Visual Environment perception for obstacle detection and crossing of lower-limb exoskeletons

Manoj Ramanathan[1*], Lincong Luo [1], Jie Kai Er [1], Ming Jeat Foo [1], Chye Hsia Chiam [1], Lei Li [1], Wei Yun Yau [1,2] and Wei Tech Ang [1,3]

***Abstract*— Lower limb exoskeletons offer support for patients suffering from mobility disorders due to injury, stroke, etc. But these devices are not used in day-to-day life and environments due to their limited human-computer interface to perceive and handle different terrains and tasks. In this paper, we introduce a simple vision-based environment perception pipeline for lower-limb exoskeletons for obstacle crossing tasks. The proposed pipeline consists of three stages, namely, ground plane and obstacle detection, estimating obstacle location and dimensions, and obstacle tracking. To reduce noisy artifacts and reliably detect obstacles, we propose a similarity metric based on color, gradient orientation, and 2D surface normal. Depth map of the detected obstacle region is utilized for estimating the obstacle location and dimensions. Also, we consider two obstacle tracking modes for obstacle crossing, visual tracking using a RGB-D camera and positional tracking using a SLAM camera. The proposed vision-based perception pipeline is integrated with an exoskeleton, where we propose a control scheme that can vary step length adaptively to successfully cross detected obstacles. We conduct offline and online experiments to validate the proposed perception pipeline and provide insights on the same. Our experiments show that the proposed pipeline allows exoskeletons to understand their environment and successfully cross obstacles.**

I. INTRODUCTION

Rehabilitative and Assistive robotics offers a unique challenge for visual perception. Lower-limb exoskeletons such as LOKOMAT [1], Ekso [2], Indego [3] are designed to help people with mobility disorders due to spinal cord injuries (SCI), stroke, etc. Stroke-related reduced mobility and disabilities are more common in our population than SCI-related ones. This is because of the increased likelihood of suffering a stroke with an increase in age [4]. There is a distinct difference between SCI and stroke patients' reduced mobility. Most stroke patients are affected on only one side of the body, which exoskeletons need to mobilize only the affected leg while the abled leg can be controlled by the patients. Due to the residual strength on the abled side of the body, it is very difficult to propose and achieve a precision control strategy to move around in daily environments. In this paper, we mainly deal with lower-limb exoskeletons where only one leg is affected.

Terrain understanding and environment perception are crucial elements of an exoskeleton, which are necessary to ensure the stability, balance, and safety of the patients. But existing exoskeletons are not popular for usage in a daily home environment as tasks such as obstacle crossing, walking on a soft carpet, stair climbing is still considered complex. Recent progress in computer vision has shown that it could be used as a viable alternative for perception in several robots and devices such as domestic robots [5], security robots [6], and even humanoid robots [7]. Any robotics vision algorithm is faced with several bottlenecks such as real-time operation with limited onboard computational resources, and the ability to deal with constrained observational conditions derived from the robot geometry, limited camera resolution, sensor/object relative pose, combine language and visual cues, etc in addition to traditional vision problems [8]. In this paper, we develop a simple vision-based environment perception pipeline for robots specifically integrated with a lower-limb exoskeleton to achieve obstacle crossing. We have used PhoeniX exoskeleton [9] for our experiments.

The proposed vision pipeline consists of three main blocks, namely, Ground and obstacle detection, estimation of obstacle dimensions, and obstacle tracking. We use the RGB, depth, and IMU stream data of Intel Realsense L515 in the proposed pipeline and integrate it into the exoskeleton. We rely on planar segmentation methods like Polylidar 3D [10] to provide an initial set of planes and obstacles in them. Using the camera's IMU position vector, we determine the ground plane. The accuracy of obstacle detection is dependent on the quality of the depth map provided by the cameras. Thus detected obstacles are either actual obstacles in the exoskeleton path or noisy artifacts due to erroneous depth perception. It is essential to differentiate between obstacles and noisy artifacts for reliable environment perception and successful integration with exoskeletons. For this purpose, we propose a similarity metric based on color, gradient orientation, and 2D surface normal orientation. Based on the metric, we can reliably differentiate obstacles and noises and accordingly control the exoskeleton.

Our visual perception pipeline estimates the distance to obstacle and obstacle dimensions from the depth data. The model is capable of handling multiple obstacles and can determine if they need to be considered as one if they are too close to each other. Based on the obstacle dimensions, our model can also determine if it is crossable or not. In

This research is supported by the grant no. W1925d0046 from the National Robotics Programme (NRP), Singapore.

* Corresponding Author and email: Manoj Ramanathan, mramanthan@ntu.edu.sg, Manoj Ramanathan and Luo Lincong are equal contributing authors.

[1] Authors affiliated to Rehabilitation Research Institute of Singapore, Nanyang Technological University (NTU), Singapore

[2] Author affiliated to Institute for Infocomm Research, A*STAR, Singapore.

[3] Author affiliated to School of Mechanical and Aerospace Engineering, NTU, Singapore.

these cases, the exoskeleton can intuitively turn in a different direction. It is essential to track the obstacle until the exoskeleton can successfully cross it. We consider two modes of tracking, visual tracking, and positional tracking. In visual tracking, the L515 camera is facing completely downwards, therefore it can track the obstacle until the exoskeleton is ready to cross it. In positional tracking, we combine L515 with a visual SLAM camera, Realsense T265. L515 camera is looking ahead at more of the operating environment. We implement an obstacle tracking strategy, which uses T265's global position (translation and rotation vectors). This allows us to track the obstacles even they are not in the field of view. We integrate the vision-based perception pipeline with our exoskeleton, the proposed control scheme allows us to adapt our obstacle approach by varying step length. The adaptive variation of step length allows to successfully implement an obstacle crossing strategy for exoskeleton users. We conduct experiments both online (with an exoskeleton) and offline (without exoskeleton) to show accurate detection of obstacles (distance from exoskeletons and dimensions). Also, we conduct successful online experiments to validate our obstacle approach and crossing strategy with users. Also, we compare both tracking strategies to identify and provide insights and limitations of each of them.

The rest of the paper is organized as follows: Section II provides related work in the field of vision-based environment perception and terrain understanding in an exoskeleton. Section III details our vision-based environment perception pipeline to detect obstacles of interest, estimate the location and dimensions, and obstacle tracking. Section IV talks about the integration to exoskeleton, our novel obstacle approach, and crossing strategy by varying step length. Section V provides the results, insights, and discussion on the online and offline experiments of the visual perception model. Section VI provides conclusions to the paper.

II. Related Work

Significant progress has been made on vision-based obstacle detection leading to several applications such as autonomous vehicle navigation [11], assisted navigation of visually impaired people [12], etc. These applications aim to assist the robot or/and its users to complete navigation tasks easily. Lower-limb exoskeletons like LOKOMAT [1], Ekso [2], Indego [3] have been developed for providing rehabilitative services for people who suffer from mobility disorders due to injuries or diseases. These devices have twin objectives to provide mobility and stability for users and maintain the autonomy of users.

Vision-based algorithms have been considered for exoskeleton application mainly for environment perception [13], [14], [15], terrain understanding [16], and intent mode recognition [17], [18]. Real-time switching of locomotion modes based on the environment perceived is the main objective of [13], [14], [15]. A chest-mounted RGB camera was used in [13], [14], [15] to collect and train a CNN model for user environment recognition. The initial model [13] trained on 3 classes (ground, stairs ascent, and stair descent) recorded a performance of 94.85%. But the extended model [14] with 12 classes achieved only 73.1%. To ensure the safety and balance of patients, it is necessary that the results of these approaches have to be near-perfect [14] and thus these methods have not been integrated into exoskeletons. [16] also focused specifically on stairs and determining the step length and step height of stairs. But [16] used Microsoft Kinect, which requires an external power supply and is bulky to be fit into an exoskeleton. The external power supply requirement would limit the mobility of the exoskeleton. Methods like [17], [18] used depth sensing as means of intent recognition and achieved an accuracy of 94.1%. But the camera was located in the lower leg portion for depth sensing and classification, which might not be suitable when applying to the exoskeleton as it will move a lot. All these vision-based methods have not been tested or implemented on an actual exoskeleton. Instead, simple online walking tests where the subjects would walk normally with a mounted camera were conducted. None of these methods have focused on obstacle detection and crossing tasks for a lower-limb exoskeleton. In contrast, we focus on developing a vision-based perception model to detect obstacles and successfully integrate it into an exoskeleton for obstacle crossing.

Recently, [19] developed and validated VALOR, a visually assisted exoskeleton for obstacle crossing using an RGB-D camera (Realsense D415). While estimating obstacle location and dimensions, they assumed that depth measurement is accurate without considering the effect of measurement error. But such assumptions are not valid in daily environments. Factors such as surface variations, lighting, etc, could result in noisy depth data. Also, they use few and well-separated obstacles during validation. In contrast, we deal with noisy depth data and propose a similarity metric to differentiate actual obstacles from noise and handle multiple obstacles if they are close to each other.

III. Visual Environment Perception Pipeline

This section provides the details on our proposed visual environmental perception pipeline, which consists of three stages, namely, ground and obstacle detection, estimating obstacle location and dimension, and obstacle tracking. The obstacle information from the pipeline is utilized by the obstacle approach and crossing strategy of the exoskeleton. The accuracy of the model would have to be near-perfect for successful, reliable integration with the exoskeleton so that the safety and balance of users can be guaranteed. For vision-based perception, we use an Intel Realsense L515 camera that provides RGB, depth, confidence maps and IMU streams. In the following subsections, we provide the details on each stage of our vision-based environment perception pipeline.

A. Ground and obstacle detection

The pipeline's first stage focuses on detecting ground and obstacles present in it. Planar segmentation methods like Polylidar 3D [10] provide the input for our proposed perception pipeline. Polylidar 3D [10] extracts non-convex

polygons representing flat surfaces in an environment, and interior holes in these surfaces represent obstacles from 3D point cloud data extracted from RGB, depth images. [10] characterizes detected planar regions with corresponding boundary contours and surface normal, whereas each obstacle is characterized only by its boundary contour. Using the boundary contours, binary masks can be obtained for each plane and obstacle detected. From the detected planar regions, it is necessary to determine the ground plane. For this purpose, the L515 camera's positional vector (accelerometer value) is utilized as it will be aligned to the gravitational pull from the ground. From the detected planes, we determine the plane with the minimum angle between its surface normal and the camera's positional vector. If the minimum angle estimated is less than 30 degrees, the plane is considered as ground. Otherwise, no ground is detected, and the mount on the exoskeleton has to be adjusted such that the ground is visible to the camera.

The next step is the detection of actual obstacles that are in the exoskeleton's pathway. All interior hole regions identified by [10] are not actual obstacles. Factors such as surface conditions, lighting, camera motion, etc affect the depth map perception and can result in artifacts. Therefore, it is essential to identify actual obstacles from these noisy artifacts since the exoskeleton's gait pattern needs to be adjusted for successful obstacle approach and crossing. To differentiate between actual obstacles and artifacts, we compare each of the detected holes with the detected ground plane surface using their corresponding binary masks. The noisy artifacts would be very similar to the ground and we aim to remove these using our proposed similarity metric. The detected holes are compared based on both color or appearance similarity (from RGB image) and depth or shape similarity (from depth images). For appearance similarity, we normalize and L_0 gradient minimize [20] the RGB image to account for lighting changes, etc. The color histograms are computed for the detected hole and ground plane regions. For shape similarity, we consider two histograms namely, gradient orientation and 2D surface normal. From the depth map, we compute the gradient image in x and y directions. The gradient orientation and 2D surface normal for each pixel in the detected holes and ground regions are computed. Histograms of gradient orientation and surface normal are computed for the regions. Finally, each of the detected holes is compared to the ground region based on the three histograms using the euclidean distance metric. Appropriate and reliable distance thresholds have been estimated for each histogram based on our experiments. The proposed similarity scheme considers shape and appearance similarities separately. That is, a hole is considered as an actual obstacle if it satisfies either shape or appearance similarity criteria. Using this proposed similarity scheme, we can differentiate between actual obstacles and noisy artifacts. Also, we use the binary masks of the actual obstacles to filter out obstacles that are straight ahead in the exoskeleton's heading direction. The exoskeleton's gait pattern and control strategy have to be modified only for these obstacles.

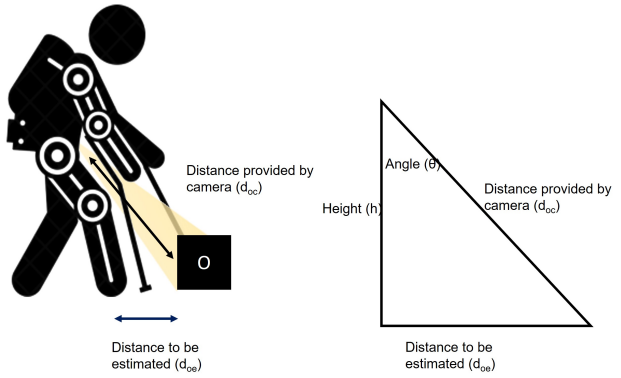

Fig. 1. Distance and height of obstacle estimated when camera is integrated with exoskeleton

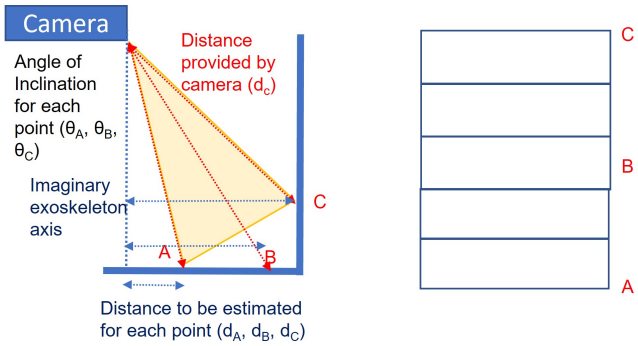

Fig. 2. Variation in angle of inclination for different points in field of view. Division of depth image into horizontal bands

B. Estimate Obstacle distance and dimensions

This subsection deals with how we use the depth data to determine the distance between exoskeleton and obstacle and the dimensions of the obstacles. From the obstacle's binary mask, the corresponding depth map region can be extracted to determine the distance between obstacle and the camera d_{oc}, shown in Figure 1. But we are interested in d_{oe} and h, the distance between obstacle and exoskeleton and the difference in height of obstacle and camera. From the Figure 1, it is clear that we can determine both these variables from angle of inclination (θ) and d_{oc}.

There is another challenge to determining d_{oe} and h for all the points in the depth image. This occurs due to the angle of inclination of the camera. The camera's field of view can be considered as a cone. When the camera is facing straight ahead, it automatically adjusts for the angle at which each point in the field of view is present and provides the depth for each point. In this case, we can say that both d_{oc} and d_{oe} are equal. But when the camera is inclined, the camera will still adjust for the angle at which each point in the field of view is present. But in this case, d_{oc} and d_{oe} are not equal. This can be clearly illustrated with Figure 2, where points A, B, and C are in the same field of view but are at different inclination angles. Our experiments have shown that the angle variation can be as large as 45 degrees between the top and bottom of the images. Thus, it is necessary to determine each of

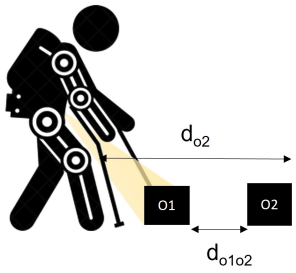

Fig. 3. When multiple obstacles are present in the camera based on the distance between two obstacles d_{o1o2} and distance of the farthest obstacle d_{o2} our model predicts if they need to combined or not. Also we determine if we can cross the obstacle.

these angles to determine d_{oe} and h for all the points. But it is a significant challenge to determine the angle θ for all points in the depth image. Therefore, we employ a simple strategy based on the camera's IMU to determine the angle of inclination for all points in the depth image.

In our installation, the RGB-D camera is attached to the exoskeleton's hip on a mounted platform. The camera is pointed downwards at the floor. But due to the movement of the exoskeleton, there will be a change in the angle of inclination of the camera. We use the camera's IMU to determine the angle of inclination throughout all the points in the depth image. We consider the IMU angle of inclination as the angle made by the center of the field of view cone's axis and divide the depth image into number of horizontal bands (N_{hb}). The points on the central horizontal band are considered to have an angle of inclination θ equal to the angle provided by the camera IMU. For bands above and below the central horizontal band, θ values are increased and decreased in fixed interval sizes (θ_i). Both N_{hb}, θ_i can be varied according to our requirements. From our experiments, we have seen that N_{hb}**,** θ_i works well to determine the obstacle distance from exoskeleton d_{oe} and height difference of obstacle h. Once this compensation is completed for the entire depth image, d_{oe} and h for the complete obstacle boundary can also be used to determine the obstacle dimensions. The minimum and maximum d_{oe} and h along the obstacle boundary are used by the exoskeleton's control module to determine the required step length for crossing the obstacle.

Most researchers [19] assume that there are few and well-separated obstacles present in the exoskeleton's path. Such assumptions do not hold valid in normal day-to-day operating environments. Thus it is necessary to handle these cases if we would like to extend exoskeleton usage outside constrained lab environments. When more than one obstacle is detected in the field of view as shown in Figure 3, we use the obstacle dimensions to determine if they are to be treated as one big obstacle or not. The distance between the two obstacles d_{o1o2} and the distance to cross the farthest obstacle d_{o2} are used to determine if both obstacles $o1$ and $o2$ need to be combined. To combine two obstacles $o1$ and $o2$, they should satisfy the conditions given in equation 1 and 2.

$$d_{o1o2} < 1.5 \times FL \tag{1}$$

$$d_{o2} < SL \tag{2}$$

where d_{o1o2} is the distance between the two obstacles, d_{o2} is the distance of the exoskeleton from the farthest point among both obstacles, FL is the foot length of the exoskeleton user, SL is the step length taken by the user (heel to foot distance). These constraints will ensure the safety and balance of exoskeleton users when crossing obstacles. When two obstacles are combined, the dimension of the combined obstacle is provided to the exoskeleton control module. It is also necessary to determine if a detected obstacle can be crossed or not. If the obstacle is too big, then the exoskeleton should not cross it and instead turn in a different direction. The condition in equation 2 intuitively checks if the obstacle is crossable or not. But this is true only when the obstacle is big enough to be detected within the camera's field of view. Thus, when the obstacle is too big, the ground surface detected is too small and the obstacle is not detected. In this case, similar to determining obstacles of interest in the exoskeleton's direction, we can use the detected ground mask to determine if the exoskeleton can move forward. The available ground plane area for walking forward can be estimated from the ground mask and based on which we can determine if the exoskeleton needs to turn.

C. Obstacle Tracking

In this subsection, we mainly discuss how the obstacles are tracked until they are crossed in both camera configurations. In the first configuration, the L515 camera is facing downwards and can detect obstacles until the exoskeleton foot is very close to the obstacle. Due to this, there is no explicit tracking required as visual confirmation is available until we cross the obstacles. In the second configuration, we use a mount as shown in Figure 4, to house both T265 and L515 cameras such that the L515 camera still can see the ground but T265 is looking ahead. Due to the angle of inclination, the L515 camera can detect obstacles much earlier but visual confirmation can be achieved while crossing the obstacles. As discussed earlier, the camera inclination can result in *'Disappearing Obstacles'* problem which is why visual confirmation cannot be obtained for crossing. The visual SLAM camera (Realsense T265) provides its global and local position. When tracking is enabled it provides translation and rotation of the camera in x,y, and z directions. These translation vectors can be used to determine the 3D movement of the camera and exoskeleton. The 3D motion is utilized for tracking the obstacle location even when it is not in the field of view. The exoskeleton control module informs the perception model before every new step is taken and the position of T265 camera at that instant is stored as the current location. Once the exoskeleton moves, the position of T265 camera begins to change and can be compared with the stored previous location to determine the total motion of the camera. Once an obstacle is detected

in the field of view, we track the obstacle using the 3D motion provided by the T265 camera. Based on the tracking, we can determine if the obstacle is crossed or not even when it is not in the field of view. The positional tracking can be used to determine how close is the obstacle to the exoskeleton and when to execute obstacle crossing. When the exoskeleton changes direction and turns to either side, the tracked obstacles are removed from memory as they are not in the direction of exoskeleton movement. The performance of this configuration is dependent on how well T265 can track in the surrounding environment.

IV. Integration with Exoskeleton - Obstacle Approach and Crossing Strategy

In this section, we discuss the integration of the vision-based perception pipeline to a PhoeniX exoskeleton [9] and how the information perceived by the camera is used to update and modify the exoskeleton's approach towards the obstacle and eventually cross it. Our obstacle approach strategy is reliant on our control module's ability to achieve scalable step length. To test the perception pipeline with the exoskeleton, the camera is mounted on the hip joint with the help of a mount. The mount allows us to test both visual and positional tracking camera configuration. The exoskeleton SDK provides the interface to obtain the status of the exoskeleton and send torque commands to joint actuators. In this study, a trajectory tracking module combining a PID controller and gravity compensation is employed to calculate the required joint torque. All the modules including the visual perception pipeline and the control module are run on NVIDIA Jetson Nano (GPU: 128-core Maxwell, CPU: Quad-core ARM A57 @ 1.43 GHz, Memory: 4 GB 64-bit LPDDR4 25.6 GB/s) board with ROS software, which is powered by a voltage regulator and battery supply. All these are housed within the exoskeleton so that it can move around freely without being attached to any wall sockets. The control board is connected via a WiFi adapter to a network that can be used for remote control if necessary. The exoskeleton control model uses the information from the visual perception model before every new step is taken. Based on the information from the perception model, the control module determines the best-suited step to be taken. In the following subsection, we discuss the control module's ability to vary step length, followed by how it is used in our obstacle approach and crossing strategy.

A. Adaptive variation of Step Length

The exoskeleton's normal gait pattern would allow it to cover a fixed step length. For an exoskeleton to adapt to its operating environment, it is necessary for it to vary its step length. From our experience, we have observed that the knee joint almost remains straight during touchdown. Thus, we changed the step length by scaling the terminal hip joint angle in the double stance phase. According to the kinematics

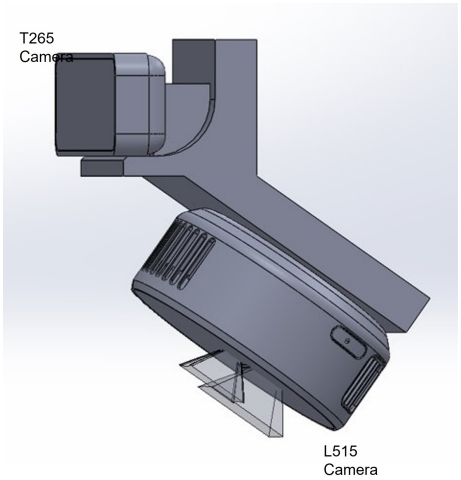

Fig. 4. Mount used for housing RealSense T265 and RealSense L515 camera. This mount is attached to the exoskeleton via a shaft and angle of inclination can be changed.

of the exoskeleton, the step length can be expressed as:

$$\begin{aligned} SL =& L_t \cdot (\sin(\mu \cdot \theta^h_{sw}) + \sin(\mu \cdot \theta^h_{st})) + \\ & L_s \cdot (\sin(\mu \cdot \theta^h_{sw} + \theta^k_{sw}) + \sin(\mu \cdot \theta^h_{st} + \theta^k_{st})) \end{aligned} \quad (3)$$

where L_t, L_s are the length of thigh and shank links, θ^h_{sw}, θ^k_{sw}, θ^h_{st}, θ^k_{st} are the hip and knee joint angle of swing and stance legs at the terminal of swing phase. μ is the constant used to scale the terminal hip joint angle. By varying μ, we can modify the exoskeleton's approach towards the obstacle.

B. Obstacle Approach and Crossing Process

The proposed vision-based perception pipeline provides information on the obstacle in the exoskeleton's direction using a camera. The pipeline provides the distance of the obstacle from the exoskeleton and its dimension. From the obstacle dimension, we can estimate how far should the exoskeleton's crossing step cover for successfully crossing the obstacle. Before every new step is taken by the exoskeleton, the current obstacle information from the perception module is utilized by the control module. The distance measures provided are utilized for the obstacle approach and crossing process.

As mentioned above, we focus on stroke patients who have residual strength on one side. Due to this only one of their leg is affected. For maintaining balance and ensuring the safety of users, we cross obstacles with the affected leg first. Our obstacle approach and crossing follow this protocol. Fig.5 shows the three stages of our exoskeleton's obstacle approach and crossing, namely, normal walking with variable step length, the first crossing step with the affected leg, and the second crossing step with the abled leg. Once an obstacle is detected, the exoskeleton control will switch to the obstacle approach phase. In this phase, we follow a normal gait pattern but vary the step length according to the obstacle distance. By varying the step length, the exoskeleton gradually gets closer to the obstacle. Before each step, we use the exoskeleton's kinematic information

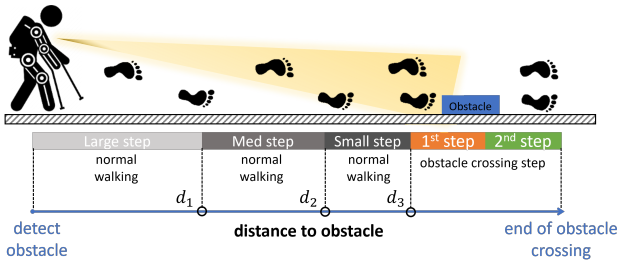

Fig. 5. Obstacle crossing diagram. The normal walking with different step length is performed when the obstacle distance is larger than d_3. When the distance is less than d_3, the first crossing step is triggered, and the second crossing step follows.

to compute the distance between the toe and obstacle. When this estimated distance is less than d_3, the exoskeleton controller is triggered to assist the affected leg to step over the obstacle. After this stage is completed, the user can control their abled leg to step over the obstacle in the following step. During the obstacle approach and crossing phase, the exoskeleton does not completely take over control but allows for user interference. In this study, since the users' abled leg is allowed to move freely in its swing phase, they also have a certain freedom to control the distance between the toe and the obstacle. The users' participation can make full use of the advantage of an individual's perception and decision-making ability to improve the success of obstacle crossing.

V. Experimental Results and Discussion

In this section, we detail the experiments conducted, results obtained, and provide insights based on them. To validate the proposed vision-based perception pipeline, we conduct both offline and online tests. For offline experiments, we mounted the camera on a stand and moved it to simulate slight camera motion. For validating the obstacle tracking stage, the camera was strapped onto the user with a belt and was asked to move around their environment where the pipeline can perceive obstacles. As the first step, the pipeline detects the ground plane (See Figure 6 for results) using the camera's positional vector. Our experiments show that ground detection shows very good accuracy in the proposed pipeline. During these experiments, we identified the thresholds for the similarity metric proposed for obstacle detection. For reliably differentiating noise and obstacles, each of the holes is compared with the ground based on three similarity metrics, color, gradient orientation, and 2D surface normal. For orientation and surface normal histogram, we consider angle range of $-90 \leq \theta \leq 90$ and $-180 \leq \theta \leq 180$ respectively. Also, we divide them into 18 and 36 bins respectively. The similarity distance metric threshold was identified to be 0.35, 0.25, and 0.2 for color, surface normal and orientation respectively. Figure 7 shows results of obstacle detection using the proposed similarity metric. The proposed scheme considers shape and appearance similarity separately and can differentiate obstacles from noise. Using this method, we were able to differentiate obstacles even in cases when they do not satisfy all similarity thresholds. For instance, when the obstacle is the same color as the ground or when the wooden block obstacle is viewed from the top such that only one flat surface is visible and it's surface normal or orientation is very similar to the ground. Also, the thresholds identified are based on the set of obstacles we experimented with, lighting conditions, etc. In all our results, the ground plane is shown with green contours, and detected holes are shown in orange contours.

Using the identified obstacles regions in the depth map, we determine the obstacle location and its dimensions, while accounting for the camera's inclination and multiple obstacles. To test the accuracy of the obstacle location, the distance from the camera is measured using tape and marked on the floor in centimeters. Using the setup, we conduct a reliability test by placing the obstacle at different distances and estimating the obstacle exoskeleton distance d_{oe} from section III. The obstacles were placed at 3 distances (30, 35, 40 cm) and trials were repeated 10 times to determine the accuracy of the detection. The proposed pipeline was able to detect obstacles each time and the maximum deviation from the actual distance was 3.2cm. In our exoskeleton control strategy, we provide a safety buffer distance to account for this deviation in distance estimation. Also, we show in the supplementary video, our results when the camera inclination angle is not compensated and how multiple obstacles are treated as a single obstacle when they are close together. We can see that when camera inclination angle is not accounted for, the error in d_{oe} estimation is too high for successful integration to the exoskeleton. The obstacle distance mentioned in the video is in millimeters.The proposed solution has been tested with only a few and select obstacles that are relatively small in shape, static in nature and can be seen in office/lab environment like bottle, pouch, and wooden blocks.

For validating the obstacle tracking stage, experiments were conducted in 2 levels. Firstly, the camera was strapped on to user's stomach and asked to walk around without an exoskeleton. In this configuration, both visual and positional tracking worked accurately to track obstacles and estimate their distance from the user. The results of the tracking can be seen in the supplementary video. We can see that positional tracking based on visual odometry can detect obstacles even it is not visible to the camera. The detected obstacle position is provided at the top of the video. Secondly, users wore the exoskeleton on which the camera was mounted. We observed that visual tracking worked better than positional tracking after integration to the exoskeleton. The positional tracking is dependent on the T265 visual SLAM camera's ability to track in the global environment location. During the test with the exoskeleton, it was observed that visual odometry with T265 camera failed and recorded very less movement compared to the actual movement of the camera. Some of the reasons that contributed to odometry failure include non-availability of sparse key features to track, lighting conditions, vibrations of the camera, etc. Also, such positional tracking works better when wheel odometry is available which is not possible with an exoskeleton. Since the position of obstacles has to be tracked accurately, we did not consider this configuration

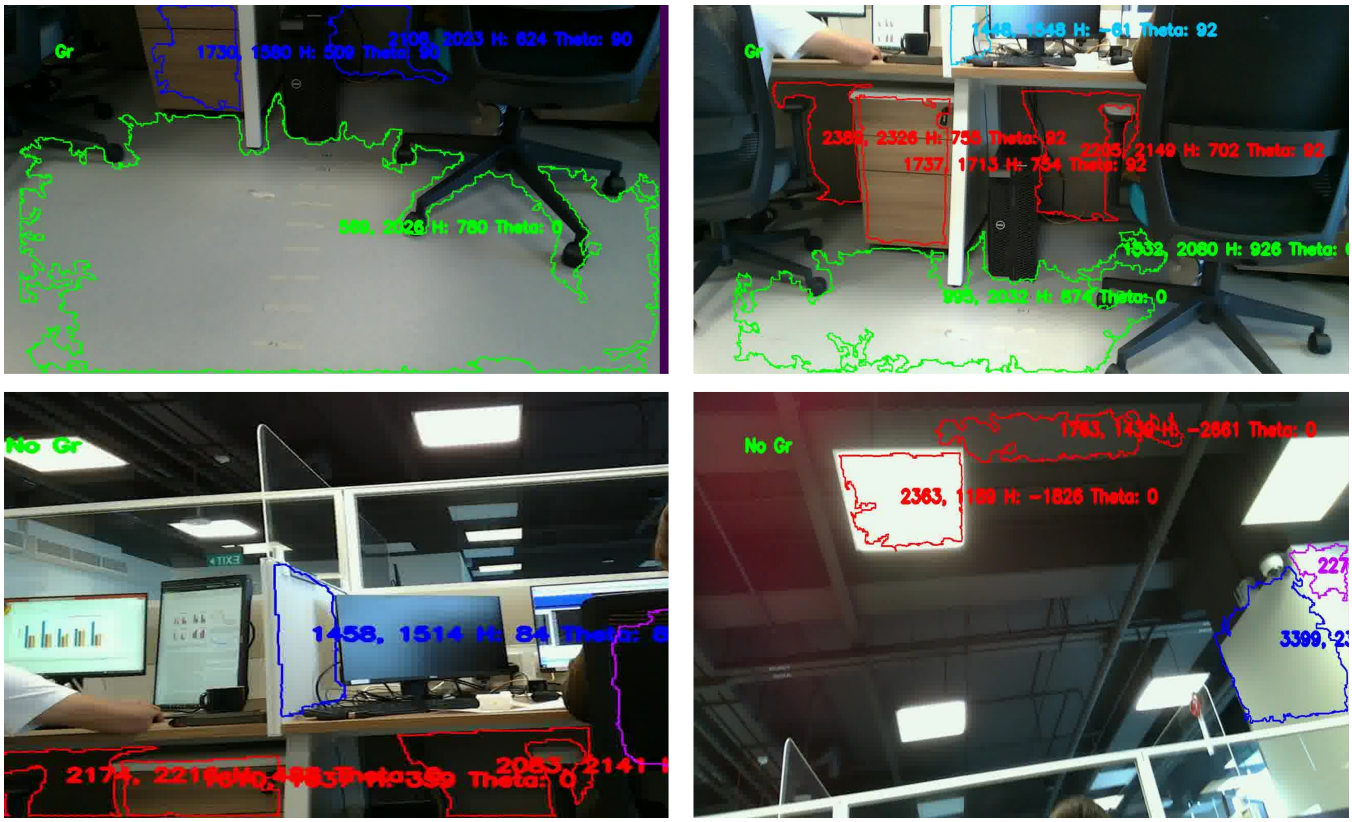

Fig. 6. Results of ground detection. Top row shows ground plane detected in green color and 'Gr' is written on top left of the frame. Bottom row shows when ground plane is not detected, 'No Gr' is written on top left of frame.

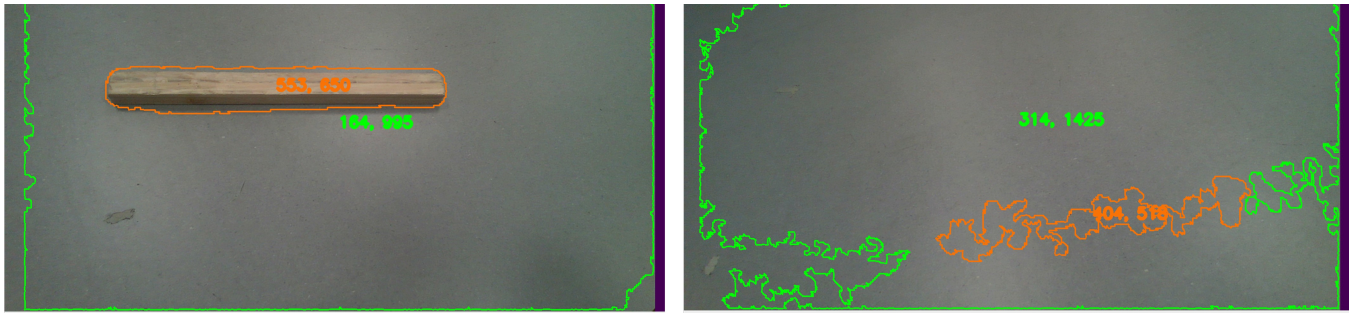

Fig. 7. Holes detected using the proposed perception pipeline. On left, wooden block is detected as hole. It is detected as obstacle based on the similarity value obtained $color = 0.834, orientation = 0.221, surfacenormal = 0.229$. On right, noisy artefact is detected as hole. It is not detected as obstacle based on the similarity value obtained $color = 0.036, orientation = 0.108, surfacenormal = 0.153$. The numbers in orange and green represent the minimum and maximum distance estimated by the camera for the regions marked with those colors

for the proposed exoskeleton usage to ensure the safety and balance of users.

For final integration to the exoskeleton, we considered only the proposed pipeline with visual tracking (only L515 camera). The exoskeleton was worn by healthy subjects who have previous experience of wearing it. During the experiments, either right or left leg was considered the affected leg which required actuation by the exoskeleton, while the other leg could be controlled by the user (to simulate stroke patients). The proposed perception pipeline was integrated into the exoskeleton control strategy and the obstacle approach and crossing strategy were tested and validated. Our experiments showed that the proposed perception pipeline was able to operate in real-time and provide information on obstacles to the exoskeleton before each new step is triggered. The proposed similarity metric was able to differentiate obstacles and noise. Figure 8 shows obstacle crossing initiated by the perception pipeline. The supplementary video shows the complete obstacle approach and crossing scheme. In this configuration, the pipeline can detect wooden blocks as obstacles that are 47cm away from the exoskeleton as that is when they enter the field of view of the camera. Also, the distance is dependent on the type and dimension of the obstacle used. One major drawback

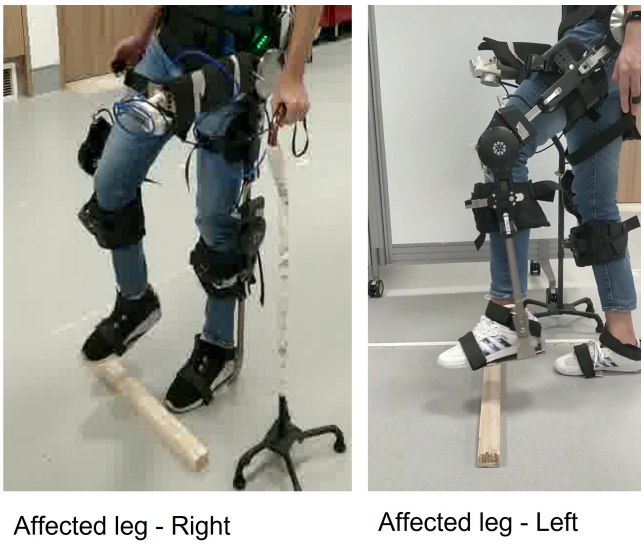

Fig. 8. Perception pipeline initiates obstacle crossing. First, the affected leg always crosses the obstacle followed by the abled leg.

was the non-detection of obstacles when the foot was very close to the obstacle. This is because the obstacle and foot are considered as one and [10] detects the ground plane excluding them. From our experiments, we noticed this error happens if the distance between the foot and obstacles is less than 3cm (See in the supplementary video). In our experiments, we used the abled leg to appropriately control the obstacle approach to avoid this error. As a part of our future work, we intend to improve our obstacle detection method so that such cases can also be handled.

VI. Conclusion

In this paper, we propose a vision-based environment perception pipeline that can run in realtime and provide information on obstacles ahead of a lower-limb exoskeleton. The proposed pipeline detects the ground plane and obstacles in it. The pipeline employs a similarity metric based on color, gradient orientation, and 2D surface normal to differentiate noisy artifacts and actual obstacles. The pipeline investigates two obstacle tracking modes, visual and positional tracking. The obstacle information is passed on to an exoskeleton, that employs a control strategy to adaptively vary step length to modify its approach and successfully cross obstacles. The experiments conducted show that the proposed pipeline can accurately detect obstacles and can be successfully integrated into an exoskeleton for obstacle crossing task.

Acknowledgements

This research is supported by the grant no. W1925d0046 from the National Robotics Programme (NRP), Singapore.

References

[1] A. Domingo and T. Lam, "Reliability and validity of using the lokomat to assess lower limb joint position sense in people with incomplete spinal cord injury," *Journal of neuroengineering and rehabilitation*, vol. 11, p. 167, 12 2014.

[2] A. Kozlowski, T. Bryce, and M. Dijkers, "Time and effort required by persons with spinal cord injury to learn to use a powered exoskeleton for assisted walking," *Topics in Spinal Cord Injury Rehabilitation*, vol. 21, pp. 110–121, 03 2015.

[3] C. Hartigan, C. Kandilakis, S. Dalley, M. Clausen, E. Wilson, S. Morrison, S. Etheridge, and R. Farris, "Mobility outcomes following five training sessions with a powered exoskeleton," *Topics in Spinal Cord Injury Rehabilitation*, vol. 21, pp. 93–99, 03 2015.

[4] M. Kelly-Hayes, "Influence of age and health behaviors on stroke risk: Lessons from longitudinal studies," *Journal of the American Geriatrics Society*, vol. 58, no. s2, pp. S325–S328, 2010. [Online]. Available: https://agsjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1532-5415.2010.02915.x

[5] S. Tibken. (2021) Samsung's ces 2021 robots will clean your house and pour you a glass of wine.

[6] design3.de. (2018) Carl - who is afraid of robots?

[7] M. Ramanathan, N. Mishra, and N. M. Thalmann, "Nadine Humanoid Social Robotics Platform," in *Computer Graphics International (CGI)*, ser. Advances in Computer Graphics, Part of LNCS book series, M. Gavrilova, J. Zhang, N. M. Thalmann, E. Hitzer, and H. Ishikawa, Eds., vol. 11542. Springer, Cham, June 2019, pp. 490 – 496.

[8] J. Ruiz-del-Solar, P. Loncomilla, and N. Soto, "A survey on deep learning methods for robot vision," *CoRR*, vol. abs/1803.10862, 2018. [Online]. Available: http://arxiv.org/abs/1803.10862

[9] PhoeniX SuitX. (2021) Phoenix medical exoskeleton, https://www.suitx.com/phoenix-medical-exoskeleton.

[10] J. Castagno and E. Atkins, "Polylidar3D-Fast Polygon Extraction from 3D Data," *Sensors*, vol. 20, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4819

[11] J. Park, J.-H. Lee, and S. H. Son, "A survey of obstacle detection using vision sensor for autonomous vehicles," in *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2016, pp. 264–264.

[12] P. Costa, H. Fernandes, P. Martins, J. Barroso, and L. Hadjileontiadis, "Obstacle detection using stereo imaging to assist the navigation of visually impaired people," *Procedia Computer Science*, vol. 14, p. 83–93, 12 2012.

[13] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Preliminary design of an environment recognition system for controlling robotic lower-limb prostheses and exoskeletons," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, 2019, pp. 868–873.

[14] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Computer vision and deep learning for environment-adaptive control of robotic lower-limb exoskeletons," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2021, pp. 4631–4635.

[15] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Environment classification for robotic leg prostheses and exoskeletons using deep convolutional neural networks," *Frontiers in Neurorobotics*, vol. 15, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2021.730965

[16] N. E. Krausz, T. Lenzi, and L. J. Hargrove, "Depth sensing for improved control of lower limb prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2576–2587, 2015.

[17] H. A. Varol and Y. Massalin, "A feasibility study of depth image based intent recognition for lower limb prostheses," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 5055–5058.

[18] Y. Massalin, M. Abdrakhmanova, and H. A. Varol, "User-independent intent recognition for lower limb prostheses using depth sensing," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 8, pp. 1759–1770, 2018.

[19] D.-X. Liu, J. Xu, C. Chen, X. Long, D. Tao, and X. Wu, "Vision-assisted autonomous lower-limb exoskeleton robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3759–3770, 2021.

[20] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, p. 1–12, dec 2011. [Online]. Available: https://doi.org/10.1145/2070781.2024208