MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

High-Performance and Tunable Stereo Reconstruction

Pillai, Sudeep; Ramalingam, Srikumar; Leonard, John

TR2016-037 May 16, 2016

Abstract

Traditional stereo algorithms have focused their efforts on reconstruction quality and have largely avoided prioritizing for run time performance. Robots, on the other hand, require quick maneuverability and effective computation to observe its immediate environment and perform tasks within it. In this work, we propose a high-performance and tunable stereo disparity estimation method, with a peak frame-rate of 120Hz (VGA resolution, on a single CPU-thread), that can potentially enable robots to quickly reconstruct their immediate surroundings and maneuver at high-speeds. Our key contribution is a disparity estimation algorithm that iteratively approximates the scene depth via a piece-wise planar mesh from stereo imagery, with a fast depth validation step for semidense reconstruction. The mesh is initially seeded with sparsely matched keypoints, and is recursively tessellated and refined as needed (via a resampling stage), to provide the desired stereo disparity accuracy. The inherent simplicity and speed of our approach, with the ability to tune it to a desired reconstruction quality and runtime performance makes it a compelling solution for applications in high-speed vehicles.

IEEE International Conference on Robotics and Automation (ICRA)

High-Performance and Tunable Stereo Reconstruction

Sudeep Pillai[1], Srikumar Ramalingam[2] and John J. Leonard[1]

http://people.csail.mit.edu/spillai/fast-stereo-reconstruction

***Abstract*— Traditional stereo algorithms have focused their efforts on reconstruction quality and have largely avoided prioritizing for run time performance. Robots, on the other hand, require quick maneuverability and effective computation to observe its immediate environment and perform tasks within it. In this work, we propose a high-performance and tunable stereo disparity estimation method, with a peak frame-rate of 120Hz (VGA resolution, on a single CPU-thread), that can potentially enable robots to quickly reconstruct their immediate surroundings and maneuver at high-speeds. Our key contribution is a disparity estimation algorithm that iteratively approximates the scene depth via a piece-wise planar mesh from stereo imagery, with a fast depth validation step for semi-dense reconstruction. The mesh is initially seeded with sparsely matched keypoints, and is recursively tessellated and refined as needed (via a resampling stage), to provide the desired stereo disparity accuracy. The inherent simplicity and speed of our approach, with the ability to tune it to a desired reconstruction quality and runtime performance makes it a compelling solution for applications in high-speed vehicles.**

I. Introduction

Stereo disparity estimation has been a classical and well-studied problem in computer vision, with applications in several domains including large-scale 3D reconstruction, scene estimation and obstacle avoidance for autonomous driving and flight etc. Most state-of-the-art methods [1] have focused its efforts on improving the reconstruction quality on specific datasets [2], [3], with the obvious trade-off of employing sophisticated and computationally expensive techniques to achieve such results. Some recent methods, including Semi-Global Matching [4], and ELAS [5], have recognized the necessity for practical stereo matching applications and their real-time requirements. However, none of the state-of-the-art stereo methods today can provide meaningful scene reconstructions in real-time ($\geq$ 25Hz) except for a few FPGA or parallel-processor-based methods [6], [7], [8]. Other methods have achieved high-speed performance by matching fixed disparities, fusing these measurements in a push-broom fashion with a strongly-coupled state estimator [9]. Most robotics applications, on the other hand, require real-time performance guarantees in order for the robots to make quick decisions and maneuver their immediate environment in an agile fashion. Additionally, as requirements for scene reconstruction vary across robotics applications, existing methods cannot be reconfigured to various accuracy-speed operating regimes.

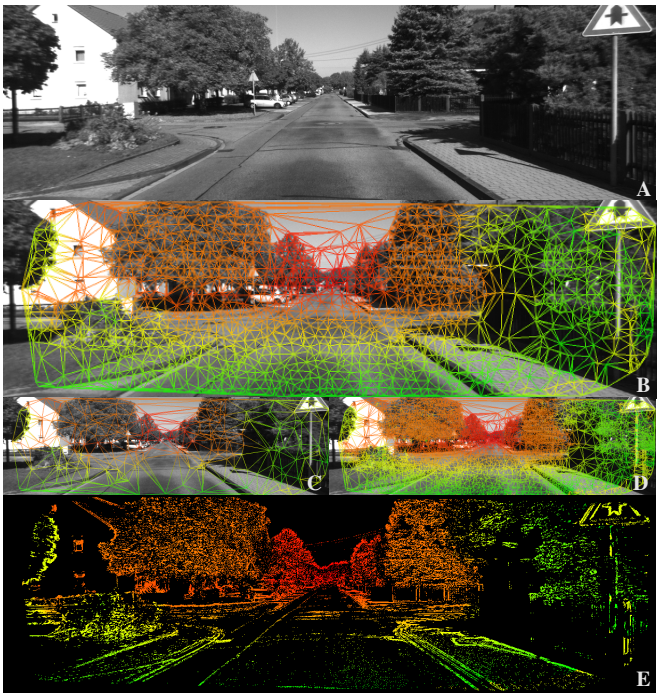

Fig. 1: The proposed high-performance stereo matching method provides semi-dense reconstruction (**E**) of the scene, capable of running at a peak frame-rate of 120Hz (8.2 ms, VGA resolution). Our approach maintains a piece-wise planar representation that enables the computation of disparities (semi-densely, and densely) for varied spatial densities over several iterations (**B**-2 iterations, **C**-1 iteration, **D**-4 iterations). Colors illustrate the scene depths, with green indicating near-field and red indicating far-field regions. Figure best viewed in digital format.

In this work, we propose a high-performance, iterative stereo matching algorithm, capable of providing semi-dense disparities at a peak frame-rate of 120Hz (see Figure 1). An iterative stereo disparity hypothesis and refinement strategy is proposed that provides a tunable iteration parameter to adjust the accuracy-versus-speed trade-off requirement on-the-fly. Through experiments, we show the strong reliability of disparity estimates provided by our system despite the low computational requirements. We provide several evaluation results comparing accuracies against current stereo methods, and provide performance analysis for varied runtime requirements. We validate the performance of our system on both publicly available datasets, and commercially available stereo sensors for comparison. In addition to single view disparity estimates, we show qualitative results of large-scale stereo reconstructions registered via stereo visual odometry, illustrating the consistent stereo disparities our approach provides on a per-frame basis.

[1]Sudeep Pillai and John J. Leonard are with the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge MA 02139, USA {spillai, jleonard}@csail.mit.edu. Their work was partially supported by the Office of Naval Research under grants N00014-10-1-0936, N00014-11-1-0688 and N00014-13-1-0588 and by the National Science Foundation under grant IIS-1318392.

[2]Srikumar Ramalingam is with Mitsubishi Electric Research Labs (MERL) ramalingam@merl.com, and his work was supported by Mitsubishi Electric Corporation.

II. Related Work

Classical stereo matching methods have mostly considered dense reconstructions, and are generally divided into two categories, *local* and *global* methods. The naive approach to stereo matching involves finding corresponding pixels in the left and right images that have similar color or intensity. Since the intrinsics and extrinsics of the stereo cameras are known, the matching search space is limited to the epipolar line with a pre-defined disparity level, assuming a maximum distance observed.

Dense Methods As one may expect, the above formulation results in a noisy disparity map, due to the high pixel-level ambiguity in matching. This is addressed by matching fixed size windows instead, reducing the noise and inherent ambiguity in the stereo imagery. Additionally, the resulting disparity is smoothed, allowing neighboring pixels to have similar disparities. Despite several advances in adaptive-supports, slanted window matching and edge-preserving filtering approaches [10], local methods suffer from being unable to estimate disparities at low-textured regions.

For the past decade, global methods have dominated stereo benchmarks [2], [3]. They differ from local methods in that their smoothness regularization assumptions are no longer limited to a fixed window size, but extend throughout the image. Typically, the disparity estimation is modeled as an energy minimization, given by:

$$E(D) = \sum_{p \in I_l} c(p, p - d_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}} s(d_p, d_q) \quad (1)$$

where $c(p, p - d_p)$ is the pixel matching cost for a disparity level d_p, $s(d_p, d_q)$ is the smoothness regularization or penalty enforced between pixels p and q that are neighbors defined by $\mathcal{N}$. The above energy minimization formulation allows several optimization strategies to be employed including (i) graph-cuts (ii) belief-propagation (iii) dynamic programming. For a more thorough description of state-of-the-art stereo matching, we refer the reader to [10].

Sparse and Semi-Dense Methods Sparse stereo matching methods have been prevalent in robotics applications primarily due to their low-computational complexity [11]. These methods, including monocular keypoint-based SLAM techniques, have been combined with tessellation or meshing techniques to represent the scene as piece-wise planar [12], making it a fairly rich representation for navigation and scene reconstruction purposes with a significantly low memory footprint.

Recently, there has been an increased interest in semi-dense representations for mapping, navigation [13], [14], [15], [16] and object detection [17]. Qualitatively, these semi-dense methods can be a compelling middle-ground, between dense stereo and sparse stereo matching methods, potentially paving the way to newer representations for navigation and reconstruction. LSD-SLAM [14], has recently shown large-scale 3D reconstructions by fusing the depth estimates for high-gradient pixels from short and wide-baseline frames in monocular videos, without the use of any interest point matches. However, monocular methods suffer from the well-known scale-drift problem (corrected using an IMU), and rely on the availability of several images to provide metrically accurate reconstructions. Recently, a semi-dense stereo reconstruction of high gradient pixels was shown using a Line-Sweep algorithm [16], which uses cross-ratio constraints on locally planar region. Our method relies on Delaunay triangulation and support point re-sampling, leading to better accuracy and improved computational performance. Furthermore, our method can reconstruct heavily occluding objects like poles, which will be challenging for Line-sweep.

Depth-priors and Plane-based Stereo Our work closely relates to that of ELAS [5] that takes a generative approach, using tessellated support points from sparse stereo matching as a depth prior to enable efficient sampling of disparities in a dense fashion. Most recently, MeshStereo [18] has been proposed, where the global stereo model is designed for view interpolation via a similar 3D triangular mesh. The authors model the difficult depth discontinuity problem as a two-layer MRF, where the upper layer models the splitting of depth discontinuities, while the lower layer regularizes the depths via a region-based optimization. In this work, we take a discriminative approach to stereo matching, and continue to maintain the piece-wise planar assumption while re-tessellating poorly reconstructed regions in the interpolated disparity image that correspond to having a high matching cost. Furthermore, we propose an iterative method that continues to re-tessellate and approximate complex surfaces with more piece-wise planar regions, with every additional iteration.

Similar to Patch-Match Stereo [19], our method implicitly computes disparities with sub-pixel precision, without the need for an additional post-processing step [20] that fits a parabolic curve within the cost volume. As duly noted in [10], parabolic fitting leads to noisy sub-pixel estimation across heavily slanted surfaces. We do note that our approach is reminiscent of plane-sweeping algorithms that include fronto-parallel and slanted windows to their label space for improved disparity estimation along varied surfaces [21], however, we draw candidate planes and disparities from the tessellations constructed with sparse keypoint-based stereo matches that in turn reduces the search space drastically.

High-speed Stereo Matching To the best of our knowledge, we are unaware of any semi-dense stereo method that can perform full disparity range estimation at speeds of $\geq$ 100Hz, without the use of GPUs, FPGAs or other specialized-hardware. We consider disparity estimation for the approximate piece-wise planar case, as this representation can be especially useful in robotics applications where obstacles are to be observed and avoided in real-time. We propose an iterative stereo matching method, that maintains a spatially-adaptive piece-wise planar representation, significantly speeding up stereo disparity estimation by a factor of 32x compared to popular stereo implementations [4], while providing sufficiently accurate disparity estimates.

III. High-Performance and Tunable Stereo Reconstruction

This section introduces the algorithmic components of our method (see Alg. 1). We propose an tunable (and iterative) stereo algorithm that consists of four key steps: (i) Depth prior construction from Delaunay triangulation of sparse key-point stereo matches (ii) Disparity interpolation using piece-wise planar constraint imposed by the tessellation with known depths (iii) Cost evaluation step that validates interpolated disparities based on matching cost threshold (iv) Re-sampling stage that establishes new support points from previously validated regions and via dense epipolar search. The newly added support points are re-tessellated and interpolated to hypothesize new candidate planes in an iterative process. Since we are particularly interested in collision-prone obstacles and map structure in the immediate environment, we focus on estimating the piece-wise planar reconstruction as an approximation to the scene, and infer stereo disparities in a semi-dense fashion from this underlying representation. Unless otherwise noted, we consider and perform all operations on only a subset of image pixels that have high image gradients $\Omega_I \subset \Omega$, and avoid reconstructing non-textured regions in this work.

A. Spatial Support via Sparse Stereo Matching

Many state-of-the-art stereo algorithms start by exhaustively computing a pixel-level cost volume $\mathcal{O}(HWN_D)$, for a fixed number of disparities N_D (usually 128). Instead, we employ a similar strategy to [5], and first construct a piece-wise planar scene depth estimate to quickly inform a coarse depth prior or mesh. First, a sparse set of support keypoints $S = \{s_1, \ldots, s_n\}$ are detected via FAST features [22] (sampled from 12x10 spatial-bins), and matched along their epipolar lines as in [11] (see SparseStereo in Alg. 1). We define each support point $s_n = (u_n, v_n, d_n)^T$, similar to [5], as the concatenation of their image coordinates $(u_n, v_n) \in \mathbb{N}^2$, and their corresponding disparity $d_n \in \mathbb{N}$. Using these support points as vertices with known depths, a piece-wise planar mesh is constructed via Delaunay-Triangulation. (see Figure 2, DelaunayTriangulation in Alg. 1).

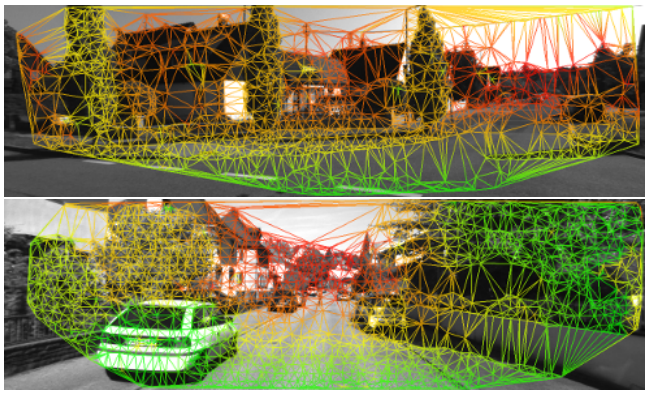

Fig. 2: Depth prior determined via Delaunay triangulation of sparse support points. Vertices in the mesh correspond to the sparse support points, or re-sampled support, while the triangular regions represent the piece-wise planar scene reconstruction.

Algorithm 1 Iterative Stereo Reconstruction

Input: (I_l, I_r, Ω_I): Input gray-scale stereo images and high-gradient regions
Output: D_f: Disparities at high-gradient regions (Semi-Dense)
Globals: Refer to Table I for description of variables

// Initialize final disparity and associated cost
$D_f \leftarrow [0]_{[H\times W]}, C_f \leftarrow [t_{hi}]_{[H\times W]}, sz_{occ} \leftarrow 32$
// S: Set of N support points
$S_1 \leftarrow$ SparseStereo(I_l, I_r)
// Tessellated mesh with estimated disparities
$\mathcal{G}(S_1) \leftarrow$ DelaunayTriangulation(S_1)
for $it = 1 \rightarrow n_{iters}$ **do**
  // Dense piece-wise planar disparity
  $D_{it} \leftarrow$ DisparityInterpolation$(\mathcal{G}(S_{it}))$
  // Cost evaluation given interpolated disparity
  $C_{it} \leftarrow$ CostEvaluation(I_l, I_r, D_{it})
  // Refine disparities
  $C_g, C_b \leftarrow$ DisparityRefinement(D_{it}, C_{it})
  // Prepare for next iteration, if not last iteration
  if $it \neq n_{iters}$ **then**
    // Re-sample regions with high matching cost
    $S_{it+1} \leftarrow$ SupportResampling(C_g, C_b, S_{it})
    // Tessellated mesh with estimated disparities
    $\mathcal{G}(S_{it+1}) \leftarrow$ DelaunayTriangulation(S_{it+1})
    // Decrease occupancy grid size by factor of 2
    $sz_{occ} = \max(1, sz_{occ}/2)$
  end if
end for

Name	Scope	Description
I_l, I_r	L	Input gray-scale stereo images
H, W	G	Dimensions of input image I_l
Ω, Ω_I	G	Set of all pixels in image, and subset of high-gradient pixels
S	L	Sparse support pixels with valid depths
$\mathcal{G}(S)$	L	Graph resulting from Delaunay Triangulation over S
X	L	Re-sampled or detected support pixels with unknown depths
D_f	G	Final disparity image
C_f	G	Cost matrix associated to D_f
D_{it}	L	Intermediate disparity (interpolated)
C_{it}	L	Cost associated to D_{it}
C_g	L	Cost associated with regions of high confidence matches
C_b	L	Cost associated with regions of invalid disparities
N_D	G	Maximum number of disparities considered
sz_{occ}	G	Occupancy grid size used for re-sampling
t_{lo}, t_{hi}	G	Lower and upper cost threshold for validating disparities
n_{iters}	G	Number of iterations the algorithm is allowed to run

TABLE I: Description of symbols used in the proposed stereo matching algorithm, and their corresponding scope (G:Global or L:Local) within the implementation.

B. Disparity Interpolation

We refer to the planar regions in the delaunay triangulation as candidate planes, as they are constructed from the sparse set of support points whose disparities are estimated via epipolar search. These candidate planes provide a strong measure of an underlying surface, and can be used to quickly verify the hypothesized planes. Inspired by previous work on candidate-plane validation [19], we leverage this efficient verification step to iteratively hypothesize candidate regions in the disparity image, thereby limiting the effective disparity search space to fewer than 3-5 disparity levels (not limiting to integer-valued disparities as most dense methods do).

At every intermediate step, we treat the stereo disparity image D_{it} as being constructed in a piece-wise planar manner via the Delaunay tessellated mesh. Each 3D planar surface

Algorithm 2 CostEvaluation

Input: (I_l, I_r, D_{it}): Left/Right stereo image, and interpolated disparity
Output: C_{it}: Matching cost corresponding to D_{it}

1: **for** $(u, v) \in \Omega_I$ **do**
    // Interpolated disparity at (u, v)
2: $d \leftarrow D_{it}(u, v)$
    // Census-based 5x5 window matching
3: $C_{it}(u, v) \leftarrow$ CensusMatchingCost$(I_l(u, v), I_r(u - d, v))$
4: **end for**

or triangle, can be described by its 3D plane parameters $(\pi_1, \pi_2, \pi_3, \pi_4) \in \mathbb{R}^4$ given by

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0 \tag{2}$$

For a stereo setup with a known baseline B, and known calibration ($u = fX/Z$, $v = fY/Z$, and $d = fB/Z$), the above equation reduces to

$$\pi'_1 u + \pi'_2 v + \pi'_3 = d \tag{3}$$

where $\pi' = (\pi'_1, \pi'_2, \pi'_3) \in \mathbb{R}^3$ are the plane parameters in disparity space.

In order to estimate interpolated disparities on a pixel-level basis, we first construct a lookup-table that identifies the triangle and its plane coefficients for each pixel (u, v) in the left image. Subsequently, the parameters π' for each triangle are obtained by solving a linear system as done in [5], and are re-estimated every time after the Delaunay triangulation step. The resulting piece-wise planar tessellation can be used to linearly interpolate regions within the disparity image using the estimated plane parameters π' (see DisparityInterpolation in Alg. 1).

C. Cost Evaluation

The interpolated disparity image resulting from every tessellation provides a set of candidate depths that could potentially contain valid scene points. In order to validate these interpolated disparities, we perform Census window-based matching on a 5x5 patch [23], [24] between the left and right stereo images. The resulting matching cost is normalized and retained to be validated in the next step (see CostEvaluation in Alg. 2).

D. Disparity Refinement

The interpolated disparities computed from the tessellation may or may not necessarily hold true for all pixels. For high-gradient regions in the image, the cost computed between the left and right stereo patch for the given interpolated disparity can be a sufficiently good indication to validate the candidate pixel disparity. We use this assumption to further refine and prune candidate disparities based on the per-pixel cost computed in the previous step, as characterized by validated (C_g) and invalidated (C_b) cost regions. Thus, we can invalidate every pixel p in the left image, if the cost associated $c(p, p - d_i)$ with matching the pixel in the right image with a given interpolated disparity d_i is above a maximum permissible cost t_{hi}. The same approach is used to validate pixels that fall within a suitable cost range ($< t_{lo}$) whose correspondence certainty is high. This step also allows

Algorithm 3 DisparityRefinement

Input: (D_{it}, C_{it}): Interpolated Disparity and associated matching cost
Output: C_g, C_b: Costs associated with regions of high and low matching confidence disparities

1: $H' \leftarrow \frac{H}{sz_{occ}}, W' \leftarrow \frac{W}{sz_{occ}}$
    // C_g: Cost matrix of confident supports: $(u, v, d, cost)$
2: $C_g \leftarrow [0, 0, 0, t_{lo}]_{[H' \times W']}$
    // C_b: Cost matrix of invalid matches: $(u, v, cost)$
3: $C_b \leftarrow [0, 0, t_{hi}]_{[H' \times W']}$
4: **for** $(u, v) \in \Omega_I$ **do**
    // Establish occupancy grid for resampled points
5: $u' \leftarrow \frac{u}{sz_{occ}}, v' \leftarrow \frac{v}{sz_{occ}}$
    // If matching cost is lower than previous best final cost
6: **if** $C_{it}(u, v) < C_f(u, v)$ **then**
7: $D_f(u, v) \leftarrow D_{it}(u, v)$
8: $C_f(u, v) \leftarrow C_{it}(u, v)$
9: **end if**
    // If matching cost is lower than previous best valid cost
10: **if** $C_{it}(u, v) < t_{lo}$ **and** $C_{it}(u, v) < C_g(u', v', 4^\dagger)$ **then**
11: $C_g(u', v') \leftarrow (u, v, D_{it}(u, v), C_{it}(u, v))$
12: **end if**
    // If matching cost is higher than previous worst invalid cost
13: **if** $C_{it}(u, v) > t_{hi}$ **and** $C_{it}(u, v) > C_b(u', v', 3^\dagger)$ **then**
14: $C_b(u', v') \leftarrow (u, v, C_{it}(u, v))$
15: **end if**
16: **end for**

†Matrices are 1-indexed

Algorithm 4 SupportResampling

Input: (C_g, C_b, S_{it}): Matching costs for confident/invalid matches
Output: S_{it+1}: New support points for tessellation

1: $S_{it+1} \leftarrow S_{it}, X \leftarrow \emptyset$
2: **for** $(u, v) \in \Omega_I$ **do**
    // Perform sparse epipolar stereo for resampled invalid pixels
3: **if** $C_b(u, v) \neq 0$ **then**
4: $X \leftarrow \{X, (u, v)\}$
5: **end if**
    // Resample confident pixels and add to support
6: **if** $C_g(u, v) \neq 0$ **then**
7: $S_{it+1} \leftarrow \{S_{it+1}, (u, v)\}$
8: **end if**
9: **end for**
    // Re-estimate disparities via epipolar search
10: $S_{matched} \leftarrow$ SparseEpipolarStereo(I_l, I_r, X)
11: $S_{it+1} \leftarrow \{S_{it+1}, S_{matched}\}$

incorrectly matched regions to be resampled and re-evaluated for new stereo matches as the interpolated costs of regions around the falsely matched corners are driven sufficiently high. Additionally, the disparities corresponding to the least cost for each pixel is updated with every added iteration, ensuring that the overall stereo matching cost is always reduced (see Step 6 in Alg. 3). For more details regarding this step see DisparityRefinement in Alg. 3.

E. Support Resampling

The disparity refinement step establishes pixels or regions in the image whose disparities need to be re-evaluated, while also simultaneously providing reliable disparities to further utilize in the matching process. With a discretized occupancy grid of size ($sz_{occ} \times sz_{occ}$), pixels with the highest matching cost within a 32x32 (sz_{occ} is initialized to 32) window are established, and re-sampled. These re-sampled pixels are

strong indicators of occluding edges, and sharp discontinuities in depth, making them viable candidates for epipolar-constrained dense stereo matching. Subsequently, the re-sampled keypoints are densely matched via epipolar search, and new support points $S_{matched}$ are established as a result. Another valuable feature is the ability to inform disparities at greater resolution and accuracy with every subsequent iteration; the discretization of the occupancy grid is reduced by a factor of 2 so that pixels are more densely sampled with every successive iteration (see SUPPORTRESAMPLING in Alg. 4).

F. Iterative Reconstruction

The stereo matching proceeds to reduce the overall stereo matching cost associated with the interpolated piece-wise planar disparity map. High-matching cost regions are re-sampled and re-estimated to better fit the piece-wise planar disparity map to the true scene disparity. With every subsequent iteration, new keypoints are sampled, tessellated to inform a piece-wise planar depth prior, and further evaluated to reduce the overall matching cost. With such an iterative procedure, the overall stereo matching cost is reduced, with the obvious cost of added computation or run-time requirement (see Figure 3).

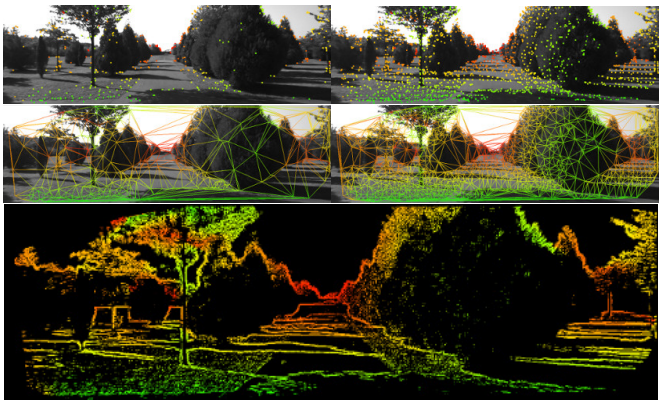

Fig. 3: Depth prior estimated with every subsequent iteration (Rows 1-2, Column 1: After 1 iteration, Column 2: After 2 iterations). As expected, the density of support points increase, with the piece-wise planar representation better fitting to the true scene disparity map. Row 3 illustrated the final semi-dense reconstruction after 2 iterations. Figure best viewed in digital format.

IV. EXPERIMENTS

In this section, we evaluate the proposed high-performance stereo matching method. We evaluate the matching accuracy and runtime performance of our proposed method on the popular KITTI dataset [3] and on 2 different stereo cameras, namely the Point Grey Bumblebee2 1394a[1], and the ZED Stereo Camera[2]. The KITTI dataset contains rectified gray-scale stereo imagery at a resolution of 1241x376 (0.46 MP), captured from 2 Point Grey Flea2 cameras mounted with a baseline of 0.54m. We compare against stereo matching algorithms that are commonly used in robotics applications - the popular implementation of Semi-Global Matching [4] in OpenCV (Semi-Global Block-Matching or SGBM), ELAS [5] and Line-Sweep [16]. We provide a thorough analysis of the trade-offs between matching accuracy and run-times achievable by our proposed method, across varied hardware and environmental setups.

[1] http://www.ptgrey.com/stereo-vision-cameras-systems

[2] https://www.stereolabs.com/zed/

A. Evaluation on KITTI dataset

Disparity Estimation Accuracy In order to evaluate our proposed semi-dense method against existing methods, we only consider disparities in the image that have large image gradients or edges. Currently, semi-dense methods cannot be fully evaluated on the KITTI dataset, since the test server interpolates missing disparities, introducing several errors in the disparity estimates and overall accuracy. For all valid and non-occluding semi-dense edges, we report the absolute difference between the proposed method and existing state-of-the-art stereo implementations. In our experiments on the provided KITTI stereo evaluation kit, we find that greater than 89.9% of edge pixels had a disparity value of less than 3 pixels with respect to ground truth for the single pass variant (*Ours-1*). As seen in table II, with increased number of iterations, the same algorithm improves overall performance (*Ours-2*: 90.2%, *Ours-4*: 91.4%). For the stereo setup on the KITTI dataset, 3 pixels correspond to $\pm$3cm at a depth of 2 meters and $\pm$80cm at a depth of 10m. In addition, we compare against recent work [16] on semi-dense reconstruction on the KITTI dataset, and achieve significantly better disparity accuracy using our approach compared to 81.2% of [16] . In Table II below, we compare the disparities computed by our proposed method, and compare against existing stereo matching implementations, including Semi-Global Matching, ELAS, and Line-Sweep. We do note that the main reason for reduced accuracy compared to state-of-the-art methods is due to the local nature of the algorithm, as compared to the global regularization methods used in SGBM and ELAS. We visualize the results of our proposed method in Figure 4 with the corresponding ground truth disparities.

Method	**Accuracy (%)**			
	< 2px	< 3px	< 4px	< 5px
SGBM [4]	89.0	93.9	95.6	96.5
ELAS [25]	92.7	96.1	97.3	97.9
Line-Sweep [16]	72.6	81.2	84.7	86.7
Ours-1†	83.1	89.9	92.9	94.7
Ours-2†	83.5	90.2	93.2	94.9
Ours-4†	85.4	91.4	94.0	95.5

†The number next to the method indicates the number of iterations the algorithm is allowed to run.

TABLE II: Analysis of accuracy of our system on the KITTI dataset [3], as compared to popular stereo implementations including OpenCV's Semi-Global Block-Matching [4], ELAS [25] and Line-Sweep [16]. The number next to the method indicates the number of iterations the algorithm is allowed to run. The accuracy results are evaluated **only** over high-gradient (semi-dense) regions in the image.

Stereo Reconstruction In this section, we show the qualitative performance of our stereo disparity estimation approach via stereo reconstructions fused over multiple frames

from a moving camera. We use the stereo imagery from the KITTI dataset, and the corresponding ground truth poses to reconstruct scenes over a short window time frame to qualitatively illustrate the stereo matching consistency our approach provides. In Figure 5, we show our reconstruction results from various sequences. The reconstructions of building facades, cars, road terrain, and road curbs are well-detailed with little noise. Furthermore, unstructured and thin occluding edges such as trees, and their trunks are also well reconstructed. See video via the following link[3].

Runtime Performance Most existing stereo matching algorithms have focused their efforts on the accuracy, without much regard for the runtime performance of these systems. In this work, we focus on the potential benefits and trade-offs of stereo matching accuracy and runtime performance. Due to the iterative nature of our proposed method, we show that our approach can be tuned to various accuracy and runtime operational levels, particularly beneficial for robotics applications. In our experiments (Table III and IV), we evaluate the runtime performance of our proposed method across several standard image resolutions ranging from WVGA (320x240) to HD1080 (1920x1080). For the common stereo image resolutions (800x600), our approach provides a speed-up factor of 32x for the single-pass stereo matching case, and a factor of 12x for the two-pass stereo matching case, as compared to OpenCV's SGBM [4] implementation.

Method	Accuracy (%)	Run-time (Hz/ms)	Speed-up
SGBM [4]	93.9	2.8 Hz / 351.9 ms	1x
ELAS [25]	**96.1**	6.2 Hz / 160.9 ms	2.1x
Line-Sweep [16]	81.2	14.2 Hz / 70.0 ms	5x
Ours-1†	89.9	**92.2 Hz / 10.8 ms**	**32.4x**
Ours-2†	90.2	**34.6 Hz / 28.9 ms**	**12.2x**
Ours-4†	91.4	**17.2 Hz / 58.2 ms**	**6.0x**

TABLE III: Analysis of run-time performance of our system on the KITTI (1241 x 376 px, 0.46 MP) dataset [3], as compared to popular stereo implementations including OpenCV's Semi-Global Block-Matching [4] and ELAS [25]. The number next to the method indicates the number of iterations the algorithm is allowed to run. We achieve comparable performance, with a run-time speed-up of approximately **32 x**. Accuracy is reported for disparities that are within 3 pixels of ground truth.

Method		Image Resolution (px)				
		320x240	640x480	800x600	1280x720	1920x1080
SGBM [4]	Runtime (ms)	53.4	216.7	360.0	763.7	1873.7
ELAS [25]		22.7	107.2	170.3	332.7	650.9
Ours-1†		**3.0**	**8.2**	**10.9**	**18.2**	**35.9**
Ours-2†		**6.4**	**19.2**	**27.4**	**46.0**	**81.0**
Ours-4†		**18.7**	**64.9**	**99.2**	**172.9**	**287.2**

TABLE IV: *Running Time vs. Image Resolution:* We compare the runtime performance of our proposed approach (*Ours*) with existing state-of-the-art solutions for varied image resolutions. As shown in the table, our proposed stereo algorithm performs an order of magnitude faster that other popular approaches for high-resolution (720P) stereo imagery. The number next to the method indicates the number of iterations the algorithm is allowed to run.

[3] http://people.csail.mit.edu/spillai/projects/fast-stereo-reconstruction/pillai_fast_stereo16.mp4

B. Evaluation on Stereo Hardware

With the advent of the USB3 standard, high-framerate stereo cameras have now started to become mainstream. These devices open the door to newer data throughput capacities, however, existing state-of-the-art stereo algorithms fail to meet such high throughput requirements. To this end, in addition to the KITTI dataset, we benchmark our proposed method on two different stereo platforms including the BumbleBee2, and the newly introduced USB3-driven ZED Stereo Camera. The Bumblebee2 (12cm baseline) operates at 48 FPS providing gray-scale stereo imagery at a resolution of 648x488, while the ZED Camera is configured to operate at 60Hz with a resolution of 1280x720. In our experiments, we compare the disparities estimated from our approach against that of SGBM and report results on its accuracy and runtime performance (see Table V).

Method	Accuracy (%)	
	BumbleBee2	ZED
ELAS [25]	81.1	**91.6**
Line-Sweep [16]	83.9	77.2
Ours-1†	89.6	87.5
Ours-2†	**90.8**	87.3

TABLE V: Analysis of accuracy of our system on the BumbleBee2 and ZED Stereo Camera, with Semi-Global Block-Matching [4] considered as ground truth. We compare against other stereo implementations including ELAS and Line-Sweep and report the accuracy for disparities that are within 3 pixels of ground truth. The number next to the method indicates the number of iterations the algorithm is allowed to run.

C. Implementation

We use the high-speed sparse-stereo implementation of [11], and the Delaunay Tessellation is performed via the Triangle[4] library for the initial set of support tessellation. Besides the 5x5 Census-based block matching that is implemented using specialized SSE instructions [11], the rest of the code is implemented on a single-CPU thread in C++, without any specialized instruction sets or GPU-specific code. All the results of our code are tested on an Intel(R) Core(TM) i7-3920XM CPU @ 2.90GHz. We do note that while our current implementation refines disparities every iteration in batch, this step can be highly-parallel and asynchronous due to the recursive nature of the refinement over the tessellated structure.

V. Discussion

Several robotics applications adhere to strict computational budgets and runtime requirements, depending on their task domain. Some systems require the need to actively adapt to varying design requirements and conditions, and adjust parameters accordingly. In the context of mapping and navigation, robots may need to map the world around them, in a slow but accurate manner, while also requiring the ability to avoid dynamic obstacles quickly and robustly. Such systems require the ability to dynamically change

[4] https://www.cs.cmu.edu/~quake/triangle.html

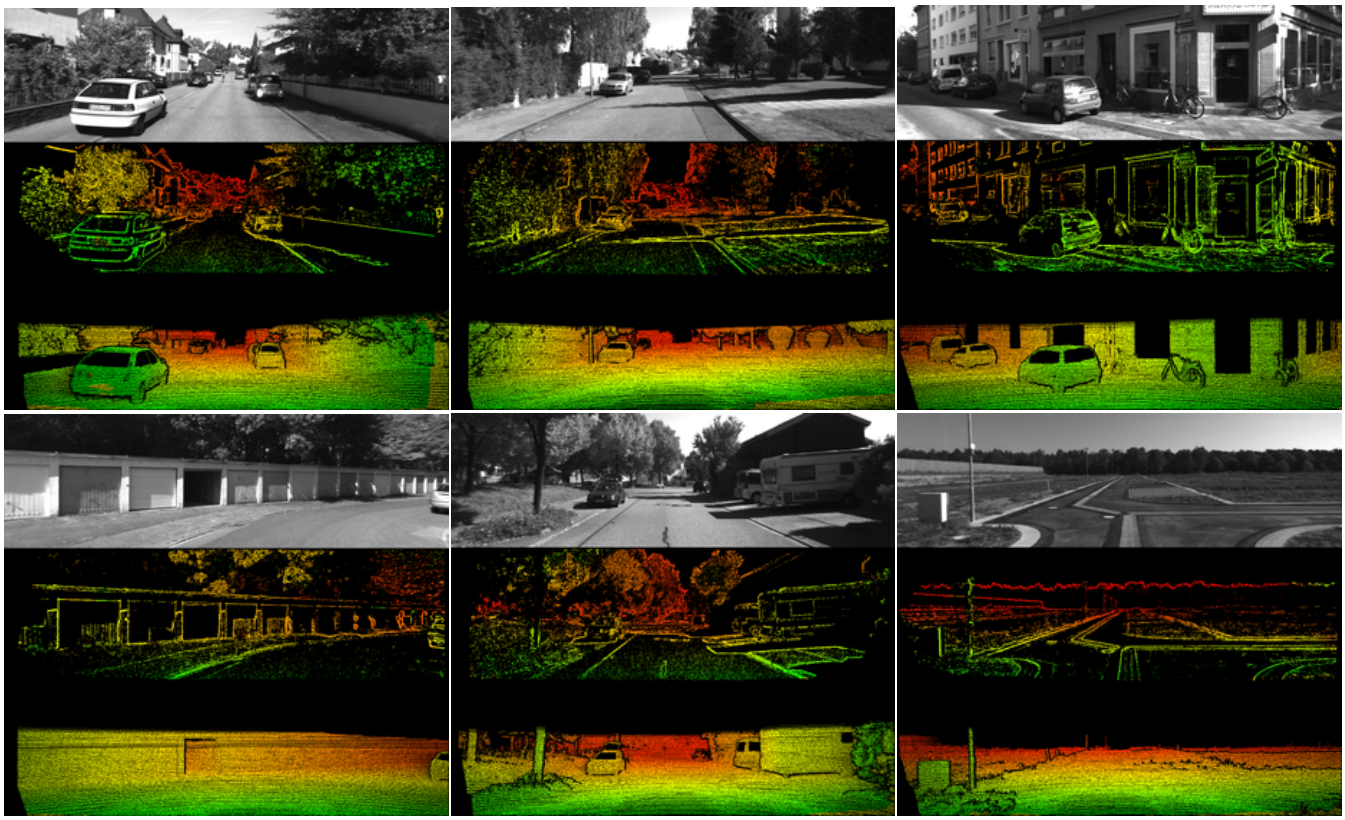

Fig. 4: Illustrations of our proposed stereo disparity estimation method (*Ours-2*, **row 2**) on the KITTI dataset with corresponding ground truth estimates (**row 3**) obtained from projecting Velodyne data on to the left camera. Despite its short execution time, our approach shows accurate estimates of disparities for a variety of scenes. The ground truth estimates are provided as reference, and are valid points that fall below the horizon. Similar colors indicate similar depths at which points are registered.

the accuracy requirements in order to achieve their desired runtime performance, given a fixed compute budget; this work is an attempt to provide such capability.

Another potential application of this approach could be to generate rapid and high-fidelity reconstructions, given a sufficiently coarse trajectory plan or foveation. Given a reasonable exploration-exploitation strategy, our approach can provide promising flexibility in exploiting accurate and rich scene information, while also being able to adjust itself to rapidly handle dynamic scenes during the exploration stage.

VI. Conclusion

Most existing stereo matching methods have been designed to ensure high accuracy guarantees for disparity estimation, however, sacrifice their runtime performance as a result. In this work, we propose a novel and high-performance, iterative and semi-dense stereo matching method, capable of running at a peak framerate of 120Hz, with comparable accuracies to existing and popular stereo matching solutions. By maintaining a piece-wise planar assumption, we develop a stereo matching strategy that recursively tessellates the scene into piece-wise planar regions so that it appropriately reconstructs it, given a fixed runtime requirement as provided by the user. By evaluating the matching costs for candidate planes, our approach quickly identifies planar regions, and repeats the process for non-planar regions by introducing more stereo matches within these regions and re-tessellating them. We compare against stereo matching algorithms that are commonly used in robotics applications and provide promising results of the trade-offs between matching accuracy and run-times achievable by our proposed method, across varied stereo dataset and hardware setups.

References

[1] J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int'l J. of Computer Vision*, vol. 47, no. 1-3, 2002.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005.

[5] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Computer Vision–ACCV 2010*. Springer, 2011.

[6] D. Honegger, H. Oleynikova, and M. Pollefeys, "Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2014.

[7] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation," in *Embedded Computer Systems (SAMOS), 2010 International Conference on*. IEEE, 2010.

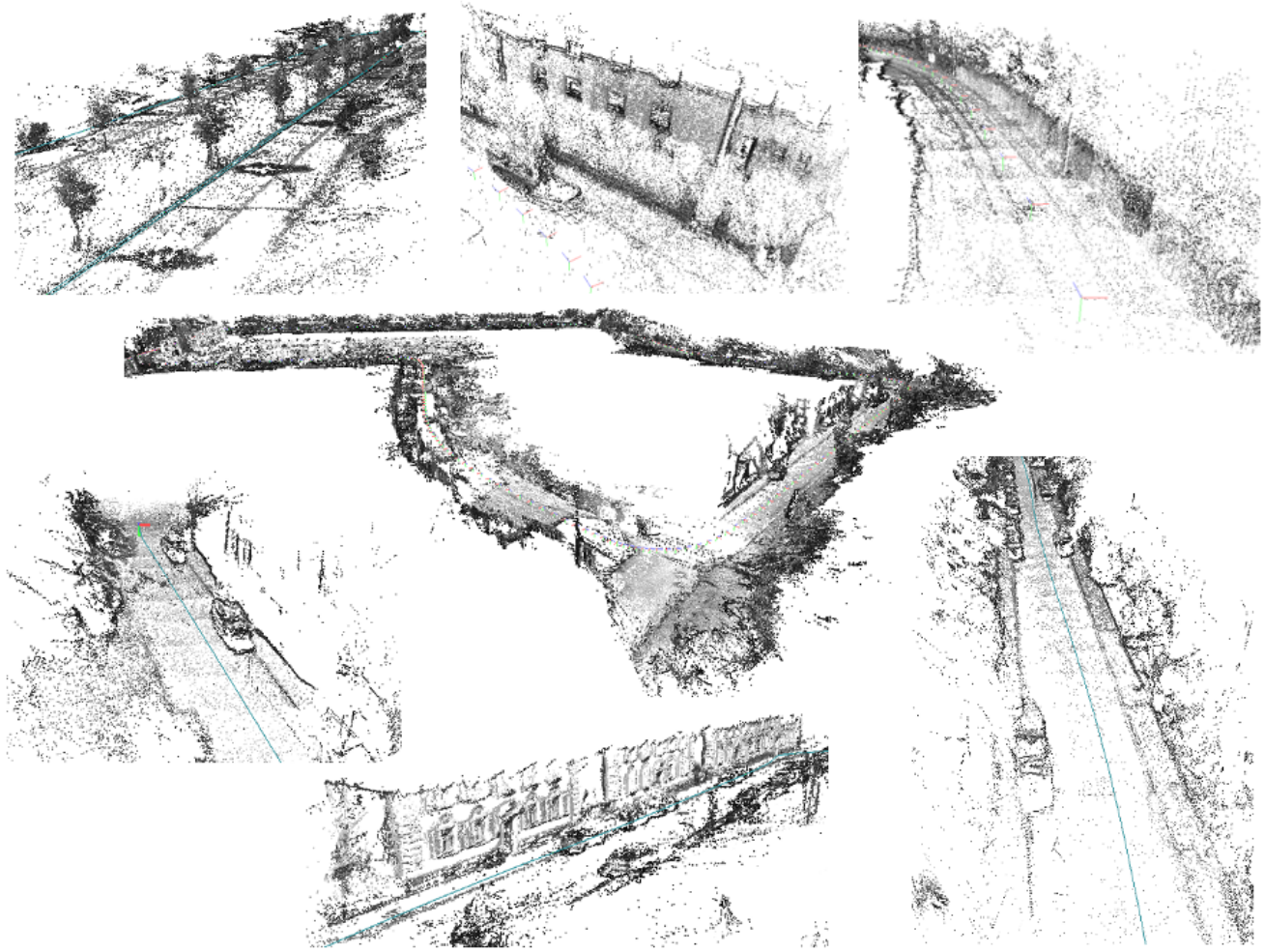

Fig. 5: Illustrated above are various scenes reconstructed using our proposed stereo matching approach. We use the ground truth poses from the KITTI dataset to merge the reconstructions from multiple frames, qualitatively showing the consistency in stereo disparity estimation of our approach.

[8] S. K. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," in *Computer Vision Systems*. Springer, 2009.

[9] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2015.

[10] M. Bleyer and C. Breiteneder, "Stereo Matching - State-of-the-Art and Research Challenges," in *Advanced Topics in Computer Vision*. Springer, 2013, pp. 143–179.

[11] K. Schauwecker, R. Klette, and A. Zell, "A new feature detector and stereo matching method for accurate high-performance sparse stereo matching," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[12] A. Concha and J. Civera, "DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[13] O. Veksler, "Dense features for semi-dense stereo correspondence," *Int'l J. of Computer Vision*, vol. 47, no. 1-3, 2002.

[14] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 2014.

[15] R. Mur-Artal and J. Tardos, "Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[16] S. Ramalingam, M. Antunes, D. Snow, G. Hee Lee, and S. Pillai, "Line-sweep: Cross-Ratio for Wide-Baseline Matching and 3D Reconstruction," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[17] S. Pillai and J. Leonard, "Monocular SLAM Supported Object Recognition," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[18] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui, "Meshstereo: A global stereo model with mesh alignment regularization for view interpolation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2057–2065.

[19] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch Stereo - Stereo matching with slanted support windows." in *BMVC*, vol. 11, 2011, pp. 1–11.

[20] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[21] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[22] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 2006, pp. 430–443.

[23] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 1994, pp. 151–158.

[24] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 9, pp. 1582–1599, 2009.

[25] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011.