122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society

Paper ID #11868

Simultaneous Tracking and Reconstruction of Objects and its Application in Educational Robotics Laboratories

Mr. Mingshao Zhang, Stevens Institute of Technology

Mingshao Zhang is currently a Ph.D. student in Mechanical Engineering Department, Stevens Institute of Technology. Before joining Stevens, he received bachelor's degrees from University of Science and Technology of China. His Current research interests include Microsoft Kinect, Computer Vision, Educational Laboratories, Desktop Virtual Reality and etc.

Mr. Zhou Zhang, Stevens Institute of Technology

Ph.D Candidate, Mechanical Engineering Department, Stevens Institute of Technology, Hoboken, NJ, 07030.

Email: zzhang11@stevens.edu

Yizhe Chang, Stevens Institute of Technology (School of Engineering and Science)
Dr. Sven K. Esche, Stevens Institute of Technology (School of Engineering and Science)

Simultaneous Tracking and Reconstruction (STAR) of Objects and its Application in Educational Robotics Laboratories

Mingshao Zhang, Zhou Zhang, Yizhe Chang, Sven K. Esche & Constantin Chassapis
Stevens Institute of Technology, Hoboken, New Jersey, USA

Abstract

Many educational and industrial applications that involve robots require knowing the location information for the robots. This necessitates both the ability to localize the robots globally in the absence of any prior data as well as to track the robots' current positions once their initial locations are known. Various approaches have been used to solve these problems, such as encoders, inertial navigation, range sensing and vision-based techniques. Among those state-of-the-art robot localization methods, vision-based techniques are considered as some of the most effective approaches, and they can be enhanced significantly by obtaining additional supporting information from signal processing techniques and related algorithm developments. However, many challenges associated with the use of vision-based robot tracking systems in uncontrolled environments remain. For example, hardware components of visual odometry systems tend to be expensive and difficult to implement; choosing the most suitable algorithms and analysis methods is not straightforward and those algorithms are considered to be computationally expensive.

In this paper, a visual odometry system implemented using a low-cost user-friendly 3-D scanner (the Microsoft Kinect) is presented. A traditional approach for robot tracking based on object recognition was applied, which includes building an object database, followed by extracting, describing and matching keypoints between the database and the scene. The advantages and disadvantages of using the Kinect in this approach were studied. Then, a technique for the simultaneous tracking and reconstruction (STAR) of objects was developed and tested. This technique was inspired by the simultaneous localization and mapping (SLAM) approach, and it was implemented using the Kinect and an iRobot Create platform. The prototype implementation shows that this STAR technique is feasible and suitable to be used in educational robotics laboratories. This technique also has multiple advantages compared to traditional educational laboratories, such as lower cost, more straightforward setup and less required preparation work by the laboratory instructor.

1. Introduction

Interest in robotics has increased tremendously over the past decade. Along with the rapid expansion of the commercial robotics market, robotics has also entered engineering curricula at all levels. The way robotics is currently introduced in educational applications is narrow. Most of the applications of robotics technology in education have mainly focused on facilitating the teaching of subjects that are closely related to robot design, robot programming, mechatronics and industrial robot applications. Only a small number of reported cases indicated the usage of robotic techniques in other forms, which are believed to have the potential to engage young people with a wider range of interest. These activities use robotics as a way to tell a story (e.g. mechanical puppet show [1]) or in connection with other disciplines and interest areas, such as music and arts [1].

Studies have shown that it is feasible to use robotics for teaching or developing skills in areas both closely and not so closely related to the field of robotics [2]. The empirical evidence to support the effectiveness of educational robotics is still limited. However, it is believed that educational robotics has enormous potential as a learning tool. The focus of the work presented here is on developing educational applications of robotics. For example, robotics represents an interesting practical application of fundamental engineering knowledge areas such as geometry [3], Newton's laws of motion [4] and kinematics [3]. Thus, teaching robotics can contribute to developing and improving the students' skills in problem solving, logic and scientific inquiry as well as to teaching the related basic knowledge.

2. Background

2.1. Traditional Implementation of Educational Robotic

There is an extensive list of educational initiatives that use robotic technologies as pedagogical tools. However, the way robotics is currently introduced in educational applications is unnecessarily narrow. Most of these initiatives are oriented toward teaching subjects directly related to the robotics field. Furthermore, it is difficult to keep the complexity and cost of robotic devices to a manageable level for a regular size class.

There are many challenges that need to be tackled to successfully build an educational robotics laboratory, related to both hardware and software. In this work, robot localization was chosen as the challenge since it is needed in many robotics applications. For example, robot programming experiments require the location information of the robots as feedback to the control program. It is desirable for tracking systems to be able to localize the robots globally in the absence of any prior data as well as to track the robots' current positions once their initial locations are known.

2.2. State of the Art in Robot Localization

Mobile robot localization, which is the process of determining and tracking the robot's location and orientation relative to its environment, has received considerable attention over the past few years. Judged by the hardware utilized in robot tracking systems, the most commonly deployed state-of-the-art robot localization techniques can be divided into four classes.

1. Dead reckoning: Encoders are used to calculate translational movements from rotational measurements based on integration. This class is theoretically simple and easy to be integrated into mobile robots. However, it is considered noisy and less robust than other classes [5].

2. Inertial navigation systems: Inertial sensors, accelerometers and detectors for electromagnetic fields and gravity are used to calculate motions based on integration. This class can lead to error accumulation, especially when drift-prune sensors are used [6].

3. Ranging: Laser, infrared, acoustic or radio signals are used to measure distance. This class tends to be unreliable in highly dynamic environments, limited in range and tends to be expensive [7].

4. Visual odometry: Single-camera, stereo-vision or even omni-directional imaging is used to determine the position and orientation of a robot. This class is considered to be computationally expensive. However, it has become a popular research topic lately. It can be improved significantly by embedding signal processing techniques and adopting new computer-vision algorithm developments [8].

2.3. Vision-based Methods and Benefits of using 3-D Scanners

There are three broad groups of indoor robot localization systems: map-based localization, map-building-based localization and mapless localization [9]. The first kind of system operates on user-created geometric models or topological maps, while the second kind uses sensors to construct its own maps and then uses these maps for localization. Mapless tracking systems recognize objects and calculate their motions based on visual observations, which suits our purpose (i.e. building easy-to-use educational laboratories) better than the first two categories.

In order to build a simple system and render it easy to operate even for inexperienced users, a single camera or groups of cameras fixed in the environment and a robot without additional sensors attached to it were chosen here as experimental setup. The robot motions are determined by observing and extracting relevant information about the elements in the environment (i.e., walls, objects such as desks, chairs, etc. and the robot itself). Using a 3-D scanner (e.g., the Microsoft Kinect) instead of stereo vision cameras can further simply the hardware setup because of the depth information it provides [10,11]. If appropriately placed in the environment, a single 3-D scanner could be enough to capture the entire motion of the robot.

3. Recognition-based Object Tracking

3.1. Overview of Recognition-based Tracking Methods

Of the techniques that have been tried for mapless tracking, the prominent ones include optical flow based methods [12] and object recognition [13]. The optical flow based methods first extract moving objects by using motion detection between frames, then they determine feature points on the surface of the objects, and finally they match the corresponding feature points between frames to estimate the object motion. Therefore, objects can be tracked without any prior information or constraints with respect to camera position or object motion. However, different implementations of optical flow based methods can cause different errors, such as induced noise when the illumination changes, failure to tack large motions, and accumulation of computational errors [14].

Localization by using object recognition techniques is promising because it uses natural visual features that can be extracted from the target objects [15]. The object recognition problem can be defined as a labeling problem based on models of known objects. Thus, the information of the object models must be available a priori, and the model database becomes an essential part of an object recognition system. A typical object recognition process includes building the model database, detecting the feature points on both the models in the database and those in the scene, finding corresponding feature points and verifying the result.

3.2. Background of 3-D Object Reconstruction

Geometry reconstruction is a well-studied research area in computer graphics and computer vision and extensive literature on this topic has been published. Examples include the digital Michelangelo project that used tracked 3-D scanners to digitize large statues [16], the viewpoint-based approach of 3-D shape reconstruction using video sequences [17], dense 3-D reconstruction from unregistered Internet-scale photo collections using appearance-based clustering techniques [18], surface reconstruction from unorganized 3-D points by solving a spatial Poisson problem [19], and environment and object virtualization using a single 3-D scanner [20,21].

There is a body of research focusing on object scanning, and those techniques have been proven to be suitable for building the datasets of 3-D models of objects. One of the first real-time reconstruction methods for small models was developed using a frame-to-frame iterative closest point (ICP) implementation [22]. Since the introduction of depth cameras, a large number of real-time 3-D reconstruction related applications have been reported, including a scanning system using a fixed time-of-flight (ToF) camera to track moving objects [23] and a moving handheld ToF object scanner [24].

In the project described here, an iRobot Create platform [25] was chosen for the prototype implementation, which consists of a Create and a laptop. The user programs the path of the Create using the laptop, and a recognition-based tracking technique was employed to record the actual path of this Create. In order to recognize the Create using the Kinect, a model representing it must be present in the object database. In the work presented here, a software project named KinectFusion [26] was used to generate 3-D models for the objects of interest. These generated models can then be used as object database in the following experiment development. KinectFusion enables the user to hold and move a standard Kinect sensor to rapidly create detailed 3-D reconstructions of an indoor scene. The system works by continually tracking the 6 DOF pose of the Kinect camera and fusing live depth data from the camera into a single global 3-D model in real time. The reconstructed models of the scene can also be texture mapped using the RGB camera of the Kinect. A sample implementation is shown in Figure 1, where a laboratory scene was reconstructed using the KinectFusion. Then, the points belonging to the object of interest were segmented out to form the object model.

Figure 1: Generation of 3-D object model using KinectFusion; Left: Scene with Create; Right: Point cloud model of Create (after segmentation)

3.3. Implementation of Recognition-based Tracking and its Problems

The strategy in recognition-based tracking is to compare each frame of the acquired data with the object database, find the objects of interest and calculate their positions and orientations. The localization information of the objects of interest can be acquired by aggregating the calculation results.

Common approaches for object recognition consist of three main stages, namely extracting keypoints, generating appropriate feature descriptors from them and subsequently comparing the keypoint descriptors from the scene and model database for a possible match. Here, the 3-D SIFT keypoint technique [27] was chosen for the implementation. This technique extends the feature descriptors from 2-D images (i.e., x and y) to 3-D spaces (i.e., x, y and z). It starts by extending the scale space using the 3-D Gaussian blur operator G, followed by finding the keypoints at the maxima/minima of the Difference of Gaussian (DoG). The DoG image $D(x,y,z,\sigma)$ is defined as:

$$
\begin{aligned}
&D(x,y,z,k_i\sigma) = L(x,y,z,k_i\sigma) - L(x,y,z,k_{i-1}\sigma) \\
&L(x,y,z,k_i\sigma) = G(x,y,z,k_i\sigma) * I(x,y,z) \\
&G(x,y,z,k_i\sigma) = \frac{1}{\left(\sqrt{2\pi}k_i\sigma\right)^3} e^{-\left(x^2+y^2+z^2\right)/2\left(k_i\sigma\right)^2}
\end{aligned}
$$

Here, $G(x,y,z,\sigma)$ is the Gaussian blur operator, σ is the standard deviation of the Gaussian distribution of the points' grayscale values, $I(x,y,z)$ is the grayscale value of the original input image, and k_i and k_j are scalar numbers used to define the scale of the Gaussian blur.

The next few steps include finding keypoints, eliminating bad keypoints (i.e., false keypoints induced by image noise or computational errors) and calculating the gradient direction and magnitude in the proximity of the keypoints. In the final step, the descriptors are created, which typically consist of 64 histograms aligned in a 4×4×4 grid, each with 8 azimuth directions and 4 elevation directions, thus resulting in a feature vector containing 2,048 elements for each keypoint. These resulting vectors are known as SIFT keys, and they are used for a nearest-neighbors search aimed at identifying possible matches within the source and target images.

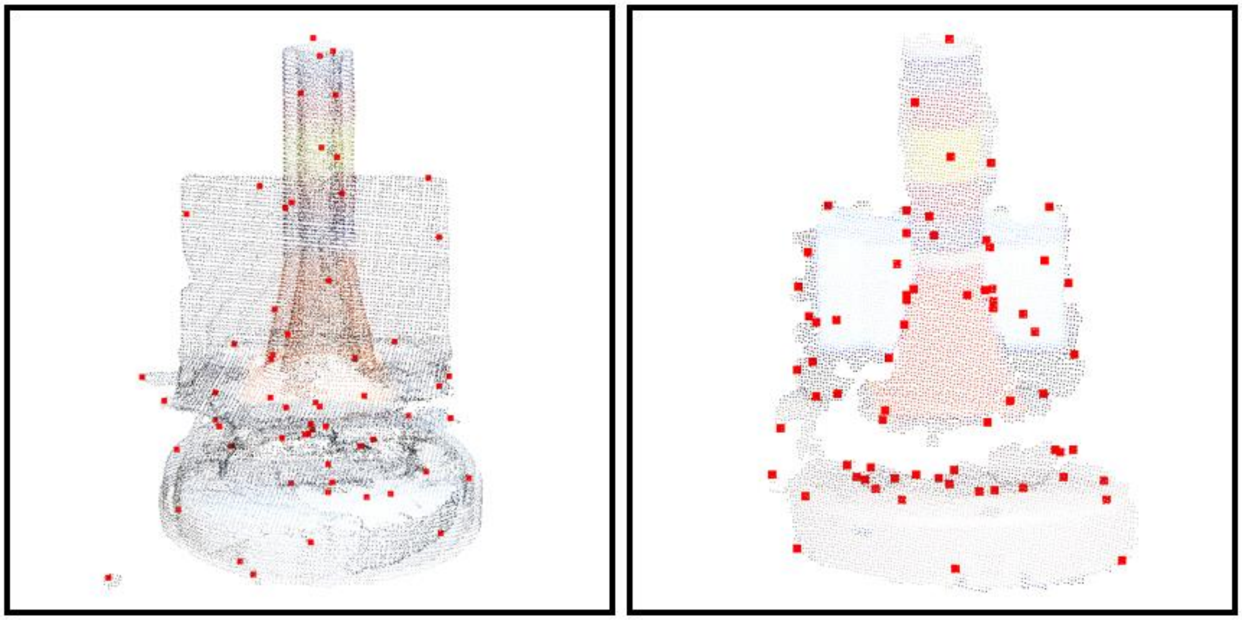

Figure 2: 3-D SIFT keypoints; Left: Create model generated by KinectFusion; Right: Create model segmented from raw data

Keypoint extraction is performed in both the model that is created by KinectFusion and the point cloud sequence captured by the Kinect. The calculated keypoints are used for generating keypoint descriptors and then tracking the motion of the Create by finding a possible match between the models in the database and the point cloud sequence. The result of the keypoint extraction is shown in Figure 2, where the red dots represent extracted feature keypoints. The overall result of using this frame-by-frame object recognition based tracking method is not ideal, whereby most of the errors are caused in the first stage of the object recognition, i.e. during the keypoint extraction. The factors that cause recognition failures can be categorized into two groups, namely keypoint extraction errors from the data recorded by the Kinect and keypoint extraction errors from the object database that was created using the KinectFusion. The first category of recognition failures is caused by the imperfection of the raw data obtained from the Kinect. Among the limitations that the Kinect inherits from it being a structured light scanner, several have a direct impact on the quality of the raw data. For example, the Kinect's strong dependency on the surface properties and the shadowing problem [28] can both cause occlusions in the raw data [29]. Also, the low accuracy and high error rate in the Kinect's depth data are caused by the round-off errors from the computations associated with the infrared sensor [30]. The second category of recognition failures arises when building the object database using the KinectFusion. One of the main problems is that KinectFusion uses the ICP algorithm [31] to track the position of the camera. The ICP algorithm assumes that the camera pose changes very slightly between two frames and that the scene is sufficiently feature-rich. Both a lack of distinguishable features and sudden movement of the handheld Kinect can cause reconstruction errors in the final database of object models.

It is believed that the tracking results can be improved using the following three approaches: by building a more accurate database, by finding more suitable recognition algorithms [30] and by preprocessing the Kinect's raw data to improve their accuracy. However, certain drawbacks of this recognition-based tracking technique still remain, namely the requirement that the database must be large enough to cover all possible objects to be tracked and the fact that it is difficult for inexperienced users to generate such a database. Therefore, a system that is capable of creating and updating the database automatically and that can also track the movement of objects of interest is desired.

4. Simultaneous Tracking and Reconstruction of Objects

4.1. Background on Simultaneous Tracking and Reconstruction

Simultaneous localization and mapping (SLAM) is a technique that uses digital devices to construct a map of an unknown physical environment or to update a map within a known environment, while simultaneously keeping track of the device's location in the environment [32,33]. The SLAM technique consists of the following steps:

1. Landmark extraction: extract from the environment features that can easily be re-observed and distinguished;
2. Data association: match observed landmarks from different scans with each other;
3. State estimation: estimate the position of the device from odometry data and landmark matching;
4. State update: update the estimated state from re-observing landmarks;
5. Landmark update: add new information to the current map.

Inspired by the SLAM scheme, a simultaneous object tracking and reconstruction technique is proposed here. This technique can be used either to track the movement of unknown objects while generating their models or to track objects known a priori while updating their models. The system works by first analyzing the raw data recorded by the Kinect and segmenting out the potentially moving objects. Then, the segmented point cloud is compared with models in the database. If the database contains models of the objects, the position of the objects is estimated and the model is updated if necessary. Otherwise, the segmented point clouds from different scans are compared, the movements of the objects between those scans are estimated, and the model is generated and stored in the database.

Various methods, including the 3-D SIFT technique mentioned above, can be used either to estimate the position of the target objects using models from the database or to estimate the movement between segmented point clouds from different scans. Here, the ICP algorithm was implemented, which works by minimizing the difference between two clouds of points [34]. The algorithm involves the following steps:

1. For each point in the source point cloud, find its closest point in the reference point cloud as a match;
2. Estimate the combination of rotation and translation (i.e. the transformation matrix) using a mean squared error cost function that best aligns those matched points;
3. Transform the source points using the obtained transformation matrix.

4.2. Prototype Implementation of STAR with Kinect and its Problems

An educational robotics laboratory was selected to validate the STAR approach proposed here. The Kinect was chosen as sensor to build an educational laboratory where the motion of an unknown object of interest is tracked and simultaneously its model is generated. Starting with the Kinect's raw data, NaN value filtering [11], motion detection and local region segmentation are applied [35]. Only the point cloud that potentially contains the moving objects is segmented out and used as input for the simultaneous object tracking and reconstruction process. First, object recognition between the first frame of the point cloud and the model dataset is performed. If a match is found, the model from the dataset is chosen as the object representation for further tracking and potential target of model updating. Otherwise, the first frame is chosen as model representation and used for further tracking and model updating, after which the final model is stored in the model dataset as a newly discovered object. The pseudo code of the STAR algorithm is shown in Listing 1.

Listing 1: STAR algorithm

for each segmented point cloud P_i do
    Calculate rigid body transformation between P_i and object model M_{i-1} using ICP
    Transform P_i into global coordinates to align with M_{i-1}
    Update voxel grid representation of object V_{i-1} by
        Update grid point count
        Filter color noise and update color information
    Generate object model M_i from voxel grid V_i

There are two approaches to performing object tracking using the ICP technique. In the first approach, each point cloud frame is compared with its preceding frame, the relative rigid body transformation is computed, and the final moving path is acquired. The benefit of using this approach is that the object motion between two consecutive frames is usually small, which makes the ICP algorithm converge in a very short time. However, some small errors (e.g., computation truncation, extensive error from certain frames and the ICP resulting in a local minimum) are accumulated, thus causing all subsequent computations to be corrupted. ICP tracking methods are undesirable for tracking objects through a long time period, unless some intermediate calibration stages are introduced into the process. The second approach for object tracking is to compare each frame with the updated object model and calculate the relative rigid body transformation, which can be used as the final moving path directly. This method is inherently more robust than the first method. However, the motion between a particular frame and the updated model is larger compared to the first method, which makes this approach more computationally costly. Though unlikely to happen, large motions cause the ICP to converge to a local minimum, thus yielding false results. In the application presented here, the second approach for object tracking was chosen. For each frame of the point cloud that potentially contains the object of interest, the previously updated object model is used to calculate the ICP convergence. The transformation matrix calculated by the ICP algorithm is recorded as tracking result and the current frame point cloud is then used to update the object model.

Figure 3 depicts the overall tracking result, where the Create was controlled to move on a circular path. The left figure represents the Create's circular path that was preset by the user, and the right figure depicts the path that was calculated using the STAR technique discussed above.

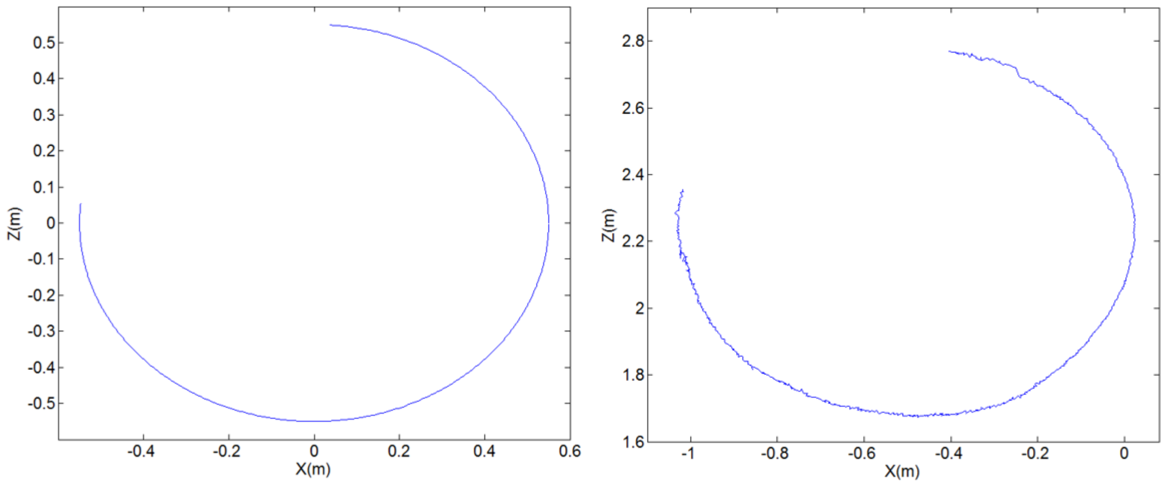

Figure 3: Tracking result of Create; Left: preset circular path; Right: path calculated by STAR algorithm

Once the position and orientation of the target object have been determined, the segmented point cloud is transformed into the global coordinate system (here chosen as the coordinate system of the first frame) using the acquired transformation matrix. The model that represents the target object is then updated using the latest frame of the point cloud, which involves three stages of calculation. First, the voxel grid representation of the model is updated. Here, the global vertices are integrated into voxels using a Truncated Signed Distance Function [36]. Second, each discrete 3-D grid location is converted into a vertex in the global coordinate system, which generates the point cloud that represents the updated object model. Finally, the color information is retrieved and added to the point cloud. The result of the updated point cloud model is depicted in Figure 4. Here, the recognition-based tracking was accomplished without generating the model of the Create a priori. Instead, the model was generated from the first two frames of the point cloud sequence and then updated using the following frames. After the ICP calculation of each frame of the point cloud, the positions and orientations of the Create are recorded as tracking result. At the end of this experiment, the model of the Create was stored in the model database for future usage.

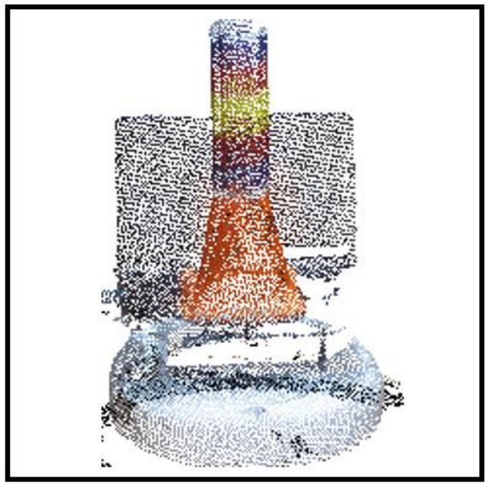

Figure 4: Updated object model using STAR technique

4.3. Benefits of Using STAR in Educational Applications

In order to implement robot tracking using the localization techniques mentioned above (dead reckoning inertial navigation and ranging), significant modifications (in most cases on both the robots and in the laboratory environment) are required. These changes add to the cost of constructing the laboratory. The complexity of the hardware setup is also owed to the fact that these platforms require experienced and well trained users to be operated and maintained. However, vision-based methods can make the hardware system simpler and easier to operate for inexperienced users.

Since most vision-based methods require marks to be added on the robots for tracking, using recognition-based methods can further reduce the complexity of the hardware setup. However, a comprehensive and high quality object model database needs to be built beforehand, which can be difficult and time consuming for the developers. Among those recognition-based tracking methods, using the STAR technique proposed here, the developed educational system only requires a straightforward setup (the user only needs to place the Kinect in a suitable position to cover the motion of the objects). This system does not require the user to prepare the model dataset, thus minimizing the preparation workload of the laboratory instructor.

In conjunction with suitable image processing algorithms, the Kinect has the potential for becoming a versatile 3-D range sensor with numerous options for educational and industrial usage. Taking educational laboratories as an example, the Kinect will be able to meet some fundamental requirements, such as a short time for learning the use of the system, ease of operation, the ability to add resources as the students' knowledge grows and robustness to prevent certain kinds of errors [37]. Furthermore, the ability to track the position, velocity, acceleration and even deformation of multiple moving objects in three-dimensional space would make the Kinect superior to most of the existing data acquisition systems currently used in educational laboratories.

5. Conclusions and Future Research

In this paper, the usage of the Kinect as alternative 3-D scanner in educational robotics laboratories was proposed. The state of the art in robot localization was studied. The benefits and limitations of using vision-based methods in robot localization were discussed. The traditional object recognition based tracking method was implemented and its advantages and disadvantages were explored. A novel simultaneous object tracking and reconstruction technique that is based on the SLAM and ICP algorithms was developed and implemented, and its advantages compared to traditional recognition based tracking methods were presented.

The next stages of research should focus on three issues. First, a more suitable ICP implementation aimed at minimizing the tracking error should be explored. Second, a better voxel grid method that improves the quality of the resulting object models should be developed. Third and finally, prototype applications should be chosen and implemented to prove the validity and impact of this work.

6. References

[1] Rusk, N., Resnick, M., Berg, R. & Pezalla-Granlund, M., 2008, “New Pathways into Robotics: Strategies for Broadening Participation,” *Journal of Science Education and Technology*, 17(1) 59-69.

[2] Benitti, F. B. V., 2012, “Exploring the Educational Potential of Robotics in Schools: A Systematic Review,” *Computers & Education*, 58(3) 978-988.

[3] Mitnik, R., Nussbaum, M. & Soto, A., 2008, “An Autonomous Educational Mobile Robot Mediator,” *Autonomous Robots*, 25(4) 367-382.

[4] Williams, D. C., Ma, Y., Prejean, L., Ford, M. J. & Lai, G., 2007, “Acquisition of Physics Content Knowledge and Scientific Inquiry Skills in a Robotics Summer Camp,” *Journal of Research on Technology in Education*, 40(2) 201-216.

[5] Borenstein, J. & Feng, L., 1994, “UMBmark: A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots,” *Technical Report UM-MEAM-94-22*, The University of Michigan.

[6] Goshen-Meskin, D. & Bar-Itzhack, I. Y., 1992, “Unified Approach to Inertial Navigation System Error Modeling,” *Journal of Guidance, Control, and Dynamics*, 15(3) 648-653.

[7] Borenstein, J., Feng, L. & Everett, H. R., 1996, “Navigating Mobile Robots: Systems and Techniques,” AK Peters, Ltd.

[8] Nistér, D., Naroditsky, O. & Bergen, J., 2006, “Visual Odometry for Ground Vehicle Applications,” *Journal of Field Robotics*, 23(1) 3-20.

[9] DeSouza, G. N. & Kak, A. C., 2002, “Vision for mobile robot navigation: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2) 237-267.

[10] Zhang, M., Zhang, Z., Esche, S. K. & Chassapis, C., 2013, “Universal Range Data Acquisition for Educational Laboratories Using Microsoft Kinect,” *Proceedings of the 2013 ASEE Annual Conference & Exposition*, Atlanta, Georgia, USA, June 23-26.

[11] Zhang, M., Zhang, Z., Aziz, E.-S., Esche, S. K. & Chassapis, C., 2013, “Kinect-Based Universal Range Sensor for Laboratory Experiments,” *Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'13*, November 13-21.

[12] Shin, J., Kim, S., Kang, S., Lee, S. W., Paik, J., Abidi, B. & Abidi, M., 2005, “Optical Flow-Based Real-Time Object Tracking Using Non-Prior Training Active Feature Model,” *Real-Time Imaging*, 11(3) 204-218.

[13] Foresti, G. L., 1999, “Object Recognition and Tracking for Remote Video Surveillance,” *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7) 1045-1062.

[14] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J. & Szeliski, R., 2011, “A Database and Evaluation Methodology for Optical Flow,” *International Journal of Computer Vision*, 92(1) 1-31.

[15] Se, S., Lowe, D. & Little, J., 2002, “Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks,” *The international Journal of Robotics Research*, 21(8) 735-758.

[16] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J. & Fulk, D., 2000, “The Digital Michelangelo Project: 3D Scanning of Large Statues,” *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 131-144, ACM Press/Addison-Wesley Publishing Co.

[17] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J. M., Yang, R., Nister, D. & Pollefeys, M., 2007, “Real-time Visibility-based Fusion of Depth Maps,” *Proceedings of IEEE 11th International Conference on Computer Vision*, pp. 1-8.

[18] Frahm, J. M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y. H., Dunn, E., Clipp, B., Lazebnik, S. & Pollefeys, M., 2010, “Building Rome on a Cloudless Day,” *Proceedings of Computer Vision – ECCV 2010*, pp. 368-381, Springer Berlin Heidelberg.

[19] Kazhdan, M., Bolitho, M. & Hoppe, H., 2006, “Poisson Surface Reconstruction,” *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*.

[20] Zhang, Z., Zhang, M., Chang, Y., Aziz, E.-S., Esche, S. K. & Chassapis, C., 2013, “Real-Time 3D Model Reconstruction Using Kinect for a Game-Based Virtual Laboratory,” *Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'13*, November 13-21.

[21] Zhang, Z., Zhang, M., Chang, Y., Esche, S. K. & Chassapis, C., 2014, “An Efficient Method for Creating Virtual Spaces for Virtual Reality,” *Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'14*, November 14-20.

[22] Rusinkiewicz, S., Hall-Holt, O. & Levoy, M., 2002, “Real-time 3D Model Acquisition,” *ACM Transactions on Graphics*, 21(3) 438-446.

[23] Weise, T., Wismer, T., Leibe, B. & Van Gool, L., 2009, “In-hand Scanning with Online Loop Closure,” *Proceedings of 12th IEEE International Conference on Computer Vision Workshops*, pp. 1630-1637.

[24] Cui, Y., Schuon, S., Chan, D., Thrun, S. & Theobalt, C., 2010, “3D Shape Scanning with a Time-of-flight Camera,” *Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1173-1180.

[25] http://www.irobot.com/About-iRobot/STEM/Create-2.aspx, accessed in March, 2015.

[26] Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux D., Hodges, S., Kim, D. & Fitzgibbon, A., 2011, “KinectFusion: Real-time Dense Surface Mapping and Tracking,” *Proceedings of 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127-136.

[27] Allaire, S., Kim, J. J., Breen, S. L., Jaffray, D. A. & Pekar, V., 2008, “Full Orientation Invariance and Improved Feature Selectivity of 3D SIFT with Application to Medical Image Analysis,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 23-28, 2008, pp. 1-8.

[28] Roscoe, M. S., Kent, K. B., Herpers, R. & Plöger, P. G., "Determining the Viability of the Kinect as a Research Tool"

[29] Draelos, M. T., 2012, "The Kinect up Close: Modifications for Short-Range Depth Imaging," Master's Thesis, North Carolina State University, Raleigh, North Carolina, USA.

[30] Zhang, M., Zhang, Z., Esche, S. K. & Chassapis, C., 2014, "Algorithm Modification Approach to Improve the Kinect's Performance in Point Cloud Processing," *Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'14*, November 14-20, 2014.

[31] Rusinkiewicz, S. & Levoy, M., 2001, "Efficient Variants of the ICP Algorithm," *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145-152.

[32] Durrant-Whyte, H. & Bailey, T., 2006, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, 13(2) 99-110.

[33] Bailey, T. & Durrant-Whyte, H., 2006, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, 13(3) 108-117.

[34] Besl, P. J. & McKay, N. D., 1992, "Method for Registration of 3-D Shapes," *Proceedings of Robotics-DL tentative, International Society for Optics and Photonics*, pp. 586-606.

[35] Zhang, M., Zhang, Z., Chang, Y., Esche, S. K. & Chassapis, C., 2015, "Kinect-based Universal Range Sensor and its Application in Educational Laboratories," *International Journal of Online Engineering*, 11(2) 26-35.

[36] Curless, B. & Levoy, M., 1996, "A Volumetric Method for Building Complex Models from Range Images," *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 303-312.

[37] Yáñez, J., Quintana, D., Quintáns, C., Fariña, J. & Rodríguez-Andina, J., 2005, "FPGA-Based System for the Education in Data Acquisition and Signal Generation", *Proceedings for the 31st Annual Conference of IEEE*, November 6-10, 2005.