2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

# Accurate Direct Visual-Laser Odometry with Explicit Occlusion Handling and Plane Detection

Kaihong Huang[1], Junhao Xiao[1], Cyrill Stachniss[2]

*Abstract*— **In this paper, we address the problem of combining 3D laser scanner and camera information to estimate the motion of a mobile platform. We propose a direct laser-visual odometry approach building upon photometric image alignment. Our approach is designed to maximize the information usage of both, the image and the laser scan, to compute an accurate frame-to-frame motion estimate. To deal with the sparsity of the range measurements, our approach identifies planar point sets within individual point clouds and subsequently extract their corresponding pixel patches from the camera image. The extracted planar image patches are used together with the non-planar pixels to estimate the frame-to-frame motion using a homography formulation capable of incorporating both types of pixel alignments. To achieve high estimation accuracy, we explicitly predict possible occlusions caused by observations taken from different locations. We evaluate our proposed approach using the KITTI dataset as well as data recorded with a Clearpath Husky platform. The experiments suggest that our approach can achieve competitive estimation accuracy and produce consistently registered, colored point clouds.**

## I. Introduction

The ability to estimate the ego-motion of a vehicle is a vital part of most autonomous navigation systems. Mobile robots and autonomous cars typically use different techniques for pose estimation, such as scan matching, visual odometry, or integrated GPS/IMU systems. Often, such vehicles are equipped with multiple sensors as the different sensing modalities have individual strength. For example, laser scanners are important for obstacle detection and tracking, while cameras are frequently used to interpret the scene using semantic segmentation or visual object detection systems.

Estimating the ego-motion using a laser scanner through point cloud alignment is often referred to as scan-matching. Scans are either matched pair-wise or with respect to a local or global map in order to compute the relative transformation between the robot's poses at the different points in time. Popular approaches for that are the iterative closest point (ICP) algorithm [1], [26] and different variants [17], [18] or correlative scan matching [15]. However, due to the limited number of laser beams in the laser scanner, the range measurements are rather sparse in the vertical direction. This can pose difficulties in the registration.

Visual odometry based on stereo or monocular cameras can also be used to estimate the ego-motion [3], [4], [10], [12]. Over the past few years, direct visual methods based on the photometric error became popular. These methods have the potential to exploit the full image information to estimate the camera motion. Recent advantages in this approach showed remarkable performance [3], [5], [9], [13], [20]. Monocular visual odometry, however, suffers inherently from the scale ambiguity problem.

Visual-laser odometry tries to exploit both the laser point cloud data and the camera image information for estimating the ego-motion. A combination of accurate but sparse spatial measurements from laser scans and dense appearance information from camera images have the potential to complement each other for the task of motion estimation. In this paper, we propose a direct laser-visual odometry approach based on the photometric image alignment method. Our approach is designed to maximize the information usage of both the image and the laser scan to achieve accurate frame-to-frame motion estimation. To address issues due to the sparsity of the range measurements, our approach identifies planar point sets from the laser data and extract the corresponding pixel patches from image data. The extracted dense planar image patches together with the sparse non-planar point cloud and pixels information are jointly used to estimate the frame-to-frame motion. For that, we rely on a homography formulation that is capable of incorporating both types of pixel alignments. We explicitly address the occlusion problem with a prediction approach tailored to deal with sparse laser point clouds.

To achieve high estimation accuracy, our approach employs a two-stage registration strategy. The first stage is aimed to ensure a proper initial pose estimate by jointly performing a coarse photometric pixel-alignments together with a geometric point cloud registration. The resulting estimate is then refined in the second stage by aligning only pixel intensities at the finest image level. The motivation behind this strategy is to combine the photometric and the geometric information while avoiding their respective pitfalls, i.e., local minima in photometric alignment and estimation bias in point cloud registration due to sparse point correspondences.

The main contribution of this paper is a novel direct approach to the joint laser-camera motion estimation. We exploit planar information, perform occlusion prediction, and a two-stage registration. Through this novel registration methods, our approach is able to obtain accurate frame-to-frame motion estimates using monocular camera image and laser range data. Experiments on the KITTI and self-recorded datasets supported this claim.

[1]K.H. Huang and J.H. Xiao are with the National University of Defense Technology, Institue of Inteligence Science, Changsha, China.

[2]C. Stachniss is with the University of Bonn, Institute of Geodesy and Geoinformation, Bonn, Germany.

This work has partly been supported by the DFG through the PhenoRob Cluster of Excellence, grant number EXC 2070.

## II. Related Work

Common work in laser-visual odometry can be categorized into two groups: *visual-odometry-based* approaches and *point-cloud-registration-based* approaches. Visual-odometry-based approaches try to apply a visual odometry pipeline with known pixel depth information coming from the laser scan. For example, the work of Shin *et al.* [19] tries to solve the visual-laser SLAM problem within the direct sparse odometry (DSO) [3] framework. They use the projected laser points as feature points instead of using the salient gradient points extracted from the images. With the depth values of the feature points known and fixed, they perform a multi-frame photometric optimization the same as the DSO to estimate the poses of the keyframes. The work of Zhang *et al.* named depth enhanced monocular odometry or DEMO [23] is similar. However, a common problem of visual-odometry based methods is that they do not consider the laser points that are outside the field of view of the camera, so much of the range measurements will be discarded with sensors like Velodyne LiDAR, which can provide a 360 degrees scan. Such setting renders the system less accurate and vulnerable to texture-less scenes.

In contrast to that, the point cloud registration (ICP) based approaches try to align the whole point cloud with the help of image information in various aspects. For example, the method in [16] and [24] simply use the visual odometry result as an initial guess to the ICP process, making the ICP less likely to be trapped in local minima.

A more advanced way to fuse the information is to use image/color information to guide and accelerate the data association process [8], [11]. The work of Joung [8] and Men [11] treat the color information as the fourth channel input to the ICP, allowing a faster convergence rate than normal ICP as reported by Men [11]. However, due to the inevitable outlier point correspondences, the true solution may not necessarily locate at the exact minimum of the ICP cost function. This is especially the case when the point cloud is sparse. Therefore, it is necessary to optimize a joint objective that rewards both tight point cloud alignment (via a geometric term) as well as consistent image appearance (via a photometric term), to obtain better estimation accuracy.

In this regard, the multi-cue photometric point cloud registration approach (MPR) by Della Corte *et al.* [2] tries to jointly register color, depth, and normal information within a unified framework by considering the depth and normal information as channels of a multi-channel image. The approach, however, requires the depth of each point to be known in order to transform the channel values, hence throws away a large amount of depth-less pixel information.

Our previous work [7] introduces a visual-laser odometry approach that estimates the 5-DoF relative orientation from image pairs through feature point correspondences and formulates the remaining scale estimation problem as a variant of the ICP problem with one degree of freedom. The image information is used indirectly through image feature points.

In this paper, we propose a direct visual-laser odometry approach that tries to avoid the aforementioned problems by considering also planar pixel patches in the image, as well as employing a two-stage registration strategy. Implementation details are described in the next section.

## III. Our Approach

In this paper, we assume the camera and laser scanner are time synchronized (e.g., by using hardware trigger) and that their relative transformation on the robot is known. Thus, one can project a 3D laser point to the camera image and directly obtain the intensity value of the corresponding image pixel.

We denote the *previous* visual-laser measurement, which consists of a point cloud $\{\boldsymbol{a}_i \in \mathbb{R}^3\}_{i=1}^{M}$ and an image $\mathcal{I}_a$, using the character $a$, while the *current* one uses $b$ with point cloud $\{\boldsymbol{b}_j \in \mathbb{R}^3\}_{j=1}^{N}$ and image $\mathcal{I}_b$. Our task is to estimate the ego-motion of the robot between $a$ and $b$, which consists of a relative rotation $\boldsymbol{R} \in \mathrm{SO3}$ and translation $\boldsymbol{t} \in \mathbb{R}^3$.

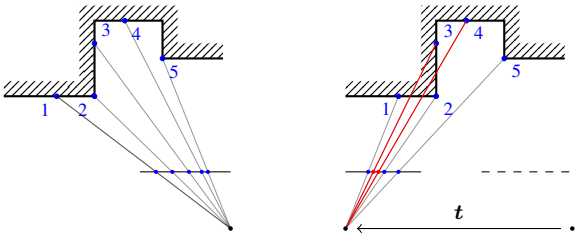
### A. Occlusion Detection for Sparse Point Cloud

Photometric alignment is based on the *constant image brightness* assumption, which assumes the intensities of corresponding pixels of a scene point in two (or more) images are equal. However, this assumption will be violated if the scene point is occluded during the viewpoint changes. The occluded points are outliers to the system and will deteriorate the estimation accuracy if they are not removed from the photometric alignment process.

To overcome the occlusion problem, we propose a novel method to predict which laser points of a sparse point cloud will be occluded under a certain camera motion. We explicitly exclude these occluded points from the motion estimation step. Compared to the standard Z-buffering approach, which is often used for dense depth images, our approach is more suitable for dealing with sparse laser point cloud data.

The key observation of our approach is that whenever parts of a point cloud are occluded in the current camera view, *the relative pixel order of the projected point cloud* in the current image will be different from the previous one. Consider Fig. 1 as an example. Assume there are five scene points, which are labeled from left to right as $1, 2, 3, 4, 5$ in the original camera view (Fig. 1a). After a camera translation, $\boldsymbol{t}$, we observe the scene again and obtain a new camera image by re-projecting the five points, as illustrated in Fig. 1b. However, points 3 and 4 are occluded in the new view. Note that, at the same time, the pixel order in the new image becomes $1, 3, 4, 2, 5$ from left to right, which is different from the original order $1, 2, 3, 4, 5$.

The phenomenon of pixel order changes is not limited to perspective projection but also holds true for the spherical projection, and we can exploit such pixel-order changes to identify occluded scene points. To be more specific, we can first compare the two sequences and find out which point-sets have been swapped, e.g. points $\{3,4\}$ and point 2 in Fig. 1. One of the point-set is occluded while the other is not. We identify the occluded one by comparing their depth values and the larger one is occluded, i.e. points 3 and 4 have larger depths than point 2, therefore they are occluded.




(a) before: 1 2 3 4 5 (b) after: 1 3 4 2 5

Fig. 1: The pixel order of projected 3D points will change if occlusions happened. (a) In the original view, we label the 3D points from left to right as $1, 2, 3, 4, 5$. (b) After the camera movement $\boldsymbol{t}$, points 3 and 4 are occluded and lead to a different pixel order in the new camera image, which is now $1, 3, 4, 2, 5$ from left to right. We exploit such pixel order changes to perform occlusion detection for sparse 3D point clouds.

Fig. 2: Example result of the occlusion detection algorithm. The red points are the image projections of the (predicted) occluded points of a laser scan. The occlusions happen mostly at the borders of objects and switch sides as the viewpoint changes.

We generalize this idea and propose Alg. 1 to perform occlusion detection for sparse 3D point clouds. Our algorithm takes a row (or a column) of points as input. A row (or column) means a subset of points having a close or the same pitch (or yaw) laser beam angle. Another input variable is a translation vector $\boldsymbol{t}$ representing the new camera position, which is expressed under the original point cloud coordinate frame. Because rotational camera movements do not induce scene occlusion, they are therefore not needed in the calculation.

Fig. 2 shows an example detection result on the KITTI dataset. The red points are the predicted occluded scene points, projected on images taken at a different location. Notice how the projected pixels lie on different objects because of the occlusion. In this example, the occluded points take up to 11% of the total visible points. It is worth noticing that the occlusion happens not only because of the camera ego-motion, but also due to the displacement between the camera and the LiDAR sensors on the vehicle. Therefore, to account for both effects, we use the camera ego-motion plus the camera-LiDAR displacement as the translation input $\boldsymbol{t}$ to Alg. 1.

**Algorithm 1** Occlusion Prediction

**Input:**
- A row of points $\mathcal{P}$
- Translational movement $\boldsymbol{t}$

**Output:**
- Occlusion mask $\mathcal{O}$

▷ **Step 1. Point projection**
Index list $\mathcal{A} \leftarrow$ **Sort** indices of $\mathcal{P}$ **by** $\{\pi_x(\boldsymbol{p}) \mid \boldsymbol{p} \in \mathcal{P}\}$
Index list $\mathcal{B} \leftarrow$ **Sort** indices of $\mathcal{P}$ **by** $\{\pi_x(\boldsymbol{p} - \boldsymbol{t}) \mid \boldsymbol{p} \in \mathcal{P}\}$
▷ **Step 2. List comparison**
Occlusion mask $\mathcal{O}(\cdot) \leftarrow$ False ▷ default: nothing occluded
Index $a \leftarrow \mathcal{A}$.pop()
Index $b \leftarrow \mathcal{B}$.pop()
**loop** size($\mathcal{P}$) times
  **if** $a == b$ **then**
    update both $a$ and $b$ ▷ in line 18
  **else**
    **if** $\mathcal{P}_z(a) > \mathcal{P}_z(b)$ **then** ▷ $a$ is behind $b$ thus occluded
      mark $\mathcal{O}(a) \leftarrow$ True
      update only $a$
    **else** ▷ $b$ is occluded
      mark $\mathcal{O}(b) \leftarrow$ True
      update only $b$
  **if** update $a$ **then**
    $a \leftarrow \mathcal{A}$.pop()
    **if** $\mathcal{O}(a)$ is True **then repeat** line 19
  **if** update b **then**
    b $\leftarrow \mathcal{B}$.pop()
    **if** $\mathcal{O}(b)$ is True **then repeat** line 22
**return** $\mathcal{O}$

### B. Coplanar Point Detection

The depth measurements are often sparse and cover only a small portion of the image pixels when projected into the camera image (see Fig. 4b). While most image pixels do not have depth information from the laser, such depth-less image pixels are either discarded (e.g., in [2]) or falsely assigned with a constant depth the same as their associated pixel (e.g., in [19]), which are both sub-optimal solutions.

We overcome this problem by exploiting planar regions in the scene, which are often abundant in structured (urban) environments. A scene plane usually corresponds to a large number of pixels, and such pixels can also be used to estimate the motion parameters even without knowing their depth values, because they can be projected to another image using plane-induced homography given the plane parameters. Therefore, to include as much as possible pixel information in the photometric term, our approach explicitly detects scene planes from the point cloud and use them for registration.

To identify which subset of the laser points are parts of a planar region, we propose a grid-based method inspired by the work by Weingarten *et al.* [21] and Xiao *et al.* [22]. The main idea is to first discretize the point cloud into a grid of cells and then, for each cell use principal component analysis (PCA) to fit a plane to the points that are inside the cell.

We also accelerate the detection process by incorporating prior knowledge about existing planes, e.g., knowledge about the ground plane or previously detected planes. Given prior

**Algorithm 2** Coplanar Point Detection

**Input:**
- Point cloud $\mathcal{IP}$
- Prior plane parameters $\{(\boldsymbol{n}, d)\}$

**Parameter:**
- Grid size $s$
- Point-to-plane distance threshold $\epsilon$

**Output:**
- Planar points list $\mathcal{P}$
- Plane normal list $\mathcal{N}$

▷ **Step 1: Prior Plane Fitting**
**for** each prior plane parameters $(\boldsymbol{n}, d)$ **do**
  Inliers $\mathcal{I} \leftarrow \{\boldsymbol{p} \in \mathcal{IP} \setminus \mathcal{P} \mid |\boldsymbol{n}^\mathsf{T}\boldsymbol{p} - d| < \epsilon\}$
  Planar points list $\mathcal{P} \xleftarrow{\text{insert}}$ Inliers $\mathcal{I}$
  Plane normal list $\mathcal{N} \xleftarrow{\text{insert}} \boldsymbol{n}$
▷ **Step 2: Discretization**
Point list $\mathcal{L} \leftarrow \{\emptyset\}$
**for** each point $\boldsymbol{p}$ in the remaining point cloud $\mathcal{IP} \setminus \mathcal{P}$ **do**
  Cell coordinates $(u, v, w) \leftarrow$ **discretize**$(\boldsymbol{p}, s)$
  Point list $\mathcal{L} \xleftarrow{\text{insert}}$ item $\{(u, v, w) : \boldsymbol{p}\}$
**Sort** point list $\mathcal{L}$ **by** $(u, v, w)$
Cell list $\mathcal{C} \leftarrow \{\emptyset\}$
Current cell $c \leftarrow \{\emptyset\}$
**for** each point $\boldsymbol{p}_i$ in the sorted point list $\mathcal{L}$ **do**
  Current cell $c \xleftarrow{\text{insert}} \boldsymbol{p}_i$
  **if** current $(u, v, w)_i \neq$ next coordinates $(u, v, w)_{i+1}$ **then**
    Cell list $\mathcal{C} \xleftarrow{\text{insert}}$ current cell $c$
    Current cell $c \leftarrow$ new cell $\{\emptyset\}$
▷ **Step 3: Plane Detection**
**for** point set $\{\boldsymbol{p}\}$ of each cell in the cell list $\mathcal{C}$ **do**
  Eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leftarrow$ **PCA**$(\{\boldsymbol{p}\})$
  **if size**$(\{\boldsymbol{p}\}) < 7$ **then** ▷ $\{\boldsymbol{p}\}$ is too sparse.
    **skip** this cell
  **if** $\lambda_1$/**size**$(\{\boldsymbol{p}\}) > \epsilon$ **then** ▷ $\{\boldsymbol{p}\}$ not planar.
    **skip** this cell
  Normal vector $\boldsymbol{n} \leftarrow$ eigenvector $\boldsymbol{v}_1$ (for eigenvalue $\lambda_1$)
  Plane normal list $\mathcal{N} \xleftarrow{\text{insert}} \boldsymbol{n}$
  Planar points list $\mathcal{P} \xleftarrow{\text{insert}} \{\boldsymbol{p}\}$
**return** $\mathcal{P}, \mathcal{N}$

plane parameters $(\boldsymbol{n}, d)$, where $\boldsymbol{n}$ is the normal vector of the plane and $d$ is the plane-to-camera-origin distance, we compute the point-to-plane distance $|\boldsymbol{n}^\mathsf{T}\boldsymbol{p} - d|$ for each point $\boldsymbol{p}$ in the new point cloud. Points with a small distant are identified as inlier points for that plane. These inlier points are removed from the point cloud and the fitting process is performed again with the next prior plane parameters until all hypotheses are tested. This process happens as the first stage of the planar point detection and can identify a large portion of the planar points. After that, all the remaining (unmatched) points are then handled by the grid-based detection process. Alg. 2 summarizes our proposed coplanar point detection method.

### C. Homography-Based Photometric Alignment

Once we extracted the (dense) planar image patches, they are used together with the (sparse) non-planar point cloud-projected pixels to estimate the motion parameters with our homography formulation.

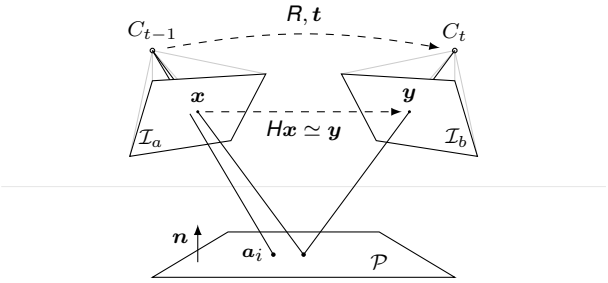


Fig. 3: Plane-induced homography relation of two images.

Fig. 3 shows a plane-induced homography relation of two images. Assume a laser point $\boldsymbol{a}_i$ is located on a 3D plane $\mathcal{P}$ with a normal vector $\boldsymbol{n}$ and a plane-to-camera-origin distance $d \overset{\text{def}}{=} \boldsymbol{n}^\mathsf{T}\boldsymbol{a}_i$. Any points $\boldsymbol{p}$ belongs to this plane will satisfy the equation

$$\boldsymbol{n}^\mathsf{T}\boldsymbol{p} = d. \tag{1}$$

Now assume that there is an previous image point $\boldsymbol{x} \overset{\text{def}}{=} [u, v, 1]^\mathsf{T}$ on $\mathcal{I}_a$, its back projected ray $\boldsymbol{r}(\lambda) \overset{\text{def}}{=} \lambda\boldsymbol{x}$ intersects the plane $\mathcal{P}$. According to Eq. (1), the intersection happens at

$$\boldsymbol{n}^\mathsf{T}(\lambda\boldsymbol{x}) = d \quad \mapsto \quad \lambda = \frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}. \tag{2}$$

Therefore, the intersection point is $\frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}\boldsymbol{x}$ and will have a 3D homogeneous coordinates

$$\begin{bmatrix} \frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}\boldsymbol{x} \\ 1 \end{bmatrix} \simeq \begin{bmatrix} d\boldsymbol{x} \\ \boldsymbol{n}^\mathsf{T}\boldsymbol{x} \end{bmatrix} \in \mathbb{R}^4. \tag{3}$$

Given the relative motion parameters $\mathsf{R}$ and $\boldsymbol{t}$, we can project this intersection point to the current image $\mathcal{I}_b$ and obtain

$$\boldsymbol{y} \simeq \begin{bmatrix} \mathsf{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} d\boldsymbol{x} \\ \boldsymbol{n}^\mathsf{T}\boldsymbol{x} \end{bmatrix} = (d\mathsf{R} + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})\boldsymbol{x}. \tag{4}$$

With the camera intrinsic matrix $\mathsf{K}$ and $\boldsymbol{x}$, $\boldsymbol{y}$ being in pixel coordinates, Eq. (4) becomes

$$\boldsymbol{y} \simeq \mathsf{K}(d\mathsf{R} + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})\mathsf{K}^{-1}\boldsymbol{x}. \tag{5}$$

Therefore, the pixels of a 3D plane in the two images, i.e. $\boldsymbol{x}$ and $\boldsymbol{y}$, are related through

$$\boldsymbol{y} \simeq \mathsf{H}\boldsymbol{x}, \tag{6}$$

where $\mathsf{H} \overset{\text{def}}{=} \mathsf{K}(d\mathsf{R} + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})\mathsf{K}^{-1}$ is a plane-induced homography.

For a non-planar laser point that does not lie on a planar region, the homography formulation in Eq. (6) is still applicable because it can be seen as a special case where the pixel by itself defines a fronto-parallel patch, i.e.,

$$\mathcal{P} = \{\boldsymbol{a}_i\} \qquad \text{and} \qquad \boldsymbol{n} = [0, 0, 1]^\mathsf{T}. \tag{7}$$

In this case, $d = [0, 0, 1]\boldsymbol{a}_i$ is the depth of $\boldsymbol{a}_i$, thus $\boldsymbol{a}_i = d\mathsf{K}^{-1}\boldsymbol{x}$ and the entity $\mathsf{H}\boldsymbol{x}$ amounts to the standard 3D

point projection as

$$\boldsymbol{H}\boldsymbol{x} = \boldsymbol{K}(\boldsymbol{R} + \frac{\boldsymbol{t}\boldsymbol{n}^\mathsf{T}}{d})d\boldsymbol{K}^{-1}\boldsymbol{x} \tag{8}$$

$$= \boldsymbol{K}(\boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t}\frac{\boldsymbol{n}^\mathsf{T}\boldsymbol{a}_i}{d}) \tag{9}$$

$$= \boldsymbol{K}(\boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t}). \tag{10}$$

Base on this homography formulation, we define our photometric cost function for estimating the motion parameters $\boldsymbol{R}$ and $\boldsymbol{t}$ as

$$E_{\text{pho},i} \overset{\text{def}}{=} \sum_{\mathbf{x}\in\mathcal{P}_{\text{img}}(\boldsymbol{a}_i)} \varphi\Big(\underbrace{\mathcal{I}'_a(\mathbf{x}) - \mathcal{I}_b\big(\pi(\boldsymbol{H}_i(\boldsymbol{R},\boldsymbol{t})\boldsymbol{x})\big)}_{\{e_{\text{pho}}\}}\Big) \tag{11}$$

where

- $\varphi(\cdot)$ is a robustification function based on the t-distribution (of five degree of freedom, as in [9]):

$$\varphi(e) \overset{\text{def}}{=} \frac{6}{5 + \frac{e^2}{\sigma^2}}\, e^2 \tag{12}$$

  with $\sigma$ being the standard deviation of all residuals $e$;
- $\mathbf{x} \overset{\text{def}}{=} [u, v]^\mathsf{T}$ is a pixel coordinates, and $\boldsymbol{x} \overset{\text{def}}{=} [u, v, 1]^\mathsf{T}$ is its homogeneous form;
- $\pi(\cdot)$ is the Euclidean normalization that transforms homogeneous coordinates into (inhomogeneous) pixel coordinates, i.e. $\pi(\boldsymbol{x}) = \mathbf{x}$, or more generally

$$\pi([u, v, w]^\mathsf{T}) \overset{\text{def}}{=} [u/w, v/w]^\mathsf{T}; \tag{13}$$

- $\mathcal{P}_{\text{img}}(\boldsymbol{a}_i)$ denotes a set of neighboring pixels around the image point of $\boldsymbol{a}_i$. If the image point of $\boldsymbol{a}_i$ is denoted as $\mathbf{a}_i \overset{\text{def}}{=} \pi(\boldsymbol{K}\boldsymbol{a}_i)$, then

$$\mathcal{P}_{\text{img}}(\boldsymbol{a}_i) \overset{\text{def}}{=} \begin{cases} \{\mathbf{x} \in \mathbb{Z}^2 \mid \|\mathbf{a}_i - \mathbf{x}\| \le r\}, & \text{if } \boldsymbol{a}_i \text{ is planar,} \\ \{\mathbf{a}_i\}, & \text{if non-planar,} \end{cases} \tag{14}$$

  where $r$ is a predefined radius;
- $\boldsymbol{H}_i(\boldsymbol{R},\boldsymbol{t}) \overset{\text{def}}{=} \boldsymbol{K}(\boldsymbol{a}_i^\mathsf{T}\boldsymbol{n}_i\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}_i^\mathsf{T})\boldsymbol{K}^{-1}$ is the homography associated to the laser point $\boldsymbol{a}_i$. For non-planar points we set $\boldsymbol{n}_i = [0, 0, 1]^\mathsf{T}$. Otherwise $\boldsymbol{n}_i$ is calculated from the laser point cloud using a method described in Sec. III-B;
- $\mathcal{I}'_a(\cdot) \overset{\text{def}}{=} \alpha\mathcal{I}_a(\cdot)+\beta$ is used to model the gain, $\alpha$, and the bias, $\beta$, between the two intensity images, to account for possible different camera exposure settings and ambient light changes. Both $\alpha$ and $\beta$ are unknown parameters to be estimated during the optimization.

### D. Two-Stage Registration

Photometric alignment is in essence a highly nonlinear optimization problem with lots of local minima. To ensure a proper initial estimate and avoid false minima, we first optimize a joint objective that rewards both consistent photometric alignment (with smoothed images) as well as tight point cloud registration. For that, besides the photometric term in Eq. (11), we also incorporate a geometric term to account for the point-to-plane point cloud registration errors as in the ICP:

$$E_{\text{geo},i} \overset{\text{def}}{=} \varphi\big(\underbrace{\boldsymbol{n}_i^\mathsf{T}(\boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}'_i)}_{\{e_{\text{geo}}\}}\big), \tag{15}$$

where $\boldsymbol{b}'_i$ is the nearest-neighbor to the transformed $\boldsymbol{a}_i$ in the point cloud $\boldsymbol{b}$, determined by using a $k$-d tree search. For non-planar points, $\boldsymbol{n}_i$ refers to the surface normal of the points.

Combining Eq. (11) and Eq. (15), we have in the first registration stage a minimization problem of the form:

$$\underset{\boldsymbol{R},\boldsymbol{t},\alpha,\beta}{\operatorname{argmin}} \frac{1}{\sigma^2_{\text{geo}}} \sum_i E_{\text{geo},i} + \frac{1}{\sigma^2_{\text{pho}}} \sum_{i\in\text{Vis}} E_{\text{pho},i}, \tag{16}$$

where

- $\sigma_{\text{geo}}$ and $\sigma_{\text{pho}}$ are the standard deviation of the residuals $\{e_{\text{geo}}\}$ and $\{e_{\text{pho}}\}$;
- $i \in$ Vis stands for laser points that are visible and not occluded in both camera images $\mathcal{I}_a$ and $\mathcal{I}_b$, which are smoothed by a Gaussian function and then downsampled.

In the second stage of the alignment procedure, the estimation of $\boldsymbol{R}$ and $\boldsymbol{t}$ is refined by performing photometric alignment at the finest resolution. Therefore, a cost function similar to Eq. (16) is used in the second stage, but with only the photometric term $E_{\text{pho}}$ and using raw images.

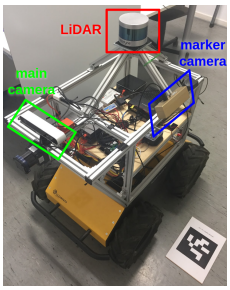
In both stages, we minimize the objective with a standard iterative Gauss-Newton optimization algorithm. Our experimental result in Sec. IV-A suggests that our two-stage registration strategy can significantly improve the estimation accuracy.

## IV. Experimental Evaluation

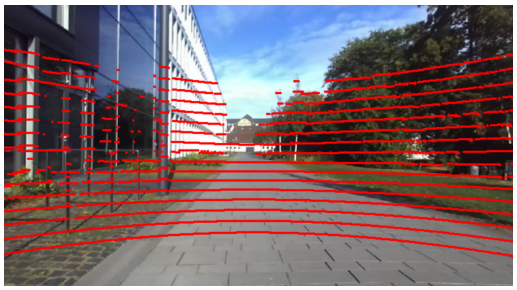
The main focus of this work is a novel direct approach to joint laser-camera odometry. The experiments are designed to show the capabilities of our method and to support our key claim that our approach is able to accurately estimate frame-to-frame motion using monocular vision and laser range data. We perform the evaluations on own robotic datasets as well as on publicly available ones.

### A. Outdoor LiDAR-Camera Dataset with Gound Truth Control Points

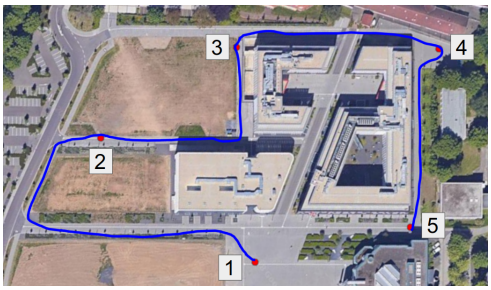
The first experiment is to verify the proposed method with a mobile robot in an outdoor environment. The robot is a Clearpath Husky mobile platform equipped with a 16-beams Velodyne VLP-16 LiDAR and a stereo-camera (we use images from only the left camera here), as shown in Fig. 4a. Along the performed experiment route, there are five geodetic control points on the ground with precisely measured coordinates around our campus, as illustrated in Fig. 4c. We place Apriltag markers [14] on top of the control points and utilize an auxiliary camera on the robot to detect these markers on the ground when the robot drives by them. In this way, we obtain the positions of the robot relative to the control points. We use these positions as ground-truth locations in the environment to evaluate the trajectory



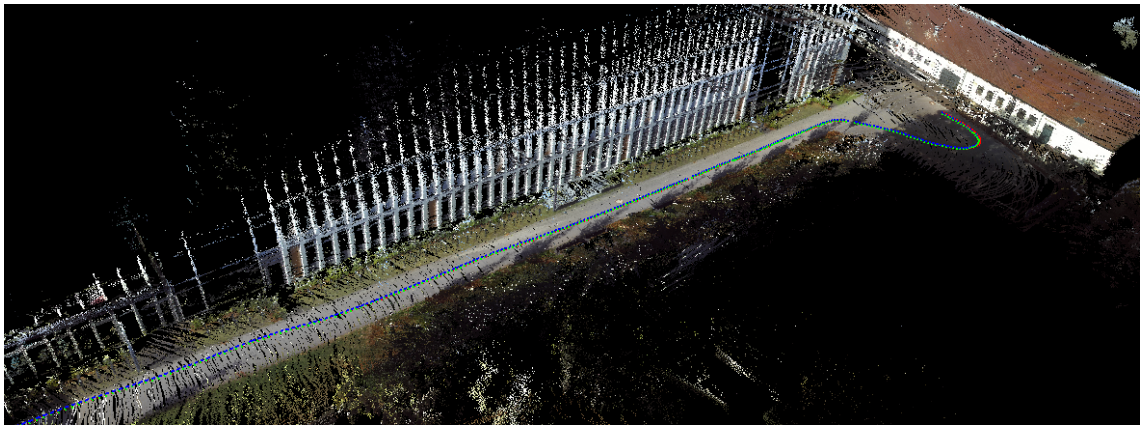

(a) Robot platform.



(b) Laser points projected into the camera image.




(c) Experimental route and control points.



(d) 3D mapping result of the path from control point 4 to 5.

Fig. 4: Outdoor experiment. We drive a Clearpath Husky robot around the campus. The experimental path passes by five precisely known geodetic control points, which are used for the ground truth evaluation.

TABLE I: Relative distance error measured at five control points.

| Segments | | 1-2 | 2-3 | 3-4 | 4-5 |
|---|---|---|---|---|---|
| Ground Truth Point Dist. (m) | | 114.12 | 93.86 | 116.87 | 103.62 |
| Estimated Trajectory length (m) | | 213.47 | 133.16 | 125.17 | 110.05 |
| Geo. Only | Dist. Error (m) | 3.20 | 1.43 | 2.60 | 2.92 |
| | Rel. Error (%) | 1.49 | 1.07 | 2.07 | 2.65 |
| Pho. Only | Dist. Error (m) | 0.92 | 2.25 | 0.26 | 0.57 |
| | Rel. Error (%) | 0.43 | 1.69 | 0.21 | 0.52 |
| Combined | Dist. Error (m) | 0.26 | 0.12 | 0.02 | 0.35 |
| | Rel. Error (%) | **0.12** | **0.09** | **0.02** | **0.32** |

estimated with our approach. Due to the orientation of the markers are somewhat uncertain, we compare the point-to-point distances and the result is shown in Tab. I.

As shown in the last row of Tab. I, our approach achieves an excellent accuracy with relative distance errors as low as 0.1%, without using loop-closing. Fig. 4d depicts a colored point cloud generated by our approach.

To see the benefit of using the two-stage registration strategy, we also include in Tab. I the results of using only the geometric term or the photometric term. The result suggests that the accuracy improvements of using two-stage registration are significant.

### *B. Comparison to State-of-the-Art Methods Using KITTI*

The second experiment performs evaluations about the motion estimation quality of our approch using the odometry datasets of the KITTI Odometry Benchmark [6]. We performed the motion estimation using the point clouds from the 64-beams Velodyne LiDAR and the monochromic images from the camera 0. The results of sequences without loop-closing are reported in Tab. II for comparison, including the reported results of Shin *et al.* [19] (a photometric-alignment based visual-laser odometry approach), as well as the state-of-the-art laser-based approach, LOAM [25]. The result shown in Tab. II suggests that our approach perform better or on par with the state-of-the-art in terms of translational error.

TABLE II: Comparison on relative translational error using the KITTI odometry dataset.

| | Sequences without loops | | | |
|---|---|---|---|---|
| Approach | 01 | 03 | 04 | 10 |
| LOAM [25] | 1.4% | 0.9% | 0.7% | 0.8% |
| Shin [19] | 1.5% | **0.9%** | 0.7% | 0.7% |
| Our | **1.0%** | 1.1% | **0.6%** | **0.7%** |

## V. Conclusion

In this paper, we presented a novel direct approach to joint laser-camera odometry. Our method exploits planar information, performs occlusion prediction, and employs a two-stage registration. This allows us to estimate frame-to-frame motions with high accuracy. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques. The evaluation result supported the claim that our approach can achieve competitive estimation accuracy.

## REFERENCES


[1] P.J. Besl and N.D. McKay. A Method for Registration of 3-d Shapes. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[2] B. Della Corte, I. Bogoslavskyi, C. Stachniss, and G. Grisetti. A General Framework for Flexible Multi-Cue Photometric Point Cloud Registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[3] J. Engel, V. Koltun, and D. Cremers. Direct Sparse Odometry. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 40(3):611–625, 2018.

[4] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 834–849, 2014.

[5] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 15–22, 2014.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[7] K.H. Huang and C. Stachniss. Joint Ego-motion Estimation Using a Laser Scanner and a Monocular Camera Through Relative Orientation Estimation and 1-DoF ICP. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[8] J.H. Joung, K.H. An, J.W. Kang, M.J. Chung, and W. Yu. 3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[9] C. Kerl, J. Sturm, and D. Cremers. Robust Odometry Estimation for RGB-D Cameras. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3748–3754, 2013.

[10] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[11] H. Men, B. Gebre, and K. Pochiraju. Color Point Cloud Registration with 4D ICP Algorithm. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[12] R. Mur-Artal, J.M.M. Montiel, and J. D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. on Robotics (TRO)*, 31(5):1147–1163, 2015.

[13] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 2320–2327, 2011.

[14] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[15] E.B. Olson. Real-Time Correlative Scan Matching. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 4387–4393, 2009.

[16] G. Pandey, J. McBride, S. Savarese, and R. Eustice. Visually bootstrapped generalized icp. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[17] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[18] J. Serafin and G. Grisetti. NICP: Dense Normal Based Point Cloud Registration. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 742–749, 2015.

[19] Y. Shin, Y.S. Park, and A. Kim. Direct visual slam using sparse depth for camera-lidar system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[20] F. Steinbrücker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *the IEEE Intl. Conf. on Computer Vision (ICCV) Workshops*, pages 719–722, 2011.

[21] J. Weingarten, G. Gruener, and R. Siegwart. A fast and robust 3d feature extraction algorithm for structured environment reconstruction. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 390–397, 2003.

[22] J.H. Xiao, J.H. Zhang, J.W. Zhang, H.X. Zhang, and H.P. Hildre. Fast plane detection for slam from noisy range images in both structured and unstructured environments. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 1768–1773, 2011.

[23] J. Zhang, M. Kaess, and S. Singh. A real-time method for depth enhanced visual odometry. *Autonomous Robots*, 41:31–43, 2017.

[24] J. Zhang and S. Singh. Visual-Lidar Odometry and Mapping: Low-Drift, Robust, and Fast. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[25] J. Zhang and S. Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41:401–416, 2017.

[26] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Intl. Journal of Computer Vision (IJCV)*, 13(2):119–152, 1994.