Robust Camera Motion Estimation using Direct Edge Alignment and Sub-gradient Method

Manohar Kuse and Shaojie Shen

Abstract— **There has been a paradigm shifting trend towards feature-less methods due to their elegant formulation, accuracy and ever increasing computational power. In this work, we present a direct edge alignment approach for 6-DOF tracking. We argue that photo-consistency based methods are plagued by a much smaller convergence basin and are extremely sensitive to noise, changing illumination and fast motion. We propose to use the Distance Transform in the energy formulation which can significantly extend the influence of the edges for tracking. We address the problem of non-differentiability of our cost function and of the previous methods by use of a sub-gradient method. Through extensive experiments we show that the proposed method gives comparable performance to the previous method under nominal conditions and is able to run at 30 Hz in single threaded mode. In addition, under large motion we demonstrate our method outperforms previous methods using the same run-time configuration for our method.**

I. Introduction

Reliable pose tracking of a robotic vehicle is a key element in its precise and stable control for an autonomous operation. To achieve this, the Global Positioning System (GPS) has been a key enabler for autonomous navigation. However, for GPS denied environments localization and navigation using other on-board sensor modalities like lasers, IMUs, monocular and stereo cameras, and RGB-D cameras have been popular research topics [1], [2]. In this work, we propose a direct (feature-less) approach for visual 3D-2D relative 6-DOF pose estimation (visual odometry) with explicit handling of non-differentiability of the cost function.

RGB-D cameras provide a simple and cost effective way to obtain scene information in 3D. A host of 3D-2D, 3D-3D techniques are available. These approaches can broadly be classified into 1) *feature-based methods* , 2) *direct methods* and 3) *Iterative Closest Point (ICP) based methods* .

Next we review the literature only for visual odometry using an RGB-D cameras. The feature based methods involve extraction of salient image features (eg., SIFT, SURF [3] etc.). These features are tracked from a reference frame to the current frame to compute a relative camera pose. These methods involve solving the PNP (Perspective N-Point Projection) problem. Finally, bundle adjustment is applied to reduce the drift [4]. Huang *et al.* [5] employed a non-linear least square solver for the minimization of the distance between inlier matched feature points. Dryanovski *et al.* [6] proposed to use a consistent scene model which is dynamically updated upon new observations using a Kalman filter. Due to the pre-selection of feature points (usually 50-500) most of the information from the image is lost. Further, these approaches are ineffective in featureless environments.

Direct methods (sometimes also referred as featureless methods) do not involve the extraction of keypoints. Instead they directly use the image-pixels for motion estimation. This enables them to use more information from the images to estimate the relative poses. The methods presented by Steinbrucker *et al.*[7], Tykkala *et al.*[8], Kerl *et al.*[2] assume photo-consistency of the scene. The estimation of the rigid transform is performed by minimization of the photo-metric error between the 3-D points (back projected using depth) in the reference image and reprojected points onto the current frame. As a result, these approaches are sensitive to illumination variation and noise in intensities. Furthermore, direct photometric approach has a small basin of attraction. It does not result in good estimates of transform between the two images (or point clouds) when the motion between the two capture locations is large, as has been noted in [2].

Iterative Closest Point (ICP) based methods directly align 3D point clouds. Stuckler *et al.*[9] has employed a method based on ICP for the alignment of point clouds obtained from an RGB-D camera. Rusinkiewicz *et al.* [10] is a survey on some other attempts which use efficient ICP-like methods for pose estimation. Fitzgibbon [11] has proposed an algorithm to align two 2D point sets. Their algorithm is also extensible to 3D point sets. [11] uses the distance transform to model the point correspondence function to align 2D curves.

A. Contribution

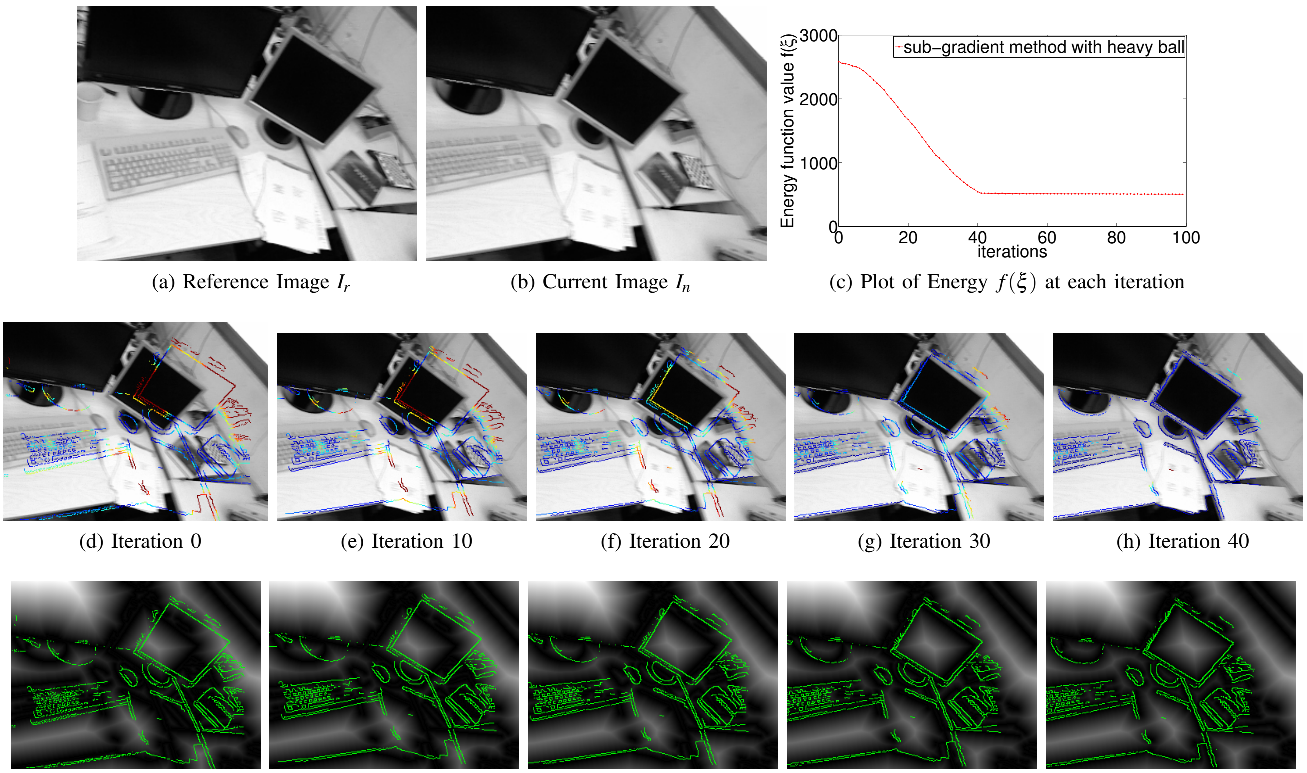
In this work, we propose a novel feature-less approach for 3D-2D relative pose estimation[1]. We propose to align a reference image with the current image by minimization of the sum of squared distances between transformed-projected (on current frame) coordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame. We make use of only the edge pixels, relying on our observation that the Distance Transform [12] of the edge-map of the current frame can be used to model the distance between edge pixels from the current frame and the re-projections from the reference frame.

Unlike the previous direct approach by Kerl *et al.* [2], our proposed approach does not rely on the photometric cost function and thus on the photo-consistency assumption.

All authors with Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay Road, Kowloon, Hong Kong. {mpkuse, eeshaojie}@ust.hk

This work was supported by HKUST project R9341 and HKUST PGS Award Scheme for International RPg Students.

[1] The source code and more results are available on our website : http://bit.ly/1KNtprg


3000
2000
1000
0
Energy function value f(ξ)
sub–gradient method with heavy ball
0
20
40
60
80
100
iterations

(a) Reference Image I_r

(b) Current Image I_n

(c) Plot of Energy $f(\xi)$ at each iteration

(d) Iteration 0

(e) Iteration 10

(f) Iteration 20

(g) Iteration 30

(h) Iteration 40

(i) Iteration 0

(j) Iteration 10

(k) Iteration 20

(l) Iteration 30

(m) Iteration 40

Fig. 1: Showing reprojections of edge-pixels in the reference frame, onto the current frame as the *sub-gradient* method progresses. The middle row shows the reprojections on the current gray image. They are false colored to represent $\upsilon_{e_i}(\xi)$. The last row shows reprojections on the distance transform image of the edge-map of the current frame. Note that the current frame and the reference frame are about 160 ms apart (5 frames) and sub-gradient method progress is shown without pyramids with initial guess as identity. Viewing in color is recommended.

The implication of this being, it is robust under changing lightening conditions. Also, being a direct method, does not involve sparse features (like SURF, SIFT etc.) and feature matches. Next we make the case for the use of a sub-gradient method (first order method for non-differentiable cost functions), instead of a Gauss-Newton method, for minimization by arguing the non-differentiability of the cost function used in previous dense methods.

We demonstrate that our method has much larger convergence basin (see Fig. 1) when compared to previous direct method [2]. We attribute the larger convergence basin of our method to the use of the distance transform to model the reprojected distances. In particular, the distance transform, by its definition extends the influence of edges much farther.

Our methods runs at 30 Hz on a PC with Intel Core i7-2600 CPU (3.4 Ghz) with 16 GB of RAM and gives comparable relative pose accuracies when evaluated against the method presented by Kerl *et al.* [2]. We evaluate the proposed method with the TUM-RGBD dataset [13].

II. Direct Edge Alignment (D-EA) Formulation

In this section, we introduce our formulation for relative camera motion estimation using RGB-D camera, which we refer to *D-EA* (*D*irect *E*dge *A*lignment) formulation. It is based on the minimization of geometric error term at each edge pixel to obtain an estimate of the rigid body transform between two frames, i.e., to find a pose (rotation and translation matrix) such that the edges of the two images align. This is in contrast to previous direct methods, notably the one proposed by Kerl *et al.* [2] which minimizes the photometric error at every pixel.

Thus, we propose an energy formulation, which is the sum of squared distances between transformed-projected (on current frame) coordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame.

A. Notations and Conventions

The RGB image collected from RGB-D sensor at timestamp k is denoted as $I_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$, where Ω represents the image domain. The depth image at timestamp k is denoted as $Z_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$. Let ${}^{k}\mathbf{P} \in \mathbb{R}^3$ denote a 3D scene point in the co-ordinate system of the camera optical center at time instance k. Since our approach computes the relative pose which are chained to estimate a trajectory, we denote the reference frame (periodically updated) with the script r and the current frame by n. Let ${}^{r}_{n}\mathbf{R}$, ${}^{r}_{n}\mathbf{T}$ together denote a rigid transformation between the reference and current frames. For convinence of notation we derive our energy formulation

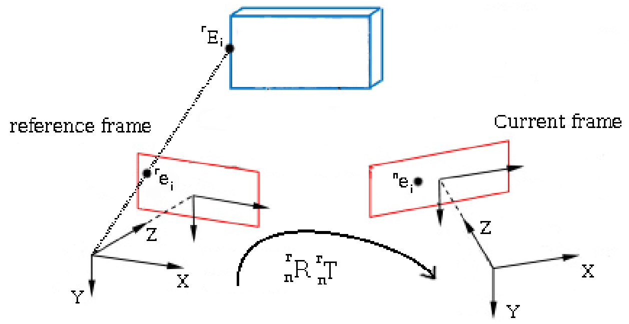


Fig. 2: Notations and Conventions

using $\mathbf{R}$, $\mathbf{T}$ as the alias to ${}^{r}_{n}\mathbf{R}$, ${}^{r}_{n}\mathbf{T}$ for a particular instance of reference and current frames.

The camera projection function $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ projects the visible 3D scene point onto the image domain. The inverse projection function $\tilde{\Pi}\ :\ (\mathbb{R}^2,\ \mathbb{R})\ \mapsto\ \mathbb{R}^3$ back-projects a pixel coordinate given the depth at this pixel coordinate:

$$ {}^{k}\mathbf{u} = \Pi({}^{k}\mathbf{P}) \tag{1}$$

$$ {}^{k}\mathbf{P} = \tilde{\Pi}({}^{k}\mathbf{u}, Z_k({}^{k}\mathbf{u})). \tag{2}$$

where ${}^{k}\mathbf{u} \in \mathbb{R}^2$ denotes the image coordinates of the 3D point ${}^{k}\mathbf{P}$.

B. Relative Motion Estimation

We propose a formulation based on geometric error terms for relative pose estimation of the current frame with respect to a previously set reference frame (the reference frame is periodically updated). Following our notations, we denote the RGB image and depth map of the reference frame as I_r and Z_r respectively. The i^{th} image point and its corresponding 3D point of the reference image are denoted as ${}^{r}\mathbf{u}_i$ and ${}^{r}\mathbf{P}_i$, respectively. Similarly, the RGB image of current frame is denoted by I_n and the i^{th} point of the current frame is denoted as ${}^{n}\mathbf{u}_i$. Thus, the proposed geometric energy function is the sum of the distances of the re-projections (of edge points from reference image) and nearest edge points in current image:

$$ f(\mathbf{R},\mathbf{T}) = \sum_i \min_j D^2\big(\ \Pi[\mathbf{R}^T({}^{r}\mathbf{P}_i - \mathbf{T})],\ {}^{n}\mathbf{u}_j\big). $$

where $D : (\mathbb{R}^2, \mathbb{R}^2) \mapsto \mathbb{R}$ denotes the Euclidean distance between those points. The best estimates for the rigid transform can be obtained by solving the following optimization problem.

$$ \underset{\mathbf{R},\mathbf{T}}{\text{minimize}}\ f(\mathbf{R},\mathbf{T}) $$
$$ \text{subject to } \mathbf{R} \in SO(3) $$

Following the theory of optimization under unitary constraints [14] which proposes to use mapping onto an appropriate manifold at each iteration step. We use the manifold defined by the Lie algebra $se(3)$ corresponding to the Lie group $SE(3)$ which maps the twist coordinates $\xi \in \mathbb{R}^6$ onto the rigid body transform denoted by a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ using the exponential map (Chp 2. of [15]):

$$ \xi = (\mathbf{t};\mathbf{w})^T \in \mathbb{R}^6. $$

where, $\mathbf{t} \in \mathbb{R}^3$ is the translation component and $\mathbf{w} \in \mathbb{R}^3$ is the rotation component. We denote the rigid body transform on any 3D point in the reference frame, ${}^{r}\mathbf{P}_i$, corresponding to ξ as $\tau({}^{r}\mathbf{P}_i, \xi)$:

$$ \tau({}^{r}\mathbf{P}_i, \xi) = \big[exp(\xi)\big]^{-1}\,{}^{r}\mathbf{P}_i = \mathbf{R}^T({}^{r}\mathbf{P}_i - \mathbf{T}). $$

Consequently, the constraint optimization formulation can be converted to an unconstrained optimization problem:

$$ \underset{\xi}{\text{minimize}} \sum_i \min_j D^2\big(\ \Pi[\tau({}^{r}\mathbf{P}_i, \xi)],\ {}^{n}\mathbf{u}_j\big). $$

In this approach, we observe that, if the image points corresponding to edge points in the reference image (denoted by ${}^{r}\mathbf{e}_i \in \mathbb{R}^2$ with corresponding 3D point, ${}^{r}\mathbf{E}_i$) are pre-selected then the function $\min_j\ D(\mathbf{u}_i, \mathbf{u}_j)$ is exactly the definition of the Distance Transform [12]. We denote the distance transform of the edge-map of the current image as $V^{(n)} : \mathbb{R}^2 \mapsto \mathbb{R}$. Thus, the energy terms for an edge-pixel of the reference frame is given by:

$$ \upsilon_{e_i}(\xi) = V^{(n)}\big(\ \Pi[\tau(\tilde{\Pi}({}^{r}\mathbf{e}_i, Z_r({}^{r}\mathbf{e}_i)), \xi)]\big) $$

$$ f(\xi) = \textstyle\sum_{\forall \mathbf{e}_i} (\upsilon_{e_i}(\xi))^2. \tag{3}$$

To summarize, the proposed direct edge alignment (D-EA) formulation is given by

$$ \xi^* = \underset{\xi}{\text{argmin}}\ \textstyle\sum_{\forall \mathbf{e}_i} (\upsilon_{e_i}(\xi))^2 \tag{4}$$

Here, ξ^* denotes the optimal value of $f(\xi)$ for the above stated optimization problem. It has to be noted that ξ^* gives an estimate of the relative transform between the reference frame and the current frame.

III. Solving D-EA with a Subgradient Method

In this section we highlight an iterative method to solve the proposed optimization problem (Eq. 4). Essentially, we employ a modified *sub-gradient* method for numerical optimization of our optimization problem and provide motivation for the same.

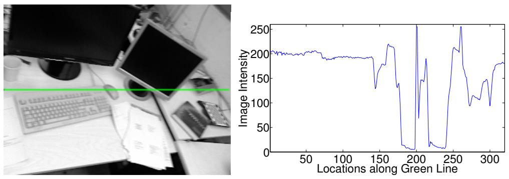
A. Non-Differentiability & Issues of Gauss-Newton Method

The approach by Kerl *et al.* [2] uses a Gauss-Newton method on the linearized energy function to numerically optimize an energy function with non-linear sum of squared terms :

$$ r(\xi, {}^{r}\mathbf{P}_i) = I_n(\Pi[exp(\xi)\ {}^{r}\mathbf{P}_i]) - I_r({}^{r}\mathbf{u}_i) $$

$$ \xi^* = \underset{\xi}{\text{argmin}} \sum_i r_{lin}(\xi, {}^{r}\mathbf{P}_i)^2. $$

The major pitfalls of the Gauss-Newton method are that it does not work with non-differentiable functions and that it has no convergence guarantee. Although their method works well in practice, we argue that their energy function is a



(a) (b)

Fig. 3: Highlighting the non-differentiability of the function I_n at object boundaries

non-differentiable function of ξ and thus does not satisfy the differentiability requirement of Gauss-Newton methods. From a mathematical stand point the concept of gradient (and Jacobian) do not make sense. Further, the linearization may not track the original non-linear function well as the differential ξ gets a higher value away from iteration's initial estimate.

The residue function $r(\xi, {}^r\mathbf{P}_i)$ is essentially a composition $I_n \circ \Pi \circ \tau(\xi, {}^r\mathbf{P}_i)$. From the theorem on continuity and composition in calculus, which states, if g is continuous at a and f is continuous at $g(a)$, only then $f \circ g$ is continuous at a. That is, $\lim_{x \to a} f(g(x)) = f(g(a))$. We argue that the function I_n being a lookup intensity function on Ω is in general non-continuous everywhere. We further illustrate the nature of I_n using fig. 3. We note that even if one thinks of smooth regions in an image as noise-free at object transition points the function I_n will still be non-differentiable. This validates our argument on non-differentiability of the residue function proposed by Kerl *et al.* [2]. On the contrary, one can argue smoothening I_n can help alleviate the problem of non-differentiability to an extent which in turn can improve the performance of Gauss-Newton iterations, but there is no theoretical guarantee on the convergence.

B. The Sub-gradient Method

We observe that our energy formulation (Eq. 4) is also a non-differentiable function of ξ. We propose to use the *sub-gradient method* which is an algorithm for minimizing a non-differential function. The sub-gradient generalizes the notion of derivatives of a non-differential function. Unlike a gradient of a multivariate function which is a vector (at a particular point), sub-gradient is a set of vectors satisfying an inequality. We refer the reader to Boyd *et al.* [16] for further details on sub-gradient calculus and sub-gradient methods.

Algorithmically, the sub-gradient method is a first order method and is very much like the gradient method but with a few notable differences. For example, the step lengths are not chosen with a line search, instead they are fixed ahead of time. In contrast with the ordinary gradient method, the sub-gradient method is not a descent method and as the iterations progress, the function value can increase.

Similar to a gradient method, the sub-gradient method starts with an initial estimate ($\xi^{(0)}$) and iteratively proceeds in the negative direction of an element in the sub-gradient set (with respect to the twist, ξ).

$$\xi^{(k+1)} = \xi^{(k)} + (-\alpha_k)\ \hbar^{(k)}. \tag{5}$$

Here, $\xi^{(k)}$ is the estimate of ξ^* in the k^{th} iteration. $\hbar^{(k)} \in \mathbb{R}^6$ denotes any sub-gradient of f at $\xi^{(k)}$ and $\alpha_k > 0$ is the step size. We use the slashed symbol ($\hbar$) to emphasis yet again the use of sub-gradients. It follows from the convergence proof of the sub-gradient method that, unlike the gradient methods, the step sizes α_k are fixed ahead of time. See section III-D on choosing step sizes. The operator $+$ denotes combination in the sense of Lie Groups given by

$$\xi_a + \xi_b := log(exp(\xi_a)\ exp(\xi_b)).$$

Also as the sub-gradient method is not a descent method we keep track of the best estimates found so far

$$f_{best}^{(k)} = min(f(\xi^{(0)}), f(\xi^{(1)}), \ldots, f(\xi^{(k)})). \tag{6}$$

C. Computation of a Sub-Gradient

In this section, we derive an expression for a sub-gradient, $\hbar$ with respect to ξ at $\xi = \xi^{(k)}$ of the proposed energy function. By definition, for $\upsilon_{e_i}(\xi)$, defined on the Euclidean set $\mathbb{R}^6$, a vector $\mathbf{c} \in \mathbb{R}^6$ is its sub-gradient if [17]

$$\lim_{\xi \to \xi^{(k)}} \upsilon_{e_i}(\xi) - \upsilon_{e_i}(\xi^{(k)}) \geq \lim_{\xi \to \xi^{(k)}} \mathbf{c} \cdot (\xi - \xi^{(k)}). \tag{7}$$

Now, $\upsilon_{e_i}(\xi) = V^{(n)} \circ \Pi \circ \tau(\xi, {}^r\mathbf{P}_i)$. As proved in [17] the chain-rule for differentiable functions is also valid for non-differentiable functions by replacing the sub-gradient in place of gradient. Thus, we write the chain rule for computing the sub-derivate of $\upsilon_{e_i}(\xi)$ with respect to ξ as

$$\mathbf{J}_{e_i} := \frac{\partial V^{(n)}}{\partial e_i} \cdot \frac{\partial e_i}{\partial E_i} \cdot \frac{\partial E_i}{\partial \xi}\bigg|_{\xi = \xi^{(k)}}. \tag{8}$$

where, by our conventions,

$$E_i(\xi^{(k)} + \delta\xi) \approxeq [\hat{\mathbf{R}}(\mathbf{I}_3 + \lfloor \delta\mathbf{w} \rfloor_x)]^T \left[{}^r\mathbf{E}_i - \hat{\mathbf{T}} - \delta\mathbf{t} \right].$$

$$\frac{\partial E_i}{\partial \xi}\bigg|_{\xi = \xi^{(k)}} = \hat{\mathbf{R}}^T \left[\ -\mathbf{I}_3 \ \middle|\ \lfloor {}^r\mathbf{E}_i - \hat{\mathbf{T}} \rfloor_x\ \right] \tag{9}$$

$\hat{R}$ and $\hat{T}$ represent the rotation and translation matrix of $exp(\xi^{(k)})$ and $\delta\xi = (\delta\mathbf{t}; \delta\mathbf{w})$, i.e., the initial estimates at every iteration. The first of the two terms is computed by forward differencing the distance transform image $V^{(n)}$ and the second term is computed from the definition of projection function.

Next we justify why $\mathbf{J}_{e_i}$ using Eq. 8 and Eq. 9 substituted for $\mathbf{c}$ satisfies Eq. 7. Once this is proved, the expression $\mathbf{J}_{e_i}$ will represent a sub-gradient of $\upsilon_{e_i}(\xi)$ evaluated at $\xi = \xi^{(k)}$. We argue that within a sufficiently small hypersphere of radius ε it is safe to assume that a linear approximation of $\upsilon_{e_i}(\xi)$ closely tracks the original non-linear function $\upsilon_{e_i}(\xi)$. Using the Taylor series expansion, we have

$$\upsilon_{e_i}(\xi) = \upsilon_{e_i}(\xi^{(k)}) + \mathbf{J}_{e_i} \cdot (\xi - \xi^{(k)})$$
$$\text{subject to } ||\xi - \xi^{(k)}|| \leq \varepsilon.$$

This is sufficient to conclude that the proposed $\mathbf{J}_{e_i}$ is a sub-gradient of υ_{e_i}, as the above expression trivially satisfies Eq. 7. The constraint $||\xi - \xi^{(k)}|| \leq \varepsilon$ can be explicitly enforced by use of the *projected sub-gradient* method which is an extension of the sub-gradient method to solve a constrained optimization problem [16]. The projected sub-gradient method works by projecting the updated estimate of ξ i.e., $\xi^{(k+1)}$ onto the hyper-sphere [2] [16].

It is relatively straightforward to derive the expression for $\hbar^{(k)}$ (sub-gradient of $f(\xi)$ at $\xi = \xi^{(k)}$):

$$\hbar^{(k)} = \textstyle\sum_{\forall \mathbf{e}_i} 2\ \upsilon_{e_i}(\xi^{(k)})\ \mathbf{J}_{e_i}. \tag{10}$$

D. Analysis on Step Size

Unlike the convergence of the gradient method, which is based on the decrease of the function value at each iteration, the convergence of the sub-gradient method is based on the Euclidean distance to the optimal set. As has been shown by Boyd *et al.* [16]. We present the convergence of a standard sub-gradient method as derived by [16] in this subsection, and in the next subsection we built upon it to derive convergence with the fast convergence strategy.

$$\begin{aligned} ||\xi^{(k+1)} - \xi^{*}||_2^2 \leq ||\xi^{(k)} - \xi^{*}||_2^2 - 2\alpha_k(f(\xi^{(k)}) - f^{*}) \\ + \alpha_k^2 ||\hbar^{(k)}||_2^2. \end{aligned} \tag{11}$$

Assume that $f(\xi)$ satisfies the Lipschitz condition i.e., $|f(\xi_a) - f(\xi_b)| \leq G||\xi_a - \xi_b||_2$. Equivalently bounded sub-gradient, $||\hbar^{(k)}|| \leq G$ (by definition of sub-gradient). Also assume R is an upper bound on the distance of the initial guess to the optimal set i.e., $||\xi^{(0)} - \xi^{*}||_2 \leq R$. One can derive the inequality

$$f_{best}^{(k)} - f^{*} \leq \frac{R^2 + G^2 \sum_{p=0}^{k} \alpha_p^2}{2\sum_{p=0}^{k} \alpha_p}. \tag{12}$$

Now suppose we set $\alpha_p := \frac{\eta}{(p+1)};\ p = 0, \cdots, k$ and η is a constant. As $k \rightarrow \infty$, $\sum_{p=0}^{k} \alpha_p^2 < \infty$ and $\sum_{p=0}^{k} \alpha_p = \infty$. This leads to convergence of the sub-gradient method under the square summable but not summable step sizes. Thus we use those step sizes:

$$\lim_{k \rightarrow \infty} f_{best}^{(k)} = f^{*}.$$

E. Fast Convergence Strategies

Although the sub-gradient methods are guaranteed to converge, their convergence speed is rather slow. However, each iteration is of very low complexity as it avoids forming and solving the normal equations. A general approach to speed up the convergence of a gradient method is to use a class of methods called *heavy ball methods*. For review of literature we direct the reader to the works of Nesterov [18], [19]. Although there are a few variants of the heavy ball methods, we propose to use the update direction as a conic combination ($\beta \in (0,1)$) of the sub-gradient in current iteration ($\hbar^{(k)}$) and previous iterations. Thus, we use the update direction $\mathbf{s}^{(k)}$ instead of $\hbar^{(k)}$.

$$\mathbf{s}^{(k)} = (1-\beta)\hbar^{(k)} + \beta \mathbf{s}^{(k-1)}. \tag{13}$$

$$\mathbf{s}^{(k)} = (1-\beta)\left(\sum_{i=1}^{k} \beta^{k-i} \hbar^{(i)}\right) \tag{14}$$

Eq. 14 can be derived by recursively applying Eq. 13. Following the logic from the previous sub-section, the convergence of the modified sub-gradient method is analyzed by the distance of the iterate to the optimal set ($||\xi^{(k+1)} - \xi^{*}||_2^2$). We obtain the basic inequality ($\mu^{(k)} := \frac{||\mathbf{s}^{(k)}||_2^2}{||\hbar^{(k)}||_2^2}$),

$$f_{best} - f(\xi^{*}) \leq \frac{R^2 + G^2 \sum_{p=0}^{k} \alpha_p^2 (1-\beta^p)^2}{2\sum_{p=0}^{k} \alpha_p \mu^{(p)}} \tag{15}$$

We can conclude from this inequality that the iterate will converge towards the optimal value when step sizes (α_p) are square summable but not summable. However, the theoretical analysis on the convergence speed compared to the simple sub-gradient method is largely inconclusive. However, in Sect. V we experimentally show the effectiveness of the use of the heavy ball scheme with the sub-gradient method for our problem.

IV. Implementation

We summarize the proposed methodology for numerical minimization of Eq. (4) in the listing Algorithm 1. The sub-routine *EdgeMap* returns a list of coordinates which are edge pixels in the provided image. We use Canny Edge detection for computing the edge map of the image. The sub-routine *DistanceTransform* computes the distance transform of the input edge-map as proposed in [12]. Whereas, the sub-routine *InverseProjectAllEdgePixels* implements Eq. (2).

We update the reference frame periodically (typically every 5-10 frames) and compute the relative poses of subsequent frames as outlined in Algorithm 1. These relative poses can be cumulated to obtain the odometry.

As illustrated in the results section, although the basin of convergence of the proposed algorithm is already much larger than the algorithms based on photometric error minimization, we implement our algorithm with image pyramids. We reason that the use of pyramids gives a better initial guess ($\xi^{(0)}$) for the top most level which helps the proposed method converge faster to produce precise results. Thus, we use 4 pyramidal levels with base resolution of 320×240 pixels.

As has been observed by Kerl [2] weighting down large residues can help alleviate the effect of outliers arising due to reflections, occlusions, edge-map misses on the computation of update direction. We use a Laplacian weighting term given by, $W(\upsilon_{e_i}(\xi)) = e^{-\upsilon_{e_i}(\xi)}$.

This solves a weighted least square problem whose sub-gradient can be computed as

$$\hbar^{(k)} = \textstyle\sum_{\forall \mathbf{e}_i} 2W(\upsilon_{e_i}(\xi))\upsilon_{e_i}(\xi^{(k)})\ \mathbf{J}_{e_i}. \tag{16}$$

V. Results

In this section we perform a series of experiments demonstrating the effectiveness of the proposed formulation for camera pose estimation. We compare our method with another direct method on RGB-D data by Kerl *et al.* [2] using

[2] The required projection can be achieved by : $\Gamma(\xi^{(k+1)}) := \frac{\xi^{(k+1)}}{||\xi^{(k+1)}||_2}\varepsilon$

Algorithm 1 RelativePoseEstimation($I_n, I_r, Z_r, \xi^{(0)}$)

```
$E_n$ = EdgeMap($I_n$)
$V_n$ = DistanceTransform($E_n$)
$E_r$ = EdgeMap($I_r$)
${}^r\mathbf{P_i}$ = InverseProjectAllEdgePixels( $E_r, Z_r$ )
$k = 1$
$\hbar^{(0)} = 0$
for $k : 1 \to M$ do
    $\hbar^{(k)}$ = GetSubGradient($V_n, {}^r\mathbf{P_i} \forall\ \mathbf{i}, \xi^{(\mathbf{k-1})}$) (Eq. (8), (10), (16))
    $\mathbf{s^{(k)}} = (1-\beta)\hbar^{(k)} + \beta\mathbf{s^{(k-1)}}$
    $\Delta\xi = \Gamma(-\alpha_k \mathbf{s^{(k)}})$
    if $||\Delta\xi||_2 < \Delta$ then
        break
    else
        $\xi^{(k)} = \xi^{(k)} + \Delta\xi$
        $f(\xi^{(k)}) = GetFunctionValue(V_n, {}^r\mathbf{P_i}\forall\mathbf{i}, \xi^{\mathbf{k}})$ (Eq. (3))
    end if
end for
return $f_{best}^{(M)} = min(f(\xi^{(0)}), f(\xi^{(1)}), \ldots, f(\xi^{(M)}))$
```

their weighted configuration in realtime parameter setting which runs in around 20 ms per frame on average (enough for 30 Hz frame rate) at a resolution 320 × 240 pixels with 6 pyramidal levels. The proposed method is tested with 4 pyramidal levels at a resolution of 320 × 240 and takes about 15 ms to evaluate the relative pose per frame pair.

The TUM-RGBD dataset [13] was used for evaluation. Both methods we tested on a PC with Intel Core i7-2600 CPU (3.4 Ghz) with 16 GB of RAM. We also note that our core system, which evaluates the relative position between a pair of images, is single threaded.

A. *Relative Pose Error (RPE)*

Strum *et al.* [13] proposed RPE to measure the local accuracy for visual odometry approaches, which they defined as,

$$\mathbf{E}_i = \left(\mathbf{Q_i^{-1}}\mathbf{Q}_{\mathbf{i}+\delta}\right)^{\mathbf{-1}}\left(\mathbf{B_i^{-1}}\mathbf{B}_{\mathbf{i}+\delta}\right)$$

where, $\mathbf{Q_1}, \cdots, \mathbf{Q_n} \in \mathbf{SE(3)}$ is the sequence of GT poses and $\mathbf{B_1}, \cdots, \mathbf{B_n} \in \mathbf{SE(3)}$ is the sequence of estimated poses indexed by time instances. δ is the relative time step. They then proposed to evaluate such statistical measures as RMSE, mean, etc. of the translation component of the sequence $\mathbf{E_1}, \cdots, \mathbf{E}_{\mathbf{n}-\delta}$. We report the RMSE values for various values of δ in Table I. We also show the relative pose error (translation component) at each frame for the sequence ‘fr1/desk’ in Fig. 4. As noted by Strum *et al.* [13], usually the comparison by translational errors is sufficient, as rotational errors show up as translational errors when the camera is moved. Thus, we do not evaluate the RPE of rotation components.

The first three sequences are the sequences for which Kerl *et al.* [2] provide results. These sequences are characterized by rather slower motion without sudden jerks. As a result, the consecutive frames (at 30 Hz) are very near to each other and the photometric error based method. Our method is comparable to [2] for these three sequences.

Sequence	D-EA		Kerl *et al.* [2]	
	$\delta = 1$	$\delta = 20$	$\delta = 1$	$\delta = 20$
fr2/desk	0.0324	0.1529	0.0333	0.2217
fr1/desk	0.0289	0.0948	0.0346	0.4286
fr1/desk2	0.0335	0.1818	0.0343	0.3658
fr1/floor	0.0355	0.1988	0.0330	0.3380
fr1/room	0.0353	0.2514	0.0307	0.3399
fr2/desk_with_person	0.0125	0.0594	0.0137	0.1516
fr3/sitting_halfsphere	0.0208	0.1462	0.0181	0.2599
fr2/pioneer_slam2	0.0593	0.4447	0.0847	0.4707

TABLE I: RMSE values of translation component of RPE for various sequences.

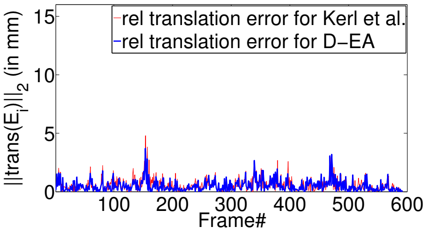


Fig. 4: Translation component of relative pose error at each frame for the sequence ‘fr1/desk’. Best viewed in color.

The next two sequences contain gaps of about 1 second each while in progress. We suspect these might be due to buffering issues in the driver. In real scenarios this might happen.

The next two sequences contain moving objects. Our method and [2] are comparable in performance for these sequences. We attribute it to the exponential weighting terms in the energy formulation.

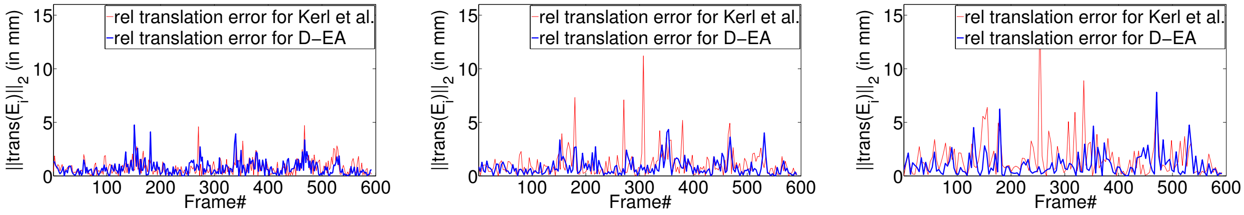
In the last sequence the RGB-D camera is placed on a pioneer ground robot and piloted manually. This sequence contains jerks in the motion. Thus the difference between two consecutive frames is quite large at times. Our method clearly outperforms [2] on this sequence.

B. *Effect of Frame Skipping*

Next we show the robustness of our method for large motion. In this experiment, we supplied our method as well as the method by Kerl *et al.*[2] with alternate frames of the sequence i.e., frame $0, 2, 4, \cdots$ and so on. We also did similar experiments by skipping 3 and 4 frames as well. We plot the translation component of the relative pose error at each frame in the sequence ‘fr1/desk’ (20.24 sec, or 593 frames). See Fig. 5. We observe that the relative pose error for our method does not vary much even when supplied 1 frame of every 4 frames, whereas the relative pose error goes on increasing for [2]. One can think of this property of our method as being robust to fast motion. Note that the runtime configuration was kept the same for all experiments in this subsection.

C. *Demonstration of Large Convergence Basin*

We select two frames from the *fr1/rpy* sequence which are about 160 ms apart (5 frames) and show the progress of the sub-gradient method iteration wise in Fig. 1. In the



(a) Processing only frames $0,2,4,6\cdots$

(b) Processing only frames $0,3,6,9\cdots$

(c) Processing only frames $0,4,8,12\cdots$

Fig. 5: Robustness for large motions. Relative pose estimation of ‘fr1/desk’ by skipping frames. Best viewed in color.

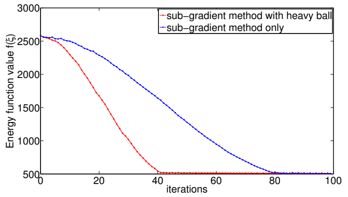


Fig. 6: Comparison of the sub-gradient method and sub-gradient method with heavy ball acceleration on the frame pair of Fig. 1.

illustration there are no pyramids in use and the initial estimate $(\xi^{(0)})$ for sub-gradient method was set as identity. We can clearly see the influence of edges extend which results in a much larger convergence basin for the proposed method.

D. Effect of Heavy Ball

We demonstrate the effectiveness of the altered gradient direction (heavy ball method). We compare the energy progress for the example reference-current frames shown in Fig. 1. The comparison of the simple sub-gradient method and the modified sub-gradient method for number of iterations to convergence is shown in Fig. 6.

VI. Conclusion

In this paper, we proposed a direct (feature-less) approach for visual 6-DOF pose estimation with the energy function based on the distance transform applied to an RGB-D camera. We demonstrate with experiments a much larger convergence basin for our method when compared to other direct approaches. We address the issue of non-differentiability of the energy function and thus make the case against the use of Gauss-Newton methods for non-differentiable functions in a strict mathematical sense. Thus, we utilize the sub-gradient method for numerical optimization, which is a class of methods to handle non-differentiable functions. We also analyze the convergence of the modified sub-gradient method (that we use) for our energy function. Our method is comparable to previous method ([2]) for sequences with slower motion and we show with experiments the robustness of our method for sequences with fast motion. This robustness is attributed to our energy function which is based on the minimization of reprojected distances rather than on photo-consistency.

References

[1] J. Engel, J. Sturm, and D. Cremers, “Camera-based navigation of a low-cost quadrocopter,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2815–2821, IEEE, 2012.

[2] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras.,” in *ICRA*, pp. 3748–3754, IEEE, 2013.

[3] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[4] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle Adjustment - A Modern Synthesis,” in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV ’99, (London, UK, UK), pp. 298–372, Springer-Verlag, 2000.

[5] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an RGB-D camera,” in *International Symposium on Robotics Research (ISRR)*, pp. 1–16, 2011.

[6] I. Dryanovski, R. G. Valenti, and J. Xiao, “Fast visual odometry and mapping from RGB-D data,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2305–2310, IEEE, 2013.

[7] 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, IEEE, 2011.

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

[9] J. Stückler and S. Behnke, “Model Learning and Real-Time Tracking Using Multi-Resolution Surfel Maps.,” in *AAAI*, 2012.

[10] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.

[11] A. W. Fitzgibbon, “Robust registration of 2D and 3D point sets,” *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, “Distance Transforms of Sampled Functions.,” *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A Benchmark for the Evaluation of RGB-D SLAM Systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[14] J. H. Manton, “Optimization algorithms exploiting unitary constraints.,” *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 635–650, 2002.

[15] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Ann Arbor: CRC Press, 1994.

[16] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient methods,” *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[17] R. T. Rockafellar, *Convex analysis*. Princeton Mathematical Series, Princeton, N. J.: Princeton University Press, 1970.

[18] Y. Nesterov, “Smooth minimization of non-smooth functions,” *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[19] Y. Nesterov, “Gradient methods for minimizing composite functions,” *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.