2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Accurate Extrinsic Calibration between Monocular Camera and Sparse 3D Lidar Points without Markers

Zhipeng Xiao[1,2], Hongdong Li[2], Dingfu Zhou[2], Yuchao Dai[2] and Bin Dai[1]

***Abstract*— It is of practical interest to automatically calibrate the multiple sensors in autonomous vehicles. In this paper, we deal with an interesting case when used low-resolution Lidar and present a practical approach to extrinsic calibration between monocular camera and Lidar with sparse 3D measurements. We formulate the problem as directly minimizing the feature error evaluated between frames following the way of image warping. To overcome the difficulties in the optimization problem, we propose to use the distance transform and further projection error model to obtain the key approximated edge points that are sensitive to the loss function. Finally, the loss minimization is solved by an efficient random selection algorithm. Experimental results on KITTI dataset show that our proposed method can achieve competitive results and an improvement in translation estimation particularly.**

I. Introduction

For long term autonomous driving, it needs algorithm to automatically calibrate the sensors. Therefore, it plays an important role for the system to accurately calibrate the extrinsic parameters without manual operations and markers. Due to the importance, many approaches have been proposed [1] [2] [3] [4] to tackle this task and remarkable performance has been achieved in real world applications. However, existing methods to extrinsic calibration generally depend on dense 3D measurements from the Lidar to enable motion estimation or feature description from depth or intensity. Most of the current autonomous vehicles (Google, Uber, Ford, Baidu, etc.) are equipped with high-end Lidar such as Velodyne 64 to achieve dense measurements. The main drawback with the current high-end Lidar such as Velodyne is the high cost, which is even comparable to the cost of the whole vehicle. Therefore there is probably an interesting case applying low-cost Lidar with sparse 3D points instead of the high-end ones. For a brief view on our work, Fig. 1 illustrates the performance of our method.

In this paper, we aim at calibrating the extrinsic parameters between a monocular camera and Lidar with sparse 3D Lidar measurements without any markers. Compared to existing methods with dense Lidar measurements, the task of using sparse Lidar measurements (e.g., around 1000 points) is particularly challenging due to the following reasons: 1) The structure (e.g., edges) and reflectance information cannot be extracted or not accurate enough to correlate well with the mutual information in the camera image. 2) The Lidar measurements are too sparse to enable accurate motion estimation for linear solution especially in the "Hand-Eye" framework. 3) In addition, with the sparse Lidar point information, the estimation of translation is even more difficult since it originally is tough though the rotation is relatively easy [5].

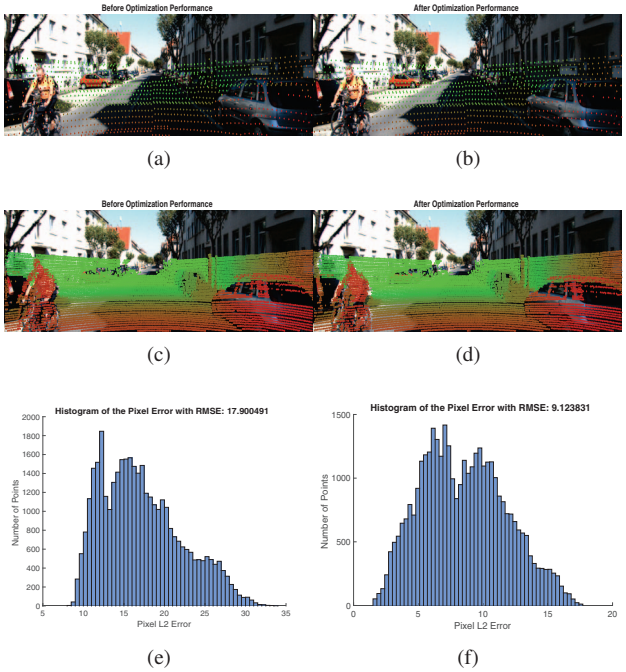


Fig. 1: Illustration of our solution to extrinsic calibration. (a)(b) show the calibration result before and after optimization with the color images overlaid by the sparse Lidar measurements. The corresponding cases for better display are shown in (c)(d). (e)(f) demonstrate the histograms of pixel intensity error for dense cases, showing the improvement by our method.

To address the above issues, we propose to directly minimizing a feature error defined in the frames in the way of image warping. In this way, the extrinsic calibration problem has been transformed to a loss minimization, for which we solve with evolutionary algorithm.

The main contributions of this paper can be summarized as: 1) A general framework for extrinsic calibration between a monocular camera and Lidar, which can be used for both dense and sparse Lidar points clouds; 2) An effective frame and point selection strategy has been proposed by analyzing the sensitivity of the loss function for rotation and translation estimation; and 3) An efficient algorithm for accurate estimation through approximate edge points and

[1]College of Mechatronic Engineering and Automation, National University of Defense Technology, P.R. China {xiaozhipeng, daibin}.cs@hotmail.com
[2]Research School of Engineering, Australian National University, Australia {hongdong.li, dingfu.zhou, yuchao.dai}@anu.edu.au

constrains on sensitive key points.

II. Related Work

Existing extrinsic calibration methods for monocular camera and Lidar can be roughly categorized into two groups: mutual information based methods and motion based methods. The mutual information based methods exploit the mutual information between different sensors to find the extrinsic calibration parameters, e.g., the intensity in color image and the reflectance or depth from Lidar. The motion based methods utilize the conjugate motion cue inside the camera and Lidar rig. When both motion estimations are available from them, "Hand-Eye" calibration [6] is used for extrinsic calibration. Again, these methods depend on cross-modality correspondences or is fragile due to algebraic optimization.

Mutual information based methods try to find the extrinsic parameters by exploiting mutual information between different sensors, such as the appearance information in color images and the reflectance or depth values from Lidar. Theoretically, extrinsic parameters can be solved from only one frame pair. Levinson *et al.* [1] proposed to calibrate camera-Lidar system by utilizing the edge features in both 2D image and 3D point clouds. Compared to [1] which only used edge information, normalized mutual information based method [2], [3] achieve better performance. The reason can be interpreted as the loss function is more smooth with fewer local minimums due to the reflectivity on both edges and non-edges areas are for computing the mutual information. In addition, Scott *et al.* proposed a method to find the good scenarios for calibration by using the Normalised Information Distance [4]. In [7], Tamas *et al.* proposed to achieve the calibration parameters by using shape registration between the 2D image and 3D points clouds, e.g., corresponding planar regions.

Motion based methods aim at solving the extrinsic transformation by exploiting the motion cues between different sensors. Theoretically, the extrinsic parameters can be computed linearly if the camera and Lidar motion are computed individually. Taylor *et al.* computed the initial calibration via "Hand-Eye" method and then refine results by using intensity consistency assumption across multiple frames, which means that the Lidar point projected into different image frames should share the same image intensity. However, this "Hand-Eye" approach suffers inevitable problem that the approximate planar motion causes the translation solution to be singular [6] and also heavily relies on the structure from motion techniques, which returns an unstable initial. Zhao *et al.* [8] provided another way to compute the transformation by registering two groups of 3D point clouds with ICP (iterative closest point). The 3D points in camera coordinates can be obtained by using structure from motion technique. Although this method is not robust and inaccurate due to image based 3D reconstruction, it also shows that this framework is promising when the image based 3D reconstruction is reliable.

III. Problem Formulation

Given a set of m sparse Lidar point clouds $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots \mathbf{p}_m\}$ and the corresponding images $\mathbf{I} = \{I_1, I_2, \cdots I_m\}$ with relative poses $\mathbf{T} = \{T_1, T_2, \cdots T_{m-1}\}$ for each pair of images, our goal is to seek the best relative transformation $\mathbf{X}$ between Lidar and camera. The loss function is formulated as:

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{N} \sum_{i=1}^{m-1} [f^{i+1}(\pi(T_i \cdot \mathbf{X} \cdot \mathbf{p}_i)) - f^i(\pi(\mathbf{X} \cdot \mathbf{p}_i))]^2, \tag{1}$$

where f is an alternative feature and SURF descriptor [9] is used in our work for robustness, m the total frame numbers, N the total number of computed points and π is the image projection function.

In this paper, the relative pose between frames is not our main concern so that the known poses are used and alternatively they can be solved by some methods, e.g. [10], [11], [12].

Note that, Eq. (1) is a little different from the classical intensity-consistency formula since its two feature items both contain the variables so that they simultaneously change in each optimization iteration, which makes the loss function contain more complex local minima. And that is why classic methods are very laborious for this problem because they belong to a kind of single matching problem where the key features are fixed and while ours is a kind of double matching problem where the features in relevant frames both vary to match each other.

IV. Proposed Approach

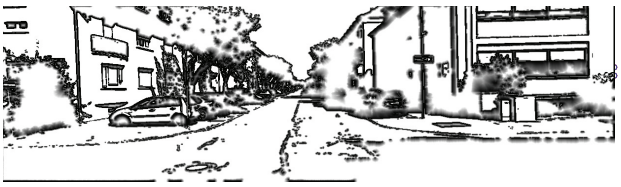
The loss function in Eq. (1) is non-linear and non-convex with a lot of local minima. Theoretically, it is real difficult to obtain an approximate global solution efficiently. Therefore, several strategies have been proposed for improvement.

A. Distinguish Frames Selection

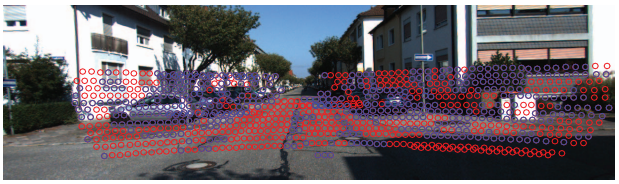
Scenario has significant influence on the function. Frankly speaking, this kind of function is invalid if the image is textureless or the intensity is evenly distributed over the whole image. For example, the camera facing a wide white wall. So, strategies should be proposed to select good scenarios where it probably includes distinguishable structure, large intensity variation, few plain area, etc. Alternatively, taking more frames for error computation is another way to smoothing the loss function, which is effective to reduce the number of local minima. However, increasing of frames will increase the computation burden. Here, we empirically choose 500 frames to balance the efficiency and performance. Additionally, in order to further enrich the scene context, we select frame pairs for loss computation rather than using consecutive frames. Thus, it can avoid using repetitive scenarios which could also make the estimation to be biased. The constrains used for selecting the frame pair include two parts:

$$\text{Angle} = \arccos(\frac{1}{2}(trace(\mathbf{R}) - 1)) \cdot \frac{180}{\pi} \geq \theta, \tag{2}$$

$$\tau_1 < |\mathbf{t}| \leq \tau_2, \tag{3}$$



(a)



(b)

Fig. 2: Illustration of Distance transform. (a) An example of distance transformed image and (b) approximate edge points in blue compared to other ordinary points in red.

where $\mathbf{R}$ and $\mathbf{t}$ represents the relative rotation and translation between the pair of frames and we set θ =3° and $\tau_1 = 0\,\mathrm{m/s}$, $\tau_2 = 2\,\mathrm{m/s}$ for the physical limits of common vehicles.

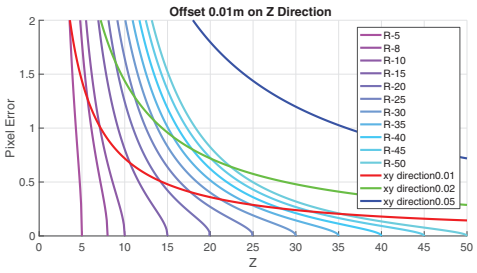
This simple criteria makes the selected frames are mostly from cross roads where abundant features are available, For example, 3D points from close buildings. Particularly, these close points in this kind of scenarios are extremely helpful to improve the translation estimation.

B. Sensitive Key Points Selection

In the field of computer vision and robotics, it is common sense that the translation estimation is more challenging than rotation in relative camera pose estimation. However, this issue also occurs in the extrinsic calibration problem. Detailed analysis is shown in V-B. Selection on distinguished points inside one frame also plays an important role on the loss function like that on distinguished frame pairs.

1) Distance Transform for Approximate Edge Points: Many methods have shown that the gradient information either in 2D image or 3D point cloud is essential to the extrinsic calibration. Unfortunately, in our case, the gradient information can be obtained in 2D image only. Therefore, we select the "Approximate Edge Points" whose projected 2D points are close to image edges. Because points projected in flat or textureless area don't contribute too much on the loss function. Here, the distance transform technique [13] is employed for seeking these approximate edge points. Figure 2 shows an example of the distance transform result. This selection makes the distribution of feature error sensitive to the variation of points so that it helps to improve the calibration performance.

2) Space Constrains for Key Points: As discussed that the loss function is more sensitive to rotation than translation, we propose to apply projection error model to analyze the translation and find these key points among the approximated edge points if it has a certain misalignment in 2D image domain by adding small noise on the translation.



(a)

Fig. 3: The relationship on sensitivity of translation in each direction and a set of sample radiuses with 0.01m offset value on Z direction for illustration.

Given a 3D point $\mathbf{P} = (X, Y, Z)$ in camera coordinate, its image position $(u, v)^T$ can be computed. By adding small noise ($\triangle\mathbf{t} = [\triangle t_x, \triangle t_y, \triangle t_z]^T$) only on translation part, its image position $(u', v')^T$ is changed to

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \frac{f\cdot(X+\triangle t_x)}{Z+\triangle t_z} + u_0 \\ \frac{f\cdot(Y+\triangle t_y)}{Z+\triangle t_z} + v_0 \end{bmatrix}. \tag{4}$$

where f, u_0, v_0 are the intrinsic parameters of a camera.

Therefore, the pixel displacement error caused by the translation noise can be computed as:

$$\begin{aligned} d_{dis} &= \sqrt{(u-u')^2 + (v-v')^2} \\ &= \begin{cases} f \cdot \frac{\sqrt{R^2 - Z^2}\cdot\|t_z\|}{Z(Z+t_z)} & [0,0,\triangle t_z]^T \\ f \cdot \frac{\sqrt{R^2 - Z^2}\cdot\|t_z\|}{Z(Z+t_z)} & [\triangle t_x,0,0]^T or [0,\triangle t_y,0]^T \end{cases} \end{aligned} \tag{5}$$

where $R = \sqrt{X^2+Y^2+Z^2}$ is the distance between point $\mathbf{P}$ and the camera center.

From the equations, it shows that the key points often appear on the two sides under a certain radius distance. Figure 3 shows the curves when the offset is set to a certain value. For example, if we want to estimate a small translation, e.g. $0.01m$ along Z-axis and $0.01m$ along both X- and Y-axis, and simultaneously it is assumed that 1 pixel offset will cause obvious intensity differences which makes it sensitive for better estimation, we have to use the points within the corresponding radius of about 10 meters.

C. Optimization

Any local optimization methods (e.g. LevenbergMarquardt) for this loss function may result in a local minimum due to its double matching complexity. Therefore, we proposed to use CMA-ES algorithm (Covariance Matrix Adaptation Evolution Strategy) [14] which is an evolutionary algorithm special for difficult non-linear non-convex optimization problems in continuous domain for solving our problem. In our task, The initial variance in each dimension for this optimizer is set as the noise variance.

D. Efficient Random Selection for Extrinsic Calibration

Computing over a long sequence of frames to enrich distinguished scenarios is extremely computational expensive. In order to improve the speed as well as to reduce the

Algorithm 1 Efficient Random Selection for Extrinsic Calibration Algorithm

Require: $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots \mathbf{p}_m\}$, $\mathbf{I} = \{I_1, I_2, \cdots I_m\}$, $\mathbf{T} = \{T_1, T_2, \cdots T_{m-1}\}$, $RndFrames$, $Iters$, $\mathbf{X}_0$

Ensure: extrinsic parameter $\widehat{\mathbf{X}}$ between Lidar and camera

1: $\mathbf{P_n} = SelectFramePairs(\mathbf{P}, \mathbf{I}, \mathbf{T})$
2: **for** $i = 1 : Iters$ **do**
3: $\quad \mathbf{FP} = randomSelectFrames(\mathbf{P_n}, RndFrames)$
4: $\quad \mathbf{T}_i = cmaes(\mathbf{FP})$
5: **end for**
6: $\widehat{\mathbf{T}} = robustRefineKsi(\mathbf{T}, Iters)$

memory consumption, an effective Random Frames Selection for extrinsic calibration algorithm is proposed. Rather than taking all the frames for optimization, we randomly take parts of the frames for optimization and repeat this process for a fixed number. Then, final result is the average value of all the trials. The main steps of the proposed Random Frames Selection algorithm is summarized in Algorithm 1.

Given point clouds and corresponding images, initialize the repetitive iterations and random numbers of frames as well as the initial extrinsic parameters. In the first *SelectFramePairs* step, frame pairs are selected consecutively in a sliding window fashion by the constrains Eq. (2) and Eq. (3), where $\mathbf{P_n}$ is a structure contains image pairs, Lidar points as well as the relative poses in camera coordinate. In next iteration step, *randomSelectFrames* firstly divides the frame pairs into *RndFrames* sets and then randomly choose one frame pair in each set, and lastly gather the required number of frame pairs to compute the loss function. Finally, all *RndFrames* frames pairs are put into *cmaes* optimizer. After all the iterations, *robustRefineKsi* post-processes the optimization results by taking the average values after omitting the outliers whose values are over 3σ.

V. Experimental Results

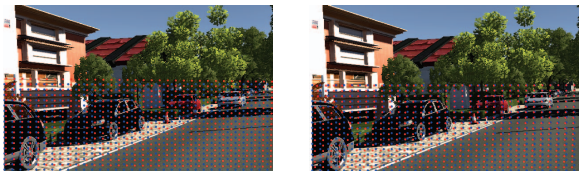
A. Experiments Settings

KITTI odometry dataset [15] is used to verify the effectiveness and robustness of the proposed approach. Specifically, sequence **00** is selected here because it includes over 4000 frames which can provide enough frames for our frame pair selection. The proposed approach is implemented under MacOs in Matlab 2015b environment on a standard laptop with 16GB RAM and Intel Core i7, 2.2GHz. Note that the demand for the speed of calibration is not strict because this calibration can be operated by a single thread in the backend.

In addition, the geodesic distance [16] which is used for measuring the noise added on rotation and translation is defined as

$$d(P, Q) = (\| \log(A_1^T A_2) \|_F^2 + \|b_2 - b_1\|_F^2)^{1/2}, \quad (6)$$

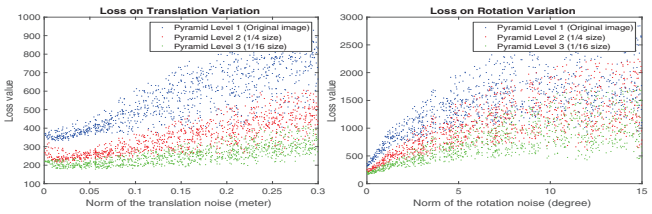
where P and Q are the two transformation matrices, A and b are the rotation matrix and translation vector respectively. The $\| \cdot \|_F$ is the Frobenius norm. Note that the units for both parts are not compatible, we just exploit either rotation or translation measurement for better evaluation.



(a) 2D misalignment by adding rotation noise.

(b) 2D misalignment by adding translation noise.

Fig. 4: Different sensitivities on rotation and translation. Blue points are projected by ground truth extrinsic transformation matrix while red ones are projected by contaminated transformation matrix.



(a) (b)

Fig. 5: Experimental results on one Virtual KITTI sequence. Variation trend of loss with different level noise on translation (a) and rotation (b). X-axis represents the norm of the noise in translation and rotation respectively, and Y-axis represents the loss value.

B. Sensitivity Evaluation on Rotation and Translation

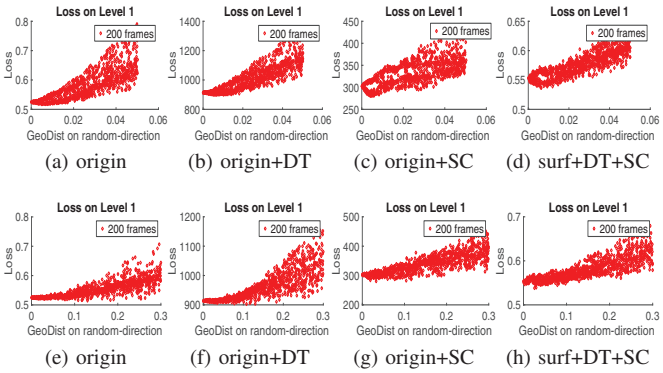
To evaluate the sensitivity on rotation and translation to Eq. (1) in controlled circumstance, Virtual KITTI dataset [17] is employed. The ground truth calibration parameters are set manually and Lidar points cloud are generated by transforming the ground truth 3D points in camera coordinate to Lidar coordinate.

Figure 4 shows qualitative results that the translation estimation is more difficult than rotation. From Fig. 4(a), the misalignment is obvious by adding small rotation noise (e.g., 1°) and is also uniformly distributed over the whole image. However, this is totally different for the case of translation in Fig. 4(b) with a small translation error (e.g., $0.1\,\mathrm{m}$) where the misalignment is extremely small for the points far from the camera but is relatively larger for the points close to the camera.

To illustrate the numerical results, the variation trend of loss is plotted in the following.

1) Evaluation for Translation Estimation: For translation, normalized direction vectors $\triangle\mathbf{t}$ are randomly generated and the amplitude λ of $\triangle\mathbf{t}$ is generated uniformly in the interval $[-0.3\,\mathrm{m}, 0.3\,\mathrm{m}]$, where $\mathbf{t}$ is the ground truth translation. Finally the noisy translation vector is set as $\hat{\mathbf{t}} = \mathbf{t} + \lambda \cdot \triangle\mathbf{t}$. In addition, we keep the rotation matrix noise free in this experiment. Fig. 5(a) displays the change of the loss function by adding different levels of noise on the translation vector. The loss increases with the increasing image size. At the original image level, the loss increases slightly with the increasing noise. Furthermore, this increasing trend becomes even weaker on level 2 and 3.

2) Evaluation for Rotation Estimation: The similar way is used on the rotation. Normalized direction vector $\triangle\mathbf{r}$ are



(a) origin (b) origin+DT (c) origin+SC (d) surf+DT+SC

(e) origin (f) origin+DT (g) origin+SC (h) surf+DT+SC

Fig. 6: Comparison on the trend of the loss on different methods. The first row only compares the influence on rotation with 2° variance and the second row only compares the translation influence with 0.3m variance. (a)(e) represent for the original intensity matching method as Eq. 1; (b)(f) represent for the original method with distance transform (DT) to find the points close to the edges within 1 pixel; (e)(g) represent for original method under space constrains (SC) with 0.02m along xyz axis; (d)(h) for surf based method with both distance transform and space constrains.

randomly generated and angles θ are uniformly generated in the interval $[-15^\circ, 15^\circ]$. The noisy rotation matrix is $\hat{\mathbf{R}} = \mathbf{R} \cdot \triangle\mathbf{R}$, where $\mathbf{R}$ is the ground truth rotation matrix. Fig. 5(b) shows the variation trend for rotation matrix where the loss increases significantly with the increasing noise. Particularly, the loss achieves the minimum value when the noise is zero for all three pyramid images levels.

C. Evaluation on Loss Function

In order to illustrate the changes made by different approaches for the loss function, a set of comparisons using 200 selected pairs of frames with either rotation or translation noise are shown in Fig. 6.

Compared in the first row for rotation in Fig. 6(a)-(d), each dot cloud shares similar overall trend which means that the capability of finding rotation solution is almost the same, and the sharpness of the left bottom area also appears similar which means that rotation accuracy is very close to each other. This shows that the rotation is naturally sensitive to the noise no matter what method is used.

However, compared in the second row for translation in Fig. 6(e)-(h), the overall trend of each dot cloud is differently. The methods of distance transform (DT) for approximate edge points and space constrains (SC) for key points selection affect performance respectively compared to the original image intensity method, while when all the approaches are used, the overall trend of loss to be more compact, making the translation solution more convergent. Additionally, its left bottom area is more sharp than others so that it can obtain a better translation solution.

D. Optimization Results

This experiment shows the improvement on our algorithm compared to the simple intensity-matching based method used in [5] which shows outstanding performance compared to other state-of-the-art methods. We take it as the **Baseline**. All tests are computed on 500 selected pairs of frames by the constrains in Eq. (2) and Eq. (3). The rotation and translation standard variance on the ground truth are 2° and 0.1 meters respectively. The simulated initial solutions with the noise are generated in the way as in V-B. The same 100 initial solutions are used and the average result is returned for each method. For our proposed algorithm, it uses 10 iterations and 10 pairs of frames randomly selected in each iteration. When the proposed algorithm is used, the average computation time consuming for each solution is about 10 minutes, which is 5 times faster than **Baseline**. However, it is possible to accelerate by parallel technique e.g. GPU configuration.

Our algorithm takes two variants for different features. One is called **Ours+Patch** with 15×15 patch size on image intensities while the other is called **Ours+SURF** but with **SURF** descriptor of also 15×15 patch size. Table I shows the quantitative performance. Our algorithm enhances the translation estimation than **Baseline** while the rotation estimation is still worse when patches are used. This is the reason that more frames are used in **Baseline**, making the rotation less ambiguous. However, when the **SURF** features are used, the performance of rotation estimation is improved and comparable to that of **Baseline** but has higher efficiency due to the random selection strategy.

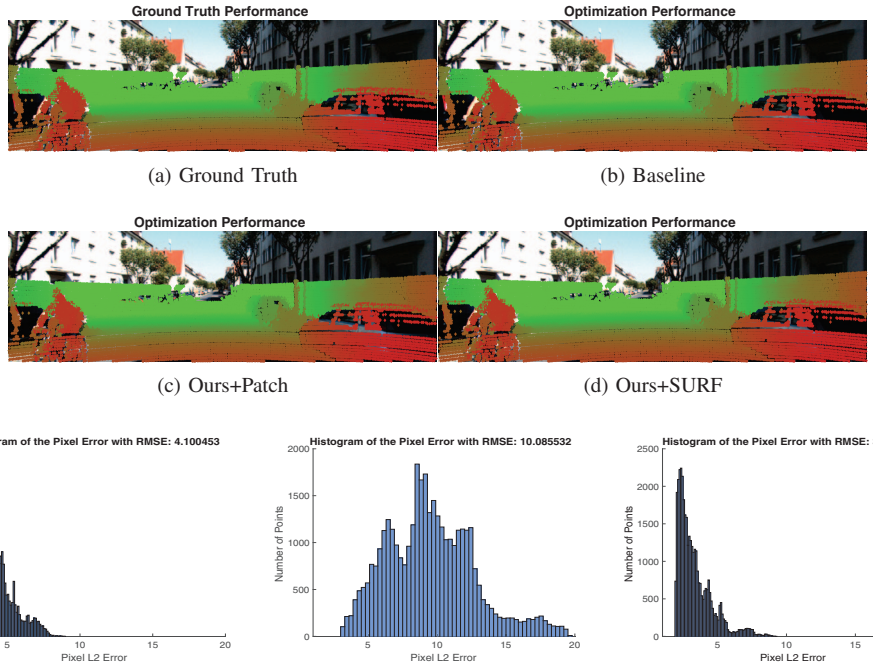
Figure 7 shows the qualitative calibration results by aligning the Lidar points onto the image using the optimized transformation matrix. For better illustration, the results are shown with the original dense Lidar points instead of the sparse Lidar points. Note that, **Ours+SURF** shows some more precise details than **Baseline** and **Ours+Patch** such as the man riding the bike as well as the left and right tree. In the bottom Fig. 7(e)-(g), the histograms of the pixel error show that the accuracy of rotation weights much heavier than that of translation. For example, **Ours+Patch** performs worse than **Baseline** because of the worse rotation estimation (shown both in the Table I and image), regarding the improvement of the translation accuracy. But when **Ours+SURF** achieves almost the same accuracy of rotation as **Baseline**, the improvement of the translation estimation will enhance the performances.

VI. Conclusion

This paper tackles extrinsic calibration between monocular camera and Lidar with sparse 3D points. To improve the original feature error minimizing loss function, key points selection and an efficient optimization strategy are proposed. Our method is potential to handle both sparse and dense Lidar measurement under a unified framework. Further bundle adjustment for jointly optimization the relative poses and extrinsic parameters will be considered for robust estimation. SLAM techniques may be applied to address the cases where rigid motion objects exist in the scenes.

ACKNOWLEDGMENT

The work is support by National Nature Science Foundation of China under Grant No. 61375050, Grant No.



(a) Ground Truth (b) Baseline

(c) Ours+Patch (d) Ours+SURF

(e) Baseline with Pixel RMSE: 4.1 (f) Ours+Patch with Pixel RMSE: 10.1 (g) Ours+SURF with Pixel RMSE: 3.6

Fig. 7: Calibration results of the compared methods. For better illustration, the results are shown with the original dense Lidar points instead of the sparse Lidar points. The bottom row shows the histograms of the pixel errors in this frame for each method.

TABLE I: L2-Norm of Error Comparison on Different Methods

Method	**X**(m)	**Y**(m)	**Z**(m)	**Pitch**(°)	**Yaw**(°)	**Roll**(°)
Baseline	0.048 ± 0.046	0.065 ± 0.061	0.051 ± 0.049	0.332 ± 0.235	$\mathbf{0.204} \pm 0.170$	$\mathbf{0.242} \pm 0.191$
Ours+Patch	0.024 ± 0.023	0.038 ± 0.037	0.025 ± 0.025	0.636 ± 0.443	0.524 ± 0.406	0.548 ± 0.421
Ours+SURF	$\mathbf{0.023} \pm 0.023$	$\mathbf{0.038} \pm 0.037$	$\mathbf{0.023} \pm 0.023$	$\mathbf{0.283} \pm 0.230$	0.313 ± 0.250	0.353 ± 0.285

91220301 and Grant No. 61420106007, and funded in part by Australian Research Council Grants of DP120103896, LP100100588, DE140100180 ARC Centre of Excellence on Robotic Vision (CE140100016) and NICTA (Data61). The first author is funded by the Chinese Scholarship Council (CSC) to be a joint PhD student from NUDT to ANU.

REFERENCES

[1] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, 2013, pp. 24–28.

[2] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Toronto, Canada, July 2012, pp. 2053–2059.

[3] Z. Taylor and J. Nieto, "Automatic calibration of lidar and camera images using normalized mutual information," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[4] T. Scott, A. A. Morye, P. Piniés, L. M. Paz, I. Posner, and P. Newman, "Choosing a time and place for calibration of lidar-camera systems," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4349–4356.

[5] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.

[6] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, 2001.

[7] L. Tamas and Z. Kato, "Targetless calibration of a lidar-perspective camera pair," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 668–675.

[8] W. Zhao, D. Nister, and S. Hsu, "Alignment of continuous video onto 3d point clouds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1305–1318, 2005.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[10] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–652.

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

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

[13] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 265–270, 2003.

[14] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3354–3361.

[16] X. Duan, H. Sun, and L. Peng, "Riemannian means on special euclidean group and unipotent matrices group," *The Scientific World Journal*, vol. 2013, 2013.

[17] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *CVPR*, 2016.