Perception Engine Using a Multi-Sensor Head to Enable High-level Humanoid Robot Behaviors

Bhavyansh Mishra[1,2], Duncan Calvert[1,2], Brendon Ortolano[1,2], Max Asselmeier[1], Luke Fina[1,2], Stephen McCrory[1,2], Hakki Erhan Sevil[2] and Robert Griffin[1,2]

***Abstract*— For achieving significant levels of autonomy, legged robot behaviors require perceptual awareness of both the terrain for traversal, as well as structures and objects in their surroundings for planning, obstacle avoidance, and high-level decision making. In this work, we present a perception engine for legged robots that extracts the necessary information for developing semantic, contextual, and metric awareness of their surroundings. Our custom sensor configuration consists of (1) an active depth sensor, (2) two monocular cameras looking sideways, (3) a passive stereo sensor observing the terrain, (4) a forward facing active depth camera, and (5) a rotating 3D LIDAR with a large vertical field-of-view (FOV). The mutual overlap in the sensors' FOVs allows us to redundantly detect and track objects of both dynamic and static types. We fuse class masks generated by a semantic segmentation model with LIDAR and depth data to accurately identify and track individual instances of dynamically moving objects. In parallel, active depth and passive stereo streams of the terrain are also fused to map the terrain using the on-board GPU. We evaluate the engine using two different humanoid behaviors, (1) look-and-step and (2) track-and-follow, on the Boston Dynamics Atlas.**

I. INTRODUCTION

Legged robots such as bipedal and quadrupedal robots boast the ability to perform various behaviors that are either impossible or highly challenging for other robot forms to achieve. Legged robot behaviors include walking over rough terrain, climbing stairs, opening doors, moving heavy loads, engaging in co-manipulation tasks, as well as freestyle athletics. However, achieving high-level behaviors such as those performed routinely by humans, requires robust and reliable awareness about the surrounding environment with reasonably high-level understanding of the world. Humans daily go through an enormous variety of environments, such as morning hygiene routines, navigating buildings, driving vehicles, and cooking food. All such tasks require humans to subconsciously track various objects and activities in their surroundings, such as, vehicles on roads, other humans in the environment, open doors, walls, stairs, etc. However, seemingly effortless tasks performed by humans can be very challenging to achieve for robots, at least from the perspective of environmental awareness. Being able to successfully track objects and events is an essential part of robot perception, and requires a pipeline for extracting and processing both metric and semantic information from sensor data.

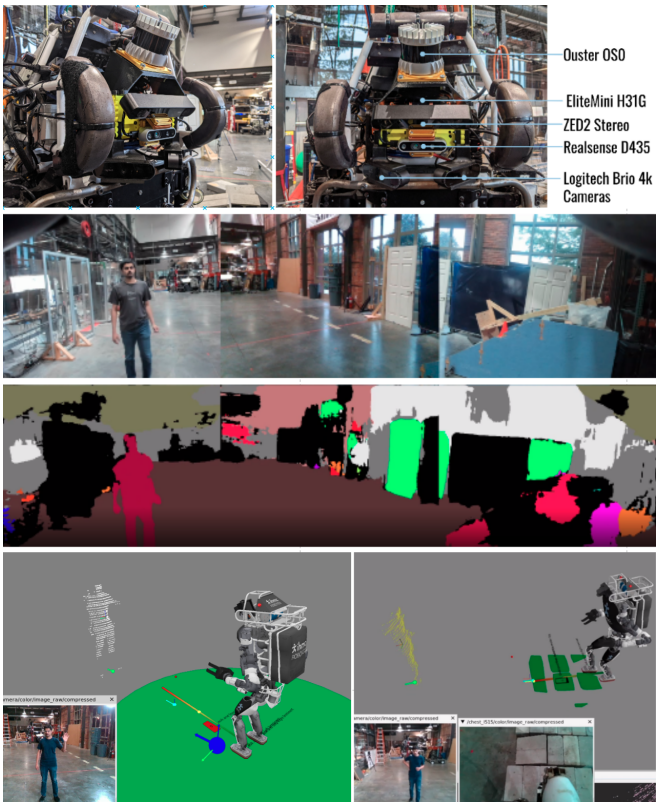

Fig. 1: Multi-Sensor Head on Atlas (first-row), combined image from Logitech Brio and D435 color streams (second-row), combined semantic mask for the image above (third-row), and screenshot of the track-and-follow behavior user interface (fourth-row) for both flat-ground (left) and rough terrain (right).

In this work, we present a semantic-metric method for perceptual awareness of legged robots in indoor environments, leveraging a custom designed and highly redundant multi-sensor head. We develop a system for tracking multiple semantically meaningful objects in 3D metric space over a large field-of-view using the novel sensor-head. An overview of the sensor head on Atlas, sensor fusion, semantic segmentation, and high-level behaviors is shown in Fig. I. We evaluate the system by performing look-and-step and track-and-follow behaviors with the Boston Dynamics DRC Atlas, both while walking on flat-ground and on rough terrain. The primary contributions of this paper are:

1) Custom and highly redundant sensor-head with three monocular cameras spanning a large FOV, an active depth sensor, a rotating LIDAR, and a passive stereo

*This work was supported through ONR Grant No. N00014-19-1-2023 and NASA Grant No. 80NSSC20M0197.

[1]Author is with the Institute of Human and Machine Cognition (IHMC), 40 S Alcaniz St, Pensacola, FL 32502, USA author@ihmc.org

[2]Author is with the University of West Florida (UWF), 11000 University Pkwy, Pensacola, FL 32514, USA author@uwf.edu

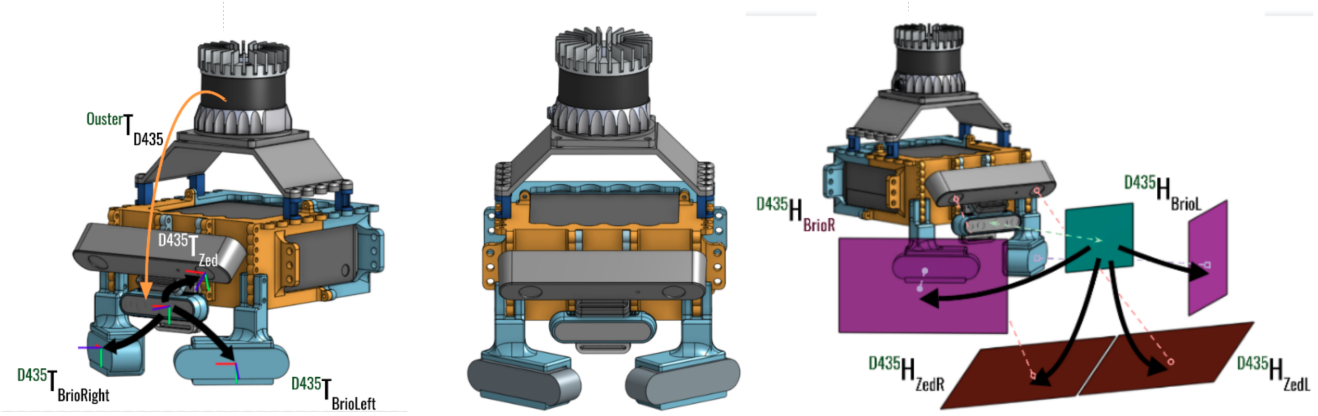

Fig. 2: Multi-Sensor Head Configuration with relative transforms and sensor mount positions for side-cameras (Logitech Brio 4K), active depth camera (Intel RealSense D435), stereo pair (ZED 2), and LIDAR on top (Ouster OS0-128). The on-board computer inside the casing is a Minisforum H31G MiniPC. The complete sensor head was designed for maximizing effective field-of-view on all sensors.

pair.

2) Fused semantic segmentation of both LIDAR point-cloud and color images simultaneously from three camera streams at 8-10 Hz.
3) Three-dimensional Multiple Object Tracking (MOT) framework to track multiple moving entities over a large FOV.
4) In our knowledge, the first work on human-following behavior for bipedal humanoid robots that can follow humans over both flat-ground and rough terrain environments.

II. RELATED WORK

Bipedal humanoid robots have the mobility to traverse complex terrains and cluttered environments, but require a higher level of complexity in their perception and planning systems to achieve this goal. Simultaneously, performing high-level and meaningful behaviors over rough terrain requires another layer of robustness associated with semantic understanding of the world. To the best of our knowledge, we are the first to present a perception engine for achieving semantically meaningful and high-level behaviors, such as person-following, for a bipedal humanoid robot while even moving over rough terrain. Therefore, the related work can broadly be classified into two main areas: (1) works on multi-sensor fusion for developing semantic understanding of the world, and (2) previous approaches to person-following behavior on legged or humanoid robots.

1) Sensor Fusion for Semantic Understanding: Semantic scene understanding has been explored by several works in the past. A theme in a subset of such papers has been to extract geometric primitives from sensor data as building blocks before further higher-level processing. The approach taken by Grotz et al. [8] first extracts geometric primitives such as planes, cylinders and spheres from RGB-D point-cloud using the Locally Convex Connected Patches (LCCP) algorithm, and fuses the geometric primitives spatio-temporally. In parallel, the input color images are used to extract semantic 2D bounding boxes using the YOLO object detection algorithm, which are then combined with geometric primitive information spatially into a scene-graph for inferring higher semantic structures in the scene. However, learning-based representations of the input were shown to outperform hand-crafted representations and policies.

Vora et al. [18] present the work on PointPainting which obtains class scores using an image-only semantic segmentation network, and then augments the pointcloud with the score vector. They show that LIDAR-only segmentation networks can then be applied to the augmented pointcloud for improved accuracy in 3D segmentation. Although, such an approach accurately segments the pointcloud, the overall computational cost grows significantly as it requires two different segmentation networks to be used, and the augmented pointcloud requires even higher memory usage.

Learning-based methods for extracting 3D objects such as pedestrians, cars, cyclists, etc. from point clouds and images have been shown to perform well on the KITTI dataset (PointRCNN [15], PointNet [13], VoxelNet [21], MV3D [3], AVOD [9], PIXOR [19], Complex YOLO [16]). Point pillars were proposed by Lang et al. [10] as an efficient organization of point cloud for end-to-end extraction of oriented bounding boxes for objects. SemanticVoxels [5] further generalized the approach taken by PointPillars for fusing semantic colored image features and geometric LIDAR features to achieve superior 3D object detection results on the KITTI dataset. Madawi et al. [4] also explore 3D semantic segmentation by fusing both LIDAR scans and color images into a Polar Grid Map (PGM) tensors. They implement custom network architectures for fusion of color and depth both before and after the feature extraction stages of the architecture, using SqueezeSeg and PointSeg as the baseline models. They achieve higher accuracy than the baseline models only with slightly higher computational costs. Although such learning-based segmentation and detection networks perform well, all single-frame techniques are prone to exhibit false detections which can lead high-level planners to make incorrect decisions. Therefore, we combine this part of the literature with that on Multi-Object Tracking (MOT) [14, 1] to discard false detections and smooth out the trajectories of various semantic objects over both space and time.

2) Person-Following on Legged and Humanoid Robots: Following humans safely and robustly is an important task in the field of Human-Robot Interaction (HRI) and is vital in disaster relief and search-and-rescue applications. However, most human-following frameworks have been either developed for structured indoor environments or wheeled robots with limited traversability.

Goldhoorn et al. [6] present experimental results on a wheeled-humanoid robot following a human target in an urban setting. They employ two novel methods using Partially Observable Monte-Carlo Planning (POMCP), with their best-performing method being a compound algorithm that uses heuristic path planning when humans are detected, and Monte Carlo simulations otherwise. Their robot was able to follow a human for over 3 kilometers over the span of 3 hours, even when the human was not visible. However, since such methods rely on a finite number of discrete actions and observations, they are usuallly unsuitable for high-dimensional systems such as legged humanoid robots in complex environments.

Zhang et al. [20] present the first person-following framework for quadrupedal locomotion. Their robot uses a LIDAR, a depth camera, an IMU, and odometry to build a local traversability-based cost map and find the pose of the tracked human. This data is fed into their motion planner for the generation of an initial coarse path, which is then optimized to minimize time and acceleration under simplified kinodynamic constraints. They evaluate their algorithm in indoor and outdoor environments using a JueYing quadrupedal robot running their module at 10 Hz.

III. SENSOR CONFIGURATION

The front of the sensor head contains three different cameras as shown in both Fig.I and Fig.2. The depth camera in the middle (Intel D435) offers a resolution of 640×480 on both color and depth images, and looks straight forward. Additionally, two monocular 1280×720 cameras (Logitech Brio) face sideways at yaws of +40 and -40 degrees from forward. The image streams from all the monocular color cameras are stitched together for being used by the human robot operator for selecting a target to follow and generally interacting with the perception engine.

The sensor head is equipped with a 360-degree LIDAR (Ouster OS0-128) consisting of 128 vertical channels and 2048 scan points per channel. This LIDAR is capable of generating scans of the environment at 10 Hz with a maximum range of 120 metres. The LIDAR is primarily used to track semantically meaningful objects in 3D, and improving the redundancy of the object detection and tracking system. Other sensor types and approaches for depth extraction could also have been used, however, we found the Ouster to be a reasonable solution for acquiring accurate and long-range depth over a wide FOV.

The sensor head also contains a passive stereo pair (ZED-2) facing down with a pitch of 20 degrees, used for generating passive depth maps. The robot also uses active depth from an Intel RealSense L515 sensor attached to the chest looking down on the terrain to extract planar regions as simplified representation of the terrain in front of the legged robot. We use the GPU algorithm presented by Mishra et al. [11] to segment the depth into multiple approximately-convex planar regions. Passive stereo depth is used as a redundancy for this task of planar region extraction as the height and surface normal are further constrained for improved footstep planning.

A dedicated on-board computer was also added to the sensor head for both interfacing with the sensors, as well as performing perception algorithms on the incoming data. The on-board computer ultimately connects to a 10 Gbps Ethernet switch on the robot for high-bandwidth transfer of processed data to all parts of the systems. Specifications of the on-board computer were chosen to be Intel Core i7 8th Gen, GTX 1050 Ti, and 32 GB of memory.

Fig. 3: Fused Semantic Segmentation and Instance Clustering for Multiple Human Class Objects in the Scene.

1) Calibration and Internal Parameters: Once the hardware for the sensor head was assembled and the on-board computer setup with necessary sensor drivers, calibration routines were performed to extract the following parameters for all 5 cameras on the head (illustrated in Fig.2): the 4×4 camera intrinsic parameter matrix M_{sensor}, the 3×3 homography matrix H_{sensor} from the sensor image plane Ω_C to the base image plane Ω_B, and the 4×4 homogeneous rigid-body transform T_{sensor} from camera frame R_C to the base frame R_B. Camera intrinsic and extrinsic parameters were calculated using Zhang's method [2], and used to generate the pinhole model for projecting LIDAR 3D points onto all the cameras as,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{d} \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (1)$$

where f_x, f_y, c_x, and c_y are focal lengths and centers of projection along x and y axes.

2) Warping and Image Fusion: We then calculate the homographies H_{sensor} between various cameras and base

image plane (D435), using ORB feature correspondences (ZED 2 and Logitech Brio cams) and warp the non-base images onto the base image plane using,

$$\begin{bmatrix} x\lambda \\ y\lambda \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (2)$$

and obtain the final stitched image $\beta(x,y)$ to be used for robot operator interfacing.

IV. APPROACH

The overall approach taken by the perception engine is to detect and track semantic entities in the environment and use them for informing high-level robot behaviors, such as building exploration, person following, manipulation, or other forms of physical interaction. The pipeline first attempts to detect all semantically meaningful objects and structures in the environment using a semantic segmentation model on the combined image stream from the multi-camera system available on the sensor head. The generated semantic mask with class labeled pixels is then used to segment the LIDAR pointcloud using the semantic class masks. The point cloud is then split into blobs belonging to different classes, which are then further separated into smaller pointclouds representing the various instances of each semantic class. The instance-separated pointclouds are further used for tracking object instances over time and providing goal locations to high-level behaviors such as the person-following behavior.

A. Semantic Segmentation of Images

Semantic segmentation is used to generate class-wise masks for classifying and segmenting various objects observed in the color image streams as shown in Fig.3. Particularly, we employ a RefineNet model with a ResNet-101 backbone [17] which enables us to achieve real-time semantic image segmentation. The RefineNet model was trained first on the combined COCO+BSD+VOC datasets, and then trained further on the ADE20K dataset for 90 epochs with learning rate decrements after every 30 epochs. A subset of ADE20k containing indoor classes was selected and augmented with operations such as randomized mirroring, cropping, downsampling, and upsampling along with padding.

The training dataset consisted of 20,210 training images, 2,000 validation images, and 3,000 testing images. More information about the classes that were included in training is available in Table II. The percentage of instances and pixels are both values obtained from the ground-truth masks in the training and validation data sets, and the IOU scores were obtained as the highest mean IOU score across all classes.

B. Terrain Mapping

Legged robots specifically need fast and reliable system for terrain surface extraction to be able to generate feasible footstep plans for locomotion. In this regard, we use an active depth sensor to extract representations of terrain surface in form of polygonal planar regions. The look-and-step behavior on the robot is responsible for planning a sequence of feasible footholds for the robot to step on using the most recent set of planar regions. We use our GPU-accelerated algorithm for planar region extraction, which segments the depth map into planar regions by dividing the depth into patches of pixels and then grouping nearby patches with similar surface normals together into polygonal regions [11].

TABLE I: Information regarding the training dataset used along with evaluation metrics

ID	Name	Frequency [%]	Pixels [%]	IOU
0	Void	66.09	32.75	0.577
1	Wall	6.81	29.70	0.669
2	Floor	5.44	11.21	0.715
3	Ceiling	3.84	7.63	0.692
4	Window	2.73	3.48	0.451
5	Cabinet	1.68	3.92	0.452
6	Person	2.97	2.28	0.676
7	Door	3.96	1.91	0.224
8	Table	2.48	2.32	0.348
9	Curtain	1.25	2.16	0.596
10	Chair	1.90	2.14	0.417
11	Stairs	0.52	0.23	0.234
12	Staircase	0.33	0.16	0.092

C. Point-Cloud Classification and Instance Clustering

For object instance extraction, the point cloud $\mathbf{X}$ is broken down into disjoint semantic subsets x_c for the different semantic classes $c \in (human, chair, table, doors, couch, ...)$. We label 3D points X_i by projecting them onto the image plane of a colored camera which overlaps in FOV with the LIDAR, and obtain the semantic class from the segmentation mask for the projected 2D coordinates (u_i, v_i). All points belonging to a particular semantic class are then collected into a single cluster x_c. The cluster is then further divided up into separate clouds for each specific instance of the class. The instance-specific cloud ${}_i x_c$ or *instance cloud* is then used as the final representation of objects in the environment. The centroids of these instance clouds are then fed into the Multiple Object Tracker (MOT) running on a separate thread. The specific MOT track selected by the robot operator is then used as the final target trajectory for the person-following behavior. Due to noise and inaccuracies at various points in the process of instance-wise clustering, the instance cloud does not always contain points evenly distributed throughout the object. This necessitates the need for tracking position of object instances over time.

D. Tracking Dynamic Object Types

Robust and reliable awareness of objects in the surroundings requires tracking object instances through time and space. The pipeline up to this point simply generates three-dimensional detections of object instances at every camera frame. Since every sensor frame of data is independent of the previous frame, a mechanism to maintain state or memory of various instances becomes necessary.

The MOT framework first calculates Histrogram of Oriented Gradients (HOG) representation of the most recent detections and caches them. The HOG feature vectors for

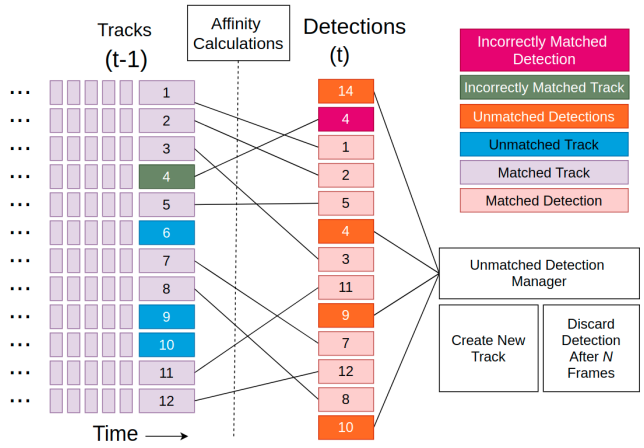

Fig. 4: The process of Multiple-Object Tracking (MOT) in which new detections are fused with their corresponding tracks after matching. In this process, an affinity metric is calculated between all detections and tracks, and used to decide if a detection can be matched to an existing track. Unmatched tracks are discarded after N frames if they cannot be matched to existing tracks.

the latest detections are then matched with those of existing tracks as shown in Fig. 4 by calculating the Hellinger distance as the affinity between them as,

$$d(H_1,H_2) = \sqrt{1 - \frac{1}{\bar{H}_1 * \bar{H}_2 * N^2} * \sum_J \sqrt{H_1(J) * H_2(J)}} \quad (3)$$

where,

$$\bar{H}_k = \frac{1}{N} * \sum_I H_k(I) \quad (4)$$

and N is the total number of bins in the histogram [12]. The framework further employs thresholds the Intersection-over-Union (IoU) of the semantic masks of consecutive detections of objects to ensure that tracks are correctly matched to detections.

For every track, an instance of a Kalman Filter is used to smooth the trajectory of any particular object instance over time. For high-level behaviors such as following an object, the object instance can be represented as simply a point moving through space in 3D.

Often, due to imperfections in identifying corresponding instances over multiple frames, the pipeline ends up losing track of previously seen objects or regaining track of lost objects. We discard unmatched tracks only after no new detections were matched with them for longer than $U = 20$ frames. However, unmatched detections give rise to newly spawned tracks and are included in the list of all tracks. This list of tracks can then be used by human operator to choose which object to follow by clicking on the stitched-image semantic mask displayed on the user-interface. As long as target objects remain at least 0.5 m away from each other, the engine is able to distinguish the different instances apart. As a result detetections were seldom observed to be incorrectly matched to tracks in practice.

E. Humanoid Robot Behaviors

For evaluating the semantic-metric perception engine, we chose a person-following behavior that was tasked with dynamically tracking and following a moving person over rough terrain. De-coupling the task of locomotion from that of target-following, we divided the overall task into two simpler behaviors named look-and-step and track-and-follow.

1) Look-and-Step Behavior: The look-and-step was designed as a low-level behavior that was responsible only for using the most recent local map of the terrain to plan a sequence of footsteps and walk to a pre-defined goal pose. The behavior depended on the GPU-based planar region extraction algorithm [11] and our A*-based footstep planner [7] to walk to the goal pose, one step at a time. The design for this behavior enabled the robot to autonomously walk both forward and backward, depending on where the goal pose was defined. Locomotion on rough-terrain could also be achieved within the same behavior design as the A*-based footstep planner was able to accommodate for footstep position, altitude, yaw, as well as partial footholds.

2) Track-and-Follow Behavior: This behavior was responsible for obtaining the target position position from the perception engine and defining the goal pose input for the look-and-step behavior. The architecture of the track-and-follow behavior was designed to be high-level and abstracted out the robot-specific tasks such as planning, control and perception. The look-and-step was used by the track-and-follow behavior as a low-level utility function. Although the perception engine is capable of acquiring target positions at 10 Hz, the track-and-follow was intentionally restricted from outputting a goal pose to the look-and-step behavior at up to 2 Hz for allowing the footstep planner and controller time to achieve the previous goal state. The planar goal pose (x,y,z,yaw) was calculated as the pose 2m away from the target along the line from target to robot, facing the target. The distance of 2m was mainly chosen as a balance between human safety and lab dimensions.

V. EXPERIMENTS AND RESULTS

We conducted several experiments with Atlas performing the track-and-follow and look-and-step behaviors while following human target around the lab, as shown in Fig.5. In all experiments, both the human target and Atlas were tagged with a motion-capture marker rigid-body on the right shoulder. In the first two trials, the terrain was made simply to be flat-ground. However, in the third trial an unstructured field of cinder blocks was placed between the human target and Atlas.

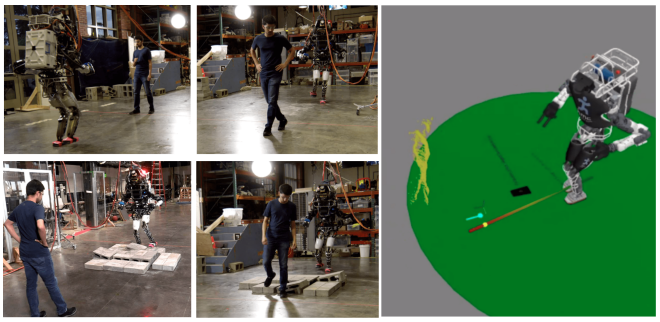

Fig. 5: Human target and Atlas while performing Track-and-Follow Behavior on flat-ground (top-row) and rough terrain (bottom-row).

Fig. 6: Motion Capture trajectories for the human target and Atlas while performing Track-and-Follow Behavior.

A. *Following Target on Flat-Ground*

The first experiment was for Atlas to follow a single person on flat-ground. A single person was tasked with moving in a figure-eight loop trajectory with a circular loop diameter of about 4 meters. Although, the perception pipeline could generate and track human targets at 30 Hz, the track-and-follow behavior was designed to only accept new goal states at 2 Hz. A safe distance of 2 m was used as an offset between the person and the robot. The trajectories in Fig. 6 show that Atlas performed better at maintaining the offset along the X-axis of the motion-capture reference frame, than on the Z-axis. This was mainly due to the fact that our robotics lab is significantly longer along X-axis, but limited in width along Z-axis.

B. *Following Target on Rough Terrain*

For evaluating the robustness of the behavior, Atlas was challenged with following the target on rough terrain. Since the track-and-follow behavior was designed such that the tasks of *walking* and *following* are decoupled, the rough terrain was traversed with very slightly worse accuracy in person-following, shown on the last column in Fig. 6. We calculated the Absolute Displacement Error (ADE) along X and Z axes for all the trials, given in II, as,

$$ADE_x = \frac{1}{N}\sum_{t=0}^{N} |X_{atlas}(t) - X_{target}(t)|. \quad (5)$$

The ADE value of around 2 m along X-axis was consistent with both the safety distance offset 2 m, as desired, and the fact that the line joining the target and robot was along the X-axis for majority of the time due to the lab dimensions.

TABLE II: The Absolute Displacement Error (ADE) along X and Y axes.

Trial	X-axis ADE (m)	Y-axis ADE (m)
1	2.194	0.104
2	2.128	0.290
3	2.025	0.621

VI. CONCLUSION

Throughout our experiments, we were keen on gathering observations regarding some of the limitations and possible extensions to our work. Although our perception engine is capable of dynamically tracking semantically meaningful objects over time and space, it does not maintain a complete map of the world. A probabilistic graph-based mapping back-end could be used to use the semantic landmarks for Simultaneous Localization and Mapping (SLAM) using higher-level semantic features, rather than low-level geometric features. Furthermore, such a semantic-metric understanding of the world could be extended to guide active planning algorithms for enabling robots for making semantically optimal decisions. We would also like to explore passive stereo further.

REFERENCES

[1] Chetan M. Bukey, Shailesh V. Kulkarni, and Rohini A. Chavan. "Multi-object tracking using Kalman filter and particle filter". In: *2017 IEEE International*

Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). 2017, pp. 1688–1692. DOI: 10.1109/ICPCSI.2017.8392001.

[2] Wilhelm Burger. "Zhang's camera calibration algorithm: in-depth tutorial and implementation". In: *HGB16-05* (2016), pp. 1–6.

[3] Xiaozhi Chen et al. "Multi-View 3D Object Detection Network for Autonomous Driving". In: *CoRR* abs/1611.07759 (2016). arXiv: 1611.07759. URL: http://arxiv.org/abs/1611.07759.

[4] Khaled El Madawi et al. "RGB and LiDAR fusion based 3D Semantic Segmentation for Autonomous Driving". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 7–12. DOI: 10.1109/ITSC.2019.8917447.

[5] Juncong Fei et al. "SemanticVoxels: Sequential Fusion for 3D Pedestrian Detection using LiDAR Point Cloud and Semantic Segmentation". In: *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 2020, pp. 185–190. DOI: 10.1109/MFI49285.2020.9235240.

[6] Alex Goldhoorn et al. "Continuous real time POMCP to find-and-follow people by a humanoid service robot". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 741–747. DOI: 10.1109/HUMANOIDS.2014.7041445.

[7] Robert J. Griffin et al. "Footstep Planning for Autonomous Walking Over Rough Terrain". In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. 2019, pp. 9–16. DOI: 10.1109/Humanoids43949.2019.9035046.

[8] Markus Grotz et al. "Graph-based visual semantic perception for humanoid robots". In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. 2017, pp. 869–875. DOI: 10.1109/HUMANOIDS.2017.8246974.

[9] Jason Ku et al. "Joint 3D Proposal Generation and Object Detection from View Aggregation". In: *CoRR* abs/1712.02294 (2017). arXiv: 1712.02294. URL: http://arxiv.org/abs/1712.02294.

[10] Alex H Lang et al. "Pointpillars: Fast encoders for object detection from point clouds". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12697–12705.

[11] Bhavyansh Mishra et al. "GPU-Accelerated Rapid Planar Region Extraction for Dynamic Behaviors on Legged Robots". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021 (Accepted).

[12] OpenCV. *Histogram Comparison Methods*. URL: https://vovkos.github.io/doxyrest-showcase/opencv/sphinx_rtd_theme/enum_cv_HistCompMethods.html#details-d6-dc7-group-imgproc-hist-1ga994f53817d621e2e4228fc646342d386.

[13] Charles Ruizhongtai Qi et al. "Frustum PointNets for 3D Object Detection from RGB-D Data". In: *CoRR* abs/1711.08488 (2017). arXiv: 1711.08488. URL: http://arxiv.org/abs/1711.08488.

[14] M Sharon and Ap Latha. "Multi-Object Tracking Based on Kalman Filter by Using Multi-Feature Appearance Model". In: *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2018, pp. 544–549. DOI: 10.1109/ICIRCA.2018.8597365.

[15] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[16] Martin Simon et al. "Complex-YOLO: Real-time 3D Object Detection on Point Clouds". In: *CoRR* abs/1803.06199 (2018). arXiv: 1803.06199. URL: http://arxiv.org/abs/1803.06199.

[17] Nekrasov Vladimir, Shen Chunhua, and Reid Ian. "Light-Weight RefineNet for Real-Time Semantic Segmentation". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. 2018, p. 125.

[18] Sourabh Vora et al. "PointPainting: Sequential Fusion for 3D Object Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4603–4611. DOI: 10.1109/CVPR42600.2020.00466.

[19] Bin Yang, Wenjie Luo, and Raquel Urtasun. "PIXOR: Real-time 3D Object Detection from Point Clouds". In: *CoRR* abs/1902.06326 (2019). arXiv: 1902.06326. URL: http://arxiv.org/abs/1902.06326.

[20] Zhen Zhang et al. "Efficient Motion Planning Based on Kinodynamic Model for Quadruped Robots Following Persons in Confined Spaces". In: *IEEE/ASME Transactions on Mechatronics* 26.4 (2021), pp. 1997–2006. DOI: 10.1109/TMECH.2021.3083594.

[21] Yin Zhou and Oncel Tuzel. "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection". In: *CoRR* abs/1711.06396 (2017). arXiv: 1711.06396. URL: http://arxiv.org/abs/1711.06396.