120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #6364

Universal Range Data Acquisition for Educational Laboratories Using Microsoft Kinect

Mr. Mingshao Zhang, Stevens Institute of Technology
Mr. ZHOU ZHANG
Dr. Sven K. Esche, Stevens Institute of Technology

Universal Range Data Acquisition for Educational Laboratories Using Microsoft Kinect

Abstract

Many experiments conducted in engineering and science laboratories involve the acquisition of range data such as linear or angular position, velocity and acceleration, distance, displacement, etc. This type of data acquisition (DAQ) is accomplished by sensors, DAQ measurement hardware and a computer with programmable software. This approach to DAQ can cause a series of problems hampering its implementation in educational laboratories. For instance, many sophisticated sensors (e.g. laser scanners) and the DAQ hardware are expensive, often the sensors and DAQ hardware and peripheral devices require modifications for being reused in other applications and most experimental setups need to be calibrated before each measurement. These facts tend to increase the up-front cost of the experimental devices and add to the required operating time. Therefore, low-cost range sensors such as the Microsoft Kinect could become a cost-effective and versatile DAQ alternative. Furthermore, Kinect has acceptable performance regarding sensitivity, accuracy, stability and reliability as well as low error rates, cost and power consumption.

In this paper, the concept of using Kinect as a substitute range DAQ is presented and a prototype implementation targeting educational experiments is introduced. This system has several attractive features besides low cost, including that it (in conjunction with appropriate software) can be trained to recognize and remember multiple objects, is able to track these objects simultaneously, does neither need to be customized nor modified for measurements in different applications, and uses all the surface data (as opposed to single-point tracking) to calculate the positions and deformations of objects, which results in low drifting error. Taking advantage of these desirable characteristics, Kinect is believed to have the potential for becoming an economical and versatile tool for adoption in a wide variety of educational laboratories.

1. Introduction

A data acquisition system is usually defined as an electronic instrument or group of interconnected electronic hardware apparatus used for the measurement and quantization of analog signals for digital analysis or processing of certain physical phenomena [1]. Compared to traditional hands-on only laboratories, experiments equipped with a DAQ system can hold considerable value for the students' learning outcomes. The students can develop a better understanding of the investigated phenomena because the efficiency of the data-gathering process makes it easy to repeat the experiment several times. Repeating experiments also enables the students to develop proficiency in instrumentation and measuring techniques. Running multiple experiments under different conditions enables the students to analyze a more complete set of data and to check the validity and/or limitations of the theoretical models, principles and assumptions, and ultimately to acquire a better understanding of the underlying physical phenomena. Furthermore, the efficiency gained by using DAQ systems can also assist the instructor in developing approaches to promote collaboration and teamwork activity among the students. Finally, the reduction in time of each run of the experiment can also reduce the number of passive participants (known as observers) in the experiment [2].

When designing a DAQ system for educational experiments, certain requirements are considered as fundamental. The time needed for the students to learn to use the system should be as short as possible, the system should not be too complicated to operate, the students should be able to add resources (i.e. a new apparatus) to the system during the learning process as their level of knowledge increases, and the system should be robust enough to prevent certain kinds of error [3]. Successfully designing a suitable DAQ system that satisfies the above-mentioned requirements even for experienced engineering students is quite difficult from an educational point of view. Also, the availability and continuous development of many sensors, signal conditioning circuits and actuators makes it difficult to set up a well functioning and inexpensive educational DAQ framework [4].

One of the fundamental capabilities of a DAQ system is the possibility to adapt it to different applications. For students to conduct different experiments, the instructors should be able to easily modify the existing DAQ systems instead of purchasing new ones. However, in current DAQ systems, the sensors and signal conditioning circuits need significant modifications and calibration before they can be used in new applications, which adds to the operating time. Also, many of the sophisticated sensors are expensive, which limits their accessibility to many educational institutions [5]. Therefore, the Microsoft Kinect becomes a promising alternative as range data sensor, with the associated benefits of being inexpensive, suitable for being modified as universal sensor used in different applications and capable of producing acceptable measurement results.

2. Using Kinect as Promising DAQ Alternative

The Microsoft Kinect system consists of a horizontal bar with built-in cameras, an accelerometer, a structured light projector and a microphone array. On the horizontal bar, there are three sensors: two infrared laser sensors used to acquire depth data with a range of 0.6-6.0 m and a RGB (red, green and blue color mode) camera with 640×480 pixels. The angular range of the cameras is 57°× 43° (horizontal × vertical), in which the vertical range can be extended to 54° by using the tilt motor [6,7]. The depth data is designed to have 11 bits, which provides 2048 levels of sensitivity [8].

Since its release in November 2010, Kinect has been proved to have great potential for applications in various areas. A fall-detection system for the elderly and the vulnerable was presented [9]. Similarly, the possibility to use Kinect in physical rehabilitation for young adults with motor disabilities was explored [10]. The two applications mentioned above, along with many other potential opportunities, were all developed based on the skeleton tracking feature of Kinect. Besides skeleton tracking, a virtually unlimited number of possible applications can be realized through direct computing using the 3D point cloud generated by the Kinect sensor. Examples include 3D object scanning and surface reconstruction [11], hand-gesture recognition [12], indoor mapping [13], etc. The application introduced in this paper is but one example of manipulating 3D point clouds.

The development of so many applications using Kinect in such as short time is possible because of its outstanding technical features. The Kinect has a low error value, the random error of depth measurements increases from a few mm to 4 cm as the distance increases to the maximum of the range. A properly calibrated Kinect sensor does not produce larger systematic error than a sophisticate laser scanner [14]. When the sensor is operating at a 30 Hz frame rate, the relative latency is found to be 106 ms on average, which makes real time applications possible. Also, Kinect's lateral resolution is measured to be 3 mm at a distance of 2 m, which is sufficient for comparatively less sophisticate application [15]. The accuracy of a calibrated Kinect sensor turns out to be +/-1 mm, which is very precise compared to the accuracy of +/-1 cm of a Kinect sensor that is not calibrated [16].

On the down side, Kinect has comparatively low x/y resolution and depth accuracy for applications such as 3D scanning. The density of points decreases with increasing distance, which leads to a low depth resolution at large distances (7 cm at the maximum of the range) [14]. These disadvantages need to be considered when developing related applications.

3. Algorithms Used for Motion Detection

3.1 Introduction

The Kinect system generates a point cloud with 640×480 pixels, each associated with their depth data and RGB value. In order to employ the Kinect as a range sensor as proposed, one needs to be able to use point cloud data to identify objects and track their motions. In section 3.2, two common algorithms (Lucas-Kanade [17] and Horn-Schunck [18]) will be summarized, which are used for tracking unidentified objects. These techniques are helpful when one needs to determine what the objects' motions are and when they occur, instead of finding their shapes and deformations, and they require only pixel brightness, which can be derived from the RGB data. In section 3.3, a simple technique that uses only the depth data from the point cloud will be introduced to track multiple moving objects. In section 3.4, a combination of algorithms using both depth and RGB data will be introduced, which will then be proved to be best suitable for our application.

3.2 Lucas-Kanade and Horn-Schunck Methods

Methods used to track unidentified objects involve tracking significant key points rather than extended objects. The term key points (also called corners) refers to points with a strong gradient of the pixel value (in our case, the brightness derived from the RGB value) in two orthogonal directions. These key points are unique and have a good chance of being identifiable again in the subsequent picture frames.

The Lucas-Kanade method relies only on local information that is derived from a small window around the point of interest. This is the reason that this method can be used only in sparse context (i.e. objects of simple geometry and texture, low image noise, etc.). In order to employ this method, a corner to be tracked needs to be identified first. Then, a window consisting of N×M pixels needs to be established (for simplicity, a N×N window was selected here). Finally, the motion direction vector $\begin{bmatrix} u & v \end{bmatrix}^T$ can be calculated from the following N^2 equations:

$$\begin{bmatrix} I_x(P_1) & I_y(P_1) \\ I_x(P_2) & I_y(P_2) \\ \vdots & \vdots \\ I_x(P_N) & I_y(P_N) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(P_1) \\ I_t(P_2) \\ \vdots \\ I_t(P_N) \end{bmatrix}$$

where I_x and I_y are the spatial derivatives of the brightness value of one pixel compared with the brightness values of the neighboring pixels in the two orthogonal directions for the first image and I_t is the derivative of the brightness of the same pixel between different images at different times. The system of equations can be solved using least-squares minimization.

The Lucas-Kanade method [17] is based on certain assumptions, namely brightness constancy, spatial coherence and temporal persistence [19]. The pixel data (in this case, the brightness) of a certain point does not change from frame to frame, neighboring points on the same surface should all undergo similar motions, and the motions of points change only slowly with time. The requirement of small motions is especially difficult to satisfy in a video camera running at frame rates of 30 Hz as in our target application, in which large and non-coherent motions are commonplace. For these reasons, the Lucas-Kanade method would exhibit a week performance in our case.

The Horn-Schunck method [18] is also based on the brightness constancy assumption. Furthermore, it assumes smoothness in the optical flow (i.e. the pattern of the apparent motion of objects, surfaces and edges [20]) over the whole image by minimizing the distortions in the flow. The direction vector $[u \quad v]^T$ of points of interest can be computed from the following relation:

$$
\begin{aligned}
I_x\left(I_x u + I_y v + I_t\right) - \alpha^2 \Delta u = 0 \\
I_y\left(I_x u + I_y v + I_t\right) - \alpha^2 \Delta v = 0
\end{aligned}
$$

in which Δ is the Laplace operator and α is a regularization constant. Larger values of α lead to smoother motion flow vectors.

Compared to the Lucas-Kanade method, the Horn-Schunck method produces a smoother flow and global information instead of local information. Furthermore, it uses more than two frames, thus rendering the derivatives more accurate. On the other hand, this method produces a larger number of optical flow vectors, is too sensitive to image noise [21], causes sharp boundaries between objects and their background to become distorted into smooth shapes, and the multi-time calculation using multiple frames makes this method slow [22]. These characteristics make the Horn-Schnuck method unsuitable for the application presented here.

3.3. Depth Data Method

A simple method using only the depth data of the point cloud is commonly implemented in a variety of simple applications and is available in almost all libraries for dealing with point cloud data processing [23]. This method only requires one to record and compare two depth maps $D(t)$ and $D(t + \Delta t)$, and at the point of interest, to compare the values of $D_{m,n}(t)$ and $D_{m,n}(t + \Delta t)$, where the indices m and n refer to the position of the pixel in an image. If the difference of the pixel between two maps is larger than a threshold value α, then this point is classified as a moving point. Then, the algorithms automatically merges the points located near the point of interest if they have similar depth values (i.e. have a smaller depth data difference than a threshold value β), and subsequently those points are considered as being part of the same object.

It can be seen that this method has several disadvantages. First of all, the threshold values α and β need to be defined by the user, which may require some trial and error. Then, the movement of the objects to be tracked also needs to be slow enough in order to prevent the result from becoming noisy, but not slow enough such that no depth change can be detected within two consecutive frames. Also, the moving object needs to have a significantly different depth value from those of the points near it. Otherwise, the algorithm may mistake environment points for points of the object. These shortcomings hamper the use of this simple algorithm in the universal sensor application introduced in this paper.

3.4 Hybrid Algorithm

In the application of using Kinect as range sensor, the system needs to first recognize the moving objects and then track their motions accordingly. Some applications may even involve identifying the moving objects' deformations. Furthermore, the system should be able to work at flexible environments, including the case where the background is chaotic and possibly changing over time. Under those assumptions, the two methods briefly summarized above become unsuitable and cannot produce the desired results. In this section, an efficient algorithm that uses both depth and RGB data is introduced. First, the depth data method introduced in Section 3.3 above is used to detect the potentially moving area, which is fast due to the method's simplicity. Then, a search for the moving objects that are already programmed to be recognizable to the system is conducted in the local area where motions are most likely to occur.

Algorithms for object recognition using 2D pixels generated by cameras or 3D point clouds created by 3D scanners represent mature technologies and large numbers of publications describing them are available. Also, a lot of algorithms have been modified to interact with the Kinect system, such as the 'Instance Distance Learning' algorithm used to detect objects from pre-built RGB-D datasets [24]. However, in our case of an educational range sensor application, the objects that need to be tracked usually have simple geometries and surface textures, such as spheres, cubes, cylinders, etc. In addition, those objects can be considered as rigid bodies, i.e. they experience no noticeable deformation in most scenarios. Therefore, algorithms designed to recognize complicated surfaces and geometries become overqualified and too time consuming in these cases. Algorithms using 2D or 3D data need to be traded off for better results such as introduced in [25], where 3D data are used when objects lack a discriminating visual feature (e.g. large, uniformly colored surfaces). In most cases, only a simple 2D recognition method such as shape matching using shape contexts is needed [26].

4. Experiment and Results

4.1 Pendulum Experiment

The simple pendulum experiment is an introductory level experiment in physics. The period of an ideal simple gravitational pendulum depends only on the swing length L and the strength of gravity g and is independent of the mass m of the bob. If the amplitude θ (i.e. the angle that the pendulum swings away from the vertical) is small, the period T can be calculated from $T \approx 2\pi\sqrt{L/g}$. Therefore, from measurements of the swing length L and period T, the gravitational constant g can be calculated.

The governing differential equation for the motion of a simple pendulum is

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0,$$

which is based on the assumptions that the cord is massless, the motion is planar and there is no energy loss due to friction or air resistance. If the air resistance is considered, the equation can be modified as

$$\frac{d^2\theta}{dt^2} + \mu\frac{d\theta}{dt} + \frac{g}{L}\sin\theta = 0$$

where μ denotes the damping coefficient.

4.2 Experiment Implementation

A simple pendulum was constructed using a tennis ball as swinging mass (see Figure 1). The pendulum was placed in front of a cluttered background in order to prove that the system functions in complex environments.

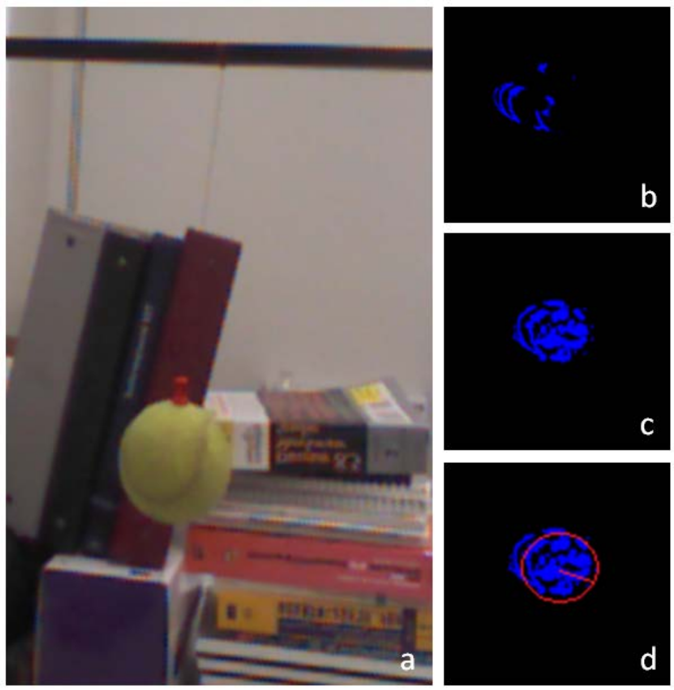

Figure 1: Experimental results

The hybrid algorithm implemented here is composed of the following three steps:

1. Use first part of hybrid algorithm, compare depth data map $D_{m,n}(t)$ and $D_{m,n}(t+\Delta t)$, highlight points with depth value change larger than preset threshold (see Figure 1b).
2. In neighborhood of points that moved, at map of time $t+\Delta t$, run object recognition algorithms to find desired object (here a sphere, see Figure 1c).
3. Use recognized points to calculate center point and radius of spherical object and direction of motion of object (see Figure 1d).

In this example of an educational experiment, the data most desired is the changing location of the center of the swinging mass $(x(t), y(t))$, from which the amplitude $\theta(t)$ can be determined. By plotting and analyzing the graph of $\theta(t)$, the students are then able to calculate the gravitational constant g and study the influence of friction on the behavior of the simple pendulum.

4.3 Results and Future Research

Taking advantage of both the efficiency of the depth method and the power of the shape recognition algorithm, the system is able to track multiple moving objects with simple geometries in real time in an acceptable manner. However, the system is still in proof-of-concept stage and has some disadvantages that hamper its broader usage as powerful range sensor in educational laboratories. From Figure 1b, apparent noise can be noticed. While most of the noise that is not related to the object is filtered out in the object recognition stage, some of it may still affect the final results. Also note that no user interface for this system has been designed yet, which prevents the accessibility of this system for most users.

The next stage of research will focus mainly on two areas. First, additional object recognition algorithms such as divide-and-conquer search [27], histograms of receptive field responses [28], geometric hashing [29], etc. will be explored and the most suitable algorithms for different application scenarios will be determined. In addition, the number of pre-recognized objects in our data set will be increased for broader and more universal usage in different experiments. Potential user interface designs include also the integration of the system into a game-based multi-user collaborative learning environment using the technology introduced previously [30]. Combining the advantages of real laboratories and virtual learning environments, the system will exhibit increased accessibility, user friendliness, flexibility, and at the end, improved learning outcomes [31].

5. Conclusion

In this paper, several algorithms used in 2D/3D object recognition and motion tracking were studied and compared, and a hybrid algorithm suitable for using Kinect as universal range sensor was developed. Based on that algorithm, a prototype application using Kinect as range DAQ in educational experiments was introduced. This low-cost system is able to identify and track multiple objects, and, in contrast to traditional DAQ systems, it does not need modification or adjustment in order to be used in different applications. The presented algorithm is efficient and can operate in almost real time. In light of these advantages, Kinect was shown to have great potential for becoming an economical and versatile tool for a variety of applications in educational laboratories.

Acknowledgements

This multi-disciplinary research project was carried out at Stevens Institute of Technology with funding from a multi-year grant by the National Science Foundation (Award No 0817463). This support is gratefully acknowledged.

References

[1] Definition of data acquisition from Wikipedia, http://en.wikipedia.org/wiki/Data_acquisition, accessed on December 12, 2012

[2] Estrada, H. & Kim, R., 1997, "Application of data acquisition systems in the undergraduate laboratories of mechanical engineering and engineering science," Proceedings of the 27th Annual Frontiers in Education Conference, November 5, 1997, Pittsburgh, Pennsylvania, USA, 454-460.

[3] Yáñez, J., Quintana, D., Quintáns C., Fariña, J. & Rodríguez-Andina, J., 2005, "FPGA-based system for the education in data acquisition and signal generation," 31st Annual Conference of IEEE, November 6-10, 2005.

[4] Costas-Pérez, L., Lago, D. & Fariña, J., 2008, "Optimization of an industrial sensor and data acquisition laboratory through time sharing and remote access," IEEE Transactions on Industrial Electronics, 55(6), 2396-2404.

[5] Blais, F., 2004, "Review of 20 years of range sensor development," Journal of Electronic Imaging, 13(1), 231-243.

[6] Karayev, S., Jia, Y., Barron, J., Fritz, M., Saenko, K. & Darrell, T., 2011, "A category-level 3-D object dataset: putting the Kinect to work," IEEE International Conference on Computer Vision Workshops, November 6-13, 2011, Barcelona, Spain, pp. 1167-1174.

[7] Kinect Sensor, http://msdn.microsoft.com/en-us/library/hh438998.aspx, accessed on December 16, 2012.

[8] Information on Kinect from Wikipedia, http://en.wikipedia.org/wiki/Kinect, accessed on December 16, 2012

[9] Kepski, M., Kwolek, B. & Austvoll, I., 2012, "Fuzzy inference-based reliable fall detection using Kinect and accelerometer," Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, Vol. 7267, Springer Berlin Heidelberg, 266-273.

[10] Chang, Y., Chen, S. & Huang, J., 2011, "A Kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities," Research in Developmental Disabilities, 32(6), 2566-2570.

[11] Izadi, S., Newcombe, R., Kim, D., Hilliges, O., Molyneaux, D., Hodges, S., Kohli, P., Shotton, J., Davison, A. & Fitzgibbon, A., 2011, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," Proceeding of the 24th Annual ACM Symposium on User Interface Software and Technology, October 16-19, 2011, Santa Barbara, California, USA, 559-568.

[12] Ren, Z., Yuan, J. & Zhang, Z., 2011, "Robust hand gesture recognition based on finger-earth movers distance with a commodity depth camera," Proceeding of the 19th ACM International Conference on Multimedia, November 28 - December 1, 2011, Scottsdale, Arizona, USA, 1093-1096.

[13] Khoshelham, K. & Elberink, S., 2012, "Accuracy and resolution of Kinect depth data for indoor mapping applications," Sensor, 12(12), 1437-1454.

[14] Khoshelham, K., 2011, "Accuracy analysis of Kinect depth data," ISPRS Workshop on Laser Scanning, Calgary, Canada, August 29-31, 2011.

[15] Livingston, M., Sebastian, J., Ai, Z. & Decker, J., 2012, "Performance measurements for the Microsoft Kinect skeleton," Proceedings of the 2012 IEEE Virtual Reality Conference, March 4-8, 2012, Orange County, California, USA, 119-120.

[16] Kinect sensor document from ROS.org, http://www.ros.org/wiki/openni_kinect/kinect_accuracy, accessed on December 16, 2012.

[17] Lucas, B. & Kanade, T., 1981, "An iterative image registration technique with an application to stereo vision," Proceedings of the 1981 DARPA Imaging Understanding Workshop, 121-130.

[18] Horn, B. & Schunck, B., 1981, "Determining optical flow," Artificial Intelligence 17(1-3), 185-203.
[19] Bradski B. & Kaehler, A., 2008, "Learning OpenCV: computer vision with the OpenCV library," O'Reilly, Cambridge, Massachusetts.
[20] Definition of 'optical flow' in Wikipedia, http://en.wikipedia.org/wiki/Optical_flow, accessed on December 16, 2012.
[21] Definition of 'image noise' from Wikipedia, http://en.wikipedia.org/wiki/Image_noise, accessed on December 16, 2012.
[22] Bruhn, A., Weickert, J. & Schnörr, C., 2005, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," International Journal of Computer Vision, 61(3), 211-231.
[23] Webb, J. & Ashley, J., 2012, Beginning Kinect programming with the Microsoft Kinect SDK, 1st Edition, Apress, New York.
[24] Lai, K., Bo, L., Ren, X. & Fox. D., 2011, "A large-scale hierarchical multiview RGB-D object dataset," Proceedings of the IEEE International Conference on Robotics & Automation, May 9-13, 2011, Seattle, Washington, USA, 1817-1824.
[25] Gould, S., Baumstarck, P., Quigley, M., Ng, A. & Koller. D., 2008, "Integrating visual and range data for robotic object detection," ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications, Marseille, France.
[26] Belongie, S., Malik, J. & Puzicha, J., 2002, "Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4), 509-522.
[27] Bentley, J., 1980, "Multidimensional divide-and-conquer," Communication of the ACM, 23(4), 214-229.
[28] Schiele, B. & Crowley, J., 2000, "Recognition without correspondence using multidimensional receptive field histograms," International Journal of Computer Vision, 36(1), 31-50.
[29] Lamdan, Y. & Wolfson, H., 1988, "Geometric hashing: a general and efficient model-based recognition scheme," Second International Conference on Computer Vision, December 5-8, 1988, Tarpon Springs, Florida, USA, 238-249.
[30] Tumkor, S., Zhang, Z., Zhang, M., Chang, Y., Esche, S. & Chassapis, C., 2012, "Integration of a real-time remote experiment into a multi-player game laboratory environment," Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'12, Houston, Texas, USA, November 9-15, 2012.
[31] Corter, J., Esche, S., Chassapis, C., Ma, J. & Nickerson, J., 2011, "Process and learning outcomes from remotely operated, simulated, and hands-on student laboratories," Computers and Education, 57(3), 2054-2067.