.semanticscholar.org/CorpusID:5884124.

[41] Sunglok Choi and Jong-Hwan Kim. "Fast and reliable minimal relative pose estimation under planar motion". In: *Image Vis. Comput.* 69 (2017), pp. 103–112. URL: https://api.semanticscholar.org/CorpusID:3415468.

[42] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 88–91. ISBN: 9780511186189, 9780521540513.

[43] Chih-Chung Chou and C. Wang. "2-point RANSAC for scene image matching under large viewpoint changes". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 3646–3651. URL: https://api.semanticscholar.org/CorpusID:16780825.

[44] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 281–282. ISBN: 9780511186189, 9780521540513.

[45] Sunglok Choi and Jong-Hwan Kim. "Robust regression to varying data distribution and its application to landmark-based localization". In: Nov. 2008, pp. 3465 –3470. DOI: 10.1109/ICSMC.2008.4811834.

[46] Sunglok Choi, Jaehyun Park, and Wonpil Yu. "Resolving scale ambiguity for monocular visual odometry". In: Oct. 2013, pp. 604–608. ISBN: 978-1-4799-1195-0. DOI: 10.1109/URAI.2013.6677403.

[47] Javier Civera et al. "1-Point RANSAC for Extended Kalman Filtering: Application to Real-Time Structure from Motion and Visual Odometry". In: *J. Field Robotics* 27 (Sept. 2010), pp. 609–631. DOI: 10.1002/rob.20345.

[48] Henrik Stewénius, Christopher Engels, and David Nistér. "Recent developments on direct relative orientation". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 60.4 (2006), pp. 284–294. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2006.03.005. URL: https://www.sciencedirect.com/science/article/pii/S092427160600030X.

[49] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004. Chap. 6. ISBN: 9780511186189, 9780521540513.

[50] SHUO Feng, JIANGMING Kan, and YEXIAO Wu. "An improved method to estimate the fundamental matrix based on 7-point algorithm". In: *Journal of Theoretical and Applied Information Technology* 46.1 (2012), pp. 212–217.

[51] Daniel Barath and Jiří Matas. "Graph-Cut RANSAC". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[52] Morton Keller. *Space resection in photogrammetry*. 2nd. Washington D.C. : Supt. of Docs. U.S. G.P.O. 1966., 1966. ISBN: 9910321787902121. URL: https://search.library.wisc.edu/catalog/9910321787902121.

[53] P.R. Wolf. *Elements of Photogrammetry: With Air Photo Interpretation and Remote Sensing*. Civil Engineering Series. McGraw-Hill, 1983. ISBN: 9780070713451. URL: https://books.google.hu/books?id=fNtTAAAAMAAJ.

[54] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 302–308. ISBN: 9780511186189, 9780521540513.

[55] J. Li and R. Chellappa. "Structure From Planar Motion". In: *IEEE Transactions on Image Processing* 15.11 (2006), pp. 3466–3477. DOI: 10.1109/TIP.2006.881943.

[56] Andrew Zisserman Richard Hartley. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2004, pp. 486–493. ISBN: 9780511186189, 9780521540513.

[57] Sunglok Choi, Jaehyun Park, and Wonpil Yu. "Simplified epipolar geometry for real-time monocular visual odometry on roads". In: *International Journal of Control, Automation and Systems* 13.6 (2015), pp. 1454–1464.

[58] Heiko Hirschmüller. "Stereo Processing by Semi-Global Matching and Mutual Information". In: *in IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (Feb. 2008), pp. 328–341.

[59] Jianbo Shi and Tomasi. "Good features to track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[60] Bruce Lucas and Takeo Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)". In: vol. 81. Apr. 1981.

[61] Marius Muja and David G. Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration". In: *International Conference on Computer Vision Theory and Applications*. 2009. URL: https://api.semanticscholar.org/CorpusID:7317448.

[62] ELTE faculty of Informatics. “INtroduction to the ELTE 3D sensor pack”. In: 2023. URL: https://www.hackademix.hu/wp-content/uploads/2023/06/Sensor_pack_summary_2023.pdf.

List of Figures

List of Algorithms