2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

A Structureless Approach for Visual Odometry

Chih-Chung Chou, Chun-Kai Chang and YoungWoo Seo

Abstract— **A local bundle adjustment is an important procedure to improve the accuracy of a visual odometry solution. However, it is computationally very expensive as it jointly optimize all the poses of cameras and locations of map points. To reduce the computational complexity of a local bundle adjustment, the state-of-the-art algorithms [1],[2],[3] were proposed to manipulate the map point variables, using extra matrix operations, from their linearized optimization solutions. Instead of relying on complex matrix manipulations, this paper proposes a novel way of addressing this complexity issue – we represent a map point as a function of two camera poses, and uses the triangulated location of the map point when needed. Our method is more efficient than ones in the full-SLAM formulation in solving the visual odometry problem in that 1) the complexity of our solution is lower than those of the state-of-the-art methods, 2) no extra matrix operations required to eliminate map points, 3) no need guesses on map points' initial locations. Experiemental results, through simulated experiments and experiments with the KITTI dataset, demonstrated that our results are more accurate than those of a full-SLAM approach with lower runtime complexities.**

I. Introduction

Visual odometry is a process of incrementally estimating the pose (positions and orientations) of a camera by analyzing images over time. Typically, for those methods based on features, it begins with extraction and matching of visual features from a stream of images. Some of the visual features are kept as map points in a 3d space based on their properties, and the locations of the matched features between the images are used to estimate the latest camera pose [4], [5], [6], [7]. Since this approach only processes consecutive images, the estimated camera pose would quickly drift primarily due to the accumulated errors from those local pose estimations. To minimize such drifts, the state-of-the-art algorithms [8], [9], [10] used a local or windowed bundle-adjustment to more accurately solve a visual odometry problem in a "full-SLAM" formulation where the poses of the camera (s) and locations of the map points in the predefined window are jointly estimated. By doing so, a new camera pose is constrained by the map points which could be observed from multiple previous poses. Thus, the local bundle adjustment could considerably improve the long-term accuracy of a visual odometry solution. However, at the same time, the local bundle adjustment increases computational complexity of a visual odometry solution primarily due to a joint optimization of camera poses and map points. For example, suppose that a local bundle-adjustment is used to estimate 10 camera poses and up to 1,000 map points. The number of variables to estimate is 3,060 ($10 \times 6 + 1,000 \times 3$). As the complexity of an optimization is typically $O(N^3)$ to the number of variables [9], this bundle adjustment of 10 different camera poses and 1,000 map points is impractically expensive. This complexity issue is a well-known challenge to a large-scale visual SLAM or odometry tasks, and would eventually limits the accuracy and applicability of a visual odometry solution.

To deal with this computational complexity, the state-of-the-art algorithms reformulate the problem to intentionally relax a joint optimization of poses and map points. In particular, the null-space trick was used to eliminate all the map point variables from the linearized system for computing EKF (extended Kalman filter) update [1] or for executing the Gauss-Newton optimization [3]. By exploiting the fact of the sparseness of the Jacobian matrix, the poses and map points can be independently updated in a bundle adjustment [9]. However, as these approaches are formulated in a full-SLAM problem where they have to exploit some matrix tricks in order to reduce the dimension of the linearized problem, it is required to have an initial guess of the 3D map point locations for the linearization. This also requires extra map point estimators to maintain an initial guess for map points and results in increasing the complexity of implementation as well. Moreover, the marginalization or separation of map point variables requires extra matrix operations that will be a substantial burden in tracking many map points. This is because the computational cost of those extra matrix operations is linearly proportional to the number of map points.

To tackle the issue of high computational complexity in a local bundle adjustment, this paper proposes a novel way of eliminating map point variables in the full-SLAM formulation. Our method uses a function of two camera poses to triangulate a map point, instead of keeping the map point as a part of state vector, and the triangulated location of the map point when the location of the map point is required for estimation. Our approach is "structureless" as the map point locations are not used as a part of the state vector. By representing map points in this way, our method can solve the visual odometry problem in a full-SLAM formulation more efficient than those of the state-of-the-art methods because 1) the complexity of our solution is a way lower than those of the state-of-the-art methods, 2) no extra matrix operations required to eliminate map points, 3) no need guesses on map points' initial locations, and lastly 4) our method is robust to the measurement noises because it is not directly dealing with noisy measurements about map points.

Chih-Chung Chou and Chun-Kai Chang are with and from the Department of Computer Science, National Taiwan University, Taipei, Taiwan, and YoungWoo Seo are with the Independent Robotics Research (IR2), San Francisco, CA, karate362@gmail.com, s921708@gmail.com, youngwoo.blank.seo@gmail.com

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

In what follows, we will survey the related work first, review the visual odometry formulation in the 3d space and a typical, visual odometry solution using non-linear optimization. And then we will detail our structureless approach to the problem of visual odometry. We will then discuss the results of simulated experiments and experiments with the KITTI dataset to verify the usefulness of the proposed method.

II. Related works

Visual odometry primarily concerns about the pose of a camera based on image analysis. A typical approach to solve visual odometry is to estimate the motion of a camera by analyzing consecutive image frames [6] and estimate the latest pose by accumulating the frame-to-frame motions. To reduce the pose drift potentially introduced by the accumulated motion errors, the keyframe-based approach was proposed [8] where 1) selected images are stored and 2) a local bundle adjustment is applied to do a batch-optimization on the key-frame poses and map points in a bounded, local sliding window. Images acquired on the fly are matched to the saved map points to estimate camera motion. To further reduce the pose drift, Mur-Artal and his colleagues [10] did an additional loop closing step using a pose-graph optimization. For these methods utilized the local bundle adjustment [8], [10], the underlying assumption is that the first camera pose in the sliding window is fixed and drops the visual measurements older than that pose. Leutenegger and his colleagues [14] remove the old poses using a marginalization technique which preserves the previously learned information as initial constraints to existing poses. Even for the direct methods [15], where the frame-to-frame motions are estimated without extracting point features, an additional local bundle adjustment was still proven to be effective for enhancing the odometry accuracy [16].

Although the local bundle adjustment processes only the poses and map points observed in a bounded window, it is still computationally expensive as it jointly optimizes poses of camera and locations of features. To reduce time complexity, Lourakis and Argyros [9] proposed the sparse bundle adjustment (SBA) that utilizes the Schur complement to convert the original linearized optimization problem $Ax = b$ into two separated linear solvers by representing the solvers of poses and map points separately. Because of the sparse nature of the Jacobian matrix A in the visual SLAM problem, the SBA solver is much more efficient than the typical joint optimization. Another way of dealing with this computational complexity issue is to apply the null-space trick, which cancels out the map point variables by projecting the linearized system to the null space of the map point Jacobian matrix. This null-space trick can be applied in both the filtering framework [1] and non-linear optimization framework [3]. This series of research work has greatly advanced the progress of the visual SLAM and odometry. However, there are still some drawbacks. Firstly, these approaches still formulate solutions in a full-SLAM setup where an initial estimation of the map points is required. Such an initial estimation requires stereo, RGB-D sensors, or extra map point trackers [17] to estimate the map point's depth. Secondly, those matrix tricks like the Schur complement or the null-space computation of the large Jacobian matrices will introduce extra computational complexity. Third, the null-space trick does not always guarantee an accurate result because of the linearization errors in the Jacobain matrices [2].

This paper proposes a remedy that accelerates and simplifies the bundle adjustment process of a full-SLAM formulation. The proposed method is based on key-frames with a local bundle adjustment, but the local bundle adjustment is done differently – a map point variable will be substituted by a point triangulation from a pair of camera poses and measurements from monocular cameras. By doing so, the map point variables are no longer being kept at the original, non-linear cost function as the state vector does not include map points as state variables. This also eliminates a need of extra map, point trackers or any matrix tricks to lightly solve the linearized, optimization problem.

III. Algorithm

The goal of this work is to develop a way of reducing the computational complexity of a local bundle adjustment for a visual odometry solution. The pipeline of a typical visual odometry solution, based on a feature tracking, begins with extracting visual features, matching the extracted features to the previously surveyed features, estimating the current camera poses based on the matched results, and lastly executing a local bundle adjustment over a sliding window to jointly optimize camera poses and map points. For such a framework, the local bundle adjustment is clearly a bottleneck for any practical solution [9] primarily because the runtime complexity of a local bundle-adjustment is dependent upon the dimension of the state to estimate. To tackle this high complexity operation, we propose a novel visual odometry solution where we use two camera poses[1] to represent the position of a map point, instead of keeping it as a state variable. This section details our structureless approach to solve the visual odometry problem. In particular, we will first review a full-SLAM formulation for solving the visual odometry in the nonlinear least square using optimization algorithms. And then we will explain how to reformulate the objective function in a structureless manner.

A. A Formulation of Visual Odometry Problem

Given a stochastic observation z_i, the SLAM problem including visual odometry can be formulated to find the maximum a posterior (MAP) of a unknown state x [11],[13]:

$$x^* = \underset{x}{\operatorname{argmax}} \; p(x|x_0) \prod_i p(z_i|x) \tag{1}$$

where x is a vector of the state variables to estimate, x_0 is the initial guess about the state, and z_i is the ith observation.

[1] We could use more than two camera poses to represent a map point, but for this paper, we will use just two poses because the solution with more than two poses needs more complex error function and the Jacobian for a nonliear least square problem.

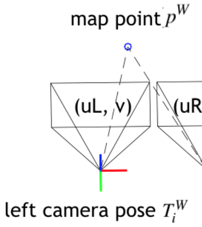


Fig. 1. Our measurement model is a stereo, perspective transformation of a 3d map point onto the left and right image planes, (u_L, u_R, v).

And $p(z_i|x)$ is the likelihood of a measurement z_i given the state x and $p(x|x_0)$ is the prior distribution of x given the initial guess x_0. Assuming that the underlying distributions are Gaussian: $p(z_i|x) \sim N(h_i(x), \Sigma_{z,i})$ and $p(x|x_0) \sim N(x_0, \Sigma_{x_0})$, one can compute a solution of the maximum a posterior in the least square sense:

$$\begin{aligned}
&\underset{x}{\operatorname{argmax}}\ p(x|x_0)\prod_i p(z_i|x) = \\
&\underset{x}{\operatorname{argmax}}\ exp(-\|x-x_0\|^2_{\Sigma_{x_0}})\prod_i exp(-\|z_i-h_i(x)\|^2_{\Sigma_{z,i}}) = \\
&\underset{x}{\operatorname{argmin}}\ \|x-x_0\|^2_{\Sigma_{x_0}} + \sum_i \|z_i-h_i(x)\|^2_{\Sigma_{z,i}}
\end{aligned} \tag{2}$$

where $\|x\|^2_\Sigma = x^T\Sigma^{-1}x$. For a typical state estimation problem including visual odometry, the state vector x is a collection of variables about the camera poses and map point locations in a predefined reference frame. For a visual SLAM, the measurement model $h_i(x)$ is typically defined as a projection of a 3d map point onto a stereo/monocular image coordinate. For the formulation of our approach, we define the 3d points and its coordinates as follows:

p^A : A 3d point p in coordinate A.
T^B_A : A transformation of a point from coordinate A to B.
R^B_A : The rotation matrix of T^B_A.
t^B_A : The translation vector of T^B_A.
ω^B_A : The axis-angle representation of R^B_A.

With these definitions, we denote a state for the full-SLAM framework as $x = (T^W_1, T^W_2, ..., T^W_N, p^W_1, p^W_2, ..., p^W_M)$, where $T^W_i = (R^B_A, t^B_A)$ is a transformation of the ith camera pose in a local coordinate to a world coordinate W and p^W_j is the jth map point of a world coordinate. In this work, we use a stereo projection function as a measurement model $h_i(x)$. Figure 1 illustrates our measurement model where a 3d point (or map point) is, through the perspective transformation, projected onto two camera coordinates. To be more specific, given the stereo camera intrinsic parameters (f_x, f_y, c_x, c_y), the stereo camera baseline b, and a left camera pose T^W_i, we define our measurement model, $h_i(x)$, as a stereo projection function $StereoProj(T^W_i, p^W)$ of a 3d point p^W_j:

$$\begin{aligned}
\begin{bmatrix} x^i_j & y^i_j & z^i_j \end{bmatrix}^T &= R^{W^T}_i (p^W_j - t^W_i) \\
u_L &= f_x \frac{x^i_j}{z^i_j} + c_x \\
u_R &= f_x \frac{x^i_j - b}{z^i_j} + c_x \\
v &= f_y \frac{y^i_j}{z^i_j} + c_y
\end{aligned} \tag{3}$$

where the resulting image coordinates, $[u_L, u_R, v]^T = z_{i,j}$ is used as a measurement from stereo. With this definition of a measurement model, we can roll out Equation 2 as it iterates all the observations over all the camera poses to map point pairs:

$$\underset{x}{\operatorname{argmin}}\ \|x-x_0\|^2_{\Sigma_{x_0}} + \sum_{i,j}\|z_{i,j}-h_{i,j}(x)\|^2_{\Sigma_{z,i,j}} \tag{4}$$

One can use any optimization algorithms like the Levenberg Marquardt [18] to iteratively solve this nonlinear least square problem. To derive an iterative optimization solution of the MAP formulation in Equation 2, we first rewrite the likelihood function as a quadratic residual function $r_i(x)$ in a matrix form:

$$\underset{x}{\operatorname{argmin}} \sum_i r_i(z_i, h_i(x))^T \Sigma_i^{-1} r_i(z_i, h_i(x)) \tag{5}$$

where the $r_i(z_i, h_i(x))$ is a differentiable residual function about the difference between the measurement z_i and a predicted measurement by the measurement model $h_i(x)$, and Σ_i is the covariance of the ith residual. A typical way of solving such a nonlinear least square problem is to linearize it first and then iteratively find an optimal value. To linearly approximate Equation 5, we only use the first two terms of the Tylor expansion:

$$r_i(z_i, h_i(x^*)) \simeq r_i(z_i, h_i(\tilde{x})) + J_i\delta x, J_i = \frac{\partial r_i}{\partial x}_{|x=\tilde{x}} \tag{6}$$

By rewriting Equation 5 with the linearization result, we will get a quadratic function to optimize:

$$\begin{aligned}
&(r_i(\tilde{x}) + J_i\delta x)^T \Sigma_i^{-1}(r_i(\tilde{x}) + J_i\delta x) = \\
&r_i(\tilde{x})^T\Sigma_i^{-1}r_i(\tilde{x}) + 2r_i(\tilde{x})^T\Sigma_i^{-1}J_i\delta x + \delta x^T J_i^T \Sigma_i^{-1} J_i \delta x
\end{aligned} \tag{7}$$

As the minimal value of a quadratic function is obtained when its gradient (i.e., Jacobian) is set to zero, we will have the following inhomogeneous linear system by taking the first-order derivative of the above equation to be zero:

$$A\delta x = b, A = \sum_i J_i^T\Sigma_i^{-1}J_i, b = -\sum_i J_i^T\Sigma_i^{-1}r_i \tag{8}$$

Now we have a way of computing the optimal value x^* using its delta state δx. In summary, we seek for the optimal value of x by iteratively computing $x^* = \tilde{x} + \delta x$ where $\tilde{x}$ is the current estimate and δx is an incremental update. This update is repeated until the summed square residual error is smaller than a predefined threshold or the maximum iteration number is reached.

B. A Structureless Approach to Visual Odometry

Solving the linear system $A\delta x = b$ in Equation 8 is an essential step in a local bundle adjustment, but it is very time-consuming and computationally expensive, particularly when the state x is in a high dimension. The state vector in most of the visual odometry solutions using a local bundle adjustment is in high dimension. This motivates us to find a more efficient way of solving this and leads to a structureless approach. The underlying idea of our method is to represent locations of map points using two camera poses, instead of keeping them as state variables. By doing so, we can save the space as much as | number of variables about map points × 3 | for the optimization. Specifically, given a stereo projection of a 3d point as a measurement by Equation 3, instead of keeping the location of a 3d point, p^W, in the state vector, we use two camera poses, T_1^W and T_2^W, observed that 3d point, and the images coordinates of the 3d point's projection on two images, $(u_{L,1}, v_1)$ and $(u_{L,2}, v_2)$, to replace the map point variable by a triangulated point, $p^{\tilde{W}}$:

$$\begin{aligned} p_1 &= \begin{bmatrix} \frac{u_{L,1}-c_x}{f_x} & \frac{v_1-c_y}{f_y} & 1 \end{bmatrix}^T, \\ p_2 &= \begin{bmatrix} \frac{u_{L,2}-c_x}{f_x} & \frac{v_2-c_y}{f_y} & 1 \end{bmatrix}^T, \\ p^{\tilde{W}} &= \tilde{\lambda_1}(T_2^1, p_1, p_2) R_1^W p_1 + t_1^W \end{aligned} \tag{9}$$

where p_1 and p_2 are two measurements in the normalized coordinate, and the $\tilde{\lambda_1}$ is the depth of the point to be triangulated, which is the solution of following least square problem:

$$(\tilde{\lambda_1}, \tilde{\lambda_2}) = \underset{\lambda_1, \lambda_2}{\operatorname{argmin}} \; \left\| \lambda_1 p_1 - (\lambda_2 R_2^1 p_2 + t_2^1) \right\|^2 \tag{10}$$

The least square solution of Equation 10 is:

$$\begin{bmatrix} \tilde{\lambda_1} & \tilde{\lambda_2} \end{bmatrix}^T = (A^T A)^{-1} A^T t_2^1, A = \begin{bmatrix} p_1 & -R_2^1 p_2 \end{bmatrix} \tag{11}$$

Note that it is difficult to deal with the matrix inverse $(A^T A)^{-1}$ when to compute the analytic derivative of the residual function. Luckily, $A^T A$ in this study is 2-dimensional, and a 2-by-2 matrix inverse can be easily and analytically computed. We can write a close-form solution of λ_1 after rewriting Equation 10 by 2-dimensional matrix inverse:

$$\begin{aligned} \tilde{\lambda_1}(T_2^1, p_1, p_2) &= \frac{g(T_2^1, p_1, p_2)}{f(T_2^1, p_1, p_2)} \\ g(T_2^1, p_1, p_2) &= \|p_2\| \, p_1^T t_2^1 - (p_1^T R_2^1 p_2)(p_2^T {R_2^1}^T t_2^1) \\ f(T_2^1, p_1, p_2) &= \|p_1\| \, \|p_2\| - (p_1^T R_2^1 p_2)^2 \end{aligned} \tag{12}$$

Putting these equations together, Equations 10 and 11, we redefine the measurement model in equation 3:

$$\begin{aligned} \begin{bmatrix} u_L & u_R & v \end{bmatrix}^T &= StereoProj(T_i^W, p_j^{\tilde{W}}) \\ p_j^{\tilde{W}} &= \tilde{\lambda_1}(T_2^1, p_{j,1}, p_{j,2}) R_{j,1}^W p_{j,1} + t_{j,1}^W \\ T_2^1 &= {T_{j,1}^W}^{-1} T_{j,2}^W \end{aligned} \tag{13}$$

Equation 13 is about a new stereo measurement model based on our structureless approach. This measurement model is in the exact same formulation as Equation 3 except the way of processing the map point variable p_j^W. The p_j^W is now replaced with the triangulated point $p_j^{\tilde{W}}$ that is a function of $T_{j,1}^W, T_{j,2}^W, p_{j,1}, p_{j,2}$. By doing so, we can remove the variables about map points from the state vector. Note that the $T_{j,1}^W, T_{j,2}^W$ are chosen from the poses to optimize $T_i^W, i = 1, 2, ..., N$ and will be updated during the optimization process. Even though this measurement model is defined over stereo camera, one can easily change it with one from monocular camera by removing one of the measurements, say u_R. Comparing this with the conventional full-SLAM formulation, instead of maintaining map point as state variables, all we need to do is just to record two poses for each map point p_j^W. Thus when one gets a new monocular or stereo measurement from any pose T_i^W for the map point p_j^W, the residual can be used to directly update $T_{j,1}^W, T_{j,2}^W, T_i^W$, without having an extra variable for p_j^W.

Now we need to explain how two poses $T_{j,1}^W$ and $T_{j,2}^W$ are chosen. We use the first and last poses as two camera pose for triangulation. We choose these two poses to minimize the triangulation error: The longer the baseline is, the more accurate the triangulation computation is. In addition, to avoid a numerically, unstable triangulation result, the relative motion between two poses should not be parallel to the direction between the camera and the map point. To measure the numerical stability, we use a heuristic of checking how much the triangulated depth is changed by one pixel. If a pair of two poses is numerical unstable, we do not use it for the optimization.

In comparison with the existing algorithms marginalizing the map points using null space trick [1], [2], [3] or Schur complement [9], the proposed method does not require an initial guess of the 3d point p^W because our method does not consider the variables of map points as parts of the formulation from the first place. Note that such a simplication by not including map-points as a part of the state vector does not sacrifice the accuracy of our method. We will later discuss this in details at the Experiments.

C. Optimization on SE(3) manifolds

From the practical perspective, there is one fact we need to clarify. The iterative solution $x^* = \tilde{x} + \delta x$ we derived earlier for the non-linear optimization would not work smoothly for visual SLAM/VO problems as it is. This is because the state x about camera poses belonging to the $SE(3)$ contains a non-Euclidean part – the rotation matrix that belongs to the $SO(3)$. As the result, the output of an addition operation in the iterative solution might not be in the $SE(3)$. Thus, for the Lie group manifolds like $SO(3)$ elements, it is necessary define a set of special operations to replace the addition and subtraction. For example, one can convert a $SO(3)$ manifold to and from an Euclidean vector using exponential and logarithmic map operations [19]:

$$\begin{aligned} &\omega \in \Re^3, R \in SO(3) \\ &R = exp(\omega), \omega = log(R) \end{aligned} \tag{14}$$

To change the representation of the state estimate x and the increment δx, we also need to define the “retract” and “local” functions to deal with element increment and decrement to replace vector addition and subtraction:

$$\begin{aligned} x_1 + x_2 &\Longrightarrow x_1 \oplus x_2 \\ -x_1 + x_2 &\Longrightarrow x_1 \ominus x_2 \end{aligned} \tag{15}$$

By the exponential map operation, the retract and local operations will satisfy the following properties:

$$\begin{aligned} exp(x_1 \oplus x_2) &= exp(x_1)exp(x_2) \\ exp(x_1 \ominus x_2) &= exp(-x_1)exp(x_2) \end{aligned} \tag{16}$$

Given this, one can define the Jacobian and the first-order Taylor approximation of a function of the Lie group manifolds as:

$$\begin{aligned} J &= \frac{\partial f(x \oplus \delta x)}{\partial \delta x}_{|\delta x \to 0} \\ f(x \oplus \delta x) &\simeq f(x) + J\delta x \end{aligned} \tag{17}$$

To replace the equation 6, we define a generalized solution for nonlinear least square in an iterative form:

$$\begin{aligned} &\delta x = \underset{\delta x}{\operatorname{argmin}} \sum_i r_i(y_i, \tilde{x} \oplus \delta x)^T \Sigma_i^{-1} r_i(y_i, \tilde{x} \oplus \delta x) \\ &r_i(y_i, \tilde{x} \oplus \delta x) \simeq r_i(y_i, \tilde{x}) + J_i \delta x, J_i = \frac{\partial r_i(x \oplus \delta x)}{\partial \delta x}_{|x = \tilde{x}} \\ &x^* = \tilde{x} \oplus \delta x \end{aligned} \tag{18}$$

For $SE(3)$, we can define its special exp and log operations [20] and deal with them in the same way, but it is not very convenient because, to satisfy the equation 16, the vectorized $SE(3)$ has to be a “twist” representation. This is different from the Euclidean translation where we used to compute the structureless residual, and will make the Jacobian derivations unnecessarily complex. Thus, in this paper, we define the vectorized $SE(3)$ in the following way:

$$\begin{aligned} &x = (\omega, t) \in \Re^6, T = (R, t) \in SE(3) \\ &T = (exp(\omega), t), x = (log(R), t) \end{aligned} \tag{19}$$

Since the vectorized $SE(3)$ is directly represented as a pair of the axis-angle for rotation and the Euclidean for translation, its retract and local functions are just a “stacked” version of $SO(3)$ retract/local and Euclidean addition/subtraction:

$$\begin{aligned} x_1 \oplus x_2 &= (\omega_1 \oplus \omega_2, t_1 + t_2) \\ x_1 \ominus x_2 &= (\omega_1 \ominus \omega_2, -t_1 + t_2) \end{aligned} \tag{20}$$

With the above definitions, the Jacobian of the estimated point depth defined in equation 12, which is an essential part of our structureless residual function, can be derived with respect to the Euclidean translation. This is way more straightforward than using $SE(3)$ exp and log operations.

IV. Experiments

To validate the usefulness of the proposed algorithm, we conducted two kind of experiments: simulated experiments and experiments with a real-world data, the KITTI data. For a simulated experiment, as we can control the level of challenge in the noises and ground truth, Section IV-A is prepared to validate the underlying idea and verify the expected results – the runtime complexity is more optimal than that of a full-SLAM approach while the accuracy is better or at least same as that of a full-SLAM approach. In Section IV-B, we evaluate, using a real-world data – KITTI data, the performance of our algorithm and compare it with that of a full-SLAM approach.

A. Simulated Experiments

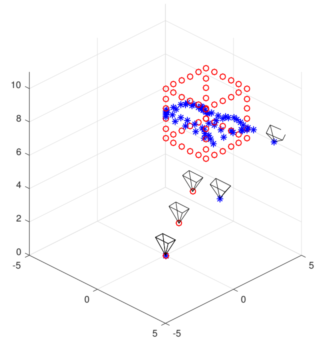
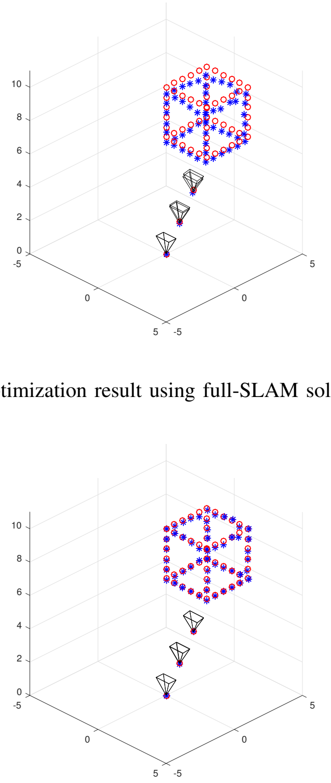


Fig. 2. A setup for simulated experiments about camera poses and landmarks. The red circles represent the ground truth of map-points' locations and the blue stars depict the noisy initial guess of the camera poses and map points. The center of the ground truth, camera poses are also depicted by red circles.

In this section, we evaluate the performance of the proposed structureless visual odometry algorithm using a simulated setup, and compare its performance with that of a full-SLAM solver. Figure 2 shows a simulated setup where landmarks in red circles are observed from three known camera poses (i.e., ground truth). For each of the landmark-pose pairs, we generate a stereo measurement (u_L, u_R, v) and added random noises to each dimension of the measurement where the noise is uniformly distributed from -3 to 3 pixels. Then, camera poses with noisy offsets are generated as the initial guess for the full-SLAM solver. The initial guess about map point positions are triangulated by using the initial poses and measurements. Then, both of the camera poses and map points are estimated by two algorithms: our “structureless solver” and “full-SLAM solver.” The proposed structureless method does not optimize the map point positions, and thus the map points are reconstructed later using an additional, map-only bundle-adjustment (BA) solver where the camera poses are treated as constants. We conduct two simulated experiments. At the first experiment, we let the algorithms use all of the available stereo measurements whereas, at the second one, we only let algorithms use one of a stereo measurement, u_L, and hold the remaining one, u_R. The first experiment is designed to see if our algorithm can achieve the same accuracy as the full-SLAM solver does, with a sufficient amount of measurements. The second experiment is prepared to see how robust our algorithm is in handling a problem of scale drifting mainly caused by insufficient and noisy measurements.

Table I and Table II show the results of this comparison. We computed distances between algorithms' outputs and ground truths positions using the root-mean square to measure the accuracy of algorithms, and the runtime. As one can see, our method is more accurate than that of a full-SLAM approach as the errors of our method is smaller than those of a full-SLAM approach. For the comparison of the runtime of Full-SLAM solver and our structureless solver, the full-SLAM solver needs more than 5 seconds to complete its optimization whereas the proposed approach needs less than 3 seconds to reconstruct the camera poses and the map points. At the last column of the table I, there are two numbers about "computation time" of our approach: the first number, e.g., 0.73, is for estimating camera poses and the second number, e.g., 1.75, is for map point reconstruction. Notice that our method only needs less than one second to optimize the camera poses.



(a) Optimization result using full-SLAM solver.

(b) Optimization result using structureless solver and then map-only solver.

Fig. 3. Simulation result using one of the stereo measurements.

For the second simulation where only one of the stereo measurement is used, the structureless approach was not only faster than the full-SLAM solver, but also shows better accuracy on estimating the camera poses and reconstruct the map points – the errors of our method is smaller than those of a full-SLAM approach. We believe that the reason a full-SLAM approach did not perform well is, when there is not sufficient amount of measurements available, estimating both camera poses and map points becomes a loosely constrained problem and a solution based on a full-SLAM formulation could overfit to the noisy measurements. On the contrary, our structureless method is quite robust to such noisy measurements because it does not keep the map point as state variables from the first place. Our method is not just better in numerical performance metrics, but also offering the following benefits: 1) No need to estimate positions of landmarks (e.g., in this simulation, it reduces the state variables from 186 (3 poses and 56 map points) to 18 (3 poses), 2) only one measurement (a part of a stereo measurement) needed to make scale converge, and 3) required number of measurements to converge is smaller than that of a full-BA solver. Figure 3 particularly emphasizes the benefit of 3) where there is only a few stereo measurements available. For this example, the result by our structureless approach converged well whereas that of the full-BA solver had a large drift for the map points.

B. Experiments using KITTI data

The previous section, with simulated experiments, proved the usefulness of the proposed algorithm. To be truly useful for any practical applications, the proposed algorithm should show an experimental result, experiments using real-world data, similar to what we observed from simulated experiments. Thus, in this section, we use a real-world data, the KITTI dataset, to evaluate the performance of the proposed structureless algorithm and compare its performance with that of the full-SLAM solver. For this experiment, we used an open source SLAM package, ORB-SLAM2 [21] that is a visual SLAM package executing the following steps: 1) extract visual features from current image, 2) match the features from the current image to the previously tracked map points which are 3d points with visual descriptors, 3) compute the current camera pose using the matched results, 4) run a local bundle adjustment which jointly optimizes the camera poses and map points over a sliding window, and 5) detect and close the loop if a place is re-visited. The proposed algorithm particularly focuses to improve the step 4), the local bundle-adjustment step. Thus, in this experiment, we depreciate the step 5) and use the steps 1), 2) and 3) from the ORB-SLAM2. For the step 4), we compare the proposed structureless algorithm with the solver in the ORB-SLAM2, which is a full-SLAM solver implemented using g2o [22], using the same camera trajectory initial guess, visual feature matching results and the monocular/stereo measurements.

Figure 4 and 5 show the results of this experiment where we computed the average of heading and position errors of the optimized camera poses in each local bundle adjustment. While computing these errors, we did not use the poses in a global coordinate, but use the current poses relative

TABLE I

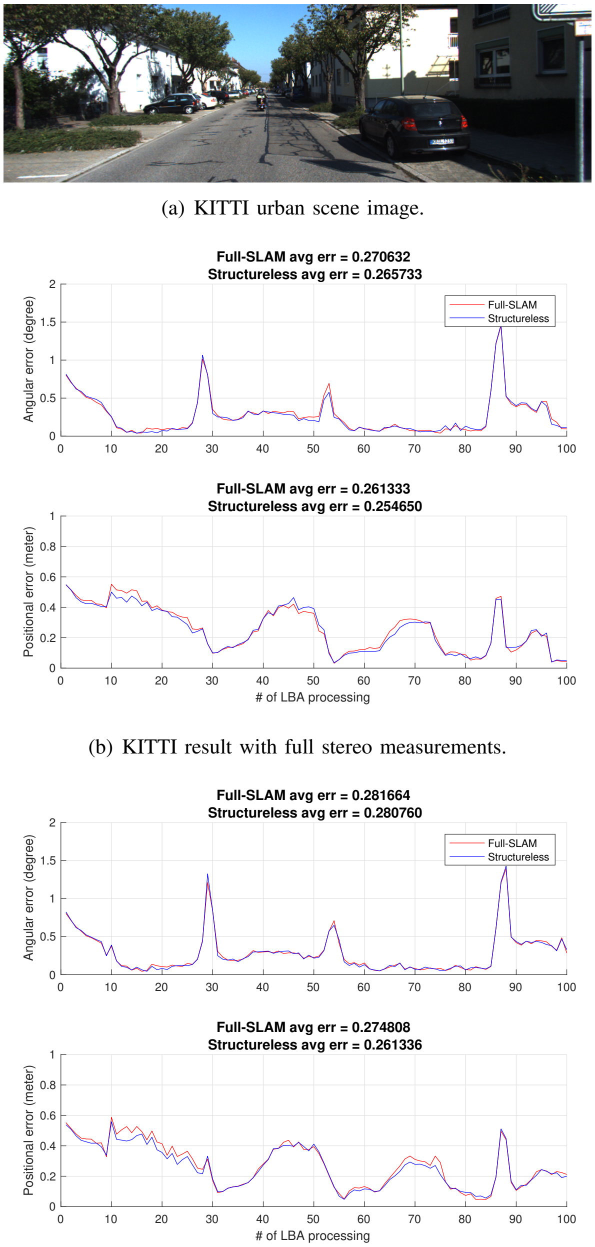
SIMULATION RESULTS USING FULL STEREO MEASUREMENTS.

	Orientation RMSE (rad)	Translation RMSE (m)	Landmark RMSE (m)	Computation time (sec)
Full-SLAM	0.0050	0.0154	0.0492	5.16
Structureless	0.0037	0.0093	0.0456	2.48 (pose: 0.73, map points: 1.75)

TABLE II

SIMULATION RESULTS USING ONE OF THE STEREO MEASUREMENTS.

	Orientation RMSE (rad)	Translation RMSE (m)	Landmark RMSE (m)	Computation time (sec)
Full-SLAM	0.0057	0.0525	0.2293	5.88
Structureless	0.0045	0.0134	0.0631	2.86 (pose: 1.05, map points: 1.81)

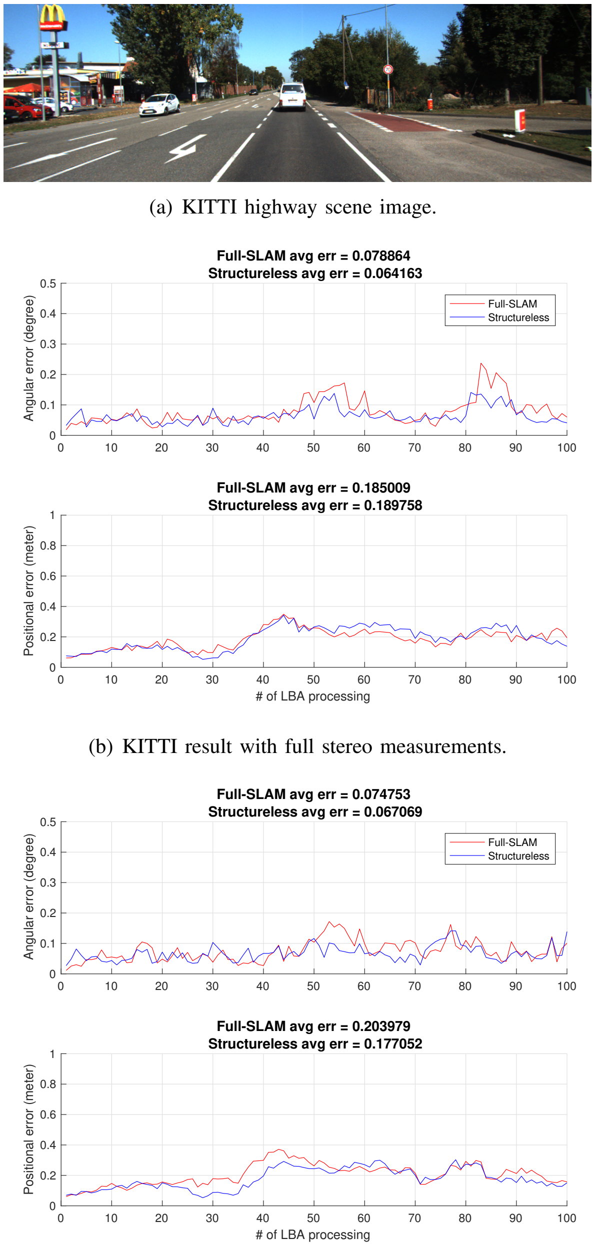


(a) KITTI urban scene image.

(b) KITTI result with full stereo measurements.

(c) KITTI result with few stereo measurements.

Fig. 4. Experimental result on the KITTI urban scene.



(a) KITTI highway scene image.

(b) KITTI result with full stereo measurements.

(c) KITTI result with few stereo measurements.

Fig. 5. Experimental result on the KITTI highway scene.

to the latest camera poses. This is because the estimated poses in a global coordinate will drift due to the accumulated visual odometry error unless the estimated pose is corrected with other measurements like ones from a differential GPS. Thus evaluating the error of relative poses enables us to eliminate the error offsets caused by the intrinsic drift, and make it straightforward to analyze the performances of these local BAs. Like in the simulated experiments, we conducted the experiments with the KITTI data in two ways: at one setup, all of the available stereo measurements (and all of the map points for the full-SLAM solver) are used to estimate the camera poses and at another setup, we only let the algorithms use at most one stereo measurement for each tracked map point. The results were similar to those of the simulated experiments in that our method is more robust than the full-SLAM one in handling the case where fewer stereo measurements are available. Furthermore, such a merit of demonstrating better performance when only fewer measurements are available, is indeed beneficial to the scenario that the car is driving straight in a high speed, where each map point could be only tracked over a very short period of time because the appearance of images changes drastically and quickly.

V. Conclusion

In this paper, we proposed a novel way of addressing the computational complexity of a local bundle adjustment in visual odometry solutions. Instead of keeping map points as state variables, the proposed method represents a map point as a function of two camera poses and uses the triangulated location of the map point when required. Experimental results showed that our method is not only accurate, but also efficient than a full-SLAM method by offering three practical benefits: 1) no need to estimate positions of landmarks, 2) monocular measurements are enough to make it scale converged, and 3) required number of measurements to converge is smaller than that of a full-SLAM approach. Moreover, our method is robust to the measurement noises because it is not directly processing noisy measurements of map points.

As future work, we will apply the proposed algorithm to the applications where computational resources are limited like a task of visual-inertial odometry for mobile-platforms such as consumer-grade drones.

References

[1] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Proceedings of the IEEE international conference on Robotics and automation*, 2007, pp. 3565–3572.

[2] M. Li and A. I. Mourikis, “Improving the accuracy of ekf-based visual-inertial odometry,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012, pp. 828–835.

[3] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,” in *Proceedings of Robotics Science and Systems*, 2015.

[4] D. Nister, O. Naroditsky, and J. R. Bergen, “Visual odometry,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 652–659.

[5] ——, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[6] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *Intelligent Vehicles Symposium (IV)*, 2010.

[7] D. Scaramuzza and F. Fraundorfer, “Visual odometry: Part i - the first 30 years and fundamentals,” *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, 2011.

[8] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[9] M. I. Lourakis and A. A. Argyros, “Sba: A software package for generic sparse bundle adjustment,” *ACM Transactions on Mathematical Software*, vol. 36, no. 1, p. 2, 2009.

[10] R. Mur-Artal, J. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[12] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 834–849.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 15–22.

[14] J. Civera, A. J. Davison, and J. M. Montiel, “Inverse depth parametrization for monocular slam,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[15] D. Fox, S. Thrun, and W. Burgard, *Probabilistic Robotics*. MIT Press, 2005.

[16] F. Dellaert and M. Kaess, “Square root sam: Simultaneous localization and mapping via square root information smoothing,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[17] D. W. Marquardt, “An algorithm for least square estimation of nonlinear parameters,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[18] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[19] Y. Wang and G. S. Chirikjian, “Nonparametric second-order theory of error propagation on motion groups,” *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[20] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras,” *arXiv preprint arXiv:1610.06475*, 2016.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g 2 o: A general framework for graph optimization,” in *Proceedings of the IEEE International Conference onRobotics and Automation*, 2011, pp. 3607–3613.