Motion Guided LiDAR-camera Self-calibration and Accelerated Depth Upsampling for Autonomous Vehicles

Juan Castorena
Ford Motor Company
22 PUBLICATIONS **102** CITATIONS

Gaurav Pandey
Ford Motor Company, Palo Alto, California
70 PUBLICATIONS **671** CITATIONS

Motion Guided LiDAR-camera Self-calibration and Accelerated Depth Upsampling for Autonomous Vehicles

Juan Castorena, Gintaras V. Puskorius and Gaurav Pandey *

Abstract

This work proposes a novel motion guided method for targetless self-calibration of a LiDAR and camera and use the re-projection of LiDAR points onto the image reference frame for real-time depth upsampling. The calibration parameters are estimated by optimizing an objective function that penalizes distances between 2D and re-projected 3D motion vectors obtained from time-synchronized image and point cloud sequences. For upsampling, a simple, yet effective and time efficient formulation that minimizes depth gradients subject to an equality constraint involving the LiDAR measurements is proposed. Validation is performed on recorded real data from urban environments and demonstrations that our two methods are effective and suitable to mobile robotics and autonomous vehicle applications imposing real-time requirements is shown.

1. Introduction

A broad range of mobile robotic (e.g., autonomous vehicle) applications exploit fusion of information from sources of multiple modalities to reduce the uncertainties or limitations inherited when any of these sources are used individually. One of the most popular and effective complementary pair of sensors for robotic perception are cameras capturing RGB or IR information and depth systems (e.g., LiDAR) providing direct 3D information of the environment. Other systems that are heavy on computation but cheap in sensing have also been devised to obtain 3D information. For example some works [1–3] rely on learned models for depth prediction using deep networks and mono, stereo or multiple passive illumination camera sensors. However, the performance of these significantly degrade in environments with poor illumination and extremes of structural variability (e.g., tunnels or hallways, vegetation). To overcome these passive sensing based limitations, methods that fuse camera and LiDAR information for depth estimation have been proposed [4–7]. These in general exploit the information gathered through the sensor measurements and uses priors or learned geometrical constraints that dictate how to fuse the information. For example, the early work proposed by Diebel and Thrun [4] demonstrated the idea of constraining a Markov random field (MRF) based optimization to yield depth-maps with edges that co-occur with those from the camera image, assuming modalities have been co-registered. He et al. [5] proposed an image guided filtering approach to guide and improve edge selection in the reconstruction of upsampled depth. The work of Lu and Forsyth [6] proposed a segmentation based approach where sparse LiDAR points are used to segment the image followed by smoothing of the sparse points within each of these segments to generate depth. The sparse-promoting regularization based work of Ferstl et al. [7] proposed to use a weighted total generalized variation (TGV) formulation to promote consistency with edges from a high-resolution image in the depth upsampled reconstruction. More recent works have also

*J. Castorena, G. Puskorius and G. Pandey are at the Robotics and AI laboratory, Ford Motor Company, Dearborn, MI.

exploited temporal information to further enhance reconstruction. For example, the work of Kamilov and Boufounos [8] formulated an inverse problem that uses a low-rank penalty on groups of related depth patches found from corresponding image patch similarity. The work of Degraux et al. [9] instead learns group-sparse representations and uses total variation (TV) regularization for depth upsampled reconstruction.

Methods that rely on fusing (in the optimal sense) information between multiple sensing modalities tend to be in general more robust than its individual sensor usage counterparts. Fusing, for example, LiDAR and camera measurements for depth upsampling predictions makes the reconstruction more invariant to illumination changes and extremes of environmental structure detail. However, most of these assume perfect alignment between data from the multiple modalities, an event that rarely occurs in practice. In fact, most fused based depth reconstruction methods are very sensitive to even small extrinsic miss-calibrations between the LiDAR-camera system. To this end, one of the first effective works to correct small miss-calibrations automatically and in an online fashion computes edges simultaneously in both modalities and uses a cost function of correlation to measure alignment between edges as in Levinson and Thrun [10]. The work of Pandey et al. [11] uses instead a mutual information function to measure the similarity between the reflectivity measurements from a LiDAR and the intensities from camera images assuming availability of reliable reflectivity measurements from a LiDAR. Other more recent automatic methods like Castorena et al. [12] considers a joint alternate optimization between data alignment and depth-upsampling reconstruction based on the idea that the better the alignment is the more accurate the depth reconstruction and vice-versa. Another method by Scott et al. [13] exploiting temporal information automatically attends specific time and place instances and formulates an objective function optimized to find the rigid body parameters that align the attended data.

This paper is focused on both the problem of depth up-sampling reconstructions for robotic applications imposing real-time requirements and on the problem of extrinsic self-calibration of a camera-LiDAR system. The contributions of the work, detailed in Section 2, are two fold: (1) our new depth upsampled reconstruction method formulated as a sparse edge promoting objective optimized with accelerated gradient descent to satisfy real-time requirements and (2) a new extrinsic self-calibration method based on alignments of motion from spatio-temporal information in both modalities and without position sensor requirements. Section 3 presents experimental validating results to depth upsampling and motion based registration with real recorded data from the KITTI benchmark and finally Section 4 concludes our findings.

2. Proposed Approach

This section describes the intuition behind the proposed motion guided method for the targetless extrinsic self-calibration of a multi-modal system composed of a LiDAR and a camera pair. Self-calibration is formulated as the problem of automatically finding the 6 DOF rigid body transformation of relative pose that registers or aligns data captured on-the-fly through the multiple sensing modalities and without any specific target requirements. Following self-calibration, a method that operates in real-time to upsample the sparse depth resulting from the re-projection of sparse LiDAR points into the camera reference is presented along with its formulation and its corresponding optimization.

2.1. Motion based self-calibration

The presented method for the extrinsic self-calibration of a LiDAR-camera system computes 6 parameters: a roll, pitch and yaw rotations and x, y and z translations characterizing a 3D rigid body transformation. The direction in which these parameters are computed is LiDAR-to-camera while the camera-to-LiDAR direction can be computed trivially given the former. The method works with any LiDAR-camera position configuration as long as there are overlapping fields of views (FOV) between the

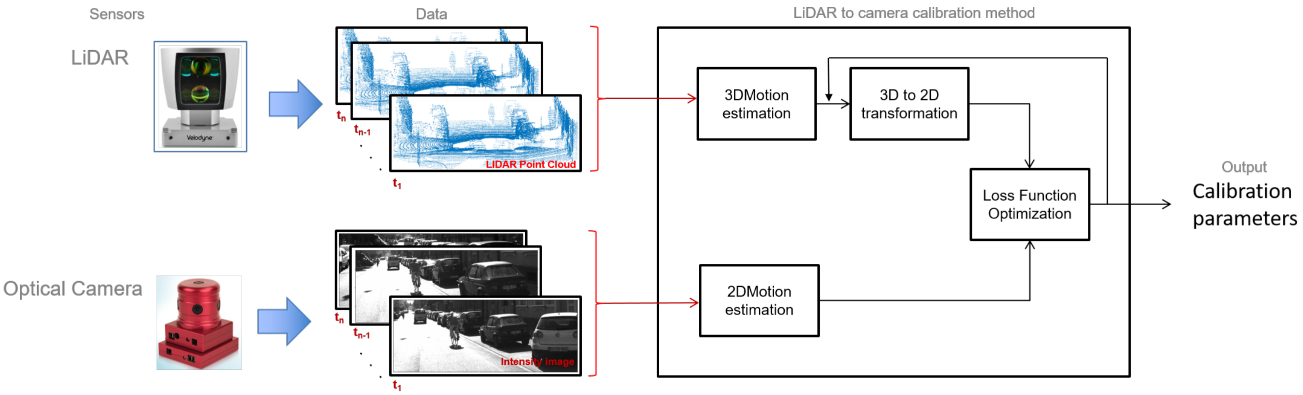


Figure 1: Motion guided LiDAR-camera self-calibration schematic.

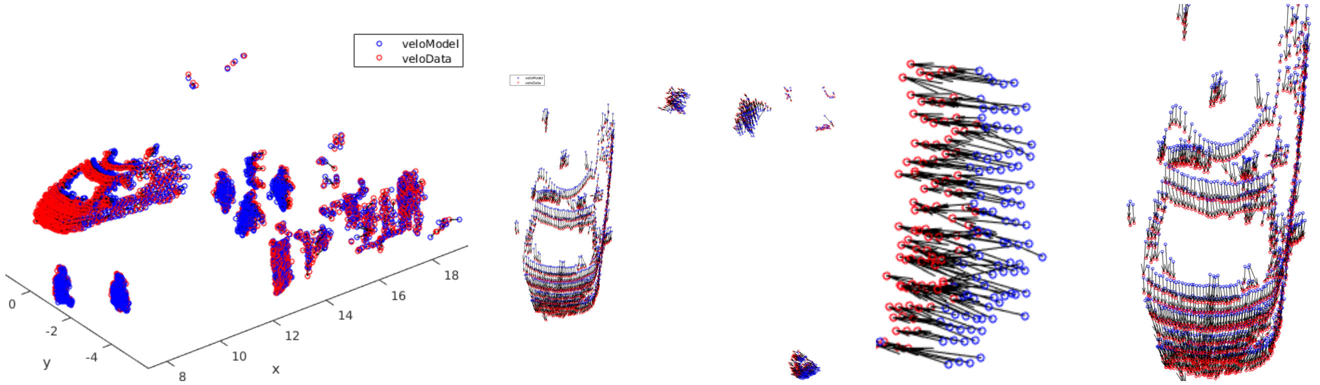
sensors being registered.

The idea of the proposed method is simple: under alignment scene motion in the modality being registered should follow scene motion in the reference modality (e.g., camera). The advantages of registering modalities from the motion cues is two-fold: (i) it avoids the challenges related to modalities measuring different units (e.g., intensity, depth, etc) and (ii) it constraints the alignments both spatially and temporally. The procedure describing this idea and summarized in Figure 1 uses a sequence of independent $L + 1$ time-synchronized 2D image and 3D point cloud pairs from camera and LiDAR correspondingly as input. These sequences are then split into sequential pairs to obtain a total of L pairs of 2D and 3D motion vectors computed independently for each modality. Given these, we optimize the self-calibration objective function formulated as shown in Eq. (1):

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\mathbf{SE}(3)} \underbrace{\left\{ \frac{1}{L}\sum_{l=1}^{L} \left\| w(\mathbf{Y}^{l}_{\text{cam}}) - w(\mathbf{Y}^{l}_{\text{lidar}},\boldsymbol{\theta}) \right\|_{\ell_2} \right\}}_{C(\boldsymbol{\theta})}. \tag{1}$$

Here, $w \in \mathbb{R}^{N\times 2}$ corresponds to the motion vector in the camera image plane defined by $w(x,\cdot) = [u(x,\cdot), v(x,\cdot)]^{\mathrm{T}}$ where $u \in \mathbb{R}^{N}$ and $v \in \mathbb{R}^{N}$ are the horizontal and vertical motion operators, respectively. The term $\mathbf{Y}^{l}_{\text{cam}} \in \mathbb{R}^{N\times 2}$ represents the sequential image pair $[\mathbf{x}^{l}, \mathbf{x}^{l+1}]$ where $\mathbf{x}^{l} \in \mathbb{R}^{N}$ is the column-wise vectorization of an image at the index l describing a time instant. Similarly, the term $\mathbf{Y}^{l}_{\text{lidar}} \in \mathbb{R}^{M\times 3\times 2}$ represents the corresponding sequential 3D pair of point clouds composed of $[\mathbf{X}^{l}, \mathbf{X}^{l+1}]$ where $\mathbf{X}^{l} \in \mathbb{R}^{N\times 3}$ is the point cloud at time instant l. In words, Equation (1) finds the optimal parameters that minimizes the distance (in the ℓ_2 sense) between a sequence of L motion vectors computed from the camera and LiDAR modalities, independently. One thing to note here is that the 3D motion vectors are re-projected into the image plane using a state of the 6DOF parameters $\boldsymbol{\theta} \in \mathbf{SE}(3)$ and the camera intrinsics and thus Eq. (1) computes motion distances in the image plane. On another side note we would like to mention that motion distance comparisons are only performed at pixel locations occupied by valid re-projection of LiDAR measurements and that each motion vector, regardless of modality, is normalized to be of unit ℓ_2-norm.

Motion for the image sequence can be estimated here by any state of the art learning based optical flow method suitable to images such as Dosovitskiy et al. [14]. However, given that current learnable optical flow methods are not invariant to camera model and type, which is highly inconvenient to our camera-LiDAR registration problem, we resort to the total variation (TV) optical flow method of Zach [15]. TV estimates motion by imposing piece-wise constant motion field constraints and has the benefit that is



(a) Side-view

(b) Top down view

(c) Pedestrian motion zoomed

(d) Vehicle motion zoomed

Figure 2: Illustrative example of the solution of 3D motion flow given a reference and target point cloud.

camera type and model invariant. For the point cloud sequence, 3D motion is estimated by sequentially taking pairs of consecutive point-clouds and computing motion in each pair along the sequence. The problem of estimating 3D motion in pairs of point clouds is posed here as a vector difference between points in the point cloud at time instant l with their corresponding point associations in the point cloud at time instant $l+1$. To compute point associations between point clouds a graph-based method that optimizes for an injective mapping between points is used. Such an assignment is formulated as the optimal transport problem that minimizes the sum of distances of point associations described in Eq. (2).

$$\hat{\mathbf{A}} = \underset{\mathbf{A}\in\mathcal{A}}{\arg\min} \left\{ \sum_{i=1}^{N}\sum_{j=1}^{M} \mathbf{A}_{i,j}\mathbf{G}_{i,j} \right\} \tag{2}$$
$$s.t. \quad \sum_{j=1}^{N} \mathbf{A}_{i,j} = 1 \quad \forall i$$
$$\sum_{i=1}^{M} \mathbf{A}_{i,j} \leq 1 \quad \forall j$$

Eq. (2) optimizes an association of points $\mathbf{x}_i \in \mathbf{X}^l$ to points $\mathbf{x}_j \in \mathbf{X}^{l+1}$ based on the cost of the Euclidean distance assignment and injective mapping constraints. The association is described by means of the matrix elements $\mathbf{A}_{i,j} \in \{0,1\}$ that associates points indexed by i to j via a non-zero entry. The matrix $\mathbf{G} \in \mathbb{R}^{+N\times M}$ is the Euclidean distance matrix (EDM) consisting of elements computed as in Eq. (3) as:

$$\mathbf{G}_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|_{\ell_2}^2 \tag{3}$$

i.e., the pair-wise distance between points $\mathbf{x}_i \in \mathbf{X}^l$ and $\mathbf{x}_j \in \mathbf{X}^{l+1}$. The first constraint in Eq. (2) forces one point in $\mathbf{X}^{l+1}$ to be associated to only one point in $\mathbf{X}^l$ while the second constraint forces a point in $\mathbf{X}^l$ to be assigned to at most a point in $\mathbf{X}^{l+1}$, thus constraining the solution to be an injective map. There are multiple methods to solve the problem in Eq. (2) suitable to the specific variation in its constraints. Here, we choose the iterative algorithmic solution of Bertsekas [16]. A representative example illustrating the result of application of this algorithm to a pair of two consecutive points clouds is shown in Figure

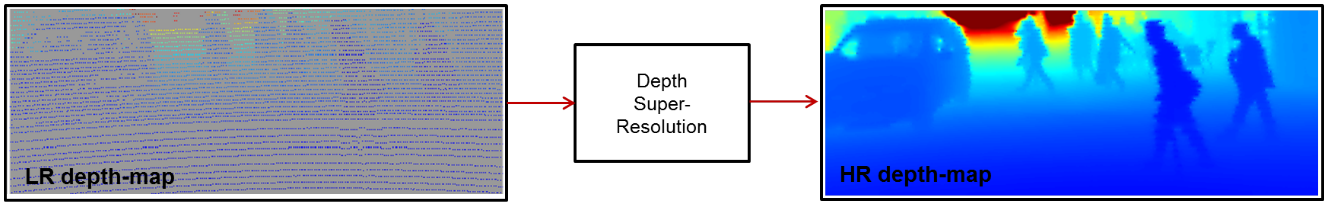


Figure 3: Accelerated depth super-resolution reconstruction schematic.

2 only for points that underwent motion. Here, blue points represent the point cloud $\mathbf{X}^l$, red points represent point cloud $\mathbf{X}^{l+1}$ and black arrows represent the motion vectors between them. Note here that no point in $\mathbf{X}^l$ is assigned to more than one point in $\mathbf{X}^{l+1}$. In general, we find that 3D motion computed in such way is effective and suitable to at least our registration problem and thus we use it throughout this paper to compute 3D motion.

To minimize the cost function in Eq. (1), we use the simplex optimization method from Nelder-Mead described in Lagarias et al. [17]. Other optimization methods could be used instead, however, it was decided to use this method because it is differentiation-free. Also, note that the chosen method is locally convergent and thus requires a rough initialization restricting search to a small neighborhood.

2.2. Depth Upsampling

The proposed method for depth upsampled reconstructions assumes availability of the extrinsic calibration parameter estimates obtained as described in Section 2.1. Reprojection of the LiDAR measurements into the image plane using the found calibration parameters and the camera intrinsics results in a sparse depth-map with defined values spread non-uniformly over the image plane. The tackled problem resolves the question of how do we upsample or fill in the missing values in a sparse depth map quickly enough to make it feasible for real-time applications. In other words, how does it go from the sparse to the upsampled depth map shown in the left and right side shown in Figure 3, respectively. Note that the left sparse and right upsampled depth map in Figure 3 represents missing information with gray; and the remaining color spectrum as the actual depth.

The problem is formulated as the optimization of an objective function that minimizes an anisotropic ℓ_1-norm on gradients subject to equality constraints with the sparsely distributed pixels containing depth values. Mathematically this can be defined as shown in Eq. (4):

$$\widehat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{X}_\Omega}{\arg\min} \underbrace{\sum_{k=\{x,y\}} \|\mathbf{D}_k\mathbf{x}\|_{\ell_1}}_{f(\mathbf{x})}, \ \text{s.t.}\ \mathbf{x}_{\Omega'} = \mathbf{y}_{\Omega'} \tag{4}$$

Here, Ω' denotes the subset of pixels in the sparse depth-map $\mathbf{y} \in \mathbb{R}^N$ with a depth measurement from the LiDAR re-projection and $\mathbf{x}$ is the upsampled depth. In Eq. (4), the gradient $\mathbf{D} : \mathbb{R}^N \rightarrow \mathbb{R}^{N\times 2}$ is the first order forward difference defined point-wise as shown in the Eq. (5):

$$[\mathbf{D}\mathbf{x}]_n = \begin{pmatrix} [\mathbf{D}_x\mathbf{x}]_n \\ [\mathbf{D}_y\mathbf{x}]_n \end{pmatrix} = \begin{pmatrix} [\mathbf{x}]_{n+N_y} - [\mathbf{x}]_n \\ [\mathbf{x}]_{n+1} - [\mathbf{x}]_n \end{pmatrix} \tag{5}$$

were $\mathbf{D}_x$ and $\mathbf{D}_y$ denotes the horizontal and vertical components, respectively. Note that the input definitions of Eq. (4) and (5) assume a column-wise vectorization of a 2D image of size $N_y \times N_x$. The interpretation of Eq. (4) is as follows: $f(\mathbf{x})$ promotes depth reconstructions that minimize the strength and occurrence of depth discontinuities (i.e., of sparse depth gradient) as imposed by the ℓ_1 norm denoted

Algorithm 1 Depth upsampling reconstruction

```
1: input: Sparse depth $\mathbf{y}$ and learning rate $\gamma > 0$.
2: set: iteration index $t \leftarrow 1$, momentum variable $q_0 \leftarrow 1$, initialization $\mathbf{s}^0 \leftarrow \mathbf{y}$ and index set of
   non-missing depth pixels $\Omega' = \text{supp}(y)$
3: repeat
4:     $\mathbf{z}^t_\Omega \leftarrow \eta_\tau(\mathbf{s}^{t-1}_\Omega - \gamma \mathbf{I}_\Omega \nabla f(\mathbf{s}^{t-1}))$
5:     $q_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4q^2_{t-1}}\right)$
6:     $\lambda_t \leftarrow (q_{t-1} - 1)/q_t$
7:     $\mathbf{s}^t_\Omega \leftarrow (1 + \lambda_t)\mathbf{z}^t_\Omega - \lambda_t \mathbf{z}^{t-1}_\Omega$.
8:     $t \leftarrow t + 1$
9: until: stopping criterion
10: set: $\widehat{\mathbf{x}}_{\Omega'} = \mathbf{y}_{\Omega'}$, $\widehat{\mathbf{x}}_\Omega = \mathbf{s}^t_\Omega$
11: return: Depth reconstruction $\widehat{\mathbf{x}}$
```

as $\| \cdot \|_{\ell_1}$ while the equality constraint ensures consistency in pixels with available depth values. Current commercially available LiDARs can offer range noise RMS at 2 cm thus supporting the high reliance of Eq. (4) on the re-projected measurements.

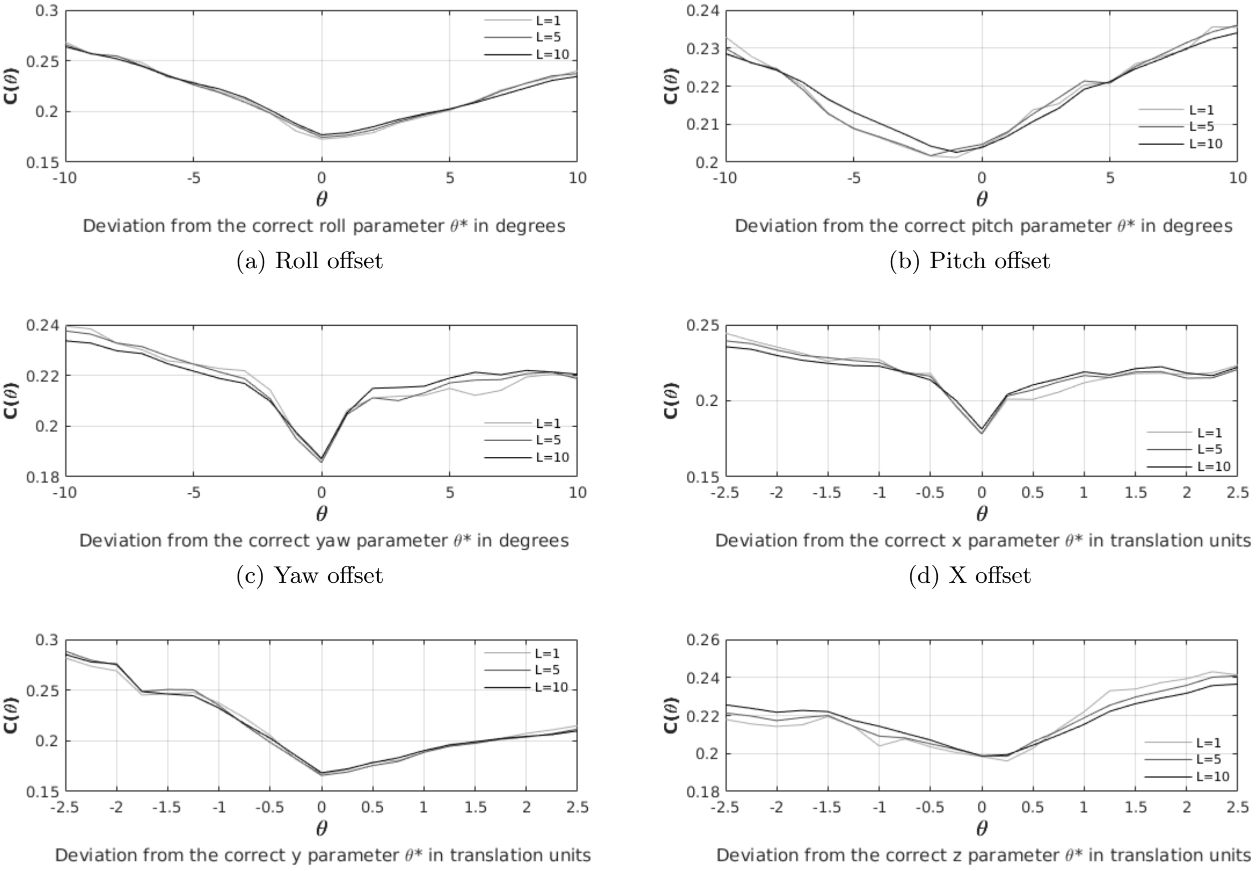
Since the cost function in Eq. (4) is convex and non-smooth, the accelerated gradient projection method of Nesterov [18] along with a proximal method for the non-smooth ℓ_1-norm is used. This proximal method is the non-linear soft-thresholding operator $\eta_\tau(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ with threshold τ defined element-wise as shown in Eq. (6):

$$[\eta_\tau(\mathbf{x})]_n = \text{sgn}([\mathbf{x}]_n)(|[\mathbf{x}]_n| - \tau)_+. \tag{6}$$

Such an operator promotes sparsity of the solution discontinuities at each iteration by collapsing values near zero to zero. With such an accelerated sparse promoting optimizer, the depth reconstruction achieves a rate of convergence of $O(1/k^2)$ (where k is the iteration number) due to the acceleration promoted by Nesterov's projection method. A summary of the algorithm is included in Algorithm 1 when, $\mathbf{I}_\Omega : \mathbb{R}^N \to \mathbb{R}^M$ is the rectangular matrix populated with 1's at entries indexed by Ω and zero otherwise. This matrix is thus a selector of pixels with missing depth values. In words, in every iteration of Algorithm 1 the depth values sparsely distributed along pixels propagate spatially to every pixel in the upsampled depth-map reconstruction provided this propagation satisfies the constraints imposed by the objective function. The way in which depth values are propagated is dictated by a minimization of the gradient of the objective function denoted by $\nabla f(x)$ and with a strength indicated by the learning parameter γ at each iteration. Such gradient based optimization and convex nature of the cost function guarantee that at each iteration it gets closer to convergence. The additional steps in lines 5,6 and 7 of algorithm 1 are Nesterov projections that further propagate depth values a little more with strength λ_t.

3. Experimentation

To validate the approaches, the KITTI benchmark [19] dataset is used. This data was collected with a vehicle outfitted with several perception and inertial sensors. The experiments use only the time synchronized data from the Velodyne HDL-64E 3D-LiDAR scanner and the left PointGray Flea2 grayscale camera. However, the approach we propose here for both motion guided registration and depth up-sampling can be scaled to multiple cameras/LiDARs configurations. The KITTI dataset provides the extrinsic calibration parameters for each sensor mounted on the vehicle. The parameters for the LiDAR and camera pair chosen for the experiments are also given with the dataset.



(a) Roll offset

(b) Pitch offset

(c) Yaw offset

(d) X offset

(e) Y offset

(f) Z offset

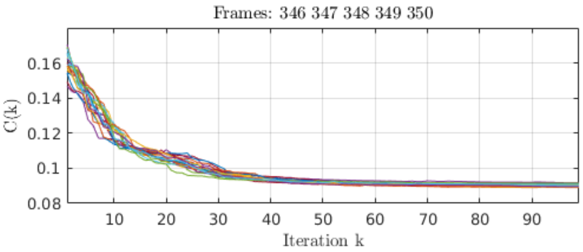
Figure 4: Effect of parameter offset in Eq. (1). a) Roll, b) Pitch, c) Yaw, d) X, e) Y, f) Z.

3.1. Motion guided self-calibration

The first experiment presented here on motion guided self-calibration shows the behavior of the objective function in Section 2.1 as a function of an offset from the true extrinsic parameters provided by KITTI. The data from KITTI used in this regard corresponds to a vehicle driving in an urban area. We manually choose corresponding image (converted to grey scale) and point cloud sequences of 2,6 and 11 consecutive frames that contain motion in the scene and evaluate the objective function in Section 2.1 by fixing 5 parameters at true value and offset one to different values and then repeat the process for all 6 DOF of roll, pitch, yaw and x, y, z. A representative example is shown in Figure 4, which includes plots of the behavior of Eq. (1) for a given parameter offset. It was observed that the proposed objective function shows a clear global minimum at the correct calibration parameters with a few minor local bumps along the way. These bumps are further smoothed as the number of frames L used in Eq. (1) is increased pushing our cost function to be both convex and smooth in the local neighborhood around the optimal value.

3.1.1 Convergence of the cost function

This experiment illustrates the convergence behavior of the simplex optimization of Eq. (1). For this purpose, the simplex optimization approach was initialized with 7 points where these were chosen ran-



(a) L=5 Motion vector pairs

Figure 5: Convergence as a function of iterations.

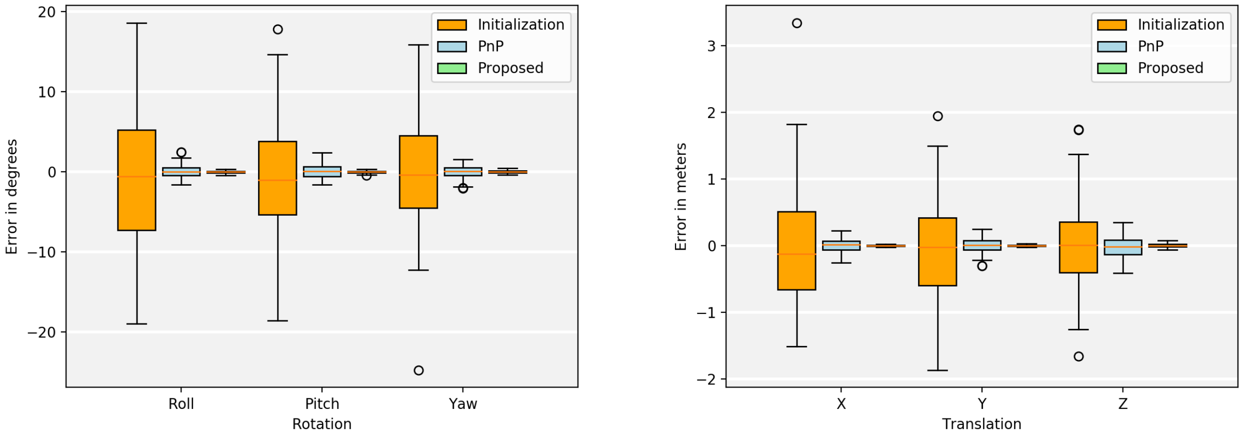
domly within 20% offset from the true. A representative example of the convergence of the optimization is illustrated in Figure 5, showing the first 100 iterations of 20 independent experiments for the $L = 5$ case. This convergence behavior is consistent throughout different scenes in an urban environment, driving behavior and number of motion vectors L. However, one of the limiting cases for convergence of the algorithm is when $L \to 1$ where along with cases of no vehicle nor scene dynamics results in motion vectors with no information to constrain data alignments. A side from these, convergence presented a rate behavior similar to that in 5 with prominent damping at 50 iterations.

3.1.2 Robustness of registration algorithm

This experiment shows the performance and robustness of the proposed motion based registration formulation against randomized initializations and compare this to the best possible registration from robust perspective n-point (RPnP) algorithm of Li et al. [20] when there is a human doing manual correspondences. A total of 100 experiments were conducted where in each registration using either the proposed method or RPnP a sequence of consecutive and time-synchronized 6 gray scale image/point cloud pairs where used (i.e., $L = 5$ for the proposed method). In each of the 100 independent registration trials, different scenes from an urban scenario with significant pedestrian traffic were randomly selected. The registration parameters of these trials were randomly initialized within offsets of $\pm 20^{o}$ for rotations and ± 3.5 m for translations from ground truth. Here, ground truth corresponds to the extrinsics provided from the KITTI benchmark.

For registration using the RPnP algorithm, a specialized user was presented with 6 image/point clouds pairs and 4 corresponding keypoints in each pair that were clearly visible were manually selected in both modalities (e.g., corners). After selection of keypoints, the user validated the selected 24 points per sequence, ran the RPnP algorithm and computed the root mean squared error (RMSE) of the result compared to the ground truth. The proposed method used in each of the 100 trials the same sequence of 6 image/point clouds pairs used in the corresponding trial for the RPnP.

Figure 6 summarize the performance results obtained from the experimentation with corresponding RMSE's listed in Table 1. The whiskers in the box-plots of Figure 6 corresponds to the range of the error while the boxplots shown in a color extend from the 1st to 3rd quartile of the error distribution. A total of three box-plots are shown per each of the six registration parameters corresponding to the random initialization labeled as ”initialization”, results of RPnP [20] algorithm labeled as ”PnP’ and the proposed method labeled as ”Proposed”. For these experiments a total of 100 trials where run for each of the registration parameters independently. Note here that in each trial not only the parameter plotted was randomized but all 6 DOF and that error is measured here as the ℓ_2-norm between result and ground truth. From Figure 6 and Table 1 we note that both algorithms were able to reduce the random initialization error to close to zero across all trials and for all 6 parameters. Both RPnP and the proposed



(a) Rotation

(b) Translation

Figure 6: Robustness of the motion guided registration algorithm. a) Rotation error results, b) Translation error results. Box-plots corresponding to the "Initialization" label is the random initialization distribution centered at ground truth, label corresponding to "PnP" is the calibration result of the RPnP method [20] while the label "Proposed" is the error performance of the proposed calibration method.

Table 1: RMSE comparisons.

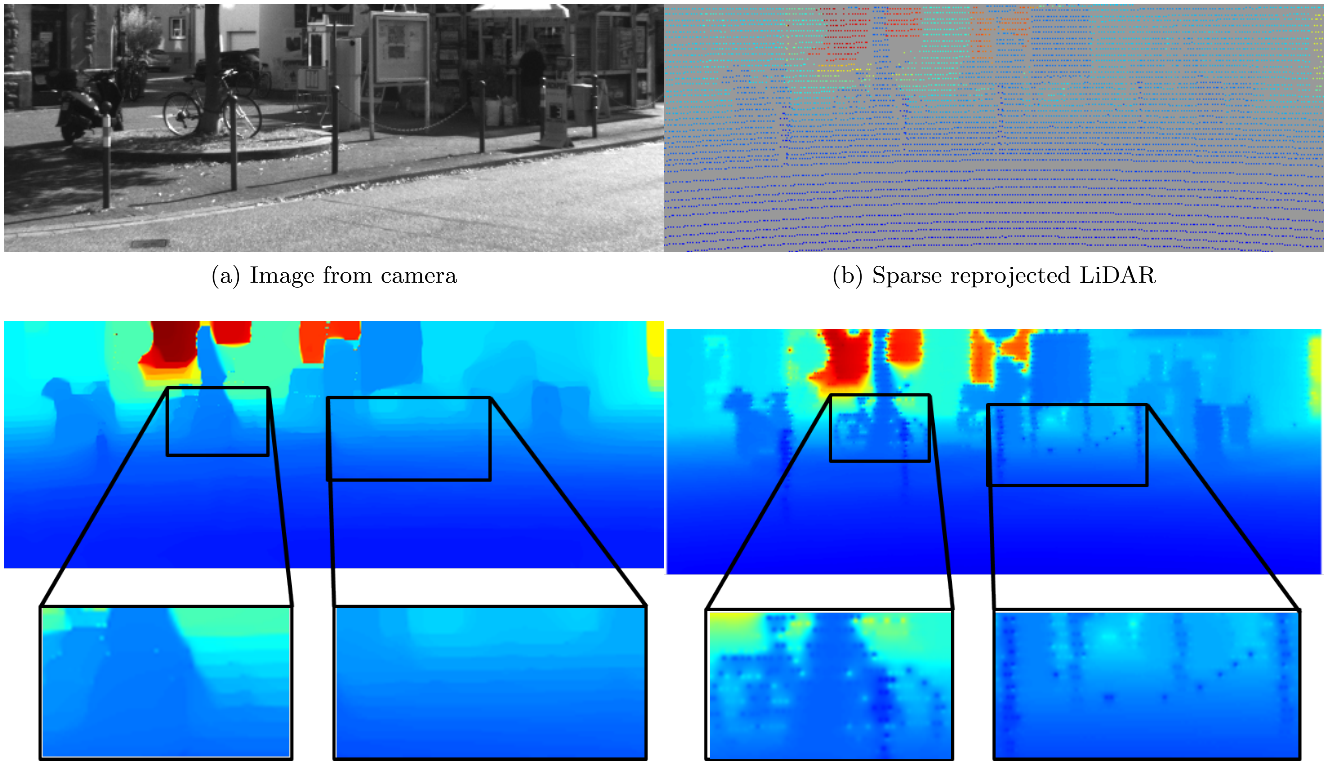
	Rotation (°)			Translation (m)		
Method	Roll	Pitch	Yaw	X	Y	Z
Initialization	8.39	7.37	7.02	0.86	0.80	0.68
PnP [20]	0.74	0.88	0.86	0.10	0.11	0.15
Proposed	**0.15**	**0.15**	**0.17**	**0.01**	**0.01**	**0.03**

algorithm achieved an overall rotation RMSE's of $< 1^{o}$ and < 15 cm for the translation. However, the proposed registration formulation outperforms the RPnP method by reducing error to within RMSE accuracies of $< 0.2^{o}$ of rotation and < 3 cm translation.

3.2. Depth super-resolution: qualitative analysis.

This experiment presents a qualitative analysis of the proposed depth upsampled reconstruction algorithm and compare it with the generalized total variation (TGV) approach described in Ferst et al. [7]. Throughout the experiments, the proposed reconstruction uses algorithm 1 with a learning rate $\gamma = 0.1$. Note that other γ values also work well. However, that value was chosen since it was found experimentally that it gives a good trade-off between depth-map reconstruction quality and convergence time.

To illustrate the performance comparison, Figure 7 shows the capabilities of our method to resolve finer details by using only the sparse depth from LiDAR in Figure 7.b as opposed to TGV that in addition uses the image in 7.a from the camera. In Figure 7.b the gray colored pixels represent pixels with missing depth measurements from the re-projection of LiDAR data while the remaining colors represent actual depth. Comparing Figures 7.c and d show that the new method is able to resolve finer details as further



(a) Image from camera

(b) Sparse reprojected LiDAR

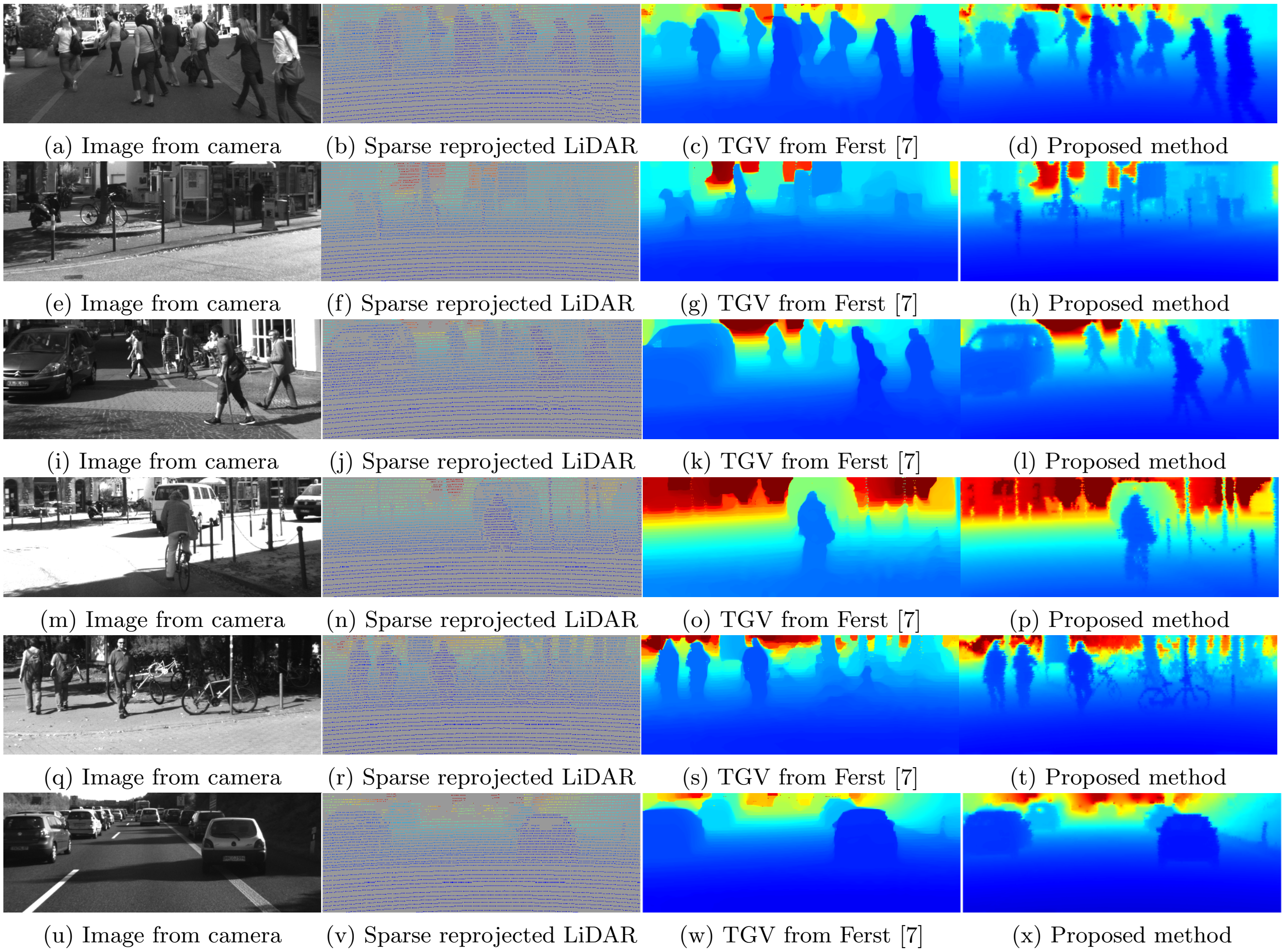
(c) TGV from Ferst [7]

(d) Proposed up-depth sample

Figure 7: Upsampled depth reconstruction comparison. Note in the zoomed patches in (c) and (d) showing a bike and a chain between two poles that our approach is capable of resolving finer details compared to TGV.

illustrated in the zoomed patches showing the bike and its wheels in the bottom left patch and the chain in between poles as seen in the bottom right patch whereas these are hardly distinguishable in the TGV method.

Figure 8 also illustrates six additional depth upsampled reconstruction examples each presented in a row. First column of Figure 8 represents an image of the scene captured from the camera and converted into gray scale. The second column of Figure 8 represents the sparse depth map while the third and fourth columns the upsampled reconstructions from TGV and the proposed algorithm, respectively. Note that the proposed approach results in better reconstructions and avoids oversmoothing edges in comparison to TGV. This is specially notorius in objects with sharp edge discontinuities as for example the pedestrian legs in Figures 8(c) versus (d) and in (k) versus (l), in the poles in (g) versus (h) and in (o) versus (p). Note also in (s) that details in trees, poles and bikes specially those in the right part of the scene are hard to resolve whereas in (t) one is able to see the tree trunks, poles and bikes present in the scene. In addition to the edge sharpness gain, the depth upsampled reconstruction by the proposed implementation took 0.033 secs per frame versus the $\sim$ 4.4 of the TGV method that makes it suitable for real-time applications such as AV's. This advantage is due to both Nesterov's optimization acceleration and the fixed equality constraint in Eq. (4) as opposed to its corresponding relaxation in the TGV method. Finally, it is worth noting that some points that appear as artifacts at the edges in this depth reconstruction are not caused by the proposed depth upsampling method but rather by the LiDAR scanning mechanism. In particular, this is caused from the horizontal difference of laser firing/detection timings across the lasers vertically



(a) Image from camera (b) Sparse reprojected LiDAR (c) TGV from Ferst [7] (d) Proposed method

(e) Image from camera (f) Sparse reprojected LiDAR (g) TGV from Ferst [7] (h) Proposed method

(i) Image from camera (j) Sparse reprojected LiDAR (k) TGV from Ferst [7] (l) Proposed method

(m) Image from camera (n) Sparse reprojected LiDAR (o) TGV from Ferst [7] (p) Proposed method

(q) Image from camera (r) Sparse reprojected LiDAR (s) TGV from Ferst [7] (t) Proposed method

(u) Image from camera (v) Sparse reprojected LiDAR (w) TGV from Ferst [7] (x) Proposed method

Figure 8: Up-sampled depth reconstruction comparisons. Note in the pedestrian extremities of (d,l), posts in (h,p), bikes and other objects in (h,t) that our approach is capable of resolving finer details compared to TGV.

oriented inside the LiDAR that causes each to fire to a slightly different horizontal location than its vertically oriented laser neighbors.

4. Conclusion

This work proposed both a novel motion based self-calibration method for multi-modal LiDAR and camera sensors and a method to generate depth upsampled reconstructions from sparsely measured re-projected LiDAR data. The experiments on recorded real-data show that the proposed motion based registration function performs well as a similarity metric between the re-projected LiDAR and camera measurements. It is efficient in constraining alignments both spatially and temporally and effective in decoupling alignments from the modality differences between intensity and 3D measurements. An additional benefit is that the newer method requires no targets and is capable of self-calibrating on the fly. Such properties, are highly attractive to general mobile autonomous robots and self-driving cars for which current practices include calibration in specialized target-based calibration facilities often expensive and impractical for mass production. The results also indicate that the proposed method for depth upsampled reconstruction achieves state of the art performance for the real-time deployment category. The algorithm renders depth reconstructions preserving smooth regions while respecting sharp edge discontinuities. The balance between performance and computational complexity of our depth upsampling makes it an attractive method perfect for mobile robotics applications imposing real-time requirements.

References

[1] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng, "Learning depth from single monocular images," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, 2005, NIPS'05, pp. 1161–1168, MIT Press.

[2] Jure Žbontar and Yann LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, Jan. 2016.

[3] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[4] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proceedings of the 18th International Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2005, pp. 291–298.

[5] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, November 2013.

[6] J. Lu and D. Forsyth, "Sparse depth super resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015.

[7] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. IEEE Int. Conf. Comp. Vis.*, Sydney, NSW, Australia, December 1-8, 2013, pp. 993–1000.

[8] U.S. Kamilov and P.T. Boufounos, "Motion-adaptive depth superresolution," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1723 – 1731, 2017.

[9] K. Degraux, U.S. Kamilov, P.T. Boufounos, and D. Liu, "Online convolutional dictionary learning for multimodal imaging," in *IEEE International Conference on Image Processing*, Beijing, China, 2017.

[10] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems*, Berlin, Germany, June 24-28, 2013, pp. 29–36.

[11] G. Pandey, J.R. McBride, S. Savarese, and R.M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 1–27, August 2014.

[12] J. Castorena, U. Kamilov, and P.T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," in *IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, Shanghai, March 20-25, 2016, pp. 2862–2866.

[13] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[15] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l1 optical flow," in *Pattern Recognition*, Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne, Eds., Berlin, Heidelberg, 2007, pp. 214–223, Springer Berlin Heidelberg.

[16] D.P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of operations research*, vol. 14, no. 1, pp. 105–123, 1988.

[17] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions.," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[18] Y.E. Nesterov, "A method for solving convex programming problem with convergence rate $o(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. of Rob. Res.*, 2013.

[20] S. Li, C. Xu, and M. Xie, "A robust o(n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, July 2012.