2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Joint Perception And Planning For Efficient Obstacle Avoidance Using Stereo Vision

Sourish Ghosh[1] and Joydeep Biswas[2]

***Abstract*— Stereo vision is commonly used for local obstacle avoidance of autonomous mobile robots: stereo images are first processed to yield a dense 3D reconstruction of the observed scene, which is then used for navigation planning. Such an approach, which we term Sequential Perception and Planning (SPP), results in significant unnecessary computations as the navigation planner only needs to explore a small part of the scene to compute the shortest obstacle-free path. In this paper, we introduce an approach to Joint Perception and Planning (JPP) using stereo vision, which performs disparity checks on demand, only as necessary while searching on a planning graph. Furthermore, obstacle checks for navigation planning do not require full 3D reconstruction: we present in this paper how obstacle queries can be decomposed into a sequence of *confident positive* stereo matches and *confident negative* stereo matches, which are significantly faster to compute than the *exact depth* of points. The resulting complete JPP formulation is significantly faster than SPP, while still maintaining correctness of planning. We also show how the JPP works with different planners, including search-based and sampling-based planners. We present extensive experimental results from real robot data and simulation experiments, demonstrating that the JPP requires less than 10% of the disparity computations required by SPP.**

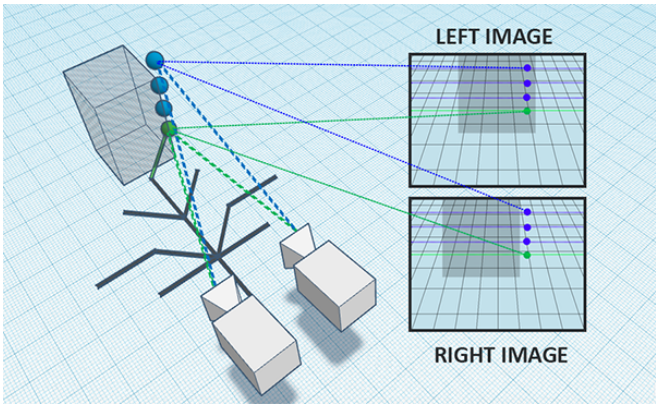


Fig. 1: Given a graph (grey lines) for planning, Joint Preception and Planning checks query edges for traversability by verifying the existence of the ground plane (green point), and the absence of obstacle surfaces for all points up to the height of the robot (blue points). All such points are first projected on to the stero images, and the disparity cost computed for the projected pairs. Existence is verified by checking if the disparity cost is $< \epsilon^+$, termed the *confident positive check*, and absence is verified by checking if the disparity cost is $> \epsilon^+$, termed the *confident negative check*.

I. Introduction

Existing approaches to local obstacle avoidance using stereo vision (*e.g.*, [1], [2]) perform sequential perception and planning (SPP), where input stereo images are processed to compute the disparity at each pixel. Such dense disparity is then used to infer depth for all image points, and hence obstacles in the world. However, the input images often include visual information that is irrelevant to the planning task at hand, thus wasting computational time at the perception step in SPP.

In this paper, we introduce a novel approach to joint perception and planning (JPP) for obstacle avoidance using stereo vision that eliminates unnecessary disparity computations. JPP treats traversability queries by the path planner as on-demand disparity checks for perception. Thus, the only disparity computations performed by perception are those necessary for planning for the obstacle avoidance task. We further simplify the problem of identifying reachable configurations of the robot by verifying confident positive disparity matches for the ground plane around the configuration pose, and by verifying confident negative disparity matches for all points within the robot's safety radius and height of that pose. Verifying confident positive and negative checks are significantly computationally faster than evaluating the exact depth: while a confidence check requires only a single disparity comparison, evaluating the exact depth requires multiple disparity comparisons along the epipolar line. The resulting obstacle checks are still exact: reachable and unreachable configurations of the robot are still correctly identified, albeit at a significantly lower computational cost. Figure 1 illustrates the decomposition of JPP into on-demand confidence disparity checks.

We empirically demonstrate JPP with an A* planner, and an RRT planner for local obstacle avoidance. Over extensive experimental evaluations, we show that JPP requires less than 10% of the disparity comparisons required by SPP. The contributions of this paper are thus three-fold: 1) We contribute a JPP formulation to integrate obstacle avoidance planning with on-demand stereo perception; 2) We present an exact simplification of the configuration-space obstacle check into a sequence of confidence checks over disparity; and 3) We empirically show that the JPP formulation results in substantial computational cost savings on a real robot. Finally, we also provide the full C++ source code implementation of JPP [1].

[1]Sourish Ghosh is with the Department of Mathematics, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. Email: sourishg@iitkgp.ac.in

[2]Joydeep Biswas is with the College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01003, USA. Email: joydeepb@cs.umass.edu

[1]Source Code: https://github.com/umass-amrl/jpp

II. Related Work

Obstacle avoidance, as an essential ability of autonomous mobile robots, has been researched in great detail, and there exist a number of approaches, with various planning algorithms (*e.g.*, [3], [4], [5]), sensor modalities (*e.g.*, [6], [7]) and efficient collision detection based on adaptive cell decomposition of the robot configuration space [8]. We thus focus on the related work most relevant to our own, covering existing approaches to obstacle avoidance using vision.

Vision based obstacle avoidance approaches can be broadly classified into monocular and stereo approaches. Approaches for obstacle detection using monocular vision include learning based methods [9] and using optical flow [10]. Appearance-based obstacle detection [11] and visual sonar [12] have also been shown to be effective for indoor ground robots.

Most stereo vision based approaches [2], [13] perform either local epipolar matching or a global disparity optimization. Recognizing the significant computational cost of dense stereo reconstruction, a few recent methods [14], [15] have tried to reduce computation time by doing sparse disparity checks. In Pushbroom Stereo [15], obstacles are detected only at a constant disparity level, and by integrating this information with an onboard IMU and state estimator, positions of obstacles at all other depths are recovered.

For a different problem of affordance based planning and a different sensor modality of LIDAR, joint perception and planning [16] has been shown to successfully reduce combined planning and perception time by limiting perception to only areas considered by the planner.

While there have been partial informative approaches [15] to obstacle avoidance using stereo vision, and joint perception and planning in other domains and with sensor modalities [16], in this paper we contribute a joint perception and planning approach for the task of obstacle avoidance using stereo vision, where we avoid unnecessary full 3D reconstruction, and further relax the problem of checking reachable configurations into confidence disparity matching.

III. Review of Epipolar Geometry

We use the left camera of the stereo pair as the sensor reference frame. Given a visible 3D point X in the reference frame of the left camera, let p and p' be its projection in the left image and right camera images, respectively. Point X, the image points p and p' (on the image planes), and the camera centers are coplanar. This is known as the epipolar constraint. For an image point p in the left image, there exists a corresponding epipolar line l' in the right image on which p' is constrained to lie. Similarly l is the epipolar line in the left image corresponding to the right image point p'. Hence, when only one of the two image points is known, the corresponding point in the other image can be found by searching along its epipolar line, resulting in a 1-D search.

Stereo camera calibration yields the left and right camera instrinsic matrices, $\mathbf{K}$ and $\mathbf{K}'$ respectively; and the rotation $\mathbf{R}$, and translation $\mathbf{t}$ from the left camera frame to the right camera frame, also known as the extrinsic parameters. The 3×4 projection matrices $\mathbf{P}$ and $\mathbf{P}'$ are defined as

$$\mathbf{P} = \mathbf{K} \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array} \right], \qquad \mathbf{P}' = \mathbf{K}' \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right] \tag{1}$$

where $\mathbf{I}$ denotes the 3×3 identity matrix. Then $p = \mathbf{P}X$ and $p' = \mathbf{P}'X$ in homogeneous coordinates.

Obstacle avoidance path planning is performed in the reference frame of the robot, where the origin coincides with the center of rotation of the robot projected on to the ground plane. Hence we find a transformation from the sensor reference frame to the robot reference frame. Let $\mathbf{R}_w$ and $\mathbf{t}_w$ denote the rotation and translation matrices of this transformation. Therefore X in the robot reference frame is expressed as $X_w = \mathbf{R}_w X + \mathbf{t}_w$.

For dense stereo reconstruction, the depth of an image point is estimated by finding its correspondence along the epipolar line in the other image. In a rectified image coordinate system, the epipolar lines become horizontal scan lines. The horizontal shift or the difference in x-coordinate of two corresponding points in rectified coordinates is thus the disparity. Depth and disparity are related as $d = \frac{fB}{z}$ where d is the disparity, z is the depth of the point from the reference camera, f is the focal length of the camera, and B is the stereo baseline.

IV. Joint Perception and Planning

Unlike Sequential Perception and Planning (SPP), Joint Perception and Planning (JPP) performs a series of sparse stereo correspondence checks based on traversability queries from the local path planner. The queries are to check if a robot pose is reachable. We explore both sample-based, and search-based path planning algorithms in the context of JPP.

Let $\mathcal{X} \subset \mathbb{R}^2$ denote the robot configuration space, which is a set of robot poses $\langle x, y \rangle \in \mathbb{R}^2$. We omit the orientation of the robot in this work, assuming that the robot can turn in place if ncessary. We partition $\mathcal{X}$ into two sets $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{obs}}$, where $\mathcal{X}_{\text{free}}$ denotes the set of poses reachable by the robot and $\mathcal{X}_{\text{obs}}$ denotes the set of poses not reachable by the robot. We define points belonging to the ground plane as those points $(x, y, z) \in \mathbb{R}^3$ such that $z = 0$. The robot safety radius is denoted by r. Any pose $\langle x, y \rangle \in \mathcal{X}_{\text{free}}$ if all the points in the set $\mathbf{P}_r = \{(x', y', 0) : (x - x')^2 + (y - y')^2 < r^2\}$ are classified as ground plane and additionally the set $\mathbf{P}_h = \{(x', y', z) : 0 \leq z \leq h, (x', y', 0) \in \mathbf{P}_r\}$ contains *only* unoccupied points, where h is the robot height. Otherwise $\langle x, y \rangle \in \mathcal{X}_{\text{obs}}$. Algorithm 1 outlines the procedure to check if a pose is reachable or not. The verification of 3D points belonging to the ground plane is done using confident positive matching while verification of points which are unoccupied in space is done using confident negative matching.

The local path planner explores a directed graph $G = \langle V, E \rangle$ on the configuration space, where $V \subset \mathcal{X}_{\text{free}}$ denotes the set of vertices and E denotes the set of edges.

Search-based Planning. The configuration space is discretized into a square grid of cell size s. The neighbors of any node $v = \langle x, y \rangle$ are $\mathbf{N}_v = \{\langle x+s, y+s \rangle, \langle x+s, y \rangle, \langle x+$

$s, y-s\rangle, \langle x, y-s\rangle, \langle x, y+s\rangle\}$. Any pose $p = \langle x, y\rangle$ is added to V if $p \in \mathbf{N}_v$ for some $v \in V$ and REACHABLEPOSE(p) outlined in Algorithm 1 is *true*. We have presented results for JPP with A* search in this paper.

Sampling-based Planning. We have implemented JPP with RRT as an instance of sampling-based planning. A uniform sampler with a goal bias b samples a new pose $p \in \mathcal{X}$ in the configuration space. From V we find a pose q such that $\|p - q\|$ is minimum ($\|\cdot\|$ denotes Euclidean distance). Then a steering function STEER : $(p, q) \mapsto m$ returns a pose $m \in \mathcal{X}$ in the direction of p from q at distance of step size s. Let $L(q, m)$ denote the set of all poses lying on the edge joining q and m. m is added to the set of vertices V if $L(q, m) \subset \mathcal{X}_{\text{free}}$. Formally, (q, m) is a valid edge if for all $n \in L(q, m)$, REACHABLEPOSE(n) is *true*.

In both instances of the planning algorithms, edges of the planning graph are validated by performing on-demand confidence matching which is described in the next section.

V. ON-DEMAND STEREO

Let $\mathbf{P}$ and $\mathbf{P}'$ (Equation 1) be the projection matrices in the rectified coordinate system of the left and right cameras respectively. Therefore any 3D point X (represented as a 3×1 matrix) in the robot reference frame is first transformed into the left camera reference frame as

$$X_l = \mathbf{R}_w^{-1}(X - \mathbf{t}_w) \tag{2}$$

and then X_l is projected as image points p and p' in the rectified stereo images I and I', where $p = \mathbf{P}X_l$ and $p' = \mathbf{P}'X_l$ (in homogeneous coordinates). Given two image points $p(u, v)$ and $p'(u', v')$ in the rectified stereo images I and I', we define the SAD cost function as

$$\mathcal{C}(p, p', w) = \sum_{\substack{q \in \tau(p,w) \\ q' \in \tau(p',w)}} |\mathbf{D}(q) - \mathbf{D}(q')| \tag{3}$$

where $\tau(p, w)$ is a window of pixels of size $w \times w$ centered around the point p and similarly $\tau(p', w)$ is defined around p'. $\mathbf{D}$ denotes the pixel descriptor of an image point. For our implementation we used the DAISY descriptor [17].

In SPP, given any point p in the left image I, its disparity d is determined by finding the best corresponding match in the right image I' by scanning along the epipolar line and finding a point q such that $\mathcal{C}(p, q, w)$ is minimum over the epipolar line. Formally

$$d = \arg\min_{d \in \mathcal{D}} \mathcal{C}(p(u, v), q(u - d, v), w) \tag{4}$$

where $\mathcal{D} = [0, d_{\text{max}} - 1]$ is the set of possible disparity values of p. Disparity refinement is done using left-right consistency checks, and low confidence matches are neglected using a threshold on the ratio of cost of the top two disparity candidates. This reference local disparity matching algorithm for SPP is used to compare paths with JPP. However, in JPP, instead of attempting to find the *exact disparity* of a point, we are interested in checking whether the point is on the surface of an obstacle as verified by a *confident positive match*, or whether the point is not on the surface of an obstacle, as verified by a *confident negative match*.

A. Confident Positive Matching

A visible 3D point $X \in \mathbb{R}^3$ under consideration of the obstacle avoidance planner is first projected on to its corresponding rectified image coordinates p and p'. The confident positive match verifies the occupancy of the point X. To do this we need to verify that p and p' are valid stereo correspondences. A function $\mathcal{L}_+(X)$ used to label point X is defined as

$$\mathcal{L}_+(X) = \begin{cases} 1, & \text{if } \mathcal{C}(p, p', w) \leq \epsilon^+ \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where ϵ^+ is a constant associated with points classified as ground plane. $\epsilon^+ = 1.1$ is found experimentally, and it depends on the length of the descriptors, and the SAD window size. A confident positive match is valid if $\mathcal{L}_+(X) = 1$ and invalid if 0. We use confident positive matching to verify points belonging to the ground plane. To increase the robustness of the confident positive matching, we apply a spatial filter around X to remove noisy estimates. We select a discretized window W with grid size w_x around X and count the number of points $X' \in W$ such that $\mathcal{L}_+(X') = 1$. Formally let $n_x = \sum_{X' \in W} \mathcal{L}_+(X')$. If $n_x > c_x n(W)$, where c_x denotes the spatial filter threshold and $n(W)$ denotes the cardinality of the set W, then we let $\mathcal{L}_+(X) = 1$ otherwise $\mathcal{L}_+(X) = 0$. Experimentally, for best results we set $W = 5$ cm $\times$ 5 cm, $w_x = 10$ mm, $c_x = 0.75$.

B. Confident Negative Matching

Confident negative matching is used to verify that a 3D point $X \in \mathbb{R}^3$ is unoccupied. The procedure is almost similar to that of confident positive matching. First X is projected into corresponding rectified image point coordinates p and p'. Then a function $\mathcal{L}_-(X)$ used to label point X is defined as

$$\mathcal{L}_-(X) = \begin{cases} 1, & \text{if } \mathcal{C}(p, p', w) \geq \epsilon^- \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where ϵ^- is a constant associated with unoccupied points. $\epsilon^- = 0.5$ is found experimentally, and it also depends on the length of the DAISY descriptors, and the SAD window size. X is classified as unoccupied if $\mathcal{L}_-(X) = 1$ and as occupied if $\mathcal{L}_-(X) = 0$. We use the confident negative checks to verify that a column of points above a 3D point (classified as ground plane) is unoccupied *i.e.*, it does not contain any obstacle. The column of points will be unoccupied if for all points X belonging to that column, $\mathcal{L}_-(X) = 1$. If for some X, $\mathcal{L}_-(X) = 0$ then it is not an unoccupied column. It is important to note here that $\epsilon^+ \neq \epsilon^-$. A similar spatial filter is also implemented for confident negative matching.

C. Reachable Poses

Using the definitions of confident positive and negative matching we can classify any pose $\langle x, y\rangle \in \mathcal{X}$ as reachable or not reachable by the robot. Algorithm 1 outlines the procedure for the classification. The general idea of this algorithm is that for any pose $\langle x, y\rangle$ to be reachable by the robot, the set of 3D points points $\{(x', y', 0) : (x - x')^2 +$

TABLE I: JPP parameters for experimental results

Name	Symbol	Domain	Value
State space grid cell size	s	> 0	5 cm
RRT goal bias	b	$(0, 1)$	0.6
Conf. positive threshold	ϵ^+	> 0	1.1
Conf. negative threshold	ϵ^-	> 0	0.5
Spatial filter window size	w_x	> 0	10 mm
Spatial filter threshold	c_x	$(0, 1)$	0.75

$(y - y')^2 < r^2\}$ (in the robot reference frame) need to be verified as points belonging to the ground plane by confident positive checks. Additionally, the column of points starting from $(x', y', 0)$ up to the robot height h *i.e.*, up to (x', y', h) should be verified as unoccupied by using the confident negative matching. r denotes the safety radius of the robot.

We consider two scenarios of the world: (1) *convex world* (2) *non-convex world*. In a convex world if any point $P(x, y, z)$ is unoccupied then the set of points $\{P'(x, y, z') : 0 \leq z' \leq z\}$ are also unoccupied. In such a case lines 11-16 in Algorithm 1 are not required *i.e.*, the confident negative matching is omitted. Omitting the extra confident negative checks speeds up JPP, but such simplification is only reasonable in environments without obstacles with overhanging parts. Table I lists the values of the parameters used by JPP for our experimental results.

Algorithm 1 Check if pose p is reachable by the robot

1: **procedure** REACHABLEPOSE($p(x, y)$)
2: $w \leftarrow$ robot width
3: $l \leftarrow$ robot length
4: $h \leftarrow$ robot height
5: $r \leftarrow \max(\frac{w}{2}, \frac{l}{2})$ ▷ robot safety radius
6: $\mathbf{P}_r \leftarrow \{(x', y', 0) : (x - x')^2 + (y - y')^2 < r^2\}$
7: **for each** $P \in \mathbf{P}_r$ **do**
8: **if** $\mathcal{L}_+(P) == 0$ **then**
9: $\mathcal{X}_{\text{obs}} \leftarrow \mathcal{X}_{\text{obs}} \cup \{p\}$
10: **return false**
11: **else**
12: $\mathbf{P}_h \leftarrow \{(P_x, P_y, z) : 0 \leq z \leq h\}$
13: **for each** $P' \in \mathbf{P}_h$ **do**
14: **if** $\mathcal{L}_-(P') == 0$ **then**
15: $\mathcal{X}_{\text{obs}} \leftarrow \mathcal{X}_{\text{obs}} \cup \{p\}$
16: **return false**
17: $\mathcal{X}_{\text{free}} \leftarrow \mathcal{X}_{\text{free}} \cup \{p\}$
18: **return true**

VI. EXPERIMENTAL RESULTS

We performed two sets of experiments to 1) evaluate the computational cost of JPP compared to SPP, and 2) to compare the path length of obstacle avoidance as evaluated by online JPP compared to an offline, high-resolution dense SPP as a reference baseline. In both experiments, we compared results using obstacle avoidance planning using both A* as well as an RRT planner. The first set of experiments were performed in simulation as well as on a real robot, a Clearpath Jackal UGV (Figure 2), equipped with two PointGrey Blackfly IMX 249 cameras, Kowa LM6HC lenses, and an Intel NUC for onboard processing. Over all experiments, the rectified stereo image resolution was scaled to 320×200 pixels. The grid cell/step size was set to 5cm.

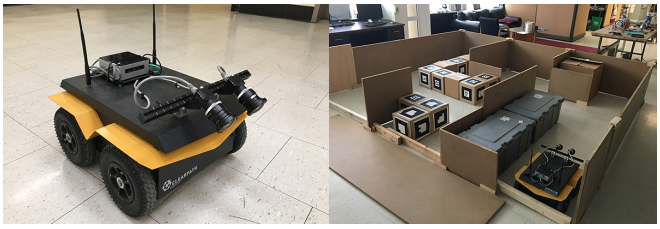
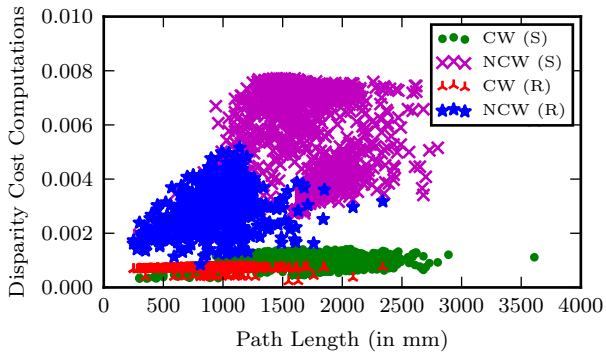


Fig. 2: Clearpath Jackal UGV robot used for real-world experiments.

A. Computational Efficiency

Simulation Tests. We created a simulator that spawns random obstacles in front of a robot equipped with stereo cameras. The obstacles were in the shape of cylinders with a fixed radius and height. We assumed a world size of 6m x 6m. 100 obstacles with base radius of 8cm and height of 40cm were spawned randomly in each simulation. In each simulation, we set an end waypoint of 2m ahead of the robot center. We ran 46,000 simulations each for RRT and A*, and logged the total number of disparity cost computations (SAD checks) for both convex and non-convex world scenarios. The goal of this experiment was to compare JPP with SPP and evaluate the number of disparity cost computations. This experiment was also designed to find experimental bounds on the number of computations required for different planners. The number of computations required by SPP is a constant for every simulation since it reconstructs a dense 3D scene by a local cost aggregation method (2560000 for an image resolution of 320 x 200 with a maximum allowable disparity of 40 pixels). The computations required by the planner is negligible compared to computations for full reconstruction. We plotted the number of computations taken by JPP as a fraction of computations by SPP with the x-axis as the path length. Figure 3 shows that for A* in simulation, the fraction of computations is less than 0.9% for the non-convex scenario and less than 0.2% for the convex scenario. For RRT the numbers are 10% and 2% respectively. Figure 4 shows a cumulative histogram of the fraction of the computations. We thus verify our hypothesis from these figures and also conclude that the complexity of JPP is a function of the complexity of the path planner.

Real World Tests. We used the Jackal to verify the bounds found in simulation and test robustness on detecting and avoiding obstacles in real world. The tests were done both for non-convex and convex world assumptions. Figures 3 and 4 clearly show that the number of computations required in real world is within the simulation bounds. For A*, the numbers are 0.7% (non-convex world) and 0.2% (convex world). For RRT the numbers are 7% and 2% respectively. Figure 6 shows that our method is robust and efficient in detecting obstacles and planning safe paths around them. Figure 6 also verifies that only sparse disparity checks ($\mathcal{L}_+$ and $\mathcal{L}_-$) are required for obstacle avoidance. We also deduce visually



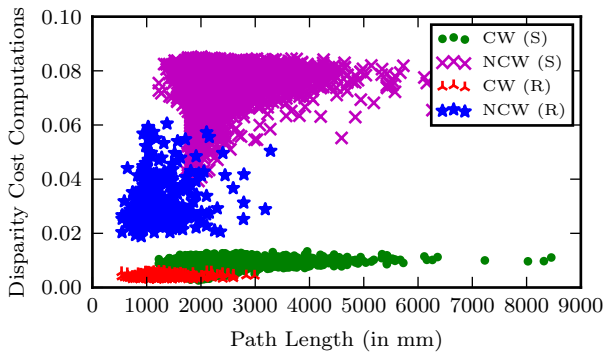
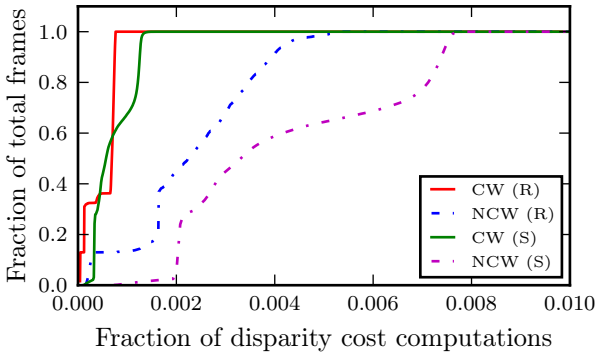


Fig. 3: Fraction of total disparity computations of JPP compared to SPP, vs. Path Length. **Top**: A* planner. **Bottom**: RRT planner. Red and blue points are from real world data while purple and green are from simulation. The number of computations is expressed as a fraction of that required by SPP. **CW**: convex world. **NCW**: non-convex world. **S**: simulation. **R**: real world.



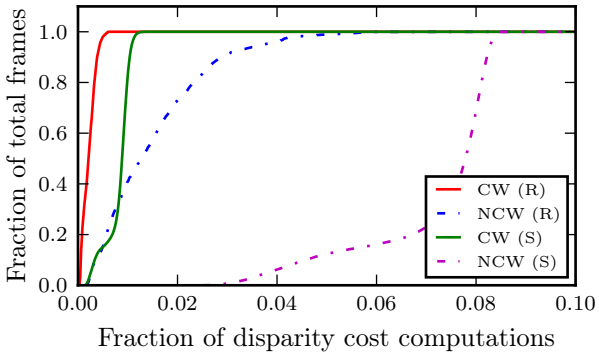


Fig. 4: Cumulative histogram of the fraction of disparity cost computations compared to SPP. **Top**: A* planner. **Bottom**: RRT planner. **CW**: convex world. **NCW**: non-convex world. **S**: simulation. **R**: real world.

that the number of computations varies based on the type of planner we use. From the plots, we can deduce that RRT does 10x more computations than A*.

B. Path Quality

To evaluate the quality of paths generated by JPP, we compared them to reference paths generated offline by SPP from dense 3D reconstructions on high resolution images of resolution 1920×1200 pixels. Note that the reference paths are indicative of the highest possible quality that can be generated from stereo vision, and cannot actually be run in real-time due to their significant computational cost: they require more than 2 minutes to generate per frame. We use the Hausdorff distance $\mathcal{H}$ to compare two paths $\mathbf{P}_1$ and $\mathbf{P}_2$. $\mathcal{H}$ is defined as

$$\mathcal{H}(\mathbf{P}_1, \mathbf{P}_2) = \max_{p \in \mathbf{P}_1} \left(\min_{q \in \mathbf{P}_2} (\|p - q\|) \right) \tag{7}$$

where p and q are points that make up the paths $\mathbf{P}_1$ and $\mathbf{P}_2$. Figure 5 shows a cumulative histogram of the Hausdorff distances for both RRT and A*. From the histogram we see that the paths generated by JPP and from high resolution 3D reconstruction are comparable as more than 80% of the cases have a Hausdorff distance of less than 0.6m.

The computational complexity of JPP is invariant of image resolution as the number of disparity checks is guided by the path planning algorithm. Recall that we project a 3D point in the robot reference frame into two image points and perform confidence matching only. In SPP we are forced to work with low resolution images since 3D reconstruction is expensive for high resolution images. We take advantage of this fact, to perform more confident and robust disparity checks using SAD with a bigger window size on higher resolution images, and utilize the saved computation time.

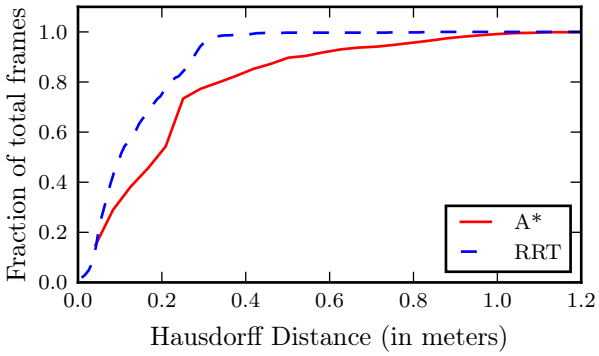


Fig. 5: Hausdorff distance of paths generated by online JPP, compared to offline high-resolution dense SPP reconstruction.

VII. Conclusion and Future Work

In this paper, we introduced a novel joint perception and planning (JPP) algorithm for obstacle avoidance using stereo vision. We showed experimentally that the JPP requires significantly fewer computational resources, while still maintaining high path quality. Since the total number of SAD disparity cost checks in JPP is invariant of the image

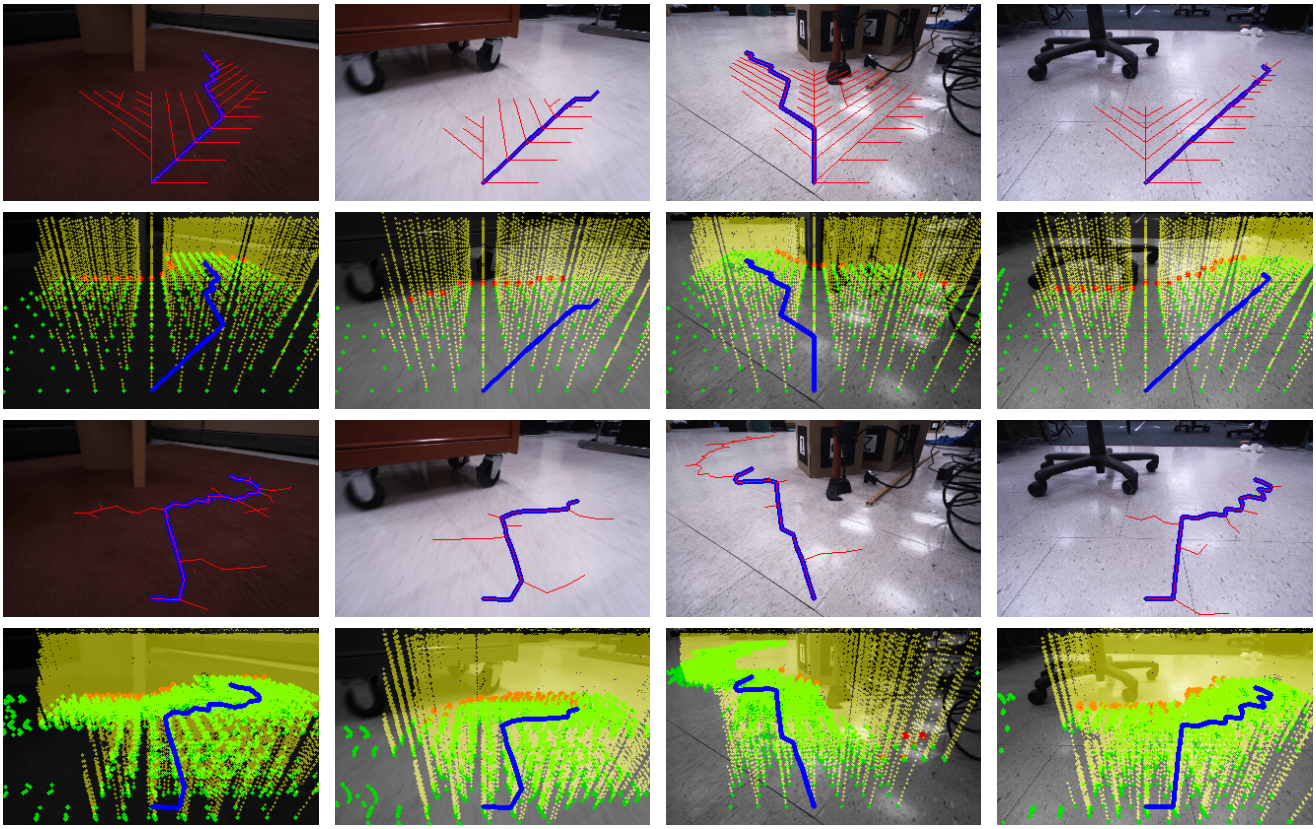


Fig. 6: Joint perception and planning on the real-world dataset collected at the Autonomous Mobile Robotics Laboratory, UMass Amherst. **Row 1**: A* planner. Red curves indicate the explored planning graph, blue curve indicates the path planned. **Row 2**: Confident positive/negative matching visualizations. Green points belong to the ground plane which are found by confident positive checks, red points indicate configurations not reachable by the robot. Yellow points represent the unoccupied points found by confident negative matching. **Row 3 and 4**: RRT planner. The colour coding is same as that of A*. It is evident from these visualizations that JPP performs sparse disparity checks as compared to dense reconstruction. Note that RRT performs more dense checks than A*.

resolution, analysis of the number of disparity computations as a function of the planning problem is a promising direction for future work.

REFERENCES

[1] D. Murray and J. J. Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 161–171.

[2] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, "Obstacle Avoidance and Path Planning for Humanoid Robots using Stereo Vision," in *Robotics and Automation, IEEE International Conference on*, 2004, pp. 592–597.

[3] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," in *Robotics and Automation, IEEE International Conference on*, 1990, pp. 572–577.

[4] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Robotics and Automation, IEEE International Conference on*, 1996, pp. 3375–3382.

[5] M. Tang, Y. J. Kim, and D. Manocha, "CCQ: Efficient Local Planning using Connection Collision Query," in *Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 229–247.

[6] K. O. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-Time Obstacle Avoidance for Polygonal Robots with a Reduced Dynamic Window," in *Robotics and Automation, IEEE International Conference on*, 2002, pp. 3050–3055.

[7] J. Biswas and M. Veloso, "Depth Camera Based Indoor Mobile Robot Localization And Navigation," in *Robotics and Automation, IEEE International Conference on*, 2012, pp. 1697–1702.

[8] L. Zhang, Y. J. Kim, and D. Manocha, "A Simple Path Non-Existence Algorithm Using C-Obstacle Query," in *Algorithmic Foundation of Robotics VII*, 2008, pp. 269–284.

[9] J. Michels, A. Saxena, and A. Y. Ng, "High Speed Obstacle Avoidance using Monocular Vision and Reinforcement Learning," in *International Conference on Machine Learning*, 2005, pp. 593–600.

[10] K. Souhila and A. Karim, "Optical Flow Based Robot Obstacle Avoidance," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, p. 2, 2007.

[11] I. Ulrich and I. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," in *AAAI/IAAI*, 2000, pp. 866–871.

[12] S. Lenser and M. Veloso, "Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 2003, pp. 886–891.

[13] R. Simmons, L. Henriksen, L. Chrisman, and G. Whelan, "Obstacle Avoidance and Safeguarding for a Lunar Rover," in *AIAA Forum on Advanced Developments in Space Robotics*, 1996.

[14] M. Kumano, A. Ohya, and S. Yuta, "Obstacle Avoidance of Autonomous Mobile Robot using Stereo Vision Sensor," in *Intl. Symposium on Robotics and Automation*, 2000, pp. 497–502.

[15] A. J. Barry and R. Tedrake, "Pushbroom Stereo for High-Speed Navigation in Cluttered Environments," in *Robotics and Automation, IEEE International Conference on*, 2015, pp. 3046–3052.

[16] W. Pryor, Y.-C. Lin, and D. Berenson, "Integrated Affordance Detection and Humanoid Locomotion Planning," in *Humanoid Robots, IEEE-RAS International Conference on*, 2016, pp. 125–132.

[17] E. Tola, V. Lepetit, and P. Fua, "Daisy: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.