ICP Stereo Visual Odometry for Wheeled Vehicles based on a 1DOF Motion Prior

Yanhua Jiang[1,2], Huiyan Chen[1], Guangming Xiong[1], Davide Scaramuzza[2]

Abstract— **In this paper, we propose a novel, efficient stereo visual-odometry algorithm for ground vehicles moving in outdoor environments. To avoid the drawbacks of computationally-expensive outlier-removal steps based on random-sample schemes, we use a single-degree-of-freedom kinematic model of the vehicle to initialize an Iterative Closest Point (ICP) algorithm that is utilized to select high-quality inliers. The motion is then computed incrementally from the inliers using a standard linear 3D-to-2D pose-estimation method without any additional batch optimization. The performance of the approach is evaluated against state-of-the-art methods on both synthetic data and publicly-available datasets (e.g., KITTI and Devon Island) collected over several kilometers in both urban environments and challenging off-road terrains. Experiments show that the our algorithm outperforms state-of-the-art approaches in accuracy, runtime, and ease of implementation.**

I. INTRODUCTION

Visual Odometry (VO) is the process of estimating the motion of a moving vehicle using video input from its onboard cameras. VO is a valid alternative or supplement to other ego-motion-estimation systems, such as wheel odometry, global positioning system (GPS), inertial measurement units (IMUs), or laser scanners.

In the past 30 years, a tremendous amount of research has focused on visual odometry using monocular [1], [2] and stereo [1], [3], [4] cameras. Most of the VO approaches work by detecting robust point correspondences between consecutive frames, by removing the wrong associations (i.e., outliers), and, finally, by estimating the incremental motion from the remaining inliers. A comprehensive tutorial on VO can be found in [5], [6].

One of the most challenging problems in VO is data association. For motion estimation in the presence of outliers, the RAndom SAmple Consensus (RANSAC) [7] has been established as the standard method; techniques, such as 5-point RANSAC [8], for monocular VO, and 3-point RANSAC [9], for stereo VO, are now widely used. However, because the number of RANSAC iterations is exponential in the number of parameters that describe the motion,[3] several works have used motion constraints to reduce the number of iterations of RANSAC. In [10], [11], non-holonomic constraints of wheeled vehicles were exploited to parameterize the motion using only one parameter, thus, significantly reducing the number of RANSAC iterations. Until now, non-holonomic constraints of wheeled vehicles have been applied only to monocular systems [11], [12].

In this work, we are interested in incrementally estimating the ego-motion of the vehicle purely from a sequence of stereo images while taking full advantage of the kinematic model. We extend the one-point algorithm proposed in [11] to stereo cameras, where the motion prior is not directly used to reject outliers but to provide a good and efficient initialization for an ICP-based 3D registration. We show that, after the 3D registration, a set of *high quality* (i.e., low depth uncertainty) inliers is preserved; then, motion estimation is done over all found inliers through minimizing the reprojection error. Compared to RANSAC-based stereo-VO methods, which are based on random sampling, our approach has the advantage of being fully deterministic (i.e., given a set of correspondences, it returns always the same motion estimate, unlike RANSAC). Additionally, it is also more accurate, given that only a small set of high-quality inliers is used to estimate the motion (unlike standard methods, which use all the inliers).

The reminder of the paper is organized as follows. In Section II, we review the related work. In Section III-B, we briefly describe the feature detection and tracking technique. In Section III-C, we explain how to compute the motion prior used to initialize the ICP. In Section III-F, we describe the proposed model-based ICP method for selecting high quality inliers. Finally, in Section IV, we present the experimental results on both synthetic and real data.

II. RELATED WORK

Works on stereo-based pose estimation from two sets of corresponding features can be divided into two categories depending on whether the feature correspondences are specified in two or three dimensions. If the two feature sets are both specified in 3D, the problem takes the name of *3D-to-3D registration* or *absolute orientation*. The solution consists of finding the transformation that minimizes the L_2 distance between the two 3D feature sets. If one feature set is specified in 3D and the other one in 2D, the problem takes the name of *3D-to-2D pose estimation* or *Perspective from n points* (PnP). The solution, in this case, consists of finding the transformation that minimizes the image reprojection error of the 3D points into the other image [13]. The minimal case involves three 3D-to-2D correspondences. This is called *perspective from three points* (P3P) and is usually implemented in a 3-point–RANSAC fashion [1].

[1]Y. Jiang, [1]H. Chen and [1]G. Xiong(86-10-68918652; xiongguangming@bit.edu.cn) are with the Intelligent Vehicle Research Center, Beijing Institute of Technology and Key Laboratory of Biomimetic Robots and Systems, Ministry of Education of China. This work was partially supported by the National Natural Science Foundation of China (Grant No. 91120010)

[2]D. Scaramuzza and [1]Y. Jiang are with the Robotics and Perception Group, University of Zurich, Switzerland—http://rpg.ifi.uzh.ch

[3]E.g., five parameters are needed to describe the unconstrained motion (6DoF) of a single camera: six minus the scale factor.

In his landmark paper [1], Nister pointed out that 3D-to-2D methods are superior to 3D-to-3D methods. The reason is that 3D points carry higher uncertainty since they are computed via stereo triangulation of noisy 2D image points. Given their superiority, 3D-to-2D methods are now widely used in VO; 3-point RANSAC (or P3P RANSAC) has become the golden standard algorithm for robust motion estimation in the presence of outliers [2], [14], [15].

3D-to-3D methods were popular in early vision-based motion-estimation systems, especially in a series of works by NASA [3]. The Iterative Closest Point algorithm [16] can be regarded as an iterative solution of the 3D-to-3D problem, which is widely used in laser-scanner–based registration problems [17]. Only a few works have applied ICP to stereo VO [18], [19]. In [18], Milella and Siegwart integrated image intensity and 3D stereo information into an ICP scheme to implement 6DOF ego-motion estimation. In [19], Tomono implemented a randomized-ICP algorithm (combined with the RANSAC paradigm) to estimate camera pose from a reference image and an 3D Map.

In the last three years, due to the rapid developments of RGB-D sensors, several works have reintroduced ICP (combined with photometric information) for motion computation [20], [21]. Newcombe et al. [22] computed camera pose via ICP registration with Truncated Signed Distance Function. In [20], Tykkälä considered VO as a 3D surface registration problem using dense structure information from images; a bi-objective cost function was proposed to minimize both photometric and depth error between subsequent image frames in order to compute the camera-motion parameters. However, all these works are limited to static indoor environments. Conversely, in this paper, we tackle challenging dynamic outdoor environments.

The use of vehicle kinematic constraints for VO has appeared in several works [10]–[12], [23]. Vatani et al. [23] used the Ackermann steering principle and the planar assumption to constraint the motion model; 2D planar motion was then estimated directly using pixel displacement from a down-looking camera. Using the car kinematic model, Zhu et al. [12] computed the motion parameters by solving a quadratic polynomial from equations established by epipolar constraint. For general wheeled vehicles, Scaramuzza et al. [10], [11] showed that, due to the existence of the Instantaneous Center of Rotation, the motion can be locally described as planar and circular, and, therefore, the motion model complexity is reduced to 1DoF, leading to a one-point minimal solver. However, their restrictive model is based on the assumption that the motion is locally planar and circular, which can often be violated in outdoor environments, even when the road looks perfectly flat. If we look at the combination of the camera and the car as a spring-mass system, when acceleration, deceleration, or sharp turns occur, the planar-motion hypothesis may fail due to the dynamic characteristics of suspensions and tires. For these reasons, in their latest work [24] the authors relaxed the constraint of locally planar and circular motion. A prior was used to compute the target distribution of the 6DoF motion, leading to increased performance compared to the one point algorithm. However, their method was developed only for monocular systems and is, therefore, not directly suitable for stereo applications.

```
//FRAME 0        Initializaton
{ $u_0^l$ }=DETECT_FAST_CORNERS( $I_0^l$ );
{$u_0^r, N$}=LEFT_RIGHT_MATCH ( $I_0^l$, $I_0^r$ );
{ $X_0$ }=STEREO_TRIANGULATE( $u_0^l$, $u_0^r$ );
//FRAME k        k=1,...
for (k=1; k++) {
    { $u_k^l$ }=LUCAS_KANADE_TRACKING ( $u_{k-1}^l$ );
    { $u_k^r, N$}=LEFT_RIGHT_MATCH( $u_k^l$, $u_k^r$ );
    { $X_k$ }=STEREO_TRIANGULATE( $u_k^l$, $u_k^r$ );
    //One point algorithm        i=1,...
    for (i=1; i $\leq$ N; i++)
        {$^i\theta$}=ONE_POINT_ESTIMATE( $^iu_k^l, ^iu_{k-1}^l, K$ );
    $\theta^\star$=argmax(Histogram($^i\theta$), i=1:$N$);
    {$R^1, t^1$}=MOTION_MODEL( $\theta^\star$ );
    //ICP refinement
    {$X_k'$}=TRANSFORM( $X_k, R^1, t^1$ );
    {$X_k^c$}=ICP_REFINEMENT( $X_k'$, $X_{k-1}$ );
    for (i=1; $i \leq N$; i++)
        {$^id$}=RMS_EUC_DISTANCE( $^iX_k^c, ^iX_{k-1}$ );
    {$d_{th}$}=HALF_NORMAL_FITTING( Hisgtogram{$^id$} );
    { in_$u_k^l$, in_$u_{k-1}^l$, in_$X_{k-1}$ }=THRESHOLDING( {$^id$}, $d_{th}$ );
    //Closed-form optimization
    {$R_k, t_k$}=EPNP( in_$u_k^l$, in_$u_{k-1}^l$, in_$X_{k-1}$ );
}
```

TABLE I: Pseudo code of our algorithm. ONE_POINT_ESTIMATE implements Eq. (5); MOTION_MODEL implements Eq. (6); in_$\star$ denotes the inliers; all notations can be found in the corresponding sections. Notice that for each new frame, new FAST corners are initialized and tracked to the next frame. For simplicity, we omitted this in the pseudo code.

III. Approach

Our VO algorithm is based on four steps. In the first step, we detect FAST corners [25] in the left image and track them using a Lucas-Kanade tracker (KLT) [26]. Stereo matches are then found based on Census transform [27]. In the second step, the motion prior is estimated from 2D feature correspondences using the one-point algorithm [11] and the scale determined through a voting scheme. In the third step, the two 3D point sets are aligned using ICP and inliers are identified without any random sampling; due to the accurate motion prior, the algorithm converges very fast. Finally, all selected inlier features are used to estimate the motion parameters using EPnP. The pseudo-code of the algorithm is given in Table I.

A. Problem Formulation and Notations

Let I_{k-1}^l, I_{k-1}^r, I_k^l, I_k^r be two left-right image pairs at times $k-1$ and k, respectively. We denote by u_{k-1}^l, u_{k-1}^r, u_k^l, u_k^r their image correspondences and by X_{k-1} and X_k the triangulated 3D points (Figure 1). Given intrinsic calibration matrix K (which is assumed constant throughout the whole sequence), the baseline b, the rotation matrix R_k, and the translation vector t_k, we can write:

$$\begin{aligned} \hat{u}_{k-1}^l &\sim P^l \hat{X}_{k-1}, \\ \hat{u}_{k-1}^r &\sim P^r \hat{X}_{k-1}, \qquad (1) \\ \hat{u}_k^l &\sim P^{l'} \hat{X}_{k-1}, \end{aligned}$$

where $\hat{\cdot}$ denotes the homogeneous coordinates and $\sim$ means equality up to scale. $P^l = K[I|0]$, $P^{l'} = K[R_k|t_k]$ are the left projection matrices at times $k-1$ and k respectively, while $P^r = K[I|[-b,0,0]^T]$ is the right projection matrix at time $k-1$. The alignment between the two $3D$ world points is then given by:

$$X_{k-1} = R_k \cdot X_k + t_k. \tag{2}$$

Normalized image coordinates of feature points are first computed as

$$\bar{u}_k = K^{-1} \cdot \hat{u}_k \tag{3}$$

and then projected onto the unit sphere (i.e., $||\bar{u}|| = 1$).

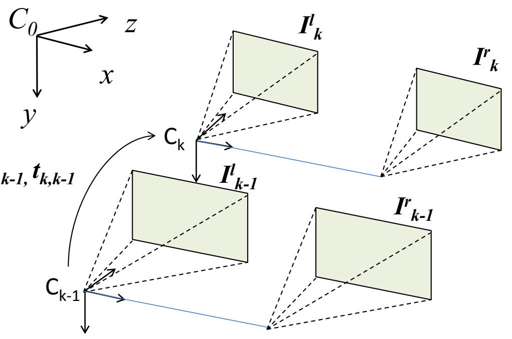


Fig. 1: C_0 refers to the initial camera position and act as the world reference frame. $R_{k,k-1}$ and $t_{k,k-1}$ denote the motion parameters from frame $k-1$ to frame k.

We solve the relative-motion estimation as a 3D-to-3D registration problem by computing the incremental motion parameters R_k and t_k that minimize the L_2 norm between the triangulated 3D points:

$$< R_k, t_k >= \arg\min_{R_k,t_k} \sum_i ||X^i_{k-1} - R_k \cdot X^i_k - t_k||. \tag{4}$$

Because 3D-to-3D registration methods are greatly affected by the depth uncertainty of the computed 3D points, it is crucial that the two points sets X_{k-1} and X_k contain only high-quality inliers, i.e., with very low uncertainty. The great advantage of 3D-to-3D methods with respect to 3D-to-2D approaches is that they do not need to reproject the 3D points in the images, saving, thus, computation. This peculiariy makes this class of methods extremely attractive for low-power embedded computers, such as those used on space rovers [3].

B. Feature Detection and Tracking

FAST corners are detected in I^l_{k-1} and tracked in I^l_k using a KLT tracker with subpixel refinement. Stereo correspondences (between I^l_{k-1} and I^r_{k-1} and between I^l_k and I^r_k) are determined using Census transform (calculated on a 9×9 pixels patch) and epipolar geometry.

To keep a minimum number of feature points in every frame, we use the bucketing technique mentioned in [14], which guarantees that the features are nearly uniformly distributed in the whole image: the image is first partitioned into cells; when the number of tracked features in each cell is lower than a threshold (20 in this work, cell size = 100×100 pixels), a new detection is triggered.

C. Motion Model

As shown in [10], [11] for any wheeled robot, the existence of the *instantaneous center of rotation* makes it possible to describe the *local* motion of the vehicle as planar and circular. Under this constraint, the motion model complexity reduces to one degree of freedom (i.e., the rotation angle), leading to a one-point minimal solver. This implies that it is sufficient to use only one feature correspondence to recover the yaw angle increment θ as well as the translation angle $\psi = \theta/2$ [11]. Consider a set of $2D$ feature correspondences $\{^i u_{k-1} \leftrightarrow^i u_k\}, i = 1 : N$, where N is the number of feature points. For each feature correspondence $^i u_{k-1} \leftrightarrow^i u_k$, the yaw angle increment can be determined using [11] as

$$^i\theta = -2\arctan \frac{^i\bar{u}_{k-1}(2) \cdot^i \bar{u}_k(1) -^i \bar{u}_{k-1}(1) \cdot^i \bar{u}_k(2)}{^i\bar{u}_{k-1}(3) \cdot^i \bar{u}_k(2) +^i \bar{u}_{k-1}(2) \cdot^i \bar{u}_k(3)}. \tag{5}$$

Based on this 1DOF motion model, as proposed in [11] we compute the best estimate of the yaw angle increment as $\theta^\star = median\{^i\theta\}$. Using $\theta^\star$, we parametrize the rotation matrix $R^1 \in SO(3)$ and translation vector $t^1 \in \mathbb{R}^3$ as

$$R^1 = \begin{bmatrix} \cos(\theta^\star) & 0 & -\sin(\theta^\star) \\ 0 & 1 & 0 \\ \sin(\theta^\star) & 0 & \cos(\theta^\star) \end{bmatrix}, t^1 = \begin{bmatrix} \sin(\psi^\star) \\ 0 \\ \cos(\psi^\star) \end{bmatrix}, \tag{6}$$

where $\psi^\star = \theta^\star/2$ according to the 1DOF motion model.

D. Model Usability Analysis

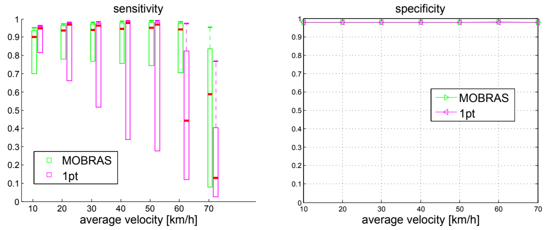
This restrictive motion model is based on two assumptions: i) locally planar and circular motion; ii) high frame-rate image input. The second requirement is relatively easy to fulfill with the current camera technology; however, the first assumption can often be violated in outdoor environments, even when the road looks perfectly flat. Indeed, if we look at the combination of the camera and the car as a spring-mass system, when acceleration, deceleration, or sharp turns occur, the planar-motion hypothesis may fail due to the dynamic characteristics of suspensions and tires. For these reasons, in their latest work [24], the authors propose a Model-based Random Sampling algorithm (called MOBRAS). They relax the constraint of locally planar and circular motion: the relative motion is modeled as a multivariate Gaussian distribution over the predominantly planar-circular motion which is computed from the image points according to (5) and (6), while the additional motion parameters (roll, pitch, and the elevation angles) are considered as zero-mean Gaussian variables. Motion hypotheses are then generated from this normal distribution and the inliers are identified through reprojection error.

In this section, we compare the outlier-removal performance of the original 1-Point algorithm [11] and MOBRAS [24] using simulation platform presented in our previous work [28]. We simulate a car moving in complicated path with many sharp turns as well as many accelerations and decelerations using different velocity level.

Since the inlier identification procedure can be regarded as a binary classification problem, we can use *sensitivity* and *specificity* as the evaluation metrics. We define True Positive (TP) as the number of *true inliers found* and False Negative (FN) as the number of *true inliers missed*. Then, we define True Negative (TN) as the number of *true outliers found* and False Positive (FP) as the number of *true outliers identified as inliers*. To compute the true inliers and true outliers, we use ground truth poses and a reprojection-error threshold of 1 pixel. Then *sensitivity* (also called true-positive rate or recall rate) and *specificity* (or true negative rate) can be computed as:

$$sensitivity = \frac{TP}{TP+FN}, \quad specificity = \frac{TN}{TN+FP} \tag{7}$$

In the ideal cases, we would like $sensitivity = 1$ and $specificity = 1$.



(a)

(b)

Fig. 2: Sensitivity and Specificity. Comparison between 1-Point [10] and MOBRAS [24] algorithms.

The relation between *sensitivity*, *specificity*, and the vehicle velocity is depicted in Figure 2. Thick red bars denote the median of the errors. The meaning of boxes is given in detail in Section IV. From the distribution of *sensitivity*, we can observe that MOBRAS finds on average more true inliers than the 1-Point algorithm, whose detection ratio decreases severely as the velocity increases. However, the *specificity* plot indicates that both methods suffer from false positives (even though very small, $specificity = 0.98$), which can corrupt the motion estimation result. However, we will show in the next sections that by leveraging the depth information from both cameras, it is possible to make *specificity* perfectly equal to 1 and, thus, improve the motion estimate.

E. Scale Computation

The 1-Point algorithm is a monocular method, which means that the metric scale (i.e., length) of the translation vector t^1 can only be obtained using additional sensor information, such as the speed from the CAN bus [10]. In stereo-vision applications, the scale is obviously a direct outcome of motion computation. However, as mentioned in section II, state-of-the-art methods are based on random schemes, such as RANSAC, and are very sensitive to the *quality* of inliers. Our goal is to avoid random schemes and rely on *good quality* inliers.

We use the depth information from triangulated features to get a fast estimate of the translation distance. Obviously, once the rotation is known, the translational component (in the absolute scale) can be recovered from a single 3D-point correspondence. For more than one correspondence, due to the presence of outliers, we cannot directly use least-square methods. Here, we propose a model-based *scaling* method. The main idea is to compute the component of the translation vector of every 3D correspondence along the direction t^1 (estimated through the 1-Point algorithm), and, then, to get the model scale s^1 by voting. Consider two triangulated point clouds at times $k-1$ and k. For every 3D point correspondence $\{^iX_{k-1}, ^iX_k\}$, we calculate its translation vector using the rotation matrix R^1 estimated form the 1-Point algorithm, that is

$$^it = ^iX_{k-1} - R^1 \cdot ^iX_k.$$

Then, as shown in Figure 3, we compute the component ^is of ^it along the direction of t^1, which can easily calculated by dot product as

$$^is = ^it \cdot t^1.$$

After obtaining the set $\{^is\}$, we select the median value $s^1 = median\{^is\}$ as the best estimate of the scale. Finally, we rescale the translation hypothesis t^1 as

$$t^1 \leftarrow s^1 t^1.$$

Note that the median should be calculated in a reasonable range, which should be determined by the vehicle speed and frame rate. In the KITTI dataset, the speed is always below $90km/h = 25m/s$ and the frame rate is $10Hz$; therefore, we restrict the scale values to the interval [0,3] meters.

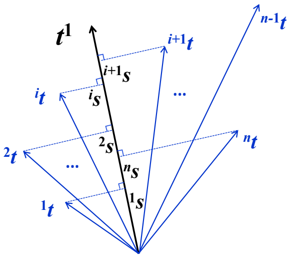


Fig. 3: Projection of the individual translation vectors onto the direction computed with the 1-Point algorithm.

F. Model-based ICP

ICP is a widely-used method for registration of 3D point clouds, which is particularly suitable for the automatic alignment of data generated by a laser scanner. ICP works by iteratively revising the transformation (translation and rotation) needed to minimize the distance between the points of two raw scans. In order to start, the algorithm needs an initial guess as input. This initial motion hypothesis is crucial for the ICP registration to be successful. In laser odometry for ground robots, this initial guess is provided through wheel odometry.

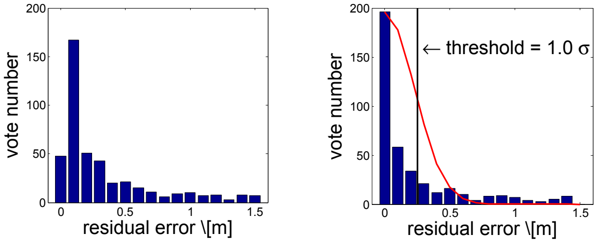
In our settings, we do not use any other additional source of information than cameras. As an initial hypothesis, we use the 1DOF motion model (R^1, t^1) computed in Section III-C, with t^1 rescaled are described in the previous section. This choice makes our ICP converge extremely fast, typically no more than 3 iterations. In the remainder of this paper, we will call our proposed algorithm MICP (Model-based ICP).

Given two 3D point sets X_{k-1}, X_k and the motion hypothesis R^1 and t^1, we first transform the point cloud X_k into X'_k as

$$X'_k = R^1 \cdot X_k + t^1.$$

Then, we implement an ICP refinement between X_{k-1} and X'_k to obtain the transformation R^c and t^c. We avoid the costly nearest-neighbor search using the known image-point correspondences. At each iteration, 3D correspondences whose residual error is larger than a given threshold are removed (in this work, we set the threshold to 2.0 meters). The process stops when the change in median residual error between two successive iterations is less than 0.1 meter.

By defining the residual errors d_i as the Euclidean distance between the model ${}^iX$ and registered one ${}^iX^c = R^c \cdot {}^i X'_k + t^c$, we can analyze the distribution $\{d_i\}$ of the residuals. First, we build a histogram of $\{d_i\}$ with a bin size of 0.1 meters. Figure 4 shows an example histogram H_d of residual errors d_i obtained from real data. As observed, the motion model has been improved remarkably by the ICP refinement, which makes the residual errors tend to be minimum.



(a) Residual histogram after 1 point registration

(b) Residual histogram after MICP registration

Fig. 4: Distribution of the residual errors after the 1-Point algorithm and after MICP.

After ICP refinement, it is straightforward to distinguish inliers and outliers by thresholding the residual errors with threshold d_{th}. However, a fixed threshold does not guarantee a sufficient level of feature quality on the entire dataset. In recent works utilizing ICP with *truncated signed distance function* for dense 3D registration, Normal distribution or *Student's t-distribution* was used to fit the residuals [22]. In this work, we fit the residual errors with a *half-normal distribution*, whose probability function is

$$P(x) = \frac{2\alpha}{\pi} e^{-x^2\alpha^2/\pi}. \quad (8)$$

Given the histogram H_d, we can estimate the parameters of the half-normal distribution using

$$\alpha = \frac{1}{\mu}, \quad (9)$$

$$\sigma^2 = \frac{\pi - 2}{2\alpha^2}, \quad (10)$$

where μ denotes the mean value of residual errors. The fitting result is shown in Figure 4(b), where the red curve represents the fitted half-normal probability function. Here, we empirically choose 1σ as the threshold to select the inliers.

G. Non-iterative optimization

An optimization step with all the selected inliers is eventually performed to refine the final estimate of inter-frame motion. We use Efficient PnP (EPnP) [29], which is a non-iterative method, whose computational complexity grows linearly with the number of correspondences used. The main idea behind (EPnP) is to express all 3D points as a weighted sum of four virtual control points, then the problem is reduced to estimating those coordinates. In this work, the optimization using EPnP costs on average only 3 milliseconds for each frame.

IV. Experiments

In this section, we test our algorithm on both synthetic and real data. In the simulation tests, MICP is compared with MOBRAS and standard P3P RANSAC, while in the real tests with state-of-the-art visual-odometry algorithms.

The performance evaluation of 6DOF motion parameters is carried out in terms of translation and rotation error, respectively. We use the error metrics proposed by the authors of the KITTI dataset [30]. Let us denote by gR and gt the ground-truth relative motion and by eR and et the estimated relative motion. We can then calculate the error transformation ΔT as:

$$\Delta T = \begin{bmatrix} \Delta R & \Delta t \\ 0 & 1 \end{bmatrix} = T_e {T_g}^{-1}, \quad (11)$$

where

$$T_g = \begin{bmatrix} gR & gt \\ 0 & 1 \end{bmatrix}, \; T_e = \begin{bmatrix} eR & et \\ 0 & 1 \end{bmatrix}.$$

Then, the translation error err_t is defined as the Euclidean norm of the translation vector Δt:

$$err_t = \|\Delta t\| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}. \quad (12)$$

The rotation error err_r is defined from the axis-angle representation of the rotation matrix ΔR as:

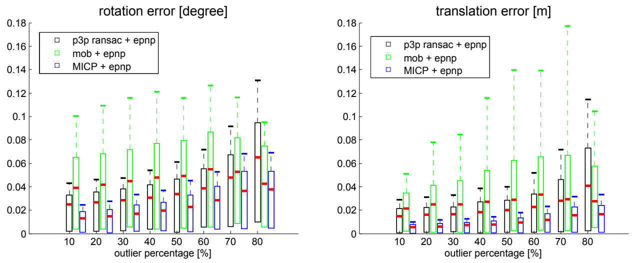
$$err_r = \arccos\left(\frac{trace(\Delta R) - 1}{2}\right). \quad (13)$$

A. Tests on synthetic data

In this section, we evaluate the performance of our algorithm on synthetic data. As in [11], we simulate a car moving in urban canyons composed of several facades. The first car location is at the origin while the second one is generated at random using the car kinematic model. To make the simulation more realistic, we set the simulation parameters as shown in Table II. To evaluate the robustness of the algorithm, we vary the fraction of outliers in the data from 10% to 80%.

Parameter	Values
Maximum yaw angle change θ_{max}	10 [degree]
Maximum pitch angle change β_{max}	1.0 [degree]
Maximum roll angle change γ_{max}	1.0 [degree]
Maximum elevation angle δ_{max}	0.5 [degree]
Variance of Gaussian noise	0.5 [pixel]
Moving distance ρ	1.0 [meters]
Maximum iteration number of RANSAC	1000
Maximum sampling number of MOBRAS	1000
Reprojection error threshold	0.5 [pixel]

TABLE II: Simulation parameters: Gaussian noise is assumed isotropically distributed in the x and y image directions. The rotation increment is modeled by three angles: yaw θ, pitch β, and roll γ. The planar component of the translation vector is modeled as a function of θ, while the elevation component by the angle δ. The translation length ρ is considered constant in the whole simulation.



(a)

(b)

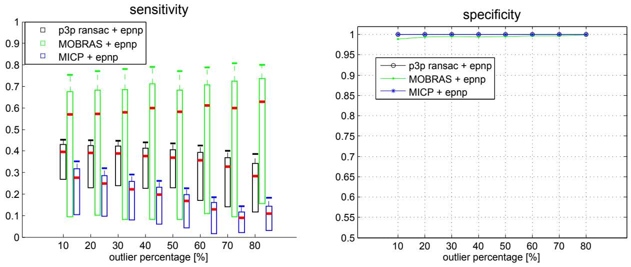
Fig. 6: Accuracy of final motion estimation versus the percentage of outliers.

We evaluate the performance of MOBRAS [24], P3P RANSAC [9] and our MICP algorithm. The evaluation consists of two parts:

1) *Sensitivity* and *Specificity* of true-inlier detection (we follow the definition of TP, FP, TN, FN, $sensitivity$, $specificity$ of Section III-C);
2) Accuracy of final motion estimation versus the percentage of outliers (for all the three algorithms, we compute the final pose from the detected inliers using EPnP [29]).

The resulting statistics for one thousand trials is shown in Figure 5 and Figure 6 using box plots. Thick red bars denote the median of the errors; the higher border of the rectangles denotes 75% percentiles while the lower border represents the minimum value of the errors; the top end of the dash lines denotes 90% percentiles. Note that we use 1000 random iterations in the P3P RANSAC method, which, according to the RANSAC statistics [7], should provide a probability of success of 99.97% (calculated assuming a fraction of outliers equal to 80%).

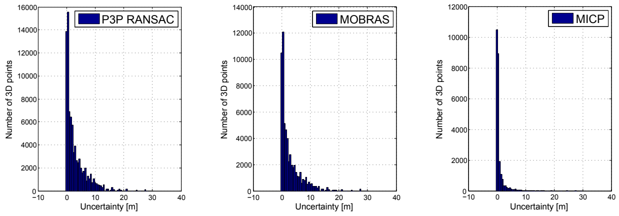
In Figure 5, it can be noticed that, although our MICP algorithm does not find a high percentage of inliers, its false positive rate is always zero (as much as P3P RANSAC). However, as observed in Figure 6, MICP+EPnP actually outperforms P3P RANSAC+EPnP and MOBRAS+EPnP, whose sensitivity is much higher. The reason for this can be found by investigating the quality of detected inliers. As shown in Table II, image noise up to 0.5 pixel is added to all the image feature locations. In the motion estimation step, 3D point locations are calculated from the noisy 2D measures. Since the depth of a 3D point is inversely proportional to the disparity of its corresponding image points, distant points will have larger uncertainty. Here, we are interested in analyzing the uncertainty distribution of the triangulated 3D points, which is shown in Figure 7. As expected, the distribution of the 3D-position uncertainty of the inliers found by MICP is significantly narrower than the distributions found with the other two methods. This confirms our previous claim that not only does MICP aim to find inliers but also *high quality* inliers, that is, features with low depth uncertainty. Conversely, points with high depth uncertainty cannot be rejected by RANSAC as the reprojection error of those points is small in any case. This explain the larger error of P3P RANSAC in the motion estimate in Figure 6.



(a)

(b)

Fig. 5: Sensitivity and specificity of true-inlier detection



(a)

(b)

(c)

Fig. 7: Distribution of the 3D position error with different algorithms: P3P RANSAC (a), MOBRAS (b), MICP (c).

B. Tests on the KITTI benchmark

We evaluate our algorithm on the KITTI benchmark, which consists of 22 stereo sequences. The camera baseline is 0.54 meter, the raw image resolution is 1392×512 pixels, and the image acquisition frame rate is on average 10 frames per second. For the training sequences 00-10, ground truth is provided, while for test sequences 11-21 there is no ground truth. The errors are evaluated as a function of trajectory length and car speed as defined in [30].

We compare both efficiency and accuracy of our algorithm with several algorithms based on RANSAC outlier-rejection schemes, including LIBVISO2 [31], TGVO [14], VO3pt [15], and VOFS [32]. We compare the results over all test sequences 11-21 (indeed, these are the only sequences for

Method	Translation	Rotation
MICP_VO	**2.13%**	**0.0065[deg/m]**
LIBVISO2	2.44%	0.0114[deg/m]
VO3pt	2.69%	0.0068[deg/m]
TGVO	2.94%	0.0077[deg/m]
VOFS	3.94%	0.0099[deg/m]

TABLE III: Overview of of average translation and rotation error on KITTI sequences 11-21.

which KITTI allows users to download results from the other contributors). All the performance statistics of these algorithms (including our MICP VO) are publicly available on the KITTI website.[1] An overview of the average translation and rotation errors calculated over the eleven test sequences for the different algorithms is shown in Table III. The results are sorted according to the average translation error. As observed, MICP_VO is superior than other algorithms in term of both translation and rotation, and the latter is typically the main source of drift in visual odometry. [2] Indeed, if we compare the trajectories estimated with the different VO algorithms (Figure 8), MICP_VO is the one that appears closer to the ground truth. More details on the error statistics can be found on the KITTI website.

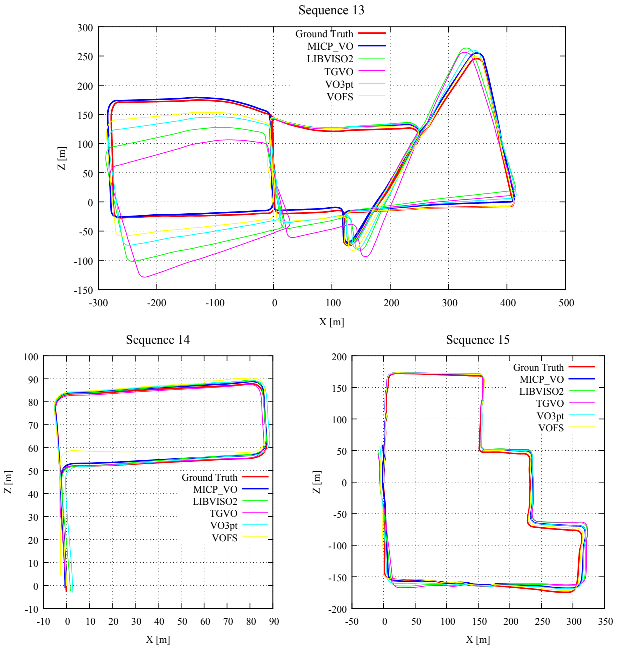


Fig. 8: Estimated Trajectories: Comparison between MICP_VO, LIBVISO2, TGVO, VO3pt, VOFS. Data from the KITTI evaluation website.

Tables IV and V show the execution-time statistics of MICP compared to the other algorithms. The test platform used all the experiments is a laptop PC with an Intel Core i7-3720QM CPU at 2.60 GHz and with 16GB of RAM. As observed, our system outperforms in execution time the other implementations, taking only 40ms (all the pipeline, including feature extraction and tracking).

	Timing [ms]		
Procedures	Min	Median	Max
Model prediction	0.58	1.179	2.483
ICP	0.925	2.977	5.798
Optimization	1.074	2.339	4.032

TABLE IV: Execution time of motion estimation pipeline. The vary of estimation time is primarily due to mutative number of features.

Algorithm	Runtime [ms]	Environment
MICP_VO	**40**	**1 core @2.6 Ghz (C/C++)**
LIBVISO2	50	1 core @2.6 Ghz (C/C++)
TGVO	60 ∗	1 core @2.5 Ghz (C/C++)
VO3pt	560	1 core @2.0 Ghz (C/C++)
LIBVISO2	510	1 core @2.0 Ghz (C/C++)

TABLE V: Runtime Comparison between different algorithms (∗ Feature Detection and Tracking are not included)

C. *Test on DEVON benchmark*

To demonstrate that our algorithm can also handle non flat, off-road, and rough terrains, we also test it on the very challenging DEVON Island Rover Navigation dataset [33]. This dataset is a collection of images and other sensor data gathered at a Mars/Moon analogue site on Devon Island, suitable for robotics research. This dataset has become extremely popular for benchmarking VO algorithms for planetary rovers. Here, we compare our MICP_VO with P3P RANSAC and LIBVISO2 on all 23 sequences provided by this dataset. For every sequence, we calculate the absolute position error relative to the traveled distance; the results are listed in Table VI. Part of the estimated trajectories are shown in Figure 9.

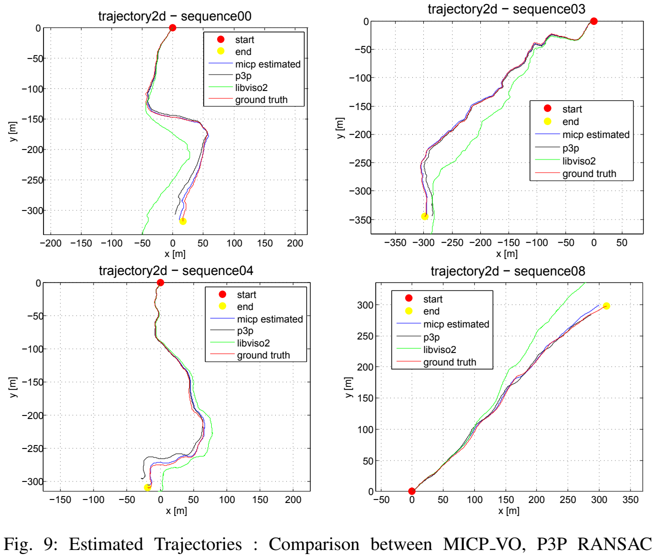


Fig. 9: Estimated Trajectories : Comparison between MICP_VO, P3P RANSAC, LIBVISO2

As observed in Table VI, in most situations (notably, in 15 out of 23 sequences) MICP_VO achieves the best estimation, which confirms our simulation results, i.e., that although the ICP is initialized from the 1-Point algorithm (which is based on a predominantly planar and circular motion) it manages to

[1] `http://www.cvlibs.net/datasets/kitti/eval_odometry.php`

[2] We didn't compare with MFI and D6DVO, because the former is a multi-frame approach while the latter is a dense method

Algorithm	Seq0	Seq1	Seq2	Seq3	Seq4	Seq5	Seq6	Seq7
MICP_VO	**3.558**	12.394	9.865	**2.0768**	**1.484**	5.496	**9.262**	**14.284**
P3P	10.728	**9.012**	12.036	3.810	8.528	**4.393**	17.113	15.464
LIBVISO2	17.072	12.569	**8.047**	7.684	4.434	20.338	22.762	16.765
Algorithm	Seq8	Seq9	Seq10	Seq11	Seq12	Seq13	Seq14	Seq15
MICP_VO	**2.6256**	12.621	**10.234**	**9.416**	24.270	9.988	**20.597**	**7.729**
P3P	12.454	**3.992**	11.344	23.702	30.302	**4.399**	50.627	8.042
LIBVISO2	12.002	15.660	21.752	23.294	38.391	17.793	23.259	14.131
Algorithm	Seq16	Seq17	Seq18	Seq19	Seq20	Seq21	Seq22	**Average**
MICP_VO	11.121	**6.134**	**7.391**	**5.061**	**9.385**	**7.701**	11.075	**9.294**
P3P	**8.311**	17.765	12.615	5.372	9.890	10.464	10.699	13.090
LIBVISO2	19.183	24.030	18.649	6.353	18.388	9.369	**3.290**	16.314

TABLE VI: Position error of experiments run on all twenty three Devon Island sequences. For every sequence, the winner is shown in bold.

relax the motion constraint, thus, achieving accurate motion estimation results in distinct types of environments. More results, such as in presence of dynamic objects (like moving cars) and different sequences, can be seen in our video attachment.

V. CONCLUSION

In this paper, we presented a novel and fast stereo visual odometry algorithm (called MICP) for wheeled vehicles moving in dynamic outdoor environments. We extended the one-point algorithm proposed in [11] to stereo cameras, where the motion prior is not directly used to reject outliers but to provide a good and efficient initialization for an ICP-based 3D registration. We showed that, after the 3D registration, a set of *high quality* inliers is preserved.

Compared to RANSAC-based stereo-VO methods, which are based on random sampling, our approach has the advantage of being fully deterministic. Additionally, it is also more accurate, given that only a small set of high-quality inliers is used to estimate the motion (unlike standard methods, which use all the inliers).

We successfully tested our algorithm on two large image datasets, spanning dozens of kilometers: the first one collected from a car while driving in a urban environment; the second one from a planetary rover navigating on a rough, outdoor terrain. Experiments show that the our algorithm outperforms state-of-the-art approaches in accuracy, runtime, and ease of implementation.

REFERENCES

[1] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR*, 2004, pp. 652–659.

[2] J. P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *IEEE IROS*, 2008, pp. 2531–2538.

[3] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, pp. 169–186, 2007.

[4] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE IROS*, 2008, pp. 3946–3952.

[5] D. Scaramuzza and F. Fraundorfer, "Visual odometry part i: The first 30 years and fundamentals," *IEEE RAM*, pp. 80–92, 2011.

[6] F. Fraundorfer and D. Scaramuzza, "Visual odometry part ii: Matching, robustness, optimization, and applications," *IEEE RAM*, pp. 78–90, 2012.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography," *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395, 1981.

[8] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. on PAMI*, vol. 26, no. 6, pp. 756–770, 2004.

[9] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *IJCV*, vol. 13, no. 3, pp. 331–356, 1994.

[10] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *IEEE ICRA*, 2009, pp. 4293–4299.

[11] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *IJCV*, vol. 95, pp. 74–85, 2011.

[12] M. Zhu, S. Ramalingam, Y. Taguchi, and T. Garaas, "Monocular visual odometry and dense 3d reconstruction for on-road vehicles," in *ECCV*, 2012.

[13] R. M. Haralick, H. Joo, and C. LEE, "Pose estimation from corresponding point data," *IEEE Transactions on Man, Systems and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

[14] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 486–492.

[15] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *IEEE ICRA*, 2012.

[16] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[17] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Auton Robot*, vol. 34, pp. 133–148, 2013.

[18] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," in *IEEE International Conference on Computer Vision Systems*, 2006, pp. 21–27.

[19] M. Tomono, "3d localization based on visual odometry and landmark recognition using image edge points," in *IEEE IROS*, 2010, pp. 5953–5959.

[20] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *IEEE ICCV Workshops*, 2011, pp. 2050–2056.

[21] F. Steinbrüker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense rgb-d images," in *IEEE ICCV Workshops*, 2011, pp. 719–722.

[22] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011, pp. 127–136.

[23] N. Nourani-Vatani, J. Roberts, and M. V. Srinivasan, "Practical visual odometry for car-like vehicles," in *IEEE ICRA*, 2009, pp. 3551–3557.

[24] D. Scaramuzza, A. Censi, and D. K., "Exploiting motion priors in visual odometry for vehicle-mounted cameras with non-holonomic constraints," in *IEEE IROS*, San Francisco, USA, 2011.

[25] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *IEEE ICCV*, 2005, pp. 1508–1515.

[26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[27] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV*, 1994, pp. 151–158.

[28] Y. Jiang, H. Chen, G. Xiong, J. Gong, and Y. Jiang, "Kinematic constraints in visual odometry of intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 1126–113.

[29] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative o(n) solution to the pnp problem," *IJCV*, pp. 1–8, 2007.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012, pp. 3354–3361.

[31] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IV*, Baden-Baden, Germany, June 2011.

[32] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE ICRA*, 2009, pp. 3539–3544.

[33] P. Furgale, P. Carle, J. Enright, and B. T.D., "The devon island rover navigation dataset," in *International Journal of Robotics Research*, 2012.