Automated ground plane detection using human motion and environmental geometry

S. Czarnuch
Memorial University
Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science and
Discipline of Emergency Medicine, Faculty of Medicine
St. John's, NL

***Abstract*-Tracking humans using computer vision has applications in healthcare delivery, security, computer interfaces and gaming. This is particularly true in a healthcare setting where people are constantly changing clothing and environments are dynamic and cluttered, preventing the use of other common methods of tracking (e.g., instrumented rooms, wearable sensors). In these challenging environments, the placement and orientation of cameras is often for practical convenience rather than technical system performance. However, knowledge of the cameras relative to the ground plane is necessary for automated human tracking to succeed. This study outlines a methodology for automatically detecting the ground plane by synthesizing human trajectories with knowledge of potential ground planes and their orthogonal sets. Potential human trajectories will be estimated using Veiwpoint Feature Histograms as a feature and a threshold 3D Euclidean velocity across frames between 1 and 2m/s. Orthogonal sets will be defined using all potential ground planes (e.g., walls, tables, floor). Results are presented for a pilot study with data collected from two different rooms and three different viewing perspectives in each room.**

Index Terms—Ground plane detection, computer vision, human trajectory, uninitialized tracking, perspective independence.

I. Introduction

In research and practice, the automated evaluation of human motion has significance in many areas including healthcare delivery, security, computer interfaces, and video gaming [1]. The emergence of inexpensive depth sensors (e.g., Microsoft Kinect [2]) has created an environment where inexpensive and portable computer vision systems [e.g., 1, 3] can be used to reliably track human motion, invariant to lighting, shape, colour and texture. Existing depth-based tracking-by-detection approaches have achieved some success commercially (e.g., the Microsoft Kinect gaming console) and scientifically [e.g., 1, 4]. Indeed, with an ideal sensor perspective, human motion tracking has been achieved with significant accuracy [1, 5-7]. However, these approaches have seen limited practical application in challenging, real-world indoor environments (e.g., dynamic, frequently changing and/or cluttered). One significant issue with existing tracking approaches in challenging environments is that the sensor cannot be placed in an ideal location (e.g., fronto-parallel perspective). Indeed, in many real-world applications (e.g., in-home activity monitoring and automated healthcare) sensors are generally placed for practical convenience rather than for system performance. In these situations, the sensor's viewing perspective is unknown, preventing the use of existing tracking approaches which require a known perspective (e.g., fronto-parallel). Accordingly, after physical placement, the sensor must be calibrated to identify the ground plane, enabling the 3D video data to be rotated to a known, useable perspective.

The calibration process, or more specifically the identification of the ground plane, can be performed manually or automatically. In the case of RGB-D data, manual calibration generally requires a physical calibration target; a three-dimensional cube of known dimensions placed in a known location [8]. Alternatively, a user can manually select points representing inliers on the ground plane [e.g., 9] using a calibration interface. These manual processes require expert knowledge and are not practical in many cases. Automated methods of sensor orientation and calibration have also been developed. In cases where the sensor may not be oriented with the ground plane a target in the field of view of the system is instrumented with a marker or label [e.g., 10]. In these approaches the target must be present during calibration and the movement of the target must be known a priori. Other automated methods can segment the ground plane (and other surfaces) but require knowledge of the global horizontal and vertical axis [e.g., 11].

In real-world applications where the sensor orientation and position are unknown and objects in the scene cannot be labeled or marked, existing automated approaches to ground plane detection do not work. This work outlines a novel approach for automatically identifying the ground plane using human motion trajectories and other large planes visible in the scene. This approach utilizes the fact that, in general, large surfaces (e.g., walls, ceiling, tabletops) are parallel or orthogonal to the ground plane, and that human trajectories are parallel to the ground plane, essentially using humans as virtual calibration targets. The outcome of this method is either the explicit identification of the ground plane if the plane is visible in the scene or the implicit identification of the ground plane normal (a plane parallel to the ground plane) if the plane is not direction observed. The proposed approach is entirely uninitialized, not requiring any information about the location, perspective and orientation of the sensor or the dimensions and occupants of the room.

II. Methods

A. Experimental Setup

A single Microsoft Kinect RGB-D sensor [12] was used to collect video data. The sensor was placed at an elevation and orientation out of alignment with the ground plane. The video stream was captured from the Kinect sensor using the OpenNI 2.0 application program interface [13]. The intrinsic transformation matrix was recorded to allow registration between the depth and RGB image feeds. Raw RGB-D data were converted to 3D point clouds using the Point Cloud Library [9].

B. Procedure

Participants were instructed to walk in front of the sensor according to a self-directed path and were allowed to move outside the field of view of the sensor. The pace of participant movement was not prescribed, though participants were not allowed to stop moving. For each trial, only a single human was visible at any point in time.

C. Automated ground plane detection

The ground plane was automatically detected from the 3D point clouds in five steps: 1) Potential ground plane detection; 2) Foreground cluster extraction; 3) Feature and correspondence estimation; 4) Foreground cluster reduction; and 5) Ground plane identification.

Potential ground plane detection: A RANdom SAmple Consensus (RANSAC) method was used to fit planes to the 3D point cloud with a minimum sample distance of 10cm. A plane was considered a potential ground plane if the number of plane inliers was greater than ten percent of the total points in the 3D point cloud as in our previous work [8]. The inliers of potential ground planes were removed from the 3D point cloud and the RANSAC method was invoked iteratively until no more potential planes were found. The vector of potential ground planes was updated for each consecutive frame captured.

Foreground cluster extraction: Euclidean clustering was used to identify potential foreground objects in the 3D point cloud with potential ground plane inliers removed. Potential ground planes were removed to improve segmentation performance and because large planes are generally not part of moving objects, similar to [14]. A cluster tolerance of 5cm was used with a minimum cluster size of 7,500 points to remove spurious outliers and outlying clusters.

Feature estimation: The Viewpoint Feature Histogram (VFH) [15] was used as a feature for each foreground cluster. The VFH, based on the Fast Point Feature Histograms descriptors [16], is fast and performs well at classifying human objects [14, 15]. The normals of each cluster were first found using a kd-tree nearest neighbor search of the extracted cluster with a search radius of 5cm. The VFH signatures were then estimated for each cluster using the point cloud data and the normals. The correspondence between clusters in consecutive frames of captured data was estimated using the Fast Library for Approximate Nearest Neighbors (FLANN). Clusters were considered a match across successive frames if the Euclidean squared distance was less than 50cm^2.

Foreground cluster reduction: The centre of mass of each foreground cluster was calculated for each frame of captured data. The 3D vector connecting consecutive centre of mass points was calculated with each new frame of data. A normal human trajectory was defined as having a velocity of 1 to 2 m/s, with a normal turning angle between consecutive frames defined as less than 25 degrees. Foreground clusters were rejected as potential humans if the length of three trajectory vectors (representing the Euclidean distance in three dimensions) was outside 1 to 2 m/s, or if the angle between consecutive vectors exceeded 25 degrees. A foreground object was considered a human if the summed distance of the trajectory vectors was greater than 1.5m and the outer bounds of the trajectory was greater than 1m.

Ground plane identification: A plane was fit to the human trajectory using RANSAC without a minimum sample distance. If a potential ground plane existed that was parallel to the trajectory plane, the ground plane was explicitly defined. Planes were considered parallel if the 3D angle between the plane normals was less than five degrees. If no plane existed that was parallel to the trajectory plane, planes orthogonal to known potential ground planes were defined (to utilize the fact that walls, ceilings, etc. are generally at right angles to each other). If these orthogonal planes were parallel to the trajectory plane, only the ground plane normal was identified.

III. Results

Video data were captured using a single Microsoft Kinect RGB-D sensor [12] connected to a Dell Optiplex 9020 MTI7-4790 3.6GHZ 16GB running 64-bit Ubuntu 14.04. Raw depth and RGB images were captured using the OpenNI 2.0 API [13], registered, and converted to point clouds [9] (see Fig. 1). Two different rooms were instrumented with sensors and three trials were completed in each room. For each trial, the sensor was moved to a predetermined location and elevation, described in Table 1, simulating several real-world applications.

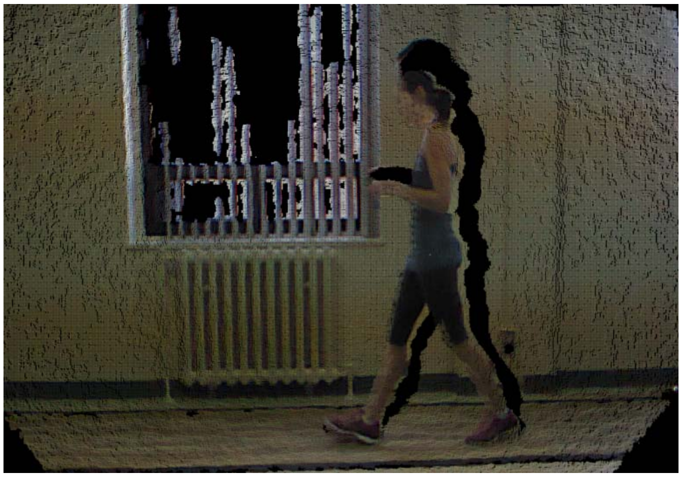

Figure 1: Raw data capture from sensor and converted to a point cloud

TABLE 1
PHYSICAL LOCATION AND PERSPECTIVE OF SENSORS FOR EACH TRIAL RELATIVE TO THE CENTRE OF THE WALKING PATH

	Sensor Elevation (cm)			Sensor Perspective (degrees from fronto-parallel)		
Room	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
1	100	200	25	0	45	-45
2	200	25	100	-45	0	45

Three participants performed the six trials, with each trial taking an average of 12.1 seconds. Image data were saved to disk at 30 frames per second resulting in 2178 frames of raw RGB and depth image data.

A. Automated ground plane detection

All image data were processed from disk after the trials were completed to identify the ground plane. The raw RGB-D data was converted offline to point cloud data. As a reflection of the exploratory nature of this study, algorithms and processing were not optimized for speed or performance.

Potential ground plane detection: A RANSAC method was invoked on the full point clouds which had an average of 223,532 points. The resulting threshold, set at ten percent of total points, for plane detection was an average of 23,532 points. The largest plane in all cases was the back wall of the test rooms, which had an average of 132,662 points. The second largest plane was the floor, with an average of 51,811 points (see Fig. 2). In some cases a third plane was fit to the window and radiator on the back wall with an average of 28,211 points. Table 2 shows the average, minimum and maximum point cloud points for the full cloud and the number of points in the remaining foreground cloud following the removal of each potential ground plane.

TABLE 2
POINT CLOUD DATA POINTS DURING GROUND PLANE DETECTION PROCEDURE

	Average	Min	Max
Total points	223532	220973	227347
Points after first plane inliers removed	90870	81749	101907
Points after second plane inliers removed	39059	28905	48959
Points after third plane inliers removed	17460	13842	21077

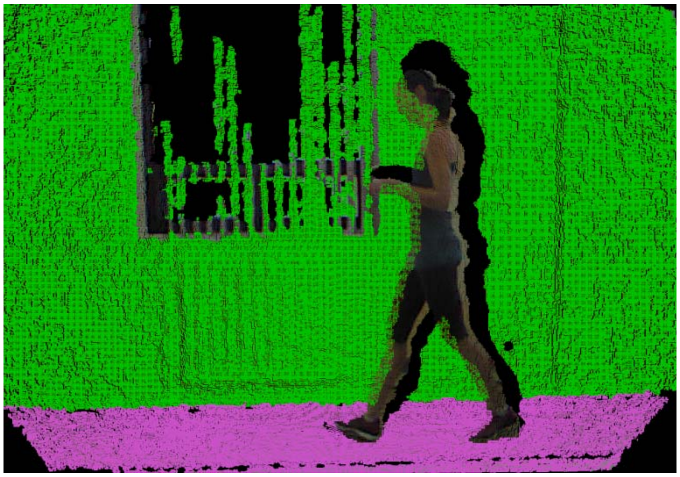

Figure 2: Potential ground planes

Foreground cluster extraction: Following removal of the potential ground plane inliers, Euclidean clustering on the foreground point cloud identified the human in the scene in every frame the human was present, with an average number of 17,996 points. In most cases, the window on the back wall was also identified as a cluster with an average of 11,412 points. The column on the back wall was also identified as a cluster in some cases with an average of 8,257 points. As seen

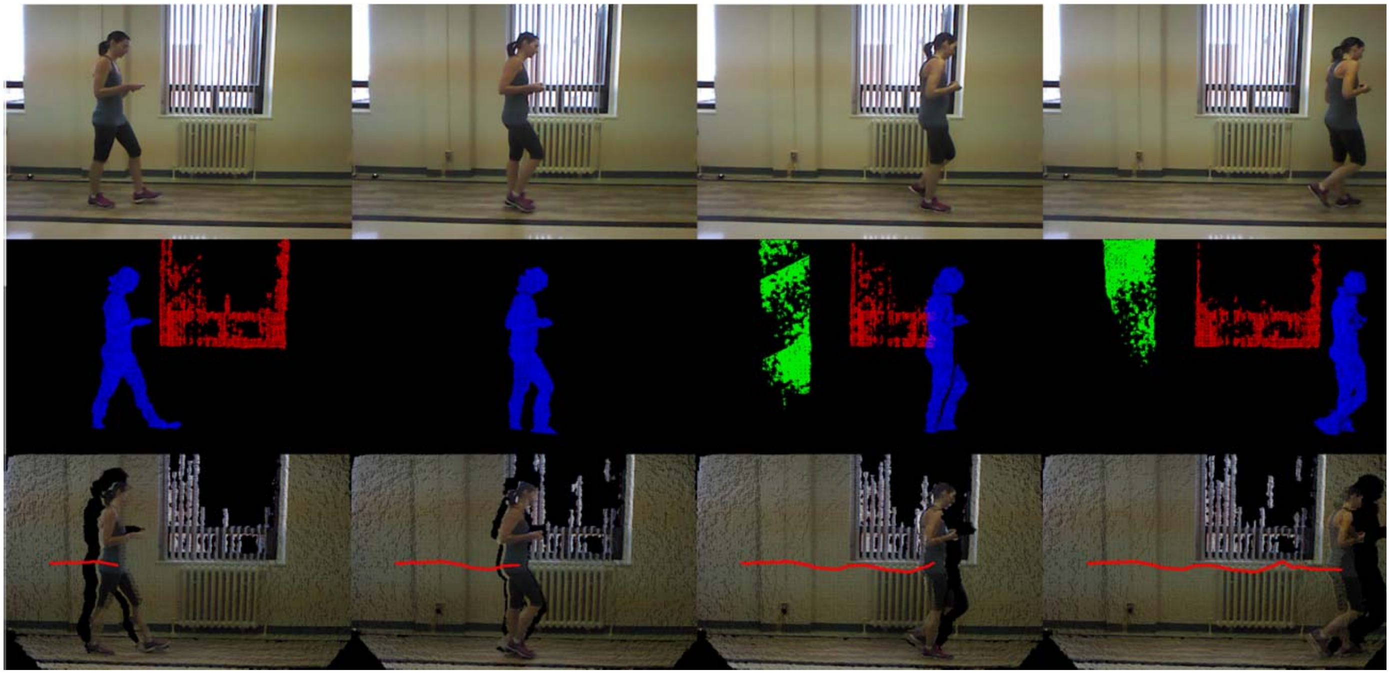

Figure 3: Sample images from a walking trial, increasing in time (left to right). Top row: raw RGB images. Middle row: foreground segmented images using Euclidean clustering. Bottom row: Point cloud visualization of registered RGB-D images with centre of mass trajectory overlay.

TABLE 3
DATA POINTS IN FOREGROUND CLUSTERS

	Average	Min	Max
Points in cluster 1	17996	15978	19806
Points in cluster 2	11412	7587	18037

in Fig. 3, Euclidean clustering is prone to noise, where in some cases only the human in the scene was clustered, while in others the window and column were also identified as clusters. Table 3 shows the average, minimum and maximum number of points in the cluster point clouds.

Feature and correspondence estimation: Viewpoint Feature Histogram signatures were successfully estimated in all cases where foreground clusters were found. Fast Library for Approximate Nearest Neighbor (FLANN) correspondence between matching clusters between frames (Table 4) were less than the threshold 50cm^2 in 82.6% of the total frames, with an average mean squared distance of 42.1cm^2. FLANN correspondence between non-matching clusters was greater than the threshold in all frames, with an average mean squared distance of 15559.5cm^2.

Foreground cluster reduction: The trajectory of all foreground objects, measured as the 3D vector connecting the centre of mass of consecutive corresponding clusters (see Fig. 3), resulted in the removal of non-human clusters in an average of 5.32 iterations. As shown in Table 5, human clusters moved an average of 1.24cm and changed direction with an average angle of 7.7degrees, while non-human clusters moved an average of 4.01cm with average angle of 62.8 degrees. Human clusters were confirmed after exceeding the 1.5m total trajectory distance with an average of 126 frames, which in all cases occurred after the outer bounds of the trajectory exceeded 1m.

Ground plane identification: A plane was fit to the human trajectory in all six trials using RANSAC. In four trials, the angle between the human trajectory plane and the actual ground plane was less than the threshold five degrees. In all four of these trials, the correct plane was selected as the ground plane, since the angle between the other planes was also greater than the threshold angle. In the remaining two trials, the angle between the human trajectory and all the potential ground planes was greater than five degrees. The orthogonal sets of planes, calculated as any plane that was perpendicular to all potential ground planes, were also greater than the five degree threshold. Both of these trials resulted in a failed convergence on a ground plane, both explicitly (from the set of potential ground planes) and implicitly (from the set of orthogonal planes).

IV. Discussion and Conclusions

The automated detection of a ground plane generally requires either knowledge of the horizontal and vertical planes or a calibration target in the scene. However, in many practical cases the location and orientation of the sensor is not known and is not guaranteed to be level with the ground plane. Additionally, the use of a calibration target is also infeasible in many real-world situations. The preliminary work presented from this study proposed using the natural motion of humans, or more specifically human trajectories, as a method of identifying the ground plane. Essentially, humans were used as a virtual calibration target.

The preliminary findings of this study suggest that this method of automated ground plane detection is not overly robust, converging on a solution in four of six trials. However, considering each individual step of the procedure elucidates the cause of the high proportion of failed convergences. The method of potential ground plane detection identified the actual ground plane in all cases, and also successfully identified the rear wall – a plane orthogonal to the actual ground plane. Once these potential ground planes were removed from the full point cloud, Euclidean clustering foreground detection successfully identified the actual human as a potential human in all cases, along with other clusters that were not potential ground planes. The Viewpoint Feature Histogram signatures were successfully used to find an 82.6% frame-to-frame correspondence between human clusters which, in conjunction with the centre of mass trajectories, allowed all foreground clusters except the actual human to be rejected.

The failed ground plane convergence occurred in both of the failed trials at the final process during the ground plane identification. In both failed trials, the actual ground plane was correctly identified. Furthermore, in both failed trials the orthogonal set of potential planes also contained a plane parallel to the ground plane. However, the human trajectory in both failed trials was almost linear, with the variation actually occurring in the vertical plane as an artifact of ambulation (i.e., the centre of mass "bobbing" up and down). When a plane was fit to the human trajectory, the best-fit plane was not parallel to the ground plane. Rather, the trajectory plane was arbitrarily oriented. Accordingly, when compared to the set of potential and orthogonal planes, convergence failed. Indeed, under these conditions convergence could equally likely have occurred with the correct plane or an incorrect plane as opposed to failing.

TABLE 3
Data points in foreground clusters

	Average	Min	Max
Points in cluster 1	17996	15978	19806
Points in cluster 2	11412	7587	18037
Points in cluster 3	8257	7559	9246

TABLE 4
Foreground cluster correspondence measures frame-to-frame

	Mean distance squared		
	Average	Min	Max
Clusters corresponding	42.1	2.37	114.7
Clusters not corresponding	15559.4	5334.4	51482

TABLE 5
Trajectory statistics between consecutive centre of mass estimates for human and non-human clusters

	Distance (cm)			Absolute angle (°)		
	Avg.	Min	Max	Avg.	Min	Max
Human clusters	1.24	0.82	3.01	7.7	0.1	78.2
Non-human clusters	4.01	0.75	18.34	62.8	1.3	167.4

The preliminary nature of this study and inconsistent results suggest several limitations. The generalizability of the successes of the proposed methodology are hampered by the small sample size and test cases. Although the approach is robust programmatically, the sequential nature of the methodological steps may not be as robust under more realistic scenarios. For example, the direct determination of ground planes with the proposed sample distance of 10cm includes, as inliers, a small set of points from adjacent clusters. In this way, the feet of any humans will be included in the plane estimation, and ultimately excluded from future cluster extraction. The implications of these limitations can only be validated with more rigorous testing under additional real-world scenarios.

Additionally, the use of a single feature, the Viewpoint Feature Histograms, for subsequent correspondence estimation may be underestimating the importance of this step. Considering the importance of correctly identifying the correspondence between humans in successive frames of data, future work will look to implement multi-feature methods of human identification and correspondence. A combination of global (e.g., VFH) and local (e.g., FPFH) features may provide more robust and reliable characterization of humans under more varied real-world conditions.

The main failures of this overall methodology occurred as a result of linear trajectories. Perhaps the most significant limitation of this work is in the translation of the human trajectories into a plane that is guaranteed to be parallel to the ground plane. An implicit assumption of this approach was that all human motion would occur on a plane parallel to the ground plane. However, provision was not made for a two-dimensional trajectory – one that would satisfy the criteria of being parallel yet not have a unique planar solution. Considering that such linear motion is not only within the boundaries of the implementation assumptions but also practically valid, future work must focus on developing a method of using this linear trajectory to identify the ground plane. For example, a weighted evaluation of the angle between the human trajectory and potential planes, in conjunction with the Euclidean proximity of the contributing point cloud to the potential planes may disambiguate potential planes and the actual ground plane. This scenario will of course not work in cases where the ground plane cannot be explicitly observed but presents one possible solution.

One other notable limitation of this study is that the implementation of this methodology was not optimized for speed or performance. Execution metrics were not taken, but observation revealed that the approach was not even close to executing in real time. This limitation was the substantive motivation behind buffering the captured images at 30 frames per second for future processing. For this approach to satisfy the main objective of the work, which is to automatically identify the orientation and location of a depth sensor in real-world applications, significant emphasis must be placed on improving the efficiency of code execution.

Notwithstanding the limitations of the current study, preliminary results support the continued efforts toward automated ground plane detection using human motion trajectories. Addressing the limitations of the current study will be the initial focus of future work, with particular focus on evaluating the robustness of the existing approach and optimizing execution. Ultimately, the evaluation of this refined approach in a true real-world context will be necessary to understand the true efficacy of the methodology.

References

[1] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," Communications of the ACM, vol. 56, pp. 116-124, 2013.

[2] Microsoft. (2015, July 27). Kinect Xbox One. Available: http://support.xbox.com/en-US/browse/xbox-one

[3] B. J. Southwell and G. Fang, "Human object recognition using colour and depth information from an RGB-D Kinect sensor," International Journal of Advanced Robotic Systems, vol. 10, p. 8, 2013.

[4] P. Kohli and J. Shotton, "Key developments in human pose estimation for kinect," in Consumer depth cameras for computer vision: Research topics and applications, A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds., ed. London: Springer-Verlag, 2013, pp. 63-70.

[5] B. Holt and R. Bowden, "Static pose estimation from depth images using random regression forests and Hough voting," presented at the 7th International Conf. on Computer Vision Theory & Application, 2012.

[6] M. S. N. Kumar and R. V. Babu, "Human gait recognition using depth camera: A covariance based approach," presented at the Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, Mumbai, India, 2012.

[7] M. Ahmed, N. Al-Jawad, and A. T. Sabir, "Gait recognition based on Kinect sensor," in SPIE 9139, Real-Time Image and Video Processing, Brussels, Belgium, 2014.

[8] S. Czarnuch and M. Ploughman, "Automated gait analysis in people with Multiple Sclerosis using two unreferenced depth imaging sensors: Preliminary steps," in Proceedings of the Newfoundland Electrical and Computer Engineering Conference, IEEE, Newfoundland and Labrador Section, St. John's, NL, 2014.

[9] pointcloud.org. (2014). Point Cloud Library. Available: http://pointclouds.org/

[10] J. Zhang, X. Yang, G.-M. Song, T.-Y. Chen, and Y. Zhang, "Relative orientation and position detections based on an RGB-D sensor and dynamic cooperation strategies for jumping sensor nodes recycling," Sensors, vol. 15, pp. 23618-23639, 2015.

[11] D. Holz, S. Holzer, R. Rusu, and S. Behnke, "Real-Time Plane Segmentation Using RGB-D Cameras," in RoboCup 2011: Robot Soccer World Cup XV. vol. 7416, T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 306-317.

[12] Microsoft. (2012, December 11). Kinect for Windows. Available: http://www.microsoft.com/en-us/kinectforwindows/

[13] OpenNI. (2012, December 11). OpenNI Modules. Available: http://openni.org/Downloads/OpenNIModules.aspx

[14] K. Litomisky and B. Bhanu, "Removing Moving Objects from Point Cloud Scenes," in Advances in Depth Image Analysis and Applications. vol. 7854, X. Jiang, O. Bellon, D. Goldgof, and T. Oishi, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 50-58.

[15] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in International Conference on Intelligent Robots and Systems, 2010, pp. 2155-2162.

[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," presented at the IEEE International Conference on Robotics and Automation, Kobe, 2009.