Visual Odomerty

SOUALHI Takieddine - 3872967

Abstract— **This document presents provides a broad introduction to VO and the research that has been undertaken from 1980 to 2011. Although the first two decades witnessed many offline implementations, only in the third decade did real-time working systems flourish, which has led VO to be used on another planet by two Mars exploration rovers for the first time.we will present a historical review of the first 30 years of research in this field and its fundamentals. After a brief discussion on camera implementation of monocular visual odometry, in the rest of this sheet we will dicuss a theoretical and experimental approach of monocular visual odometry and an implementation of a proposed algorithm**

Keywords: RANSAC optimization, Odometry, FAST algorithmEpilpolair Constraint.

I. INTRODUCTION

Visual odometry (VO) is the process of estimating the egomotion of an agent (e.g., vehicle, human, and robot) using only the input of a single or multiple cameras attached to it. Application domains include robotics, wearable computing, augmented reality, and automotive. The termVO was coined in 2004 by Nister in his landmark paper [1]. The term was chosen for its similarity to wheel odometry, which incrementally estimates the motion of a vehicle by integrating the number of turns of its wheels over time. Likewise, VO operates by incrementally estimating the pose of the vehicle through examination of the changes that motion induces on the images of its onboard cameras. For VO to work effectively, there should be sufficient illumination in the environment and a static scene with enough texture to allow apparent motion to be extracted. Furthermore, consecutive frames should be captured by ensuring that they have sufficient scene overlap. The advantage of VO with respect to wheel odometry is that VO is not affected by wheel slip in uneven terrain or other adverse conditions. It has been demonstrated that compared to wheel odometry, VO provides more accurate trajectory estimates, with relative position error ranging from 0.1 to 2 . This capability makes VO an interesting supplement to wheel odometry and, additionally, to other navigation systems such as global positioning system (GPS), inertial measurement units (IMUs), and laser odometry (similar to VO, laser odometry estimates the egomotion of a vehicle by scan-matching of consecutive laser scans). In GPS-denied environments, such as underwater and aerial,VO has utmost importance.

II. STATE OF THE ART

The problem of recovering relative camera poses and three-dimensional (3-D) structure from a set of camera images (calibrated or non-calibrated) is known in the computer vision community as structure from motion (SFM). VO is a particular case of SFM. SFM is more general and tackles the problem of 3-D reconstruction of both the structure and camera poses from sequentially ordered or unordered image sets. The final structure and camera poses are typically refined with an offline optimization (i.e, bundle adjustment), whose computation time grows with the number of images. Conversely, VO focuses on estimating the 3-D motion of the camera sequentiallyas a new frame arrivesand in real time. The problem of estimating a vehicles egomotion from visual input alone started in the early 1980s and was described by Moravec. It is interesting to observe that most of the early research in VO was done for planetary rovers and was motivated by the NASA Mars exploration program in the endeavour to provide all-terrain rovers with the capability to measure their 6-degree-of freedom (DoF) motion in the presence of wheel slippage in uneven and rough terrains. The work of Moravec stands out not only for presenting the first motion-estimation pipelinewhose main functioning blocks are still used todaybut also for describing one of the earliest corner detectors (after the first one proposed in 1974 by Hannah) which is known today as the Moravec corner detector apredecessor of the one proposed by Forstner [12] and Harris and Stephens. Moravec tested his work on a planetary rover equipped with what he termed a slider stereo: a single camera sliding on a rail. The robot moved in a stop-and-go fashion, digitizing and analyzing images at every location. At each stop, the camera slid horizontally taking nine pictures at equidistant intervals. Corners were detected in an image using his operator and matched along the epipolar lines of the other eight frames using normalized cross correlation. Potential matches at the next robot locations were found a gain by correlation using a coarse-to-fine strategy to account for large-scale changes. Outliers were subsequently removed by checking for depth inconsistencies in the eight stereo pairs. Finally, motion was computed as the rigid body transformation to align the triangulated 3-D points seen at two consecutive robot positions. The system of equation was solved via a weighted least square, where the weights were inversely proportional to the distance from the 3-D point. Although Moravec used a single sliding camera, his work belongs to the class of stereo VO algorithms. This terminology accounts for the fact that the relative 3-D position of the features is directly measured by triangulation at every robot location and used to derive the relative motion. Trinocular methods belong to the same class of algorithms. The alternative to stereo vision is to use a single camera. In this case, only bearing

information is available. The disadvantage is that motion can only be recovered up to a scale factor. The absolute scale can then be determined from direct measurements (e.g., measuring the size of an element in the scene), motion constraints, or from the integration with other sensors, such as IMU, airpressure, and range sensors. The interest in monocular methods is due to the observation that stereo VO can degenerate to the monocular case when the distance to the scene is much larger than the stereo baseline (i.e., the distance between the two cameras). In this case, stereo vision becomes ineffective and monocular methods must be used. Over the years, monocular and stereo VOs have almost progressed as two independent lines of research. In the remainder of this section, we have surveyed the related work in these fields.

III. VISUAL ODOMETRY TECHNICS

A. Monocular VO

The difference from the stereo scheme is that in the monocular VO, both the relative motion and 3-D structure must be computed from 2-D bearing data. Since the absolute scale is unknown, the distance between the first two camera poses is usually set to one. As a new image arrives, the relative scale and camera pose with respect to the first two frames are determined using either the knowledge of 3-D structure or the trifocal tensor [22]. Successful results with a single camera over long distances (up to several kilometers) have been obtained in the last decade using both perspective and omnidirectional cameras [23][29]. Related works can be divided into three categories: feature-based methods, appearance-based methods, and hybrid methods. Feature-based methods are based on salient and repeatable features that are tracked over the frames; appearance-based methods use the intensity information of all the pixels in the image or subregions of it; and hybrid methods use a combination of the previous two. In the first category are the works by the authors in [1], [24], [25], [27], and [30][32]. The first real-time, largescale VO with a single camera was presented by Nister et al. [1]. They used RANSAC for outlier rejection and 3-Dto-2-D camera-pose estimation to compute the new upcoming camera pose. The novelty of their paper is the use of a five-point minimal solver [33] to calculate the motion hypotheses in RANSAC. After that paper, fivepoint RANSAC became very popular in VO and was used in several other works [23], [25], [27]. Corke et al. [24] provided an approach for monocular VO based on omnidirectional imagery from a catadioptric camera and optical flow. Lhuillier [25] and Mouragnon et al. [30] presented an approach based on local windowed-bundle adjustment to recover both the motion and the 3-D map (this means that bundle adjustment is performed over a window of the last m frames). Again, they used the five-point RANSAC in

B. Stereo VO

Most of the research done in VO has been produced using stereo cameras. Building upon Moravecs work, Matthies and Shafer [6], [7] used a binocular system and Moravecs procedure for detecting and tracking corners. Instead of using a scalar representation of the uncertainty as Moravec did, they took advantage of the error covariance matrix of the triangulated features and incorporated it into the motion estimation step. Compared to Moravec, they demonstrated superior results in trajectory recovery for a planetary rover, with 2 relative error on a 5.5-m path. Olson et al. [9], [13] later extended that work by introducing an absolute orientation sensor (e.g., compass or omnidirectional camera) and using the Forstner corner detector, which is significantly faster to compute than Moravecs operator. They showed that the use of camera egomotion estimates alone results in accumulation errors with superlinear growth in the distance traveled, leading to increased orientation errors. Conversely, when an absolute orientation sensor is incorporated, the error growth can be reduced to a linear function of the distance traveled. This led them to a relative position error of 1:2 on a 20-m path. Lacroix et al. [8] implemented a stereo VO approach for planetary rovers similar to those explained earlier. The difference lies in the selection of key points. Instead of using the Forstner detector, they used dense stereo and, then, selected the candidate key points by analyzing the correlation function around its peaksan approach that was later exploited in [14], [15], and other works. This choice was based on the observation that there is a strong correlation between the shape of the correlation curve and the standard deviation of the feature depth. This observation was later used by Cheng et al. [16], [17] in their final VO implementation on board the Mars rovers. They improved on the earlier implementation by Olson et al. [9], [13] in two areas. First, after using the Harris corner detector, they utilized the curvature of the correlation function around the featureas proposed by Lacroix et al.to define the error covariance matrix of the image point. Second, as proposed by Nister et al. [1], they used the random sample consensus (RANSAC) RANSAC) RANSAC [18] in the least-squares motion estimation step for outlier rejection. A different approach to motion estimation and outlier removal for an all-terrain rover was proposed by Milella and Siegwart [14]. They used the Shi-Tomasi approach [19] for corner detection, and similar to Lacroix, they retained those points with high confidence in the stereo disparity map. Motion estimation was then solved by first using least squares, as in the methods earlier, and then the iterative closest point (ICP) algorithm [20]an algorithm popular for 3-D registration of laser scansfor pose refinement. For robustness, an outlier removal stage was incorporated into the ICP. The works mentioned so far have in common that the 3-D points are triangulated for every stereo pair, and the relative motion is solved as a 3-D-to-3-D point registration (alignment) problem. A completely different approach was proposed in 2004 by Nister et al. [1]. Their paper is known not only for coining the term VO but also for providing the first real-time long-run implementation with a robust outlier rejection scheme. Nister et al. improved the earlier implementations in several areas. First, contrary to all

previous works, they did not track features among frames. but detected features (Harris corners) independently in all frames and only allowed matches between features. This has the benefit of avoiding feature drift during cross-correlation-based tracking. Second, they did not compute the relative motion as a 3-D-to-3-D point registration problem but as a 3-D-to-two-dimensional (2-D) camera-pose estimation problem (these methods are described in the Motion Estimation section). Finally, they incorporated RANSAC outlier rejection into the motion estimation step. A different motion estimation scheme was introduced by Comport et al. [21]. Instead of using 3-D-to-3-D point registration or 3-D-to-2-D camera-pose estimation techniques, they relied on the quadrifocal tensor, which allows motion to be computed from 2-D-to-2-D image matches without having to triangulate 3-D points in any of the stereo pairs. The benefit of using directly raw 2-D points in lieu of triangulated 3-D points lays in a more accurate motion computation.

C. V-SLAM

Although this document focuses on VO, it is worth mentioning the parallel line of research undertaken by visual simultaneous localization and mapping (V-SLAM).. Two methodologies have become predominant in V-SLAM: 1) filtering methods fuse the information from all the images with a probability distribution [49] and 2) non filtering methods (also called keyframe methods) retain the optimization of global bundle adjustment to selected key frames [50]. The main advantages of either approach have been evaluated and summarized in [51]. In the last few years, successful results have been obtained using both single and stereo cameras [49], [52] [62]. Most of these works have been limited to small, indoor workspaces and only a few of them have recently been designed for large-scale areas [54], [60], [62]. Some of the early works in real-time V-SLAM were presented by Chiuso et al. [52], Deans [53], and Davison [49] using a full-covariance Kalman approach. The advantage of Davisons work was to account for repeatable localization after an arbitrary amount of time. Later, Handa et al. [59] improved on that work using an active matching technique based on a probabilistic framework. Civera et al. [60] built upon that work by proposing a combination of one-point RANSAC within the Kalman filter that uses the available prior probabilistic information from the filter in the RANSAC model-hypothesis stage. Finally, Strasdat et al. [61] presented a new framework for large-scale V-SLAM that takes advantage of the keyframe optimization approach [50] while taking into account the special character of SLAM.

IV. FORMULATION OF THE VO PROBLEM

An agent is moving through an environment and taking images with a rigidly attached camera system at discrete time instants k. In case of a monocular system, the set of images taken at times k is denoted by $I_{0:n} = (I_0, ..., I_n)$. In case of a stereo system, there are a left and a right image at every time instant, denoted by $I_{l,0:n} = (I_{l0}, ..., I_{ln})$ and $I_{r,0:n} = (I_{r0}, ..., I_{rn})$ Figure 1 shows an illustration of this setting. For simplicity, the camera coordinate frame is assumed to be also the agents coordinate frame. In case of a stereo system, without loss of generality, the coordinate system of the left camera can be used as the origin. Two camera positions at adjacent time instants k-1 and k are related by the rigid body transformation $T_{k,k-1}$ of the following form:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

where $R_{k,k-1}$ is the rotation matrix, and $t_{k,k-1}$ the translation vector. The set $T_{0:n}$ contains all subsequent motions. To simplify the notation, from now on, T_k will be used instead of $T_{k,k-1}$. Finally, the set of camera poses $C_{0:n} = (C_0, ...C_n)$ contains the transformations of the camera with respect to the initial coordinate frame at K=0. The current pose C_n can be computed by concatenating all

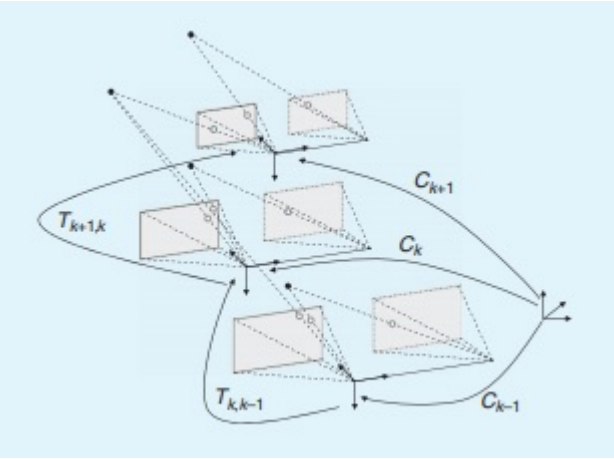

Fig. 1. fig:An illustration of the visual odometry problem

the transformations $T_{0:n}$, and, therefore, $C_n = T_n C_{n-1}$, with C_0 being the camera pose at the instant k=0, which can be set arbitrarily by the user.

The main task in VO is to compute the relative transformations T_k from the images I_k and I_{k-1} and then to concatenate the transformations to recover the full trajectory $C_{0:n}$ of the camera. This means that VO recovers the path incrementally, pose after pose. An iterative refinement over the last m poses can be performed after this step to obtain a more accurate estimate of the local trajectory.This iterative refinement works by minimizing the sum of the squared reprojection errors of the reconstructed 3-D points (i.e., the 3-D map) over the last m images (this is called windowed-bundle adjustment, because it is performed on a window of m frames. The 3-D points are obtained by triangulation of the image points. As mentioned in the Monocular VO section, there are two main approaches to compute the relative motion T_k: appearance-based (or global) methods, which use the intensity information of all the pixels in the

two input images, and feature-based methods, which only use salient and repeatable features extracted (or tracked) across the images. Global methods are less accurate than feature based methods and are computationally more expensive. (As observed in the History of VO section, most appearance-based methods have been applied to monocular VO. This is due to ease of implementation compared with the stereo camera case.) Feature-based methods require the ability to robustly match (or track) features across frames but are faster and more accurate than global methods. Therefore, most VO implementations are feature based. The VO pipeline is summarized in Figure 2. For every new image I_k (or image pair in the case of a stereo camera), the first two steps consist of detecting and matching 2-D features with those from the previous frames. Two-dimensional features

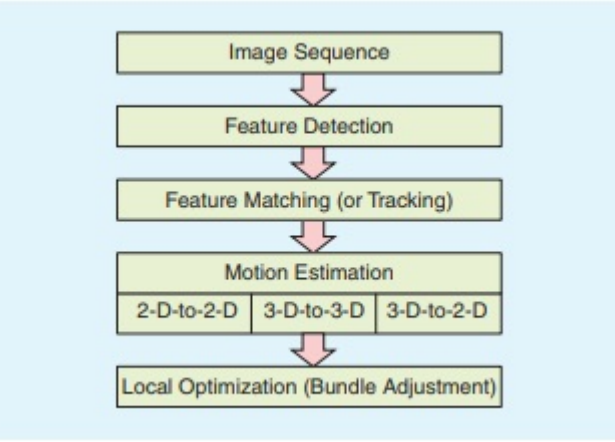

Fig. 2. fig:A block diagram showing the main components of a VO system.

that are the reprojection of the same 3-D feature across different frames are called image correspondences. (we distinguish between feature matching and feature tracking. The first one consists of detecting features independently in all the images and then matching them based on some similarity metrics; the second one consists of finding features in one image and then tracking them in the next images using a local search technique, such as correlation.) The third step consists of computing the relative motion T_k between the time instants k_{-1} and k. Depending on whether the correspondences are specified in three or two dimensions, there are three distinct approaches to tackle this problem (see the Motion Estimation section). The camera pose C_k is then computed by concatenation of T_k with the previous pose. Finally, an iterative refinement (bundle adjustment) can be done over the last m frames to obtain a more accurate estimate of the local trajectory.

To this end, the next section reviews the standard models and calibration procedures for perspective and omnidirectional cameras.

V. CAMERA CALIBRATION AND MODELS

VO can be done using both perspective and omnidirectional cameras. In this section, we review the main models.

A. *PERSPECTIVE CAMERA MODEL*

The most used model for perspective camera assumes a pinhole projection system: the image is formed by the intersection of the light rays from the objects through the center of the lens (projection center), with the focal plane [Figure 3(a)]. Let $X = [x, y, z]^t$ be a scene point in the camera reference frame and $p = [u, v]^t$its projection on the image plane measured in pixels. The mapping from the 3-D world to the 2-D image is given by the perspective projection equation:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KX = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where k is the depth factor, a_u and a_v the focal lengths, and u_0, v_0 the image coordinates of the projection center. These parameters are called intrinsic parameters. When the field of view of the camera is larger than 45, the effects of the radial distortion may become visible and can be modeled using a second- (or higher)-order polynomial. The derivation of the complete model can be found in computer vision textbooks. Let $p = [u, v, 1]^T = K^{-1}[u, v, 1]^T$ be the normalized image coordinates. Normalized coordinates will be used throughout in the following sections.

B. *Omnidirectional Camera model*

Omnidirectional cameras are cameras with wide field of view (even more than 180) and can be built using fish-eye lenses or by combining standard cameras with mirrors [the latter are called catadioptric cameras, Figure 3(b)].

Typical mirror shapes in catadioptric cameras are quadratic

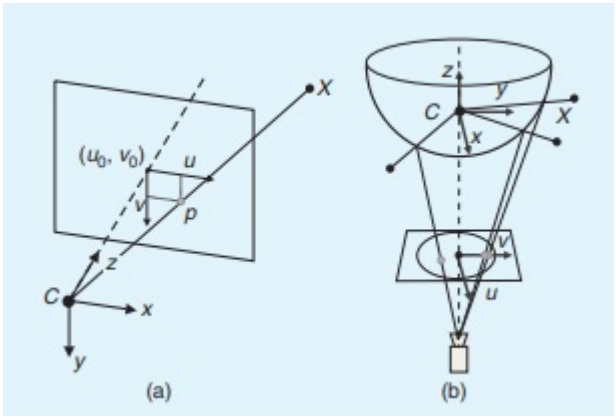

Fig. 3. fig:(a) Perspective projection, (b) catadioptric projection.

surfaces of revolution (e.g., paraboloid or hyperboloid), because they guarantee a single projection center, which makes it possible to use the motion estimation theory presented in the Motion Estimation section. Currently, there are two accepted models for omnidirectional cameras. The first one

proposed by Geyer and Daniilidis [64] is for general catadioptric cameras (parabolic or hyperbolic), while the second one proposed by Scaramuzza et al. [65] is a unified model for both fish-eye and catadioptric cameras. A survey of these two models can be found in [66] and [67]. The projection equation of the unified model is as follows:

$$\lambda \begin{bmatrix} u \\ v \\ a_0 + a_1 p + ... + a_{n-1} p^{n-1} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where $p = \sqrt{u^2 + v^2}$ and a_0,a_1.....a_n an are intrinsic parameters that depend on the type of mirror or fish-eye lens. As shown in [65], n = 4 is a reasonable choice for a large variety of mirrors and fish-eye lenses. Finally, this model assumes that the image plane satisfies the ideal property that the axes of symmetry of the camera and mirror are aligned. Although this assumption holds for most catadioptric and fish-eye cameras, misalignments can be modeled by introducing a perspective projection between the ideal and real-image plane [66].

C. Camera calibration

The goal of calibration is to accurately measure the intrinsic and extrinsic parameters of the camera system. In a multicamera system (e.g., stereo and trinocular), the extrinsic parameters describe the mutual position and orientation between each camera pair. The most popular method uses a planar checkerboard-like pattern. The position of the squares on the board is known. To compute the calibration parameters accurately, the user must take several pictures of the board shown at different positions and orientations by ensuring that the field of view of the camera is filled as much as possible. The intrinsic and extrinsic parameters are then found through a least-square minimization method. The input data are the 2-D positions of the corners of the squares of the board and their corresponding pixel coordinates in each image. Many camera calibration toolboxes have been devised for MATLAB and C.

VI. MOTION ESTIMATION

Motion estimation is the core computation step performed for every image in a VO system. More precisely, in the motion estimation step, the camera motion between the current image and the previous image is computed. By concatenation of all these single movements, the full trajectory of the camera and the agent (assuming that the camera is rigidly mounted) can be recovered. This section explains how the transformation T_k between two images I_{k-1} and I_k can be computed from two sets of corresponding features f_{k-1}, f_k at time instants $k-1$ and k, respectively. Depending on whether the feature correspondences are specified in two or three dimensions, there are three different methods: - 2-D-to-2-D: In this case, both f_{k1} and f_k are specified in 2-D image coordinates.
- 3-D-to-3-D: In this case, both f_{k-1} and f_k are specified in 3-D. To do this, it is necessary to triangulate 3-D points at each time instant; for instance, by using a stereo camera system.
- 3-D-to-2-D: In this case, f_{k1} are specified in 3-D and f_k are their corresponding 2-D reprojections on the image I_k. In the monocular case, the 3-D structure needs to be triangulated from two adjacent camera views (e.g., I_{k2} and I_{k-1}) and then matched to 2-D image features in a third view (e.g., I_k). In the monocular scheme, matches over at least three views are necessary.
Notice that features can be points or lines. In general, due to the lack of lines in unstructured scenes, point features are used in VO.

VII. 2-D TO 2-D: MOTION FROM IMAGE FEATURE CORRESPONDENCES

A. Estimating the Essential Matrix :

The geometric relations between two images I_k and I_{k-1} of a calibrated camera are described by the so-called essential matrix E. E contains the camera motion parameters up to an unknown scale factor for the translation in the following form:

$$E_k \approx t_k R_k$$

where $t_k =$

$$\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

The essential matrix can be computed from 2-D-to-2-D feature correspondences, and rotation and translation can directly be extracted from E. The main property of 2-D-to- 2-D-based motion estimation is the epipolar constraint, which determines the line on which the corresponding feature point lies in the other image (Figure 4). This constraint can be formulated by $p^{'} Ep = 0$, where $p^{'}$ is a feature location in one image (e.g., I_k) and p is the location of its corresponding feature in another image (e.g., I_{k-1}). we note that p and $p^{'}$ are normalized image coordinates. The essential matrix can be computed from 2-D-to-2-D feature correspondences using the epipolar constraint. The minimal case solution involves five 2-D-to-2-D correspondences and an efficient implementation proposed by Nister in. Nisters five-point algorithm has become the standard for 2-D-to-2-D motion estimation

B. Extracting R and t from E :

From the estimate of E, the rotation and translation parts can be extracted. In general, there are four different solutions for R, t for one essential matrix; however, by triangulation of a single point, the correct R, t pair can be identified. The four solutions are

$$R = U(W^T)V^T$$

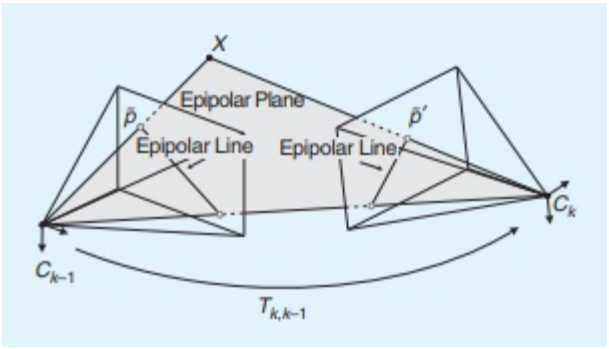

Fig. 4. fig:An illustration of the epipolar constraint.

$$t = U(W)SU^T$$

where

$$W^T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

After selecting the correct solution by triangulation of a point and choosing the solution where the point is in front of both cameras, a nonlinear optimization of the rotation and translation parameters should be performed using the estimate R, t as initial values.

below is the algorithm that will be implemented later :
Result: Estimated path
input: DataSet;
for *Every Frame k of the Data set* **do**
1) Capture new frame I_k;
2) Extract and match features between Ik_1 and I_k;
3) Compute essential matrix for image pair Ik_1, I_k;
4) Decompose essential matrix into R_k and t_k, and form T_k;
5) Compute relative scale and rescale t_k accordingly;
6) Concatenate transformation by computing $C_k = C_{k1}T_k$;
7) Repeat from 1).
end

Algorithm 1: implemented Algorithm

VIII. Matching and Robustness :

A. Feature selection and matching:

There are two main approaches to find feature points and their correspondences. The first one is to find features in one image and track them in the following images using local search techniques, such as correlation. The second one is to independently detect features in all the images and match them based on some similarity metric between their descriptors. The former approach is more suitable when the images are taken from nearby viewpoints, whereas the latter is more suitable when a large motion or viewpoint change is expected. Early research in VO is opted for the former approach [2][5] while the works in the last decade concentrated on the latter approach [1], [6][9]. The reason is that early works were conceived for small-scale environments, where images were taken from nearby viewpoints, while in the last few decades, the focus has shifted to large-scale environments, and so the images are taken as far apart as possible from each to limit the motion-drift-related issues.

*1) **Feature Detection:**:* During the feature-detection step, the image is searched for salient keypoints that are likely to match well in other images. A local feature is an image pattern that differs from its immediate neighborhood in terms of intensity, color, and texture. For VO, point detectors, such as corners or blobs, are important because their position in the image can be measured accurately. A corner is defined as a point at the intersection of two or more edges. A blob is an image pattern that differs from its immediate neighborhood in terms of intensity, color, and texture. It is not an edge, nor a corner. The appealing properties that a good feature detector should have are: localization accuracy (both in position and scale), repeatability (i.e., a large number of features should be redetected in the next images), computational efficiency, robustness (to noise, compression artifacts, blur), distinctiveness (so that features can be accurately matched across different images), and invariance to both photometric (e.g., illumination) and geometric changes [rotation, scale (zoom), perspective distortion]. The VO literature is characterized by many point-feature detectors, such as corner detectors (e.g., Moravec [2], Forstner [10], Harris [11], Shi-Tomasi [12], and FAST [13]) and blob detectors (SIFT [14], SURF [15], and CENSURE [16]). An overview of these detectors can be found in [17]. Each detector has its own pros and cons. Corner detectors are fast to compute but are less distinctive, whereas blob detectors are more distinctive but slower to detect. Additionally, corners are better localized in image position than blobs but are less localized in scale. This means that corners cannot be redetected as often as blobs after large changes in scale and viewpoint. However, blobs are not always the right choice in some environmentsfor instance, SIFT automatically neglects corners that urban environments are extremely rich of. For these reasons, the choice of the appropriate feature detector should be carefully considered, depending on the computational constraints, real-time requirements, environment type, and motion baseline (i.e., how nearby images are taken). An approximate comparison of properties and performance of different corner and blob detectors is given in Figure 1.Notice that SIFT, SURF, and CENSURE are not true affine invariant detectors but were empirically found to be invariant up to certain changes of the viewpoint. A performance evaluation of feature detectors and descriptors for indoor VO has been given in [18] and for outdoor environments in [9] and [19]. Every feature detector consists of two stages. The first is to apply a feature-response function on the entire image [such as the corner response

function in the Harris detector or the difference-of-Gaussian (DoG) operator of the SIFT]. The second step is to apply nonmaxima suppression on the output of the first step. The goal is to identify all local minima (or maxima) of the feature-response function. The output of the non maxima suppression represents detected features. The trick to make a detector invariant to scale

	Corner Detector	Blob Detector	Rotation Invariant	Scale Invariant	Affine Invariant	Repeatability	Localization Accuracy	Robustness	Efficiency
Haris	x		x			+++	+++	++	++
Shi-Tomasi	x		x			+++	+++	++	++
FAST	x		x	x		++	++	++	++++
SIFT		x	x	x	x	+++	++	+++	+
SURF		x	x	x	x	+++	++	++	++
CENSURE		x	x	x	x	+++	++	+++	+++

Fig. 5. Comparison of feature detectors: properties and performance.

*2) **Feature Matching:**:* The feature-matching step searches for corresponding features in other images. Figure-6 shows the SIFT features matched across multiple frames overlaid on the first image. The set of matches corresponding to the same feature is called feature track. The simplest way for matching features between two images is to compare all feature descriptors in the first image to all other feature descriptors in the second image. Descriptors are compared using a similarity measure. If the descriptor is the local appearance of the feature, then a good measure is the SSD or NCC. For SIFT descriptors, this is the Euclidean distance.

Fig. 6. SIFT-feature tracks.

3) Number of features and distribution: The distribution of the features in the image has been found to affect the VO results remarkably [1], [9], [29]. In particular, more features provide more stable motion-estimation results than with fewer features, but at the same time, the key points should cover the image as evenly as possible. To do this, the image can be partitioned into a grid, and the feature detector is applied to each cell by tuning the detection thresholds until a minimum number of features are found in each sub image [1]. As a rule of the thumb, 1,000 features is a good number for a 640x480-pixel image.

B. outlier removal

Matched points are usually contaminated by outliers, that is, wrong data associations. Possible causes of outliers are image noise, occlusions, blur, and changes in viewpoint and illumination for which the mathematical model of the feature detector or descriptor does not account for. For instance, most of the feature-matching techniques assume linear illumination changes, pure camera rotation and scaling (zoom), or affine distortion. However, these are just mathematical models that approximate the more complex reality (image saturation, perspective distortion, andmotion blur). For the camera motion to be estimated accurately, it is important that outliers be removed. Outlier rejection is the most delicate task in VO. An example VO result before and after removing the outliers is shown in Figure 7.

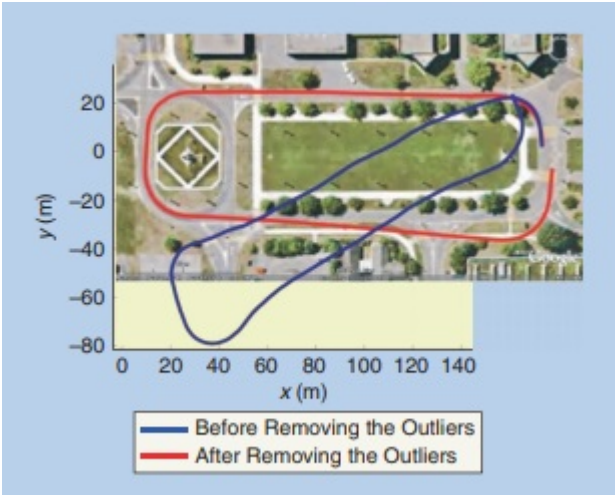

Fig. 7. Comparison between VO trajectories estimated before and after removing the outliers.

*1) **RANSAC**:* The solution to outlier removal consists in taking advantage of the geometric constraints introduced by themotionmodel. Robust estimation methods, such as M-estimation [32], case deletion, and explicitly fitting and removing outliers [33], can be used but these often work only if there are relatively few outliers. RANSAC [34] has been established as the standard method formodel estimation in the presence of outliers. The idea behind RANSAC is to compute model hypotheses from randomly sampled sets of data points and then verify these hypotheses on the other data points. The hypothesis that shows the highest consensus with the other data is selected as a solution. For two-view motion estimation as used in VO, the estimated model is the relative motion (R, t) between the two camera positions, and the data points are the candidate feature

correspondences. Inlier points to a hypothesis are found by computing the point-to-epipolar line distance [35]. The point-to-epipolar line distance is usually computed as a first-order approximation called Sampson distancefor efficiency reasons [35]. An alternative to the point-to-epipolar line distance is the directional error proposed by Oliensis [36]. The directional error measures the angle between the ray of the image feature and the epipolar plane. The authors claim that the use of the directional error is advantageous for the case of omnidirectional and wide-angle cameras but also beneficial for the standard camera case.

IX. IMPLEMENTATION

A. Preprocessing :

The image frames were undistorted using the MATLAB function provided in the input. For applications such as Visual Odometry, it is important to know the real world location of points since the nonlinear nature of the lens distortion makes the problem challenging.

B. Extraction of Camera Parameters :

Intrinsic matrix of the camera is calculated using the the MATLAB function ReadCameraModel.m. The function gives the values of focal length (f_x and f_y) and Principal point offset (c_x and c_y).

C. Feature detection and Correspondence

SURF feature detector was used for detecting features. Then descriptor or feature vectors were extracted using extractFeatures function of MATLAB computer vision toolbox. matchFeatures function was used to locate the point with matching features.

D. Estimation of Fundamental matrix :

Estimating fundamental matrix is the key step involved in the whole pipeline.The 8-point normalized algorithm was implemented to obtain the fundamental matrix. Using the matching points, a RANSAC function was implemented to randomly sample 8 set of matching points and checking the score of the resulting matrix with reference to other matching points. A fitness score was used to update the value of fundamental matrix. The perpendicular distance of matching point from the epipolar line is computed . If the distance lies within the threshold selected we increment the score by 1. Thus, the fundamental matrix gets updated if its score is greater than the earlier score

E. Estimation of Essential matrix:

Essential Matrix is computed after this by using the camera intrinsic and Fundamental matrix. It is made sure that the rank of the matrix is two by only keeping two singular values in the diagonal matrix of the svd.

F. Extracting camera pose

For a given essential matrix $E = Udiag(1, 1, 0)V^T$ and first camera matrix $P1 = [I|0]$, there are four possible choices for the second camera matrix P2 namely :
$P2 = [UWV^T|+u3]$ or $[UWV^T|-u3]$ or $[UW^TV^T|+u3]$ or $[UW^TV^T|-u3]$ where W is orthogonal matrix equal to $[0-10; 100; 001]$

G. Selection of correct pose:

Now we need to select one correct pose out of 4 poses derived from previous steps. I have used the constrained which is specific for this project to estimate the correct pose : Assuming that car is moving forward only , we can filter out the transform which has positive z translation. Rotation and Translation condition around y axis : For the current scenario, rotation can be assumed to be only around y axis. So that we can filter out the rotation matrix whose rotation component along other axes is approximately zero. While selecting the correct pose, the pose with minimum y translation is also one of the condition utilized in computing the correct pose. Finally, if none of the poses are able to satisfy all the conditions then pose of the world frame is selected as correct pose.

H. Camera pose update:

The last step before getting the final camera pose is updating the pose with respect to previous frame. For updating the pose of first frame we use the pose of the world frame and then subsequent frames uses the pose of previous frames. Simple equation for that is given below :

$$R_next = Rcurrent * R_next$$

I. Results and interpretations:

The code seems to work, but some times it drifts far away to give a wrong tracking, we assume its due to the outliers even though that we used RANSAC

X. CONCLUSION

This project was a great experience that allowed me to master the basic material of this course (computer vision for robotics) and to know how to look for new models and how to implement them.

REFERENCES

[1] C. N. Fischer, R. K. Cytron & R. J. LeBlanc. Crafting a Compiler, 2nd ed.,Boston: Pearson Education, 2010.

[2] D. Nister, O. Naroditsky, and J. Bergen, Visual odometry, in Proc. Int. Conf. Computer Vision and Pattern Recognition, 2004, pp. 652659.

[3] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature, vol. 293, no. 10, pp. 133135, 1981.

[4] C. Harris and J. Pike, 3d positional integration from image sequences, in Proc. Alvey Vision Conf., 1988, pp. 8790.

[5] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu,Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys, Building rome on a cloudless day, in Proc. European Conf. Computer Vision, 2010, pp. 368381.

[6] H. Moravec, Obstacle avoidance and navigation in the real world by a seeing robot rover, Ph.D. dissertation, Stanford Univ., Stanford, CA, 1980.
[7] L. Matthies and S. Shafer, Error modeling in stereo navigation, IEEE J. Robot. Automat., vol. 3, no. 3, pp. 239248, 1987.
[8] L. Matthies, Dynamic stereo vision, Ph.D. dissertation, CarnegieMellon Univ., Pittsburgh, PA, 1989.
[9] S. Lacroix, A. Mallet, R. Chatila, and L. Gallo, Rover self localization in planetary-like environments, in Proc. Int. Symp. Articial Intelligence, Robotics, and Automation for Space (i-SAIRAS), 1999, pp. 433440.
[10] C. Olson, L. Matthies, M. Schoppers, and M. W. Maimone, Robust stereo ego-motion for long distance navigation, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2000, pp. 453458.
[11] M. Hannah, Computer matching of areas in stereo images, Ph.D. dissertation, Stanford Univ., Stanford, CA, 1974.
[12] H. Moravec, Towards automatic visual obstacle avoidance, in Proc. 5th Int. Joint Conf. Artificial Intelligence, Aug. 1977, p. 584.
[13] W. Forstner, A feature based correspondence algorithm for image matching, Int. Arch. Photogrammetry, vol. 26, no. 3, pp. 150166, 1986.
[14] C. Olson, L. Matthies, M. Schoppers, and M. Maimone, Rover navigation using stereo ego-motion, Robot. Autonom. Syst., vol. 43, no. 4, pp. 215229, 2003.
[15] A. Milella and R. Siegwart, Stereo-based ego-motion estimation using pixel tracking and iterative closest point, in Proc. IEEE Int. Conf. Vision Systems, pp. 2124, 2006.
[16] A. Howard, Real-time stereo visual odometry for autonomous ground vehicles, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2008, pp. 39463952.
[17] Y. Cheng, M. W. Maimone, and L. Matthies, Visual odometry on the mars exploration rovers, IEEE Robot. Automat. Mag., vol. 13, no. 2, pp. 5462, 2006.
[18] M. Maimone, Y. Cheng, and L. Matthies, Two years of visual odometry on the mars exploration rovers: Field reports, J. Field Robot., vol. 24, no. 3, pp. 169186, 2007.
[19] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM, vol. 24, no. 6, pp. 381395, 1981.
[20] C. Tomasi and J. Shi, Good features to track, in Proc. Computer Vision and Pattern Recognition (CVPR 94), 1994, pp. 593600.
[21] P. Besl and N.McKay, Amethod for registration of 3-d shapes, IEEE Trans. Pattern Anal. Machine Intell., vol. 14, no. 2, pp. 239256, 1992.
[22] A. Comport, E. Malis, and P. Rives, Accurate quadrifocal tracking for robust 3d visual odometry, in Proc. IEEE Int. Conf. Robotics and Automation, 2007, pp. 4045.
[23] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge U.K.: Cambridge Univ. Press, 2004.
[24] D. Nister, O. Naroditsky, and J. Bergen, Visual odometry for ground vehicle applications, J. Field Robot., vol. 23, no. 1, pp. 320, 2006.
[25] P. I. Corke, D. Strelow, and S. Singh, Omnidirectional visual odometry for a planetary rover, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2005, pp. 40074012.
[26] M. Lhuillier, Automatic structure and motion using a catadioptric camera, in Proc. IEEE Workshop Omnidirectional Vision, 2005, pp. 18.
[27] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, Visual vehicle egomotion estimation using the Fourier-Mellin transform, in Proc. IEEE Intelligent Vehicles Symp., 2007, pp. 450455.
[28] J. Tardif, Y. Pavlidis, and K. Daniilidis, Monocular visual odometry in urban environments using an omnidirectional camera, in Proc. IEEE/ RSJ Int. Conf. Intelligent Robots and Systems, 2008, pp. 25312538.
[29] M. J. Milford and G.Wyeth, Single camera vision-only SLAM on a suburban road network, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA 08), 2008, pp. 36843689.
[30] D. Scaramuzza and R. Siegwart, Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles, IEEE Trans. Robot. (Special Issue on Visual SLAM), vol. 24, no. 5, pp. 1015
[31] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, Real time localization and 3d reconstruction, in Proc. Int. Conf. Computer Vision and Pattern Recognition, 2006, pp. 363370.
[32] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, in Proc.

IEEE Int. Conf. Robotics and Automation (ICRA 09), 2009, pp. 42934299.

[33] A. Pretto, E. Menegatti, and E. Pagello, Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation, in Proc. IEEE Int. Conf. Robotics and Automation, 2011, pp. 32893296.

[34] D. Nister, An efficient solution to the five-point relative pose problem, in Proc. Int. Conf. Computer Vision and Pattern Recognition, 2003, pp. 195202.

[35] M. Milford, G. Wyeth, and D. Prasser, RatSLAM: A hippocampal model for simultaneous localization and mapping, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA 04), 2004, pp. 403408.

[36]] B. Liang and N. Pears, Visual navigation using planar homographies, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA 02), 2002, pp. 205210.

[37] Q. Ke and T. Kanade, Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection, in Proc. Computer Vision and Pattern Recognition (CVPR), June 2003, pp. 390397.

[38] H. Wang, K. Yuan, W. Zou, and Q. Zhou, Visual odometry based on locally planar ground assumption, in Proc. IEEE Int. Conf. Information Acquisition, 2005, pp. 5964.

[39] J. Guerrero, R. Martinez-Cantin, and C. Sagues, Visual map-less navigation based on homographies, J. Robot. Syst., vol. 22, no. 10, pp. 569581, 2005.

[40] D. Scaramuzza, 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints, Int. J. Comput. Vis., vol. 95, no. 1, pp. 7485, 2011.

[41] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints, in Proc. IEEE Int. Conf. Computer Vision (ICCV), Kyoto, Oct. 2009, pp. 14131419.

[42] F. Fraundorfer, D. Scaramuzza, and M. Pollefeys, A constricted bundle adjustment parameterization for relative scale estimation in visual odometry, in Proc. IEEE Int. Conf. Robotics and Automation, 2010, pp. 18991904.

[43] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel, Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle, in Tagungsband Autonome Mobile Systeme, Reihe Informatik aktuell. Levi, Schanz, Lafrenz, and Avrutin, Eds. Berlin, Springer-Verlag, 2005, pp. 157163.

[44] K. Konolige, M. Agrawal, and J. Sol, Large scale visual odometry for rough terrain, in Proc. Int. Symp. Robotics Research, 2007.

[45] J. Tardif, M. G. M. Laverne, A. Kelly, and M. Laverne, A new approach to vision-aided inertial navigation, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2010, pp. 41614168.

[46] A. I. Mourikis and S. Roumeliotis, A multi-state constraint kalman filter for vision-aided inertial navigation, in Proc. IEEE Int. Conf. Robotics and Automation, 2007, pp. 35653572.

[47] E. Jones and S. Soatto, Visual-inertial navigation, mapping and localization: A scalable real-time causal approach, Int. J. Robot. Res., vol. 30, no. 4, pp. 407430, 2010.

[48] H. Durrant-Whyte and T. Bailey, Simultaneous localization and mapping (SLAM): Part I. The essential algorithms, Robot. Automat. Mag., vol. 13, no. 2, pp. 99110, 2006.

[49] T. Bailey and H. Durrant-Whyte, Simultaneous localisation and mapping (SLAM): Part II. State of the art, Robot. Automat. Mag., vol. 13, no. 3, pp. 108117, 2006.

[50] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in Proc. Int. Conf. Computer Vision, 2003, pp. 1403 1410.

[51] G. Klein and D. Murray, Parallel tracking and mapping for small ar workspaces, in Proc. Int. Symp. Mixed and Augmented Reality, 2007, pp. 225234.

[52] H. Strasdat, J. Montiel, and A. Davison, Real time monocular SLAM: Why filter? in Proc. IEEE Int. Conf. Robotics and Automation, 2010, pp. 26572664.

[53] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, 3-D motion and structure from 2-D motion causally integrated over time: Implementation, in Proc. European Conf. Computer Vision, 2000, pp. 734750.

[54] M. C. Deans, Bearing-only localization and mapping, Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, 2002.

[55] L. A. Clemente, A. J. Davison, I. Reid, J. Neira, and J. D. Tardos, Mapping large loops with a single hand-held camera, in Proc. Robotics Science and Systems, 2007.

[56] T. Lemaire and S. Lacroix, Vision-based SLAM: Stereo and monocular approaches, Int. J. Computer Vision, vol. 74, no. 3, pp. 343364, 2006.
[57] E. Eade and T. Drummond, Monocular SLAM as a graph of coalesced observations, in Proc. IEEE Int. Conf. Computer Vision, 2007, pp. 18.
[58] G. Klein and D. Murray, Improving the agility of keyframe-based SLAM, in Proc. European Conf. Computer Vision, 2008, pp. 802815.
[59] K. Konolige and M. Agrawal, FrameSLAM: From bundle adjustment to real-time visual mappping, IEEE Trans. Robot., vol. 24, no. 5, pp. 10661077, 2008.
[60] A. Handa, M. Chli, H. Strasdat, and A. J. Davison, Scalable active matching, in Proc. Int. Conf. Computer Vision and Pattern Recognition, 2010, pp. 15461553.
[61] J. Civera, O. Grasa, A. Davison, and J. Montiel, 1-point RANSAC for ekf filtering: Application to real-time structure from motion and visual odometry, J. Field Robot., vol. 27, no. 5, pp. 609631, 2010.
[62] H. Strasdat, J. Montiel, and A. J. Davison, Scale drift-aware large scale monocular SLAM, in Proc. Robotics Science and Systems, 2010.
[63] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, RSLAM: A system for large-scale mapping in constant-time using stereo, Int. J. Computer Vision, vol. 94, no. 2, pp. 198214, 2010.
[64] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, An Invitation to 3D Vision, from Images to Models. Berlin: Springer-Verlag, 2003.
[65] C. Geyer and K. Daniilidis, A unifying theory for central panoramic systems and practical applications, in Proc. European Conf. Computer Vision, 2000, pp. 445461.
[66] D. Scaramuzza, A. Martinelli, and R. Siegwart, A flexible technique for accurate omnidirectional camera calibration and structure from motion, in Proc. IEEE Int. Conf. Computer Vision Systems (ICVS) 2006, Jan. 2006, pp. 4553.
[67] D. Scaramuzza, Omnidirectional vision: From calibration to robot motion estimation Ph.D. dissertation, ETH Zurich, 2008.