2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Semi-Dense Visual Odometry for RGB-D Cameras Using Approximate Nearest Neighbour Fields

Yi Zhou, Laurent Kneip and Hongdong Li

Abstract— **This paper presents a robust and efficient semi-dense visual odometry solution for RGB-D cameras. The core of our method is a 2D-3D ICP pipeline which estimates the pose of the sensor by registering the projection of a 3D semi-dense map of a reference frame with the 2D semi-dense region extracted in the current frame. The processing is speeded up by efficiently implemented approximate nearest neighbour fields under the Euclidean distance criterion, which permits the use of compact Gauss-Newton updates in the optimization. The registration is formulated as a maximum a posterior problem to deal with outliers and sensor noise, and the equivalent weighted least squares problem is consequently solved by iteratively reweighted least squares method. A variety of robust weight functions are tested and the optimum is determined based on the probabilistic characteristics of the sensor model. Extensive evaluation on publicly available RGB-D datasets shows that the proposed method predominantly outperforms existing state-of-the-art methods.**

I. Introduction

Image-based estimation of camera motion, known as visual odometry (VO), plays a very important role in many robotic applications such as control and navigation of unmanned mobile robots, especially when no external navigation reference signal is available. Although a number of successful works have been presented over the past decade in this relatively mature field, the conclusion is that no method is all-powerful and working in any scenario. For example, salient feature based sparse methods such as [1], [2] do not work well when there is insufficient texture in the image for generating feature points. By taking advantage of all intensity information, direct methods like [3], [4], [5], [6] achieve better performance in textureless environments as long as the assumption of photometric consistency is sufficiently met. Engel *et al*. [7], [8] further improve the efficiency of [4] by using photometric information in a semi-dense region only. Other systems such as [9], [10], [11], [12] track the camera using an iterative closest point (ICP) algorithm over the depth information only. This, however, requires the presence of sufficient 3D structure and fails, for instance, in the situation of a planar scene. Furthermore, the ICP algorithm is known to be computationally expensive, and usually depends on GPU resources for real-time performance.

In this work we combine the merits of semi-dense processing and ICP based tracking. Compared to sparse methods, the proposed semi-dense method exploits the common structure of man-made environments, and thus can handle relatively textureless situations. It ensures computational efficiency while being able to work in the degenerate case for ICP trackers (i.e. a single plane). Instead of applying a direct method which is sensitive to illumination changes, an accurate ICP inspired geometric framework is proposed. More precisely, the estimation of the camera pose at the current frame with respect to the reference frame is cast as a 2D-3D registration problem. The 3D part is given by a 3D semi-dense map defined in the reference frame, and the 2D part is given by the semi-dense region extracted in the current frame. Similar to the classical ICP framework which aims at aligning surfaces in 3D, our method aims at non-parametric curve-to-curve registration. To improve robustness against sensor noise and outliers, we apply a probabilistic model in the spirit of [4]. The resulting maximum a posteriori (MAP) problem is equivalent to a weighted least squares problem which can be solved using the iteratively re-weighted least squares (IRLS) method.

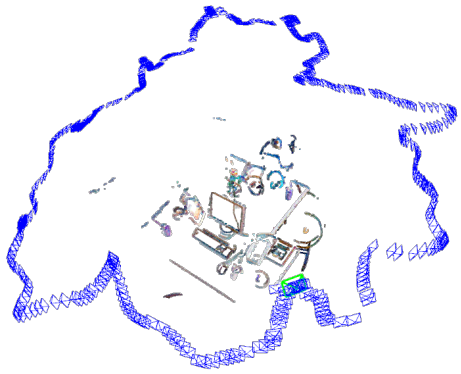


Fig. 1. An illustration of the proposed semi-dense visual odometry. All reference frames are drawn in blue and the current frame in green. The colorful structures in the centre are the reconstruction of the semi-dense region observed by some of the reference frames.

Our main contribution is four-fold:

- Introduction of the idea of approximate nearest neighbour fields, which permits the use of compact Gauss-Newton updates in the registration.
- Exploration of the optimal robust weight function for the probabilistically formulated, 2D-3D semi-dense ICP based motion estimation.
- A real-time implementation running at 25 Hz on a laptop using only CPU resources.
- Extensive evaluation under varying experimental conditions and varying algorithm setups (e.g., different meth-

All the authors are with the Research School of Engineering, the Australian National University. {yi.zhou, laurent.kneip, hongdong.li}@anu.edu.au. The research leading to these results is supported by the Australian Centre for Robotic Vision. The work is furthermore supported by ARC grants DE150101365. Yi Zhou acknowledges the financial support from the China Scholarship Council for his PhD Scholarship No.201406020098.

ods for extracting the semi-dense region and reweighting residuals) and performance comparison against state-of-the-art solutions on publicly available datasets.

The paper is organized as follows. More related work is discussed in Section II. Section III provides an in-depth review of geometric semi-dense 2D-3D registration. Section IV then presents the core of our new approach, including the idea of approximate nearest neighbour fields and keys to robust motion estimation despite occlusion, noise and outliers. An overview of our framework is given in Section V, which is followed by extensive evaluation including the exploration of the best configuration and the comparison against state-of-the-art methods in Section VI.

II. Related Work

Line, curve based and semi-dense methods: Lines are alternative features to points and have been widely used in many VO and SLAM frameworks such as [13], [14]. One reason is that lines are abundant in man-made structures and environments, and do not depend on sufficient texture. Another reason is that line features are easily parametrized and included into a bundle adjustment (BA) framework [13], [15] for the purpose of global optimization. However, straight lines are not a general feature because object contours can be arbitrary curves in 3D space. Therefore, Nurutdinova *et al*. present a method which uses parametric curves as landmarks for motion estimation and BA [16]. Furthermore, Engel *et al*. apply direct photometric registration to semi-dense regions defined as the neighbourhood of all boundaries, edges and contours [7], [8]. The most relevant work to ours is [17], which presents a direct edge alignment approach for 6-DOF tracking. They address the problem of non-differentiability of their Distance Transform (DT) based cost function by using a sub-gradient method. Conversely, we improve the differentiability of the cost function intrinsically and achieve more accurate results at a comparable computational cost.

ICP: The Iterative Closest Point (ICP) algorithm is a fundamental component of our method and it has been used exhaustively in 3D-3D registration problems. Typical issues when applying those methods are missing data, noise, outliers, and local minima in the registration process. Yang investigates globally optimal solutions to the point set registration problem [18]. However, this method is not efficient enough for real-time applications, where the frame-to-frame displacement remains small enough anyway for a successful application of local methods. The most related work is [19] which applies ICP and distance transforms to semi-dense 3D-2D registration. Chebyshev/Chamfer distance field is chosen as an approximation of the Euclidean distance field to achieve real-time performance. Without discussing how to choose a reference frame, [19] stops at solving an absolute pose estimation problem rather than providing a full VO system.

Photometric and hybrid registration methods: ICP and its close derivatives [11], [12], [9], [10] still represent the methods of choice for real-time LIDAR tracking though sometimes expensive computational resources like GPU are necessary. However, the advent of RGB-D cameras has led to a new generation of 2D-3D registration algorithms that exercise a hybrid use of both depth and RGB information. For instance, Steinbrücker uses the depth information along with the optimized relative transformation to warp one RGB-D image to the next [5], thus permitting direct and dense photometric error minimization. The similar idea is applied in [4], [7], [8].

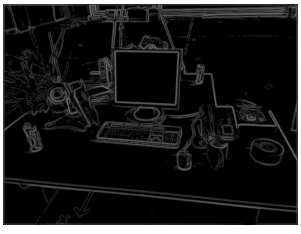
Robust M-estimators and IRLS: When system noise and outliers are taken into account, M-estimators are a popular choice for re-weighting the naïve least squares problem. The earliest tutorial about using different M-estimators in the application of conic fitting was given in [20]. Recently, Aftab investigates the full range of robust M-estimators that are amenable to IRLS [21]. In consideration of the great success of applying IRLS and M-estimators in motion estimation works such as [3], [4], [7], we utilize it in our work as well.

III. Review of Geometric Semi-dense 2D-3D Registration

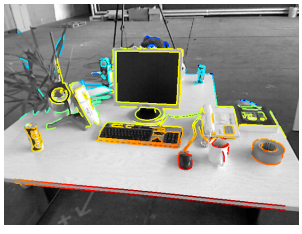
This section reviews the basic idea behind geometric semi-dense 2D-3D registration. After a clear problem definition, we will review existing registration methods, and conclude with a brief summary of the open problems addressed in this paper.

A. Problem formulation

Let $\mathcal{P}^{\mathcal{F}} = \{\mathbf{p}_i^{\mathcal{F}}\}$ be a set of pixel locations in a frame $\mathcal{F}$ defining the so-called semi-dense region. As illustrated in Fig. 2, it is obtained by thresholding the norm of the image gradient, which could, in the simplest case, originate from a convolution with Sobel kernels. Let us further assume that the depth value z_i for each pixel in the semi-dense region is available as well. In the preregistered case, they are simply obtained by looking up the corresponding pixel location in the associated depth image. For each pixel, a local patch (5×5 pixels) is visited and the smallest depth is



(a) Image gradient's norm map.



(b) 3D Semi-dense map.

Fig. 2. Image gradient is calculated in both horizontal and vertical direction at each pixel location. Euclidean norm of each gradient vector is calculated and illustrated in (a) (brighter means bigger while darker means smaller). The semi-dense region is obtained via thresholding the gradient's Euclidean norm map. By accessing the depth information of the semi-dense region, a 3D semi-dense map (b) is created, in which hot colors refer to close points while cold colors mean faraway points.

selected in the case of a depth discontinuity[1]. This operation ensures that we always retrieve the foreground pixel despite possible misalignments caused by extrinsic calibration errors (between the depth camera and the RGB camera) or asynchronous measurements (RGB and depth) under motion. An example result is indicated in Figure 2(b). We furthermore assume that both the RGB and the depth camera are fully calibrated (intrinsically and extrinsically). Thus we have accurate knowledge about a world-to-camera transformation function $\pi(\lambda \mathbf{f}_i) = \mathbf{p}_i$ projecting any point along the ray defined by unit vector $\mathbf{f}_i$ onto the image location $\mathbf{p}_i$. The inverse transformation $\pi^{-1}(\mathbf{p}_i) = \mathbf{f}_i$ transforming points in the image plane into unit direction vectors located on the unit sphere around the center of the camera is also known. If the RGB image and the depth map are already registered, the extrinsic parameters can be omitted. Our discussion from now on will be based on this assumption.

Consider the 3D semi-dense map (defined in the reference frame $\mathcal{F}_{\text{ref}}$) as a curve in 3D, and its projection into the current frame $\mathcal{F}_k$ as a curve in 2D. The goal of the registration step consists of retrieving the pose at the current frame $\mathcal{F}_k$ (namely its position $\mathbf{t}$ and orientation $\mathbf{R}$) such that the projected 2D curve aligns well with the semi-dense region $\mathcal{P}^{\mathcal{F}_k}$ extracted in the current frame $\mathcal{F}_k$. Note that—due to perspective transformations—this is of course not a one-to-one correspondence problem. Also note that we parametrize our curves by a set of points originating from pixels in an image. While there are alternatives to this (e.g. splines), the objective function outlined in this work will remain applicable to any parametrization of the structure.

B. ICP-based motion estimation

The problem can be formulated as follows. Let

$$\mathcal{S}^{\mathcal{F}_{\text{ref}}} = \left\{ \mathbf{s}_i^{\mathcal{F}_{\text{ref}}} \right\} = \left\{ d_i^{\mathcal{F}_{\text{ref}}} \pi^{-1}\left(\mathbf{p}_i^{\mathcal{F}_{\text{ref}}}\right) \right\} \tag{1}$$

denote the 3D semi-dense map in reference frame $\mathcal{F}_{\text{ref}}$, where $d_i = \frac{z_i}{\mathbf{f}_{i,3}}$ denotes the distance of point $\mathbf{s}_i$ to the optical center. Its projection into the current frame $\mathcal{F}_k$ results in the semi-dense region

$$\mathcal{O}^{\mathcal{F}_k} = \left\{ \mathbf{o}_i^{\mathcal{F}_k} \right\} = \left\{ \pi\Big(\mathbf{R}^{\mathrm{T}}\big(\mathbf{s}_i^{\mathcal{F}_{\text{ref}}} - \mathbf{t}\big)\Big) \right\}. \tag{2}$$

We define

$$n(\mathbf{o}_i^{\mathcal{F}_k}) = \underset{\mathbf{p}_j^{\mathcal{F}_k} \in \mathcal{P}^{\mathcal{F}_k}}{\operatorname{argmin}} \|\mathbf{p}_j^{\mathcal{F}_k} - \mathbf{o}_i^{\mathcal{F}_k}\| \tag{3}$$

to be a function that returns the nearest neighbour of $\mathbf{o}_i^{\mathcal{F}_k}$ in $\mathcal{P}^{\mathcal{F}_k}$ under the Euclidean distance metric. The overall objective of the registration is to find

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{o}_i^{\mathcal{F}_k} - n(\mathbf{o}_i^{\mathcal{F}_k})\|^2, \tag{4}$$

where $\boldsymbol{\theta} := [c_1, c_2, c_3, t_x, t_y, t_z]^{\mathrm{T}}$ represents the parameter vector that defines the pose of the camera. c_1, c_2, c_3 are Cayley parameters [22] for orientation $\mathbf{R}$[2], and $\mathbf{t} = [t_x, t_y, t_z]^{\mathrm{T}}$. The above objective is of the same form as the classical ICP problem, which alternates between finding approximate nearest neighbours and then registering those putative correspondences, except that in the present case, the correspondences are between 2D and 3D entities. A very similar objective function has been already exploited by [19] for robust semi-dense 2D-3D registration in a hypothesis-and-test scheme. It proceeds by iterative sparse sampling and closed-form registration of approximate nearest neighbours.

C. Distance fields

As already outlined in [19], the repetitive explicit search of nearest neighbours is too slow even in the case of robust sparse sampling. This is due to the fact that all distances need to be computed in order to rank the hypotheses, and this would again require an exhaustive nearest neighbour search. This is where distance transforms come into play. The explicit location of a nearest neighbour does not necessarily matter in order to evaluate the optimization objective (4), the distance alone may already be sufficient. Therefore, we can pre-process the semi-dense region in the current frame and derive an auxiliary image in which the value at every pixel simply denotes the Euclidean distance to the nearest point in the original semi-dense region. Euclidean distance fields can be computed very efficiently using region growing techniques. Chebychev distance is an alternative when faster performance is required. For further information, the interested reader is referred to [23].

Let us define the function $d(\mathbf{o}_i^{\mathcal{F}_k})$ that retrieves the distance to the nearest neighbour by simply looking up the value at $\mathbf{o}_i^{\mathcal{F}_k}$ inside the chosen distance field. The optimization objective (4) can now easily be rewritten as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} d(\mathbf{o}_i^{\mathcal{F}_k})^2. \tag{5}$$

Note that, in order to emulate a smooth optimization objective and bypass the effects of image discretization, the distances in the field are sampled using bilinear interpolation.

There are a few problems with objective (5):

- It is the sum of squared residual distances. The residual distance is a positive entity which means that it is hard to optimize by techniques other than gradient descent like methods[3]. Despite that it may have good convergence properties, it is known to be slow due to the cascaded update procedure, which may for instance

[1]The depths of all pixels in the patch are sorted and clustered based on a simple Gaussian noise assumption. If there exists a cluster center that is closer to the camera, the depth value of the current pixel will be replaced by the depth of that center. This circumvents resolution loss and elimination of fine depth texture.

[2]Note that the orientation is always optimized as a change with respect to the previous orientation in the reference frame. The chosen Cayley parametrization therefore is equivalent to the local tangential space at the location of the previous quaternion orientation and, therefore a viable parameter space for local optimization of the camera pose.

[3]The values of the residual distance are always positive, which renders Gauss-Newton type methods not applicable. We discuss how to enable Gauss-Newton by introducing an alternative to Euclidean distance fields in the following section.

involve a bisectioning line-search along the gradient direction within each iteration.

- As very well explained in [16], the distance transform may easily lead to wrong registrations. For instance, if only a part of a model curve is observed in the current frame, the corresponding distance field may easily converge in a wrong location, even if only translational displacements in the image plane are taken into account. A detailed illustration of this problem is given in Fig. 3 of [16]. In their work, they solve the problem through a variable lifting strategy, which however blows up the space of optimized parameters quite significantly.
- Even in the absence of the above two problems, a simple continuous minimization of the L2-norm of the residual distances would simply fail because it is easily affected by outlier associations. In [19], this problem is circumvented by switching to the L1-norm of the residual distances. While a direct continuous minimization of the L1-norm is practically feasible, it remains conceptually wrong. As claimed in [17] the plain residual distance is not necessarily differentiable around zero.

The following section will address these problems one by one.

IV. Theory

This section introduces approximate nearest neighbour fields as an alternative to distance fields, thus enabling registration through only few Gauss-Newton iterations. It also introduces the gradient directions to project the residuals, thus leading to correct registration even though only part of a model is observed. Afterwards, we follow the probabilistic formulation given by [4] and solve its equivalent weighted least squares problem. Finally, the sensor model is learned to determine the optimal weight function. Note that our tracking approach has similarities with [24]. However, our approximate nearest neighbour fields obey the Euclidean distance metric, and we provide a more concise derivation of the Gauss-Newton update steps including robustification against outliers.

A. Approximate Nearest Neighbour Field

As discussed in III-C, a full (signed) residual is needed to make the Gauss-Newton updates applicable. Thus, we replace the Euclidean distance field with a field that can retrieve the exact location of the nearest point on a curve. There is a straightforward alternative to the commonly used distance field that maintains all necessary information for computing full residuals, namely an *Approximate Nearest Neighbour Field (ANNF)*. An ANNF is given by a $w \times h \times 2$ integer matrix, where w denotes the width of the image, and h its height. The integers at coordinates $(x, y, :)$ simply denote the pixel index of the nearest neighbour, rather than the distance to it. An intuitive explanation of the ANNF is illustrated in Fig. 3(a).

What is perhaps surprising is that the ANNF can be computed equally efficiently than the distance field. The reason for this is simply given by the functioning of efficient Euclidean distance field extraction algorithms. They perform region growing starting from the semi-dense region itself. The border of the growing region updates and propagates a reference to the closest point in the seed region (i.e. the original semi-dense region). Extracting a distance field or an ANNF is simply a matter of what piece of information is retained.

Using the ANNF, the function $n(\mathbf{o}_i^{\mathcal{F}_k})$ from (3) now boils down to a trivial look-up. This enables us to again go back to objective (4), and attempt a solution via Gauss-Newton updates. Let us define the residuals

$$\mathbf{v} = \begin{bmatrix} \mathbf{o}_1^{\mathcal{F}_k} - n(\mathbf{o}_1^{\mathcal{F}_k}) \\ \cdots \\ \mathbf{o}_N^{\mathcal{F}_k} - n(\mathbf{o}_N^{\mathcal{F}_k}) \end{bmatrix}_{2N \times 1}. \tag{6}$$

By using (6) in (4), our optimization objective can be reformulated as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{v}\|^2. \tag{7}$$

Supposing that $\mathbf{v}$ were a linear expression of $\boldsymbol{\theta}$, it is clear that solving (7) would be equivalent to solving $\mathbf{v}(\boldsymbol{\theta}) = \mathbf{0}$. The idea of Gauss-Newton updates (or iterative least squares) consists of iteratively performing a first-order linearization of $\mathbf{v}$ about the current value of $\boldsymbol{\theta}$, and then each time improve the latter by solving the resulting linear least squares problem. The linear problem to solve in each iteration therefore is given by

$$\mathbf{v}(\boldsymbol{\theta}_i) + \left.\frac{\partial \mathbf{v}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \boldsymbol{\Delta} = \mathbf{0}, \tag{8}$$

and, using $\mathbf{J} = \left.\frac{\partial \mathbf{v}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}$, its solution is given by

$$\boldsymbol{\Delta} = -(\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{v}(\boldsymbol{\theta}_i). \tag{9}$$

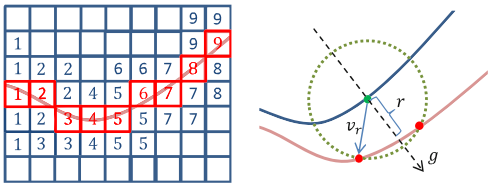
The motion vector is finally updated as $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \boldsymbol{\Delta}$.

While this may sound straightforward, there is one element that requires particular attention. The nearest neighbour of each point should remain fixed in each round of iterative least squares. This statement particularly addresses the (numerical) Jacobian computation, as even tiny variations of $\mathbf{o}_i^{\mathcal{F}_k}$ can easily lead to a potentially substantial change of the nearest neighbour $n(\mathbf{o}_i^{\mathcal{F}_k})$ (e.g. from a point in one curve to another point in a completely different curve). We circumvent this problem by fixing the nearest neighbours during the Jacobian computation. The Jacobian $\mathbf{J}$ simply becomes

$$\mathbf{J} = \left[\left(\frac{\partial \mathbf{o}_1^{\mathcal{F}_k}}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}} \quad \cdots \quad \left(\frac{\partial \mathbf{o}_N^{\mathcal{F}_k}}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}} \right]^{\mathrm{T}}_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}. \tag{10}$$

B. Projection of residual vectors

While the fixation of the nearest neighbours during the Jacobian computation has clear benefits, it also leads to one further problem. Imagine a case where we have to register one long horizontal and one short vertical line in the image plane, and there are only two degrees of freedom. The horizontal line is already registered, but the vertical one not yet. A shift along the horizontal axis would solve the



(a) ANNF. (b) Projected residual.

Fig. 3. Illustration of the Approximate Nearest Neighbour Field and the projected residual. The red numbers in (a) represent the index of the points on the red curve. Given the coordinate of a point in the map, its closest point can be easily accessed by checking the blue number, thus leading to the residual vector $\mathbf{v}_r$. The projected distance r is finally calculated by projecting $\mathbf{v}_r$ onto the direction of the local gradient g.

problem, however, the Jacobian will not provoke an overall error reduction along this dimension. This is because—with fixed nearest neighbours—the "registered" points along the horizontal edge may lead to spurious residual errors for any horizontal shift that ultimately outweighs the error reduction along the short vertical line.

As shown in Fig. 3(b), we solve this problem by projecting the residual vectors onto the local gradient directions. The new residual is given by

$$\mathbf{r} = \begin{bmatrix} \left(\mathbf{o}_1^{\mathcal{F}_k} - n(\mathbf{o}_1^{\mathcal{F}_k})\right)^{\mathrm{T}} \mathbf{g}(\mathbf{p}_1^{\mathcal{F}_{\mathrm{ref}}}) \\ \cdots \\ \left(\mathbf{o}_N^{\mathcal{F}_k} - n(\mathbf{o}_N^{\mathcal{F}_k})\right)^{\mathrm{T}} \mathbf{g}(\mathbf{p}_N^{\mathcal{F}_{\mathrm{ref}}}) \end{bmatrix}_{N\times 1}, \tag{11}$$

where the gradient of the registered point in the reference frame is denoted by $\mathbf{g}(\mathbf{p}_i^{\mathcal{F}_{\mathrm{ref}}})$, and remains fixed throughout the optimization. This is only an approximation of the local curve gradient in the current frame, which is sufficiently valid under the assumption that the frame-to-frame transformation—and notably the rotation about the principal axis of the camera—is small enough. Also, while the residual errors have now become scalars again, they remain signed entities, and thus Gauss-Newton remains applicable. The new Jacobian is finally given by

$$\mathbf{J} = \left[\left(\frac{\partial \left(\mathbf{g}(\mathbf{p}_1^{\mathcal{F}_{\mathrm{ref}}})^{\mathrm{T}} \mathbf{o}_1^{\mathcal{F}_k} \right)}{\partial \boldsymbol{\theta}} \right)^{\mathrm{T}} \quad \cdots \quad \left(\frac{\partial \left(\mathbf{g}(\mathbf{p}_N^{\mathcal{F}_{\mathrm{ref}}})^{\mathrm{T}} \mathbf{o}_N^{\mathcal{F}_k} \right)}{\partial \boldsymbol{\theta}} \right)^{\mathrm{T}} \right]^{\mathrm{T}}_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}. \tag{12}$$

Note that the projection of residual vectors onto the local gradient direction also helps to better approximate the orthogonal distance between curves, and thus address the problem raised in [16]—how to avoid wrong registrations in the case where some of the curves are observed partially.

C. Robust motion estimation

From a probabilistic point of view, the motion would be estimated by maximizing the posteriori $p(\boldsymbol{\theta}|\mathbf{r})$ in the presence of noise. Following the derivation in [4], the Maximum A Posteriori (MAP) problem is translated into the weighted least squares minimization problem,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_i \omega(r_i)(r_i(\boldsymbol{\theta}))^2. \tag{13}$$

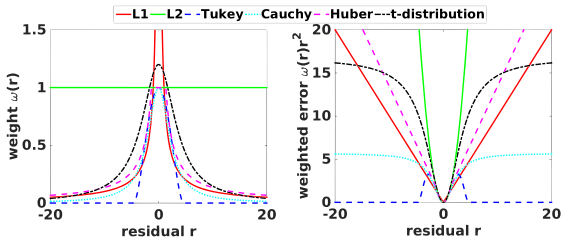


Fig. 4. Illustration of several robust weight functions and corresponding weighted squared errors. The parameters used are from [20].

The weight is defined as $\omega(r_i) = -\frac{1}{2r_i}\frac{\partial \log p(r_i|\boldsymbol{\theta})}{\partial r_i}$, which is a function of the sensor model $p(r_i|\boldsymbol{\theta})$. IRLS is used for solving (13) and we discuss how to determine the optimal weight function $\omega(\cdot)$ by learning the statistical characteristics of the sensor model in the next section.

D. Learning the sensor model

We investigate several of the most widely used robust weight functions. They are illustrated in Fig. 4 together with their corresponding weighted squared errors. The interested reader can find more details in [20], [21].

The choice of the weight function depends on the statistics of the residual, which is identified in a dedicated experiment. We start by defining reference frames in a sequence by applying the same criteria to create new reference frames as shown in the full pipeline (cf. Fig. 6). The residuals are calculated using the ground truth relative pose between the reference frame and the current frame. The residuals are collected over an entire sequence, and then summarized in a histogram as shown in Fig. 5. By fitting the various distribution models depicted in Fig. 4 to the data, we finally identify the t-distribution to be the best model to describe the residual statistics. Assuming the mean of the t-distribution is always zero, only two parameters (ν_0 and σ_0) have to be determined during the model fitting. As shown in [4], the variance σ is later on recursively updated on the online data before being used for calculating weights.

V. Framework Overview

Here we discuss how to improve the robustness of the method further by incorporating a constant velocity motion model. Finally, we describe the complete VO pipeline.

A. Constant velocity motion model

Given a sufficiently high processing rate, even a simple motion model can be very helpful to predict a good starting

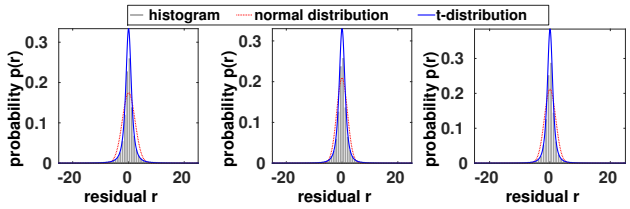


Fig. 5. Sensor model $p(r)$ is obtained by fitting the histogram with certain probabilistic distribution. T-distribution gives best fitting result for all selected sequences fr1/floor, fr2/desk, fr3/structure_texture_near.

point for the optimization which is relatively close to the optimum where the residuals are minimal. This strategy has been widely used in VO and SLAM works [1], [25], [4], and it improved the robustness of the system by effectively avoiding local minima in the optimization. Instead of assuming a prior distribution for the motion as in [4], we follow [1] and implement a simple decaying velocity model. It effectively improves the convergence speed and the tracking robustness, especially when the displacement between the reference frame and the current frame is relatively large.

B. Complete VO system

The complete VO system is designed based on the above robust motion estimation method. Two main threads are running in parallel, which are marked with dashed lines in Fig 6. In the motion estimation thread, only the RGB image is used for the extraction of the semi-dense region and the subsequent ANNF computation. The objective is constructed and then optimized via the Gauss-Newton method. The reference frame needs to be updated once the current frame is too far away. Thus, we track the disparity between the semi-dense region in the reference frame and the corresponding pixels in the registered current frame. If the median disparity is larger than a given threshold, a new reference frame is created by the 3D semi-dense map (3DSDM) preparation thread, in which the depth information is loaded and corrected by the foreground reasoning operation described in Section III-A.

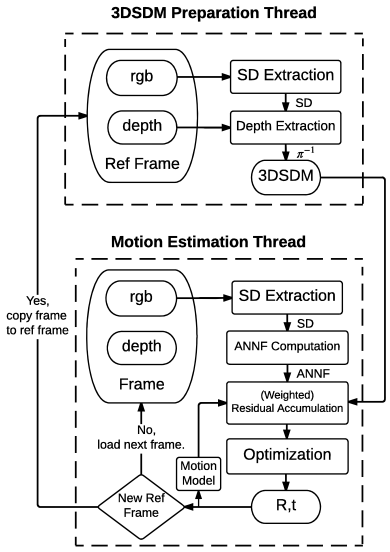


Fig. 6. Flowchart of the proposed method. The two main threads are marked with dashed lines. SD refers to semi-dense region, 3DSDM represents 3D semi-dense map and ANNF means approximate nearest neighour field.

VI. Experimental Evaluation

We now proceed to the evaluation of the proposed method. We start by exploring various configurations in which performance under different semi-dense region extractors and different weight functions are assessed and compared. Then we provide a comparison between the standard Euclidean distance field based method and our method, showing the advantage of the ANNF. We furthermore evaluate our algorithm on a set of benchmark datasets and compare the performance with several state-of-the-art camera tracking solutions. Finally, a semi-dense reconstruction result of two indoor scenes is provided which qualitatively demonstrates that the proposed method is able to work in relatively large-scale environments.

Method	fr2/desk		
	RMSE($\mathbf{R}$)	RMSE($\mathbf{t}$)	Run-time
Sobel	0.562	0.018	**0.00824**
Smoothed + Sobel	0.581	0.018	0.00948
Gradient	0.565	0.017	0.01720
Smoothed + Gradient	**0.560**	**0.016**	0.01863

TABLE I
DIFFERENT METHODS OF SEMI-DENSE REGION EXTRACTION.

Method	fr2/desk		
	RMSE($\mathbf{R}$)	RMSE($\mathbf{t}$)	Run-time
Least Squares	0.899	0.024	**0.03318**
ℓ_1 norm	0.587	**0.016**	0.04211
Tukey	0.879	0.023	0.04077
Huber	0.591	**0.016**	0.04084
Cauchy	0.769	0.019	0.04193
t-distribution	**0.560**	**0.016**	0.04260

TABLE II
DIFFERENT ROBUST WEIGHT FUNCTIONS.

Note that the relative pose errors listed in all tables are given in terms of either root-mean-square error or median error. The unit for rotation and translation error are $\deg/s$ and m/s respectively. The best performance is always highlighted in bold.

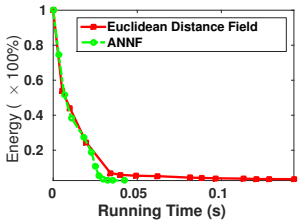
A. Performance: Different gradient extractors

A good extraction of the semi-dense region is key to good motion estimation accuracy. Thus, we provide a comparison in Table I where several different methods are applied for calculating the image gradients. "Smoothed" refers to a 5×5 Gaussian kernel, which is used for smoothing the image. "Gradient" refers to a Sobel-like gradient computation method which uses a 5×5 kernel. It shows the impact that each method makes on the accuracy v.s. required computational time. Note that the t-distribution based IRLS is used. The run-time for computing the semi-dense region is expressed in seconds.

B. Performance: Different weight functions

In order to confirm experimentally that the chosen weight function is optimal, we compare the performance for all robust weight functions over several sequences of the TUM benchmark datasets. Comparison on fr2/desk is provided as an example in Table II. "Smoothed + Gradient" is used for extracting the semi-dense region. The run-time is counted in seconds and includes the extraction of the semi-dense region, the ANNF computation and the following optimization.

(a) The projection of the 3DSDM.



(b) Convergence speed.

Fig. 7. Illustration of comparison between Euclidean distance based method and the novel ANNF based method. (a) shows the projection of the 3DSDM on the current frame before the optimization and (b) demonstrates the convergence speed of the two methods. Although both of them converge at similar rates, the ANNF based method is more efficient. The optimization of the two methods start from exact the same energy ((b) depicts the normalized energy).

C. Euclidean distance field v.s. ANNF

As discussed in III-B, by using ANNFs, we are able to calculate the signed orthogonal residual between the registered curves, thus enabling Gauss-Newton updates to solve the problem. Here we confirm that this leads to faster convergence compared to gradient descent (over the Euclidean distance field). The result in Fig. 7 demonstrates that the ANNF based method converges much faster than the standard Euclidean distance field based method. Note that "Smoothed + Gradient" and t-distribution based IRLS are used.

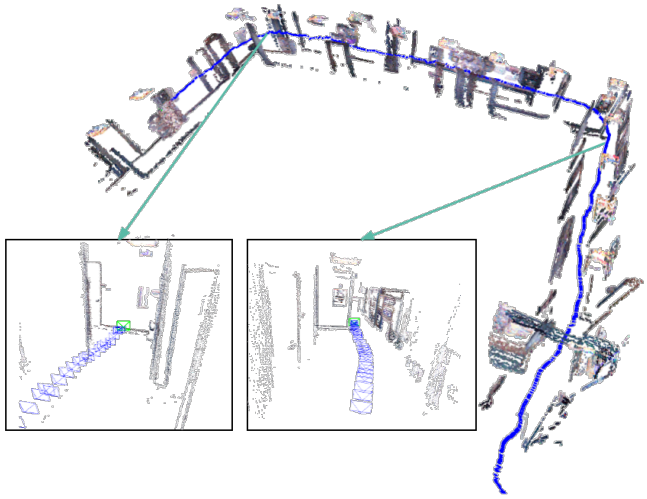
D. Comparison against the state of the art

We compare the performance of our method against three state-of-the-art, open-source motion estimation frameworks: DVO [4], LSD [7], [8] and ICP [10]. For best performance, we apply "Smoothed + Gradient" for extracting the semi-dense region, t-distribution based IRLS and the constant velocity motion model. All methods are evaluated on published and challenging indoor benchmark datasets from the TUM RGB-D [26] series. The datasets we picked for evaluation and the corresponding results are listed in Table III. DVO, LSD-SLAM and our method perform comparably efficiently on a laptop with only CPU resources (30 Hz, 30 Hz and 25Hz respectively) while ICP achieves 60 Hz on a GPU (NVIDIA Tesla K40). It can be easily observed that our method provides the best overall performance on TUM datasets.

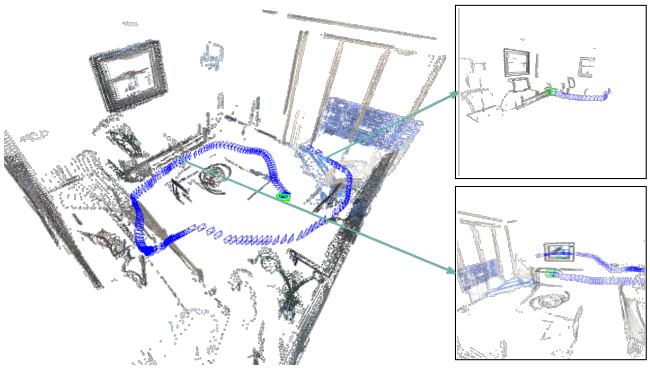
During the evaluation on the TUM datasets, we discovered that almost all underperforming registration results for our method are related to motion blur in the image. The reason is that the semi-dense region cannot be accurately extracted from blurry images, thus also harming the resulting 3D semi-dense map. Consequently, the motion estimation based on the inaccurate 3D semi-dense map will not be accurate. Deblurring techniques or adaptive thresholds could alleviate this problem, but a much more straightforward solution consists of simply discarding frames for which the semi-dense region reveals a sudden jump in cardinality.

E. Semi-dense reconstruction

In order to show that our method is capable to work in relatively large-scale environments, we provide reconstruction results on two sequences from the TAMU RGB-D datasets [14] and ICL-NUIM synthetic datasets [27]. As shown in Fig. 8, the semi-dense reconstruction is much more visually expressive than sparse point clouds.



(a) Corridor reconstruction on the TAMU dataset [14].



(b) ICL living room [27].

Fig. 8. Reconstruction result of indoor scenes.

VII. Conclusions

We present a robust real-time semi-dense visual odometry algorithm for RGB-D cameras. The camera motion is estimated through a non-parametric 2D-3D geometric curve registration approach. The introduction of ANNFs enables the use of Gauss-Newton optimization. To improve robustness against occlusions, noise and outliers, the ICP-based pipeline is formulated as a maximum a posteriori problem, which is subsequently transformed into a weighted least squares problem. Furthermore, we study the statistical properties of the sensor model, which leads to the optimal choice from various robust M-Estimators. Experiments show that our geometric registration alternative outperforms state-of-the-art camera tracking solutions in most cases. The method may be pushed even further by a more accurate and robust method for extracting contours, a more elaborate motion estimation filter, as well as a sliding window refinement. We will furthermore explore the implementation of hybrid cues (geometric and photometric) into our framework.

Odometer	Error	fr1/xyz	fr1/floor	fr2/xyz	fr2/rpy	fr2/desk	fr3/cabinet	fr3/long office hosuehold	fr3/structure texture near	Average
DVO	RMSE(R)	1.831	2.221	0.493	0.685	1.023	4.912	0.840	**0.938**	1.618
	Median(R)	**1.259**	0.705	0.407	0.538	0.830	4.457	0.635	**0.740**	1.200
	RMSE(t)	0.040	0.074	0.013	0.018	0.026	0.145	0.024	**0.018**	0.045
	Median(t)	0.030	0.016	0.011	0.011	0.021	0.130	0.018	**0.015**	0.032
LSD SLAM	RMSE(R)	3.973	5.071	0.463	5.208	3.482	12.114	**0.631**	10.446	5.174
	Median(R)	2.946	3.352	0.314	4.398	0.854	10.615	0.483	3.457	3.302
	RMSE(t)	0.053	0.121	0.009	0.015	0.102	0.272	0.026	0.178	0.097
	Median(t)	0.042	0.089	0.005	0.011	0.058	0.270	0.018	0.067	0.070
ICP	RMSE(R)	1.812	5.252	1.307	2.760	4.393	6.640	5.733	6.019	4.240
	Median(R)	1.346	2.181	0.893	1.951	3.071	6.027	3.960	1.929	2.670
	RMSE(t)	**0.031**	0.209	0.027	0.053	0.108	0.171	0.131	0.132	0.108
	Median(t)	**0.024**	0.069	0.021	0.041	0.078	0.153	0.099	0.056	0.068
Our Method	RMSE(R)	**1.533**	**0.746**	**0.324**	**0.356**	**0.560**	**2.692**	0.673	1.128	**1.001**
	Median(R)	1.373	**0.587**	**0.251**	**0.283**	**0.425**	**1.451**	**0.411**	0.789	**0.696**
	RMSE(t)	0.041	**0.013**	**0.005**	**0.005**	**0.016**	**0.063**	**0.016**	0.024	**0.023**
	Median(t)	0.028	**0.012**	**0.004**	**0.004**	**0.012**	**0.034**	**0.010**	0.017	**0.015**

TABLE III

PERFORMANCE COMPARISON ON TUM DATASET.

REFERENCES

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Mixed and Augmented Reality, 6th IEEE and ACM International Symposium on*, 2007, pp. 225–234.

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

[3] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 2050–2056.

[4] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3748–3754.

[5] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 719–722.

[6] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," in *Australasian Conf. on Robotics and Automation*, vol. 2, 2011, pp. 2–2.

[7] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1449–1456.

[8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[9] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 10th IEEE international symposium on*, 2011, pp. 127–136.

[10] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *3rd RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, (Sydney, Australia)*, 2012.

[11] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, "Tracking a depth camera: Parameter exploration for fast ICP," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3824–3829.

[12] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[13] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," *Image and Vision Computing*, vol. 27, no. 5, pp. 588–596, 2009.

[14] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 688–694.

[15] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *European Conference on Computer Vision*. Springer, 2008, pp. 802–815.

[16] I. Nurutdinova and A. Fitzgibbon, "Towards pointless structure from motion: 3D reconstruction and camera parameters from general 3d curves," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2363–2371.

[17] M. P. Kuse and S. Shen, "Robust camera motion estimation using direct edge alignment and sub-gradient method," in *IEEE International Conference on Robotics and Automation (ICRA-2016), Stockholm, Sweden*, 2016.

[18] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proceedings of the 14th International Conference on Computer Vision (ICCV)*, 2013, pp. 1457–1464.

[19] L. Kneip, Z. Yi, and H. Li, "SDICP: Semi-dense tracking based on iterative closest points," in *Proceedings of the British Machine Vision Conference (BMVC)*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 100.1–100.12. [Online]. Available: https://dx.doi.org/10.5244/C.29.100

[20] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[21] K. Aftab and R. Hartley, "Convergence of iteratively re-weighted least squares to robust M-estimators," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 480–487.

[22] A. Cayley, "About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic," in *Reine Angewandte Mathematik*, 1846.

[23] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, "2D euclidean distance transform algorithms: A comparative survey," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, p. 2, 2008.

[24] J. J. Tarrio and S. Pedre, "Realtime edge-based visual odometry for a monocular camera," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 702–710.

[25] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys, "Live metric 3D reconstruction on mobile phones," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 65–72.

[26] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580.

[27] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.