Direct RGB-D Visual Odometry via Gaussian Gradient Features

Zhigang Yao[1]

Abstract— **Point-wise tracking in either geometric or photometric paradigm greatly relied on pixel saliency in a local region or a sub-window, where the former requires reliable feature detection and the latter is prone to suffer from redundancies, especially for dense tracking. To estimate a rigid-body motion, in addition, target points evenly distributed in the image plane are preferred due to transformations or distortions in projections. Inspired by existing studies, this paper gives a direct Visual Odometry (VO) of RGB-D cameras by using features measured by low-order Gaussian derivative functions such as Gaussian gradient operator which is commonly known as a band pass filter enables removing significantly image spatial redundancies whereas could remain certain correlations between adjacent pixels. By filtering the image with such a local contrast feature metric, it improves further the possibility for sampling more reliable points from scenarios that are lack of structure or texture and is beneficial to continuous tracking. Without elaborating optimization or hybrid Jacobian computation, the proposed approach reaches relatively acceptable performance with a globally heuristic framework built on the general coarse-to-fine and inverse compositional estimation. To evaluate its validity in direct VO, comparative study of this work is conducted via several experimental results on a group of TUM datasets.**

I. Introduction

Studies on VO have almost been half a century long, and a couple of decades since it was formally coined in early 2000s [1]. Plenty of work, in years gone by, had been done for making solutions relatively more robust, more precise and faster in motion estimation of the robot or any form of the agent when sensing the environment with a single camera or multi-one. Researches of this issue can be divided into several categories in terms of types of visual sensing devices, theoretical background of solution pipeline and some other considerations.

Up to now, monocular [2] or binocular [3] vision is the most general sensing mode and is the original hardware form in VO studies [1]. Both is involved in triangulation for estimating depth of the 2D projection of a space point. Cameras based on structured light [4] came afterwards somehow contributed a solution for making depth known in sensing stage even though the measuring range is basically limited. RGB-D camera such as Microsoft Kinect, typically, is driven by the structured light technique and is capable of generating both RGB and depth signals in time domain. This kind of camera is widely employed in research filed [2], [5]–[7] due to its advantages in cost and usability considerations. In this regard, RGB-D is also the sensing mode this paper focuses on. Event camera came out as a new sensing option shortly after the booming of researches on RGB-D. Event camera is inspired by bio-vision's reflect on dynamic changing of the scene. In comparison with traditional intensity signal, it gives in pixel-wise asynchronous measurements of brightness changing of the target scene [8]. Capturing only motions in the scene enables it owning high dynamic range, running at relatively high speed, reducing motion blur issues and having some other beneficial properties. Correspondingly, novel imaging scheme requires as well unconventional solutions to take advantage of its potential features. In addition, there are also some other special perception tools such as the panoramic vision [9] generally equipped with multiple cameras. Multi-frame conducted by panorama cameras are basically of different scene locations around the camera rig giving richer information than the conventional and requiring to deal with more details for multi-camera collaboration in return [10].

Visual sensing devices, by far, are variant but outputs are basically signals of 2D domain. It allows researches sharing different computing pipelines or methodologies but what behind them is identical to each other. Existing studies have theoretically been supported by geometric constraints and vector filed continuity since VO issues were solved via optimizing a posterior likelihood function. The former is constructed by point-wise data association and is known as geometry consistency, while the latter is established via constant brightness assumption and is referred to as photometric consistency.

Point-wise data association is basically implemented via feature matching between the reference image and the one with parallax, with which epipolar constraint is found between the two yielding a re-projection error. One of the state of the art work of matching-based approaches is ORB-SLAM [3] built on the PTAM [11]. It employs ORB feature points to carry out relatively fast matching in comparison with other point features. Combined with loop closure detection and optimizations in both local and global graphs, ORB-SLAM shows highly accurate camera trajectories in using only visual sensors. Up until today, the main limit for geometric methods is the time consumption which currently is unavoidable for feature detection and matching. This prerequisite may failed to establish in case the scene is lack of salient regions or sharp textures. Once the triangulation or re-projection error is built by using such a data association, depth of space point or camera pose can be estimated via 2D-2D decomposition, 3D-2D PnP and 3D-3D ICP optimizations or bundle adjustment (BA) [1].

On the other hand, the optical flow [12] defined on continuous vector filed created data association in an other

*This work was not supported by any organization.

[1]Author Name is with the Affiliation Name, Post code, City name, Country. aopawy@gmail.com

Finished on 15/12/2022.
Uploaded to arXiv.org.

way, i.e. the direct visual odometry relied on photometric error minimization. The DVO [5] is one of the representative direct methods for RGB-D cameras. To enrich information available for motion estimation, they use full image and a depth image in aligning consecutive color images. What more meaningful in their paper is the probabilistic investigation of the photometric residuals contributing a relative more robust weighting function via student t-distribution which I followed as well in this paper.

Existing studies of RGB-D VOs, so far, have been embracing various possibilities such as joint minimizations considering both geometric and photometric consistencies [13], multiple feature types [14] and learning-based approaches [15] that mainly focused over recent years.

What proposed in this paper focuses mainly on point-based VO with the lightweight RGB-D camera. The main highlights of this study can be summarized as two: 1) Gaussian kernel is the only metric for sampling points locally to track, 2) which reaches a compromise between reliable estimation and efficient computation speeding up notably the runtime of a direct VO. Additionally, extended evaluation is also performed on some specific challenging scenarios such as scenes with little to no visible structure and texture.

By the end of this brief introduction, section II will give a short discussion on some related work of this paper. Problem notations is given in section III, and the Gaussian feature metric is described concisely in section IV. Section V details the experimental study with few reference VOs, and section VI concludes the paper.

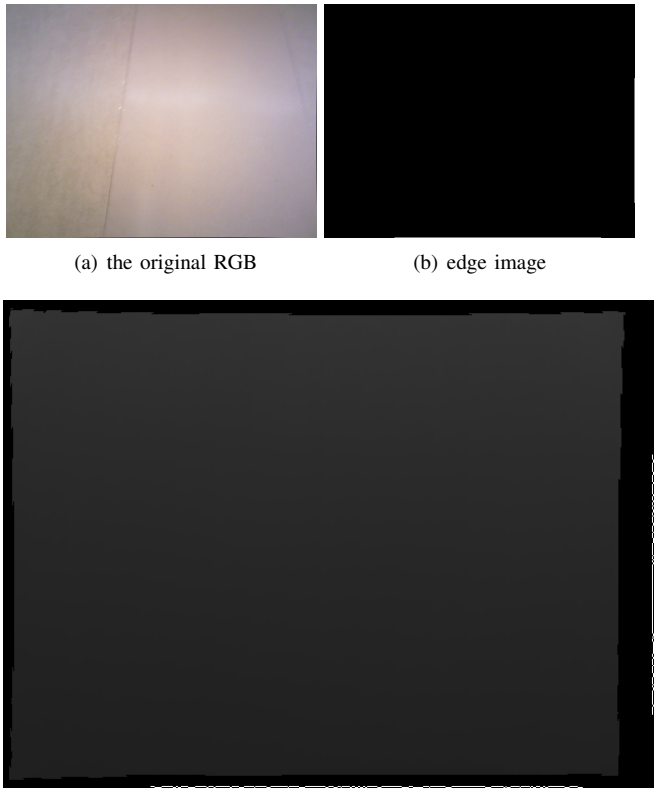
II. Related work

Typically, computing gradient is a representative application of low-order differential operator in image processing. Examples of processing images in gradient domain or computing features via gradients can largely be found in literatures, including, but not limited to image fusion, segmentation or structures detection. It is also introduced into VO problems for advantages such as relatively robust to illumination variation [16].

The significance of gradients, however, is fully characterized by how it is involved in optimizations. Existing studies recognized that only pixels with intensity gradient are beneficial to motion estimation. LSD-SLAM [7] considers mainly depth estimation for pixels with sufficiently large intensity gradients. Later in SVO [2], mapping is in a semi-dense manner via joint optimization of high gradient intensities and the samples normal to the edge lied on demonstrating how phase or orientation plays in the optimization.

Being a representative of gradient-characterized feature, existing studies have investigated more on edges or local edgelets in either color space or depth domain [17]–[19]. Similar to matching-based ICP but in a direct way, with assumptions of both full calibration and synchronization of the RGB and depth, searching with variant distance metrics in a local region the nearest neighbour of the one projected from the reference image can give a good initial guess of correspondences [19]. Alternatively, locations of features detected or sampled in the current frame can give an initialization of the inverse compositional optimization heuristically [20].

On the other hand, limit of edge-based approaches is simply the availability of the feature or structure, i.e. it may fail to extract an edge due to improper threshold or in some unideal or challenging cases. Here is an example from the TUM sequence, see figure 1.



(a) the original RGB (b) edge image

(c) depth image with edges on it

Fig. 1: Image (a) is the 1341840843.310587.png of the fr3/nostructure_notexture_far [21], followed by an edge image generated with the canny detector employed in [20] and an illustration of these edges on the depth image synchronized (depth/1341840843.311514.png).

There are two edge patterns in image 1(b) locating respectively on the middle bottom and the right border of the image[1]. These two line structures, visually, seem to be imaged by lens borders of the camera rather than the real edge projected from the scenario. It can be found via depth values of these edges, see figure 1(c).

In regard to these, considering concurrently properties of both rigid-body and projections of 3D points, particles scattered on the 2D plane with relatively high gradient magnitude are preferred to be extracted for estimating camera motions. To tackle this, low-order isotropic Gaussian derivative operators are used in this study due to [22]: 1) Gaussian operators could remove a large amount of image spatial

[1]One can zoom the .pdf to or so 330% for clear view.

redundancies while certain correlations between neighboring pixels would remain; 2) Gaussian response measures local luminance changing strength benefiting down-sampling image in a local region; 3) the principal components of natural images strongly resemble the directional derivatives of 2D Gaussian functions. Inspired by existing study [22], this paper detect points to track in terms of responses of the Gaussian partial derivatives, as detailed in IV.

III. Notations

The direct method, as discussed in previous sections, explicitly establishes promising connections of pixels in time domain and is not related to any preprocessing of point feature matching. This section gives a straightforward formulation of its minimization expression.

A. Formulation

Giving a 3D point $\boldsymbol{P} = [X, \quad Y, \quad Z]^{\mathrm{T}} \in \mathbb{R}^3$, its projections in pixel frames at time $k-1$ and k are respectively formulated by

$$\begin{aligned} \boldsymbol{p}_{k-1} &= [u, \quad v]^{\mathrm{T}} \\ &= \frac{1}{Z}\boldsymbol{K}\boldsymbol{P} \end{aligned} \tag{1}$$

and

$$\boldsymbol{p}_k = \frac{1}{Z'}\boldsymbol{K}\mathbf{T}\boldsymbol{P}, \tag{2}$$

where matrix $\boldsymbol{K}$ is the camera intrinsic defined by focal length f_x, f_y and optical center c_x and c_y, and $\boldsymbol{T} \in SE(3)$ is the transformation matrix indicating the rigid-body motion of the camera which is defined as

$$\begin{aligned} \mathbf{T}(\boldsymbol{\xi}, \boldsymbol{P}) &= exp(\boldsymbol{\xi}^{\wedge})\boldsymbol{P} \\ &= exp\left(\begin{bmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}\right)\boldsymbol{P} \\ &= \mathbf{R}\boldsymbol{P} + \mathbf{t}. \end{aligned} \tag{3}$$

In equation (3), $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ are simply rotation and translation components decomposed from the homogeneous matrix $\mathbf{T}$. The generalized velocity $\boldsymbol{\xi} \in \mathfrak{se}(3)$ is the Lie algebra of matrix $\mathbf{T}$ and is given by

$$\begin{aligned} \boldsymbol{\xi} &= [\boldsymbol{\phi}, \quad \boldsymbol{\rho}]^{\mathrm{T}} \\ &= [\upsilon_1, \quad \upsilon_2, \quad \upsilon_3, \quad \upsilon_4, \quad \upsilon_5, \quad \upsilon_6]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where $\boldsymbol{\phi} \in \mathfrak{so}(3)$ and $\boldsymbol{\rho} \in \mathbb{R}^3$ are Lie algebra w.r.t rotation and translation velocity, respectively.

Let $\mathbf{T}\boldsymbol{P} = \mathbf{T}(\boldsymbol{\xi}, \boldsymbol{P}) = \boldsymbol{P}' = [X', Y', Z']^{\mathrm{T}}$, equation (2) is further formulated as

$$\begin{aligned} \pi(\mathbf{T}(\boldsymbol{\xi}, \boldsymbol{P})) &= [\hat{u}, \quad \hat{v}]^{\mathrm{T}} \\ &= \frac{1}{Z'}\boldsymbol{K}\boldsymbol{P}' \\ &= \boldsymbol{p}_k. \end{aligned} \tag{5}$$

The inverse projection of equation (5) is then defined as

$$\begin{aligned} \pi^{-1}(\boldsymbol{p}_k, Z(\boldsymbol{p}_k)) &= \boldsymbol{K}^{-1}\left[Z'\boldsymbol{p}_k \quad Z'\right]^{\mathrm{T}} \\ &= \begin{bmatrix} Z'\frac{\hat{u}-c_x}{f_x} \\ Z'\frac{\hat{v}-c_y}{f_y} \\ Z' \end{bmatrix}. \end{aligned} \tag{6}$$

From equation (5) and (6), the warping function or the observation of photometric consistency can be denoted as

$$w(\boldsymbol{\xi}, \boldsymbol{p}) = \pi\left(\mathbf{T}\left(\boldsymbol{\xi}, \pi^{-1}\left(\boldsymbol{p}, Z(\boldsymbol{p})\right)\right)\right) \tag{7}$$

where $\boldsymbol{p}$ is simply a general representation of pixel coordinates.

Supposed that having an image $\mathcal{I}_k$ of $N \times N$ at time k and a pixel locates at $\boldsymbol{p}_i \in \mathcal{I}_k$, one can get its location in image $\mathcal{I}_{k+1}$ with equation (7). According to optical flow definition, pixel intensity $\mathcal{I}_k(\boldsymbol{p}_i)$ keeps constant with in a small time interval. Based upon this assumption, the photometric residual of a single channel image can be defined as

$$\begin{aligned} e_i &= \mathcal{I}_k(\boldsymbol{p}_i) - \mathcal{I}_{k+1}\left(\pi\left(\mathbf{T}\left(\boldsymbol{\xi}, \pi^{-1}\left(\boldsymbol{p}_i, Z(\boldsymbol{p}_i)\right)\right)\right)\right) \\ &= \frac{1}{Z_i}\boldsymbol{K}\boldsymbol{P}_i - \frac{1}{Z_i'}\boldsymbol{K}\mathbf{T}\boldsymbol{P}_i \\ &= \frac{1}{Z_i}\boldsymbol{K}\boldsymbol{P}_i - \frac{1}{Z_i'}\boldsymbol{K}\,exp(\boldsymbol{\xi}^{\wedge})\boldsymbol{P}_i. \end{aligned} \tag{8}$$

The residual given by equation (8) is the function of $\mathbf{T}$ or its Lie algebra $\boldsymbol{\xi}$, i.e. $e(\mathbf{T}) = e\left(exp\left(\boldsymbol{\xi}^{\wedge}\right)\right)$. One step further, the mahalanobis distance for the entire image is given by

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = \sum_{i=0}^{N^2-1}\left(e_i\left(\boldsymbol{\xi}\right)\right)^2 \tag{9}$$

where $e_i \sim N(0, 1)$ as assumed.

Finally, a relatively optimal $\mathbf{T}$ or $\boldsymbol{\xi}$ that maximizes the posterior probability $p(\boldsymbol{\xi}|\boldsymbol{e})$ [5] can be solved via the minimization of equation (9), i.e.

$$\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} \boldsymbol{J}(\boldsymbol{\xi}) = \arg\min_{\boldsymbol{\xi}} \sum_{i=0}^{N^2-1}\left(e_i\left(\boldsymbol{\xi}\right)\right)^2. \tag{10}$$

To solve the non-linear equation (10), the Gauss-Newton or its variant Levenberg-Marquardt is typically employed.

B. Scale estimator

Existing study [5] has already found that the normal distribution does not properly fit the residuals. Instead, student t-distribution is proved more suitable to model the errors. By following this investigation, the minimization of residuals is transformed to as

$$\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} \sum_{i=0}^{N^2-1} w_i\left(e_i\right)\left(e_i\left(\boldsymbol{\xi}\right)\right)^2. \tag{11}$$

In equation (11), $w_i\left(e_i\right)$ is the weight of each e_i derived from the t-distribution

$$\begin{aligned} w_i\left(e_i\right) &= \frac{log\, p\left(e_i\right)}{\partial e_i}\frac{1}{e_i} \\ &= \frac{\nu+1}{\nu + \left(\frac{e_i-\mu}{\sigma}\right)^2} \end{aligned} \tag{12}$$

where μ and σ^2 are respectively the mean and the variance of the t-distributed data, and ν is the degrees-of-freedom of the distribution.

By applying weight function (12), outliers with low probabilities can be effectively filtered out. A typical configuration of the mean μ in existing studies is either set to be the median of all intensity errors or zero which is adopted in this paper. The standard deviation can be iteratively solved or approximated using the Median Absolute Deviation (MAD)

$$\sigma = 1.48\, med\left(\{|e_i - \mu|\}\right), \tag{13}$$

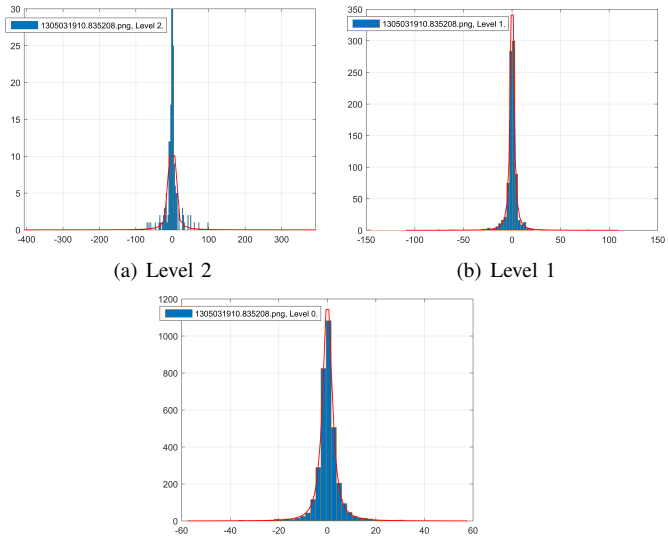
where $med\,(\cdot)$ is the median operator.

Equation (13) calculates the scale via a constant normalizer which is an empirical value and might be affected by camera parameters such as auto-exposure for some cases [23]. Alternatively, the variance can be iteratively calculated via

$$\sigma^2 = \frac{1}{n}\sum_i e_i^2 \frac{\nu+1}{\nu + \left(\frac{e_i}{\sigma}\right)^2}, \tag{14}$$

where $n = N^2 - 1$ or the total number of residuals.

Basically, equation (14) can be solved within 4 or 5 iterations. In this paper, both scale estimators are investigated for different scenarios. Alternatively, prior to this, a fitting analysis is done to determine that 2 layers are employed for the coarse-to-fine estimation, see figure 2.



(a) Level 2 (b) Level 1

(c) Level 0

Fig. 2: Fitting analysis.

Figure 2(a) and figure 2(c) are respectively the smallest and the original size images of the pyramid, where figure 2(a) illustrates that the second down-sampling of the original image is not fitted as well as the larger so that only 2 layers are used for pose computations. This choice might lead to weak performance while dealing with large motion. However, the fact is that it's hard to make the fitting absolutely good for all scenarios with a constant number of layers or parameters due to natural variations of the scenes. In this paper, 2-layer analysis is simply used for demonstrating the proposed approach. Readers interested in adaptive fitting are recommended referring to [24], [25].

IV. Plain Local Point Selection

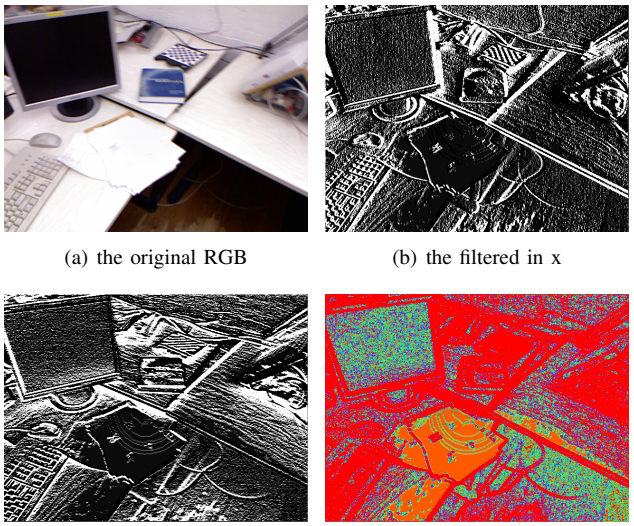
Following simply the blind image quality assessment approach [22], the Gaussian gradient magnitude (GGM)

$$\mathcal{G}_{\mathcal{I}} = \sqrt{\left[\mathcal{I} \otimes \boldsymbol{h}_x\right]^2 + \left[\mathcal{I} \otimes \boldsymbol{h}_y\right]^2} \tag{15}$$

is calculated by using the Gaussian partial derivative filters

$$\begin{aligned} \boldsymbol{h}_d\,(x, y|\sigma) &= \frac{\partial}{\partial d}\boldsymbol{g}(x, y|\sigma) \\ &= -\frac{1}{2\pi\sigma^2}\frac{d}{\sigma^2}\, exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \end{aligned} \tag{16}$$

where "$\otimes$" indicates linear convolution, $\boldsymbol{h}_d, d \in x, y$ represents the Gaussian partial derivative filter along the orientation of x or y respectively derived from the isotropic Gaussian function $\boldsymbol{g}\,(x, y|\sigma) = \frac{1}{2\pi\sigma^2}\, exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$ in which σ is the standard deviation [2]. An original image and filter outputs of this image are displayed in figure 3.



(a) the original RGB (b) the filtered in x

(c) the filtered in y (d) the magnitude in HSV

Fig. 3: Image (a) is the 1305031453.359684.png of the fr1/desk [21], followed by response images in x (b) and y (c) directions. Image (d) is the gradient magnitude in HSV space.

Further, the Laplacian of Gaussian (LOG) filter is formulated by a second-order partial derivatives of the Gaussian function

$$\begin{aligned} \boldsymbol{h}_{LOG}\,(x, y|\sigma) &= \frac{\partial^2}{\partial x^2}\boldsymbol{g}\,(x, y|\sigma) + \frac{\partial^2}{\partial y^2}\boldsymbol{g}\,(x, y|\sigma) \\ &= \frac{1}{2\pi\sigma^2}\frac{x^2+y^2-2\sigma^2}{\sigma^4}\, exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \end{aligned} \tag{17}$$

The LOG filter is also isotropic and responds to intensity contrast in a small spatial neighborhood. These are beneficial to, in a local cell, sample points that are locally salient and relatively robust to motions within a small time interval.

[2]Note that it is not the same item as the one in weighting function (12).

To select points evenly scattered in pixel plane, a block-wise down sampling is applied onto the entire image. The GGM or LOG is at first calculated via equation (15) or (17) for an image followed by dividing the image into blocks of 8×8. The pixel with the largest quantitative value in each block is picked as a representative or salient point to track, while blocks full with same scalar values are ignored.

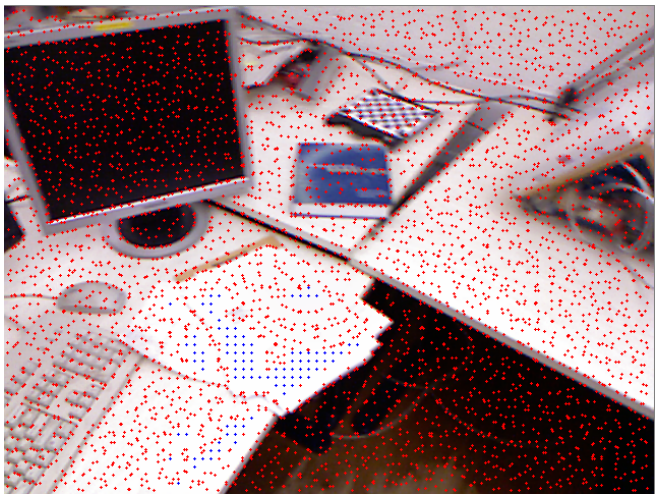


Fig. 4: Illustration of the RGB image with sampling particles on it.

Points in red displayed in figure 4 are candidates to track, while the blue indicate either magnitudes of 0 or a local region with no variation.

V. Experiments

To validate the proposed approach, a comparative study is conducted on a set of TUM datasets [21] with few open-source VO implementations for reference.

A. Protocol

Built on existing studies [5], [20], [22], the engineering implementation of the proposed approach is programmed via C++ and would be as well publicly available at `givenlaterviagithub` to guarantee re-implementation. Experimental platform is a Win-7 laptop computer (lenovo E460) with an Intel Core i5-6200 CPU running at 2.30GHz, 4G RAM and an Intel HD Graphic 520 graphic unit. The integrated development environment employed is Visual Studio 2013.

Configuration details of the TUM datasets employed in the experiments are given in table I, in which the "long_office", "**ns**" and "**nt**" indicate respectively "long_office_household", "no structure" and "no texture" simply for editing the paper. The associations is the number of RGB and depth image pairs synchronized. Each VO approach then outputs a trajectory consisted of poses of the same amount.

Table I involves 4 categories of the TUM dataset, where the first 3 groups in the table are full of category Testing and Debugging, Handheld SLAM and Robot SLAM. The last group in the table is from category Structure vs. Texture of the TUM dataset considering mainly scenarios lacking of both structure and texture, where "**ns_nt**_far" is relatively short in length, and "**ns_nt**_near_withloop" is a relatively long length loop-closure frame sequence of approximately 11.739 m containing intentionally little to no visible structure and texture. Edge-based approach has already demonstrated validity on scenario that is rich of texture such as "fr3/nostructure_texture_near_withloop" [19] on which the EdgeVO also performs the best and GGM_TD the second among all approaches applied in this paper. Therefore, these 2 datasets are chosen to check if the proposed approach is able to successfully conduct a result. All sequences of fr1 and fr2 in table I are captured with a Microsoft Kinect-v1 producing asynchronously RGB and depth images of 640×480. In table I, sequences of fr3 consisted of images in the same size are recorded via Asus Xtion sensor according to the description of TUM dataset. TUM benchmark also gives groundtruth trajectories generated via a multi-camera motion capture system. Readers are recommended to refer [21] for more details about different data categories.

TABLE I: TUM datasets involved in the experiment.

Sequence	Duration (s)	Length (m)	Associations
fr1/xyz	30.09s	7.112m	792
fr1/rpy	27.67s	1.664m	694
fr2/xyz	122.74s	7.029m	3615
fr2/rpy	109.97s	1.506m	3221
fr1/360	28.69s	5.818m	744
fr1/floor	49.87s	12.569m	1227
fr1/desk	23.40s	9.263m	573
fr1/desk2	24.86s	10.161m	620
fr1/room	48.90s	15.989m	1352
fr2/360_hemisphere	91.48s	14.773m	2670
fr2/360_kidnap	48.04s	14.286m	1413
fr2/desk	99.36s	18.880m	2893
fr2/large_no_loop	112.37s	26.086m	3299
fr2/large_with_loop	173.19s	39.111m	5057
fr3/long_office	87.09s	21.455m	2488
fr2/pioneer_360	72.75s	16.118m	830
fr2/pioneer_slam	155.72s	40.380m	2198
fr2/pioneer_slam2	115.63s	21.735m	1645
fr2/pioneer_slam3	111.91s	18.135m	2266
fr3/**ns_nt**_far	15.79s	2.897m	455
fr3/**ns_nt**_near_withloop	37.74s	11.739m	1093

Reference VO algorithms in experiments are simply EdgeVO [20] and DVO, i.e. one sparse and one dense approach. For DVO, a fork of dvo core [26] without VTK and ROS dependencies are used in this paper. In the last of the experiment, evaluation tools of the TUM benchmark are used for validation by computing the Root Mean Squared Error (RMSE) values of both the Relative Pose Error (RPE) and the Absolute Pose Error (ATE) of the trajectory. Further, trajectory visualization in this paper relies on the evo package [27].

B. Results

In experiments, both trajectory and running time elapsed for calculating the whole trajectory are recorded on each dataset where the latter is involved in figuring out the frame rate, i.e. the Frames Per Second (FPS). Firstly, the RMSE

TABLE II: RMSE values of RPE and ATE. Cells displayed via **value** and **value** are respectively the minimum and the 2^{nd} to minimum values. Cells indicated by ~~failed~~ mean that the approach failed to conduct a result.

Sequence	Translation RMSE of RPE (m)					RMSE of ATE (m)				
	dvo_core	EdgeVO	GGM_MAD	GGM_TD	LOG_MAD	dvo_core	EdgeVO	GGM_MAD	GGM_TD	LOG_MAD
fr1/xyz	0.135952	**0.068184**	0.069516	0.074873	**0.063192**	0.093377	**0.046531**	0.048101	0.050951	**0.04551**
fr1/rpy	0.102993	0.092403	**0.074861**	**0.090192**	0.117354	0.069933	0.061494	**0.050311**	**0.057253**	0.071287
fr2/xyz	0.360045	**0.080887**	0.257913	0.179553	**0.104721**	0.247475	**0.054867**	0.173528	0.119671	**0.066893**
fr2/rpy	0.374406	**0.073716**	0.137555	**0.109836**	0.125108	0.254608	**0.051298**	0.087257	**0.069125**	0.08634
fr1/360	0.445659	0.435583	**0.294198**	0.350412	**0.318093**	0.296239	0.290507	**0.188482**	0.225164	**0.2072**
fr1/floor	0.340762	**0.321855**	0.34455	0.395078	**0.331876**	0.240302	**0.212191**	0.23332	0.274262	**0.225152**
fr1/desk	**0.109197**	1.279893	**0.401341**	0.885872	2.042864	**0.060523**	0.930582	**0.268203**	0.597989	1.354174
fr1/desk2	**0.14537**	0.305618	0.225425	0.186796	**0.161218**	**0.079268**	0.209148	0.148419	0.121983	**0.104473**
fr1/room	0.476805	0.407284	**0.389037**	**0.368439**	0.409254	0.337426	0.279021	**0.274572**	**0.257675**	0.287909
fr2/360_hemisphere	6.267317	**1.318145**	**1.214921**	2.913162	1.882698	3.93945	0.86538	**0.639938**	1.293073	**0.776003**
fr2/360_kidnap	**2.308621**	~~failed~~	~~failed~~	~~failed~~	~~failed~~	**1.246565**	~~failed~~	~~failed~~	~~failed~~	~~failed~~
fr2/desk	0.660515	0.252002	**0.229928**	0.248258	**0.227475**	0.426563	0.166819	**0.145306**	0.163146	**0.132406**
fr2/large_no_loop	**2.188121**	6.110764	13.132706	11.869643	**0.967221**	0.800331	**0.794389**	8.785756	7.978082	**0.410647**
fr2/large_with_loop	6.572068	**1.95654**	**2.191336**	4.87555	2.706425	4.08959	**1.107616**	**0.920794**	2.526663	1.261545
fr3/long_office_household	0.508268	**0.211991**	0.337617	0.431788	**0.290999**	0.342806	**0.111337**	**0.162122**	0.240976	0.176762
fr2/pioneer_360	**1.683595**	~~failed~~	5.379506	4.427134	**3.891514**	**0.971953**	~~failed~~	3.196815	2.660321	**2.528023**
fr2/pioneer_slam	**4.20158**	~~failed~~	**2.445337**	6.056413	5.326067	**2.641875**	~~failed~~	**1.525171**	3.974027	3.256649
fr2/pioneer_slam2	**2.171842**	~~failed~~	5.259017	**2.772176**	6.232297	**1.10388**	~~failed~~	3.082678	**1.945854**	3.3591
fr2/pioneer_slam3	2.296052	~~failed~~	2.667803	**2.051939**	**2.10509**	1.31667	~~failed~~	1.727229	**1.247296**	**1.279341**
fr3/**ns_nt**_far	0.787483	~~failed~~	0.77997	**0.634789**	**0.482448**	0.426217	~~failed~~	0.396676	**0.3228**	**0.219453**
fr3/**ns_nt**_near_withloop	**1.351384**	~~failed~~	1.618603	1.506444	**1.331465**	**0.819074**	~~failed~~	**0.847782**	0.855994	0.867309

values of the translation drift and the ATE are calculated via TUM evaluation tool, see table II.

Table II demonstrates a quantitative comparison between various methods. The translational drift and the ATE are given first simply because they are basically consistency with each other except for "fr2/360_hemisphere", "fr2/large_no_loop", "fr3/long_office_household" and the last two datasets. The "fr2/360_hemisphere" and "fr2/360_kidnap" are relatively challenging scenarios where the former make the camera a 360 pivot-turn, and the latter is similar and is covered several times. Only GGM_MAD conducted a circular trajectory in x-y plane on the "fr2/360_hemisphere" but is overall far from the groundtruth, see figure 5, while only **dvo_core** obtain a result on the "fr2/360_kidnap" demonstrating the validity of the reference. The groundtruth trajectory of

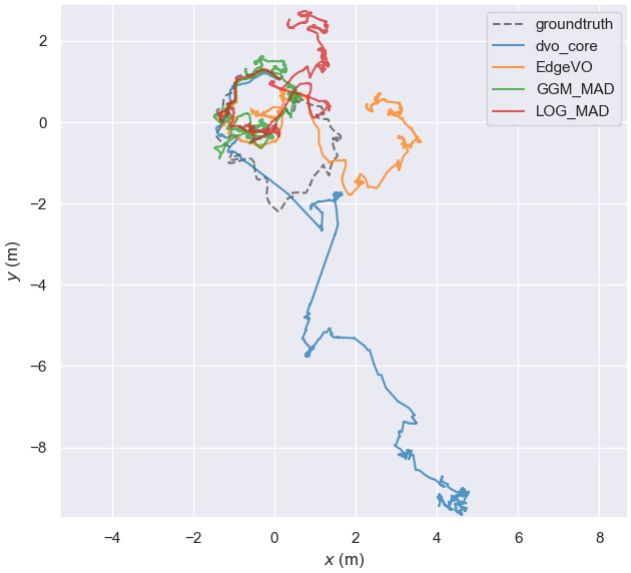
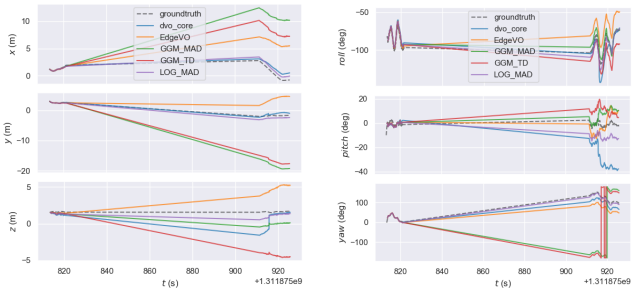
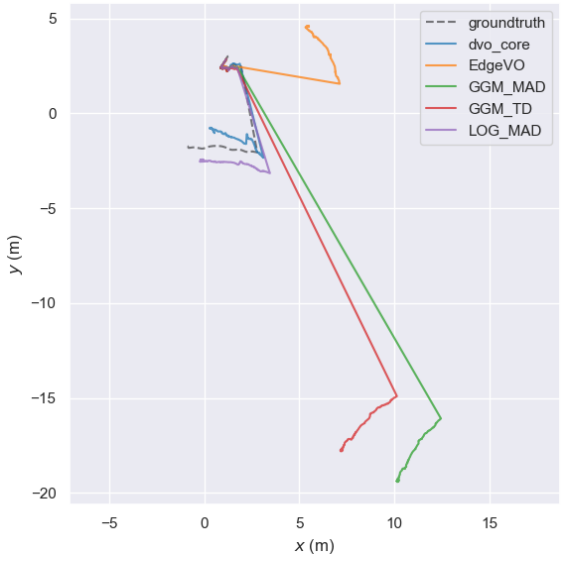


Fig. 5: Trajectories on "fr2/360_hemisphere".



(a) xyz view (b) rpy view



(c) Trajectory

Fig. 6: Trajectories on fr2/large_no_loop.

"fr2/large_no_loop" is 26 meters long, on which the LOG feature got the translational drift less than 1 m. This result is also validated via "xyz" and "rpy" analyses, see figure 6. For the last two datasets related to structure and texture,

TABLE III: Rotation RMSE values of RPE (deg)

Sequence	Rotation RMSE of RPE (deg)				
	dvo_core	**EdgeVO**	**GGM_MAD**	**GGM_TD**	**LOG_MAD**
fr1/xyz	7.427633	3.51981	**2.58761**	**3.244429**	4.796773
fr1/rpy	9.651925	**6.509513**	**8.57167**	8.805735	9.866002
fr2/xyz	11.597177	**2.865161**	15.829005	9.709086	**5.082213**
fr2/rpy	8.479948	**3.27682**	10.418238	8.225424	**2.873666**
fr1/360	14.350917	16.429954	**10.691339**	**11.168372**	11.216206
fr1/floor	**6.694833**	**8.124217**	8.762889	10.438528	9.13847
fr1/desk	**5.221469**	117.66927	**13.847863**	23.952766	60.795276
fr1/desk2	8.244604	9.923616	6.188858	**5.812032**	**6.144647**
fr1/room	19.055491	16.589381	**14.755445**	**14.427663**	15.742407
fr2/360_hemisphere	57.805863	12.438128	**8.99062**	15.98977	**10.86044**
fr2/360_kidnap	**55.261693**	~~failed~~	~~failed~~	~~failed~~	~~failed~~
fr2/desk	20.159219	6.201685	**5.195663**	**5.382411**	5.979194
fr2/large_no_loop	**28.561812**	38.291841	44.566058	38.335487	**6.962546**
fr2/large_with_loop	25.015993	32.079851	**19.841802**	29.884188	**13.859673**
fr3/long_office_household	12.617962	**5.901304**	10.404824	11.226037	**7.554759**
fr2/pioneer_360	**33.589914**	~~failed~~	55.886116	**51.262256**	75.512202
fr2/pioneer_slam	**66.853673**	~~failed~~	88.183387	**71.818741**	97.677631
fr2/pioneer_slam2	**53.922031**	~~failed~~	106.984782	**60.04889**	109.205836
fr2/pioneer_slam3	49.193033	~~failed~~	56.071329	**30.644957**	**34.518307**
fr3/**ns_nt**_far	28.176971	~~failed~~	26.027608	**23.098693**	**18.547029**
fr3/**ns_nt**_near_withloop	**40.636638**	~~failed~~	57.463985	53.247918	**38.259102**

each approach conducts a relatively large error. Approaches shown inconsistent ranking of the RPE and ATE possess results close to each other in either RPE or ATE which can be found as well in the RMSE of the rotational drift given in table III.

The sequences of robot SLAM are also challenging for all the approaches. Datasets of this category are relatively long where fr2/pioneer_slam is the longest of all. Each approach with a result generated can only follow a few meters at the beginning part of the groundtruth. To demonstrate the result more generally, the x-y plot of few trajectories of different lengths and categories are given in figure 7[3] where four of them follow relatively well the groundtruth and the others don't. Figure 7(e) and 7(f) give respectively trajectories of "fr2/pioneer_slam" and "fr2/pioneer_slam3", from which one can clearly found that only few meters of trajectories are relatively coincide with the groundtruth due to weak following in RPY for the SLAM category. Cases with poor structure and texture are shown in figure 7(g) and 7(h) illustrating large amount of frame-by-frame errors accumulated for both the reference and the proposed. Reports about these 2 datasets are rarely found in literatures.

TABLE IV: Frame rate via FPS.

Approach	Min. FPS	Max. FPS	Ave. FPS
dvo_core	0.427004	0.675441	0.544489
EdgeVO	0.427475	0.634118	0.535528
GGM_MAD	0.839581	1.191072	0.944219
GGM_TD	0.841329	1.21326	0.988791
LOG_MAD	0.874166	1.351039	1.122969

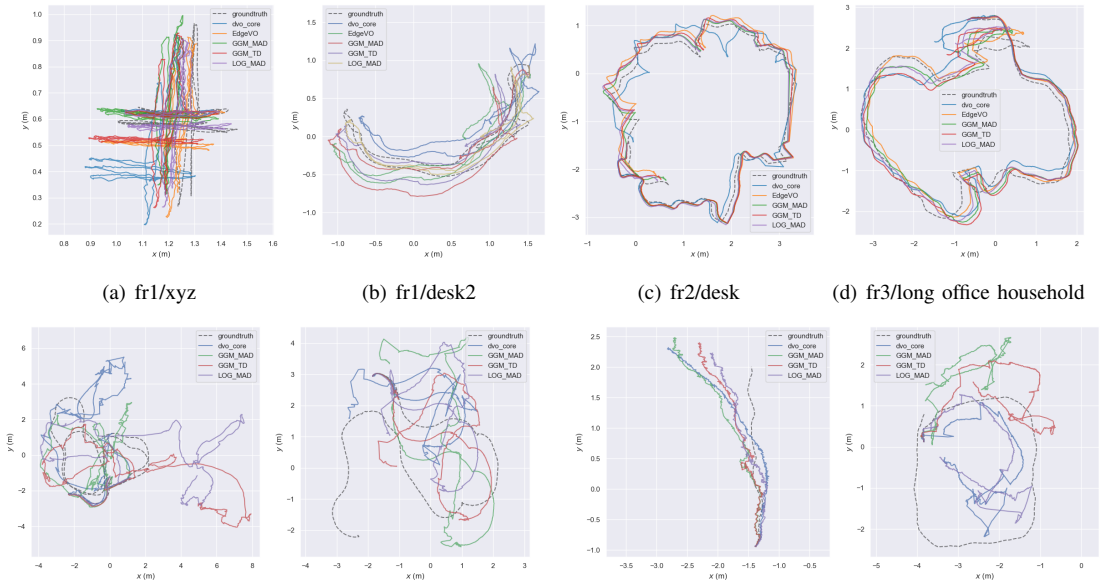
Additionally, the frame rate of each approach is also recorded via FPS, see table IV. Only FPS value that an approach conducts successfully a result on a test sequence is involved in calculating the average one. Both GGM and LOG took relatively longer time on the sequence of structure and texture than on the others where they reach a frame rate around 1.0 and 1.15 FPS respectively.

[3]Having a clear look with 300% zoom.

VI. Conclusions

Following on existing studies, this paper proposes to use low-order partial derivatives of Gaussian function down-sampling pixels for estimating motions of RGB-D cameras via direct tracking. To validate the feature metric, plenty of frame streams for variant evaluation purposes are employed making a comparison study against reference approaches. Experiment results have demonstrated the validity and running time performance of the proposed approach.

Building on a large amount of experiments, the future work of this paper would firstly aims at adaptive fitting of photometric residuals as mentioned at the end of section III-B. Parameter estimation of error distribution largely affects the accuracy of the trajectory generated. In the experiment of using the 3-layer pyramids with some customized parameters, the GGM is already able to give relatively good results for some cases as it reaches respectively the RMSE of the translational drift and the ATE to 0.093502 and 0.096328 on sequence "fr1/desk" for instance. Similar behaviours can also be found in using the LOG metric showing that the statistic parameter is of great interest in direct motion estimation. Secondly, hybrid attempts via GGM and Edges for example can further improve the overall accuracy. It reaches the RMSE of the translational drift to 0.070 and 0.064 on "fr2/xyz" and "fr2/rpy" respectively. However, this is not happening to every of the categories meaning that redundancies were increased for some cases and telling in return there is still work open in finding or learning detailed criteria for selecting candidate particles to track except for phase consideration in point-based tracking or optimization as it is discussed in section II. Alternatively, tracking in GGM or LOG domains could also be investigated.



(a) fr1/xyz (b) fr1/desk2 (c) fr2/desk (d) fr3/long office household

(e) fr2/pioneer_slam (f) fr2/pioneer_slam3 (g) fr3/**ns_nt**_far (h) fr3/**ns_nt**_near_withloop

Fig. 7: Trajectories illustration.

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80–92, 2011.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 15–22.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] H. Strasdat, A. J. Davison, J. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in 2011 International Conference on Computer Vision, 2011, pp. 2352–2359.

[5] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 3748–3754.

[6] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense rgb-d mapping," in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 5724–5731.

[7] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[8] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 1, pp. 154–180, 2022.

[9] A. Levin and R. Szeliski, "Visual odometry and map correlation," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 1, 2004, pp. I–I.

[10] Z. Javed and G.-W. Kim, "Omnivo: Toward robust omni directional visual odometry with multicamera collaboration for challenging conditions," IEEE Access, vol. 10, pp. 99 861–99 874, 2022.

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

[12] M. I. Baker, S., "Lucas-kanade 20 years on: A unifying framework," International Journal of Computer Vision, vol. 56, pp. 221–255, February 2004.

[13] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. Guerrero, "Dense rgb-d visual odometry using inverse depth," Robotics and Autonomous Systems, vol. 75, pp. 571–583, 2016.

[14] P. F. Proença and Y. Gao, "Probabilistic rgb-d odometry based on points, lines and planes under depth uncertainty," Robotics and Autonomous Systems, vol. 104, pp. 25–39, 2018.

[15] H. Z., C. S. W., J.-W. B., and I. Reid, "Visual odometry revisited: What should be learnt?" in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 4203–4210.

[16] J. Zhu, "Image gradient-based joint direct visual odometry for stereo camera," in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 4558–4564.

[17] Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3934–3942.

[18] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," IEEE Robotics and Automation Letters, vol. 2, no. 4, pp. 2263–2270, 2017.

[19] Y. Zhou, H. Li, and L. Kneip, "Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment," IEEE Transactions on Robotics, vol. 35, no. 1, pp. 184–199, 2019.

[20] K. Christensen and M. Hebert, "Edge-direct visual odometry," CoRR, vol. abs/1906.04838, 2019. [Online]. Available: http://arxiv.org/abs/1906.04838

[21] 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, 2012, pp. 573–580.

[22] W. Xue, X. Mou, L. Zhang, A. C. Bovik, and X. Feng, "Blind image quality assessment using joint statistics of gradient magnitude and laplacian features," IEEE Transactions on Image Processing, vol. 23, no. 11, pp. 4850–4862, 2014.

[23] S. Li and D. Lee, "Fast visual odometry using intensity-assisted iterative closest point," IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 992–999, 2016.

[24] S. Y. Lee, "The use of a log-normal prior for the student t-distribution," Axioms, vol. 11, no. 9, 2022.

[25] D. Hosszejni, "Bayesian estimation of the degrees of freedom parameter of the student-t distribution—a beneficial re-parameterization," September 2021. [Online]. Available: https://arxiv.org/pdf/2109.01726.pdf

[26] M. Pizenberg, "dvo core," Mar. 2019. [Online]. Available: https://github.com/mpizenberg/dvo/tree/76f65f0c9b438675997f595471d39863901556a9

[27] M. Grupp, "evo: Python package for the evaluation of odometry and slam," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo