BEST PAPER WINNER

A Microsoft Kinect-based Gesture System to Enhance 3D Visual Communication in Virtual Collaboration

Author

Dr. Yi-hsiang Chang, Illinois State University, Normal, IL
Ms. Narda Hamilton, University of North Dakota, Grand Forks, ND
Dr. Dave Yearwood, University of North Dakota, Grand Forks, ND

Abstract

Effective synchronous internet-based collaboration is essential for companies to succeed in today's globalized business environment. However, the function of existing computing tools for 3D object manipulation in collaborative product development sessions is still limited: In addition to the problem that the 3D object is presented via two-dimensional screens, the user interface to manipulate these objects is not intuitive for novices, and the presentation is usually one way instead of two-way, simultaneous interaction. To address these concerns, we present a low-cost virtual collaboration environment built upon the Microsoft Kinect system and driven by a gesture-based interface. The rationale of the system design and results from initial tests will be presented. In addition, limitations related to vision-based cameras for 3D reconstruction will be reported, and the paper will conclude by identifying several future research directions.

Introduction

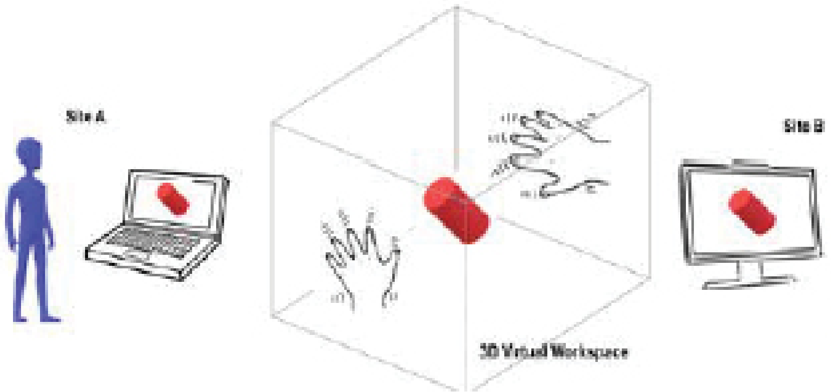
With the breakthrough of internet-based computation technology in the last decade, the "follow-the-sun" business model (Carmel, Dubinsky, & Espinosa, 2009) for the extended enterprise becomes much easier to realize. Given that more product research and development are done via the collaboration of a virtual team across different time zones or locations, the practice of "collaborative product commerce" (Bardhan, 2007; Hung, Chang, Yen, Kang, & Kuo, 2011) is no longer solely practiced by major corporations in Aerospace or Automotive industries. The result is that small to medium size enterprises can also take advantages of modern communication tools to form strategic alliances and enter new markets.

For example, company X located in Minneapolis, USA, might have won the contract from an European Agriculture Machinery company to design the electrical control panel for its next generation tracker. After evaluating the digital mockup of the tracker's panel sent by its client, company X found that a switch would not fit into the pre-specified hole. With the help of modern computing tools, company X could hold a synchronous conference call with its client in Europe to discuss the possibility of moving the hole to a different location. The participants in the conference might evaluate the Engineering Change Request from company X's engineers, determine the specification of the switch, its geometry and location on the panel, and verify if the proposed change would affect the ergonomic criteria before a final decision could be made.

Issues of Existing Technology

Nevertheless, there has not been much progress made in terms of the user interface for the type of collaboration mentioned, due in part to creative approaches, and possible hardware and software limitation. The majority of technical breakthrough made so far is mostly at the backend: Broader bandwidth, higher definition of video and audio, greater speed of streaming, etc.. But the user interface for synchronous internet-based collaboration remains pretty much the same as it was twenty years ago. The presentation of information, including that of three-dimensional objects, is mainly done through flat images or video, and controlled through two-dimensional devices such as conventional mice or touch screens (Moscovich & Hughes, 2008). The spatial manipulation of three-dimensional objects, especially rotation in the 3D space, is still done by a triad (Nielson & Olsen Jr, 1987). While this Cartesian coordinate-like triad allows the user to translate or rotate the object along the X, Y, or Z axes, it is not intuitive to untrained participants. Last but not the least, people at the listening end in a conference call most likely have no way to interact with the content until the control (the sharing of mice, whiteboard, software, or the presenter's computer desktop) is passed to them from the other end (Rekimoto, 1998; Armstrong et al., 2005). And further, the lack of spontaneous interaction could also impact real-time contributions that might be vital in collaborative exchanges.

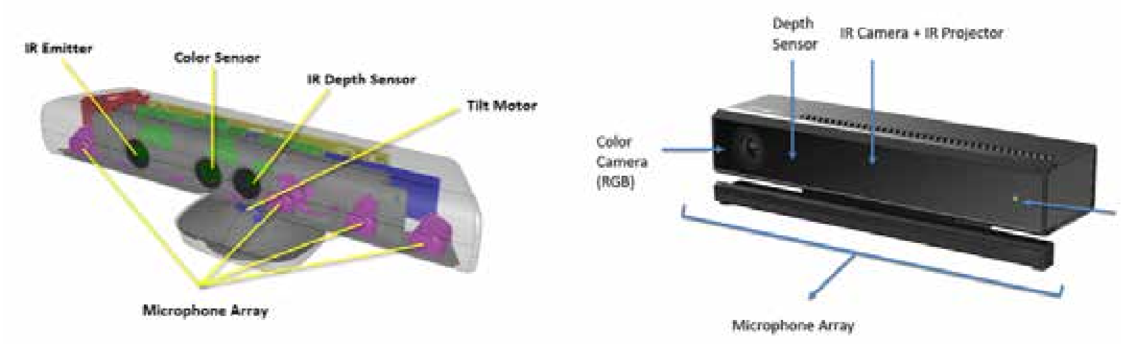
Figure 1. A Microsoft Kinect-based 3D Virtual Workspace for Synchronous Collaboration



In this paper we report an ongoing research project on a gesture-based, "natural interface" (Figure 1) designed to enhance the user experience of object manipulation in 3D space during the collaborative conference sessions. Through its ability to track human's skeletal movement ("Developing with Kinect," n.d.), individuals can manipulate objects with their bare hands in this proposed Microsoft Kinect-based virtual collaboration environment instead of using additional mice or touch devices. Moreover, the "hands" of participants visible in the virtual environment also can serve as the visual aid to "highlight" specific features on an object, or "direct" the audience's attention to the focal point at the time. Illustrated in Figure 1, the participants of the synchronous collaboration sessions at different sites can use their hands to interact with the object of discussion directly, besides communicating through conventional video or audio conference tools.

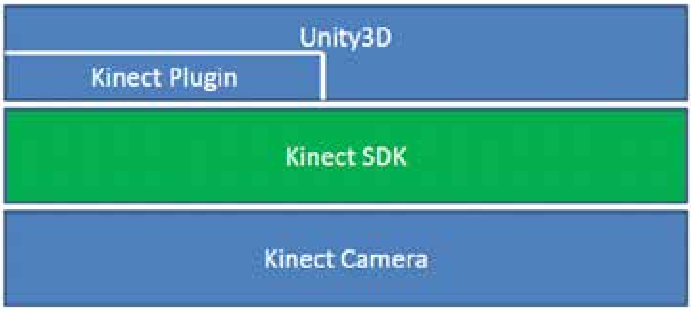
System Design Consideration for 3D Object Manipulation

Researchers (Gallo, De Pietro, Coronato, & Marra, 2008; Moeslund, Störring, & Granum, 2001) have reported their work of natural interface design for spatial navigation in virtual environments. In these studies, infrared cameras were used to track the motion of a stylus or a Nintendo Wii wand, to "point" to a specific direction or an object for further manipulation. To avoid the passing of control during the collaboration session, the vision-based camera technology used in the Microsoft Kinect system was selected to eliminate the need of additional hardware.



As seen in Figure 2, both the Kinect V1 camera and V2 camera utilize one RGB camera and one IR depth camera. The spatial coordinates of joints in individuals' skeletal were then calculated by comparing images from both cameras. While some researchers utilized the movement of an individual's body members to control digital or physical objects (Gallo, Placitelli, & Ciampi, 2011; Sanna, Lamberti, Paravati, & Manuri, 2013), others used the skeletal data to study human kinematics (Dutta, 2012; Gabel, Gilad-Bachrach, Renshaw, & Schuster, 2012). Because of the broader bandwidth of its USB 3.0 interface (5gbps vs. 60mbps of USB 2.0), the Kinect V2 camera can collect a larger amount of spatial data to describe more detailed movements. The tracking algorithm in the Kinect V2 system, along with its higher resolution cameras (1920x1080 vs. 640x480 for RGB camera, and 512x424 vs. 320x240 for depth camera), provide additional joint information to help recognize precise skeletal movement such as hands. With four joints (thumb, wrist, tip, and palm, vs. wrist and tip only in V1) identified for each hand, more sophisticated gestures such as "hand close", "hand open", "hand movement", and "wrist rotation" can be designed for 3D object manipulation, e.g. grab, release, translation, and rotation respectively. The number of users could be tracked by the camera is also triple (six total vs. two in V1), which allows more participants at one site to interact simultaneously.

Figure 3. The system architecture of the proposed virtual collaboration environment



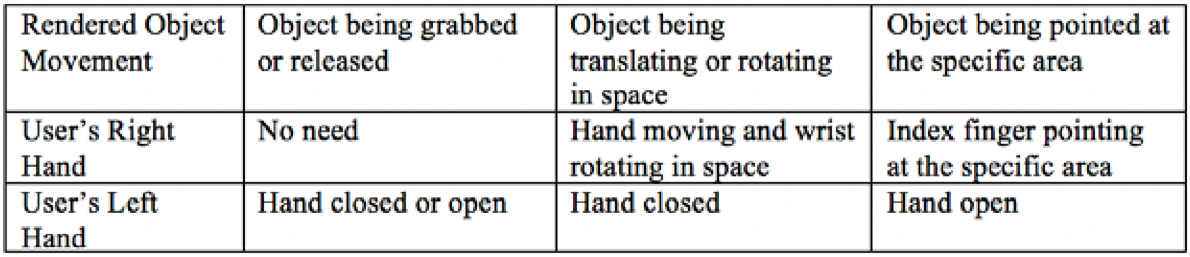
The virtual collaboration environment reported in this paper was built and tested. Figure 3 depicts the architecture of the proposed system: In addition to the Microsoft Kinect V2 camera and Kinect V2.0 SDK, a commercially available API, Unity v5.3.4 Personal, was used for rendering. The software development environment was Microsoft Visual Studio 2013, and the language used was C#. The computation platform was a Lenovo W520 mobile workstation, equipped with an Intel i7-2920XM processor, 16GB memory, and an Nvidia Quadro 2000M video card with 2GB Ram which supported both Open GL and Microsoft DX 11. The operating system was Microsoft Windows 8.1 installed on a Solid State Drive.

Further examination, to design the gesture system, could involve behavior observation of conference participants who will be able to use their hand gestures to interact with the object of discussion without holding additional hardware. Seeing how various individuals in a product design brainstorming session interact with the physical prototype could provide useful information regarding the current implementation of the proposed system around three sets of gesture, namely "object grab/release;" "object translation/rotation;" and "object pointing".

Results of Current Implementation

In an operational scenario, once the user hands engage with the Kinect V2 camera by entering its zone of detection, the spatial coordinates of the user's hand joints were used to create a "right hand" on the screen, which the user could use to grab the object, move it in the space, turn it around, release the object, and point at specific areas. Currently the user's left hand is used for grab and release the object, while the right hand is used to control the object's spatial location and orientation. Table 1 listed the existing combination of gestures in order to manipulate the object in the virtual workspace.

Table 1. The object's spatial manipulation and corresponding hand gestures



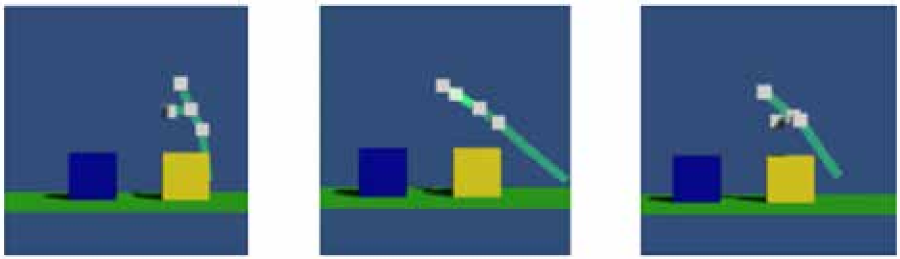
Rendered Object Movement	Object being grabbed or released	Object being translating or rotating in space	Object being pointed at the specific area
User's Right Hand	No need	Hand moving and wrist rotating in space	Index finger pointing at the specific area
User's Left Hand	Hand closed or open	Hand closed	Hand open

There were also several things to consider to provide a better user experience. Besides the color, material, and texture rendition of the object, the rendering engine Unity provided features such as gravity and collision to simulate the physical world behavior. These features could allow users to have a more "natural" or "intuitive" experience, since the virtual object behaved similarly as in their real life experiences. However, these physically based rendering features also challenged the computing hardware used. A test of the gesture system in real time revealed some unanticipated results, namely a lot of lagging or jittering as more features were turned on. These results were possibly due to a need for more computing power, and optimized gesture detection algorithms are highly recommended to alleviate these problems.

Limitations of Vision-based Cameras

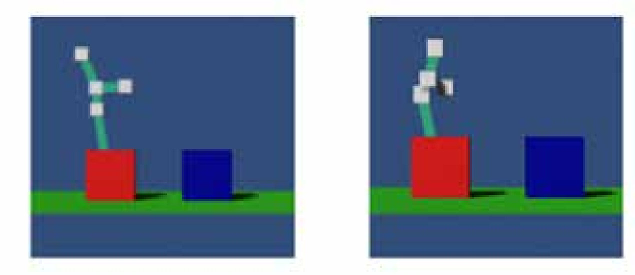
While the proposed system appears to be promising, there were some limitations of vision-based camera such as Microsoft Kinect's sensors. Of particular significance is that certain orientations of the hands could cause miscalculation, as illustrated in Figure 4. Whenever the index finger of the right hand turned to an orientation, the locations of four hand joints (gray squares lined with green lines) were not separated enough to determine their spatial location, the virtual hand could suddenly turn to the opposite direction. Similar situations also occurred in most of the vision-based camera systems. For example, the IR-based Leap Motion controller, while it recognized all ten figures of the human hand. Such a glitch could confuse the users and might cause unnecessary miscommunication.

Figure 4. Problematic orientations (from left to right): Pointing forward, sideway and backward



A similar situation could be observed with the hand open-close gesture, as shown in Figure 5. The skeletal in the left screen shot is an open left hand facing the camera, with the thumb pointing to the right, while the skeletal in the right screen shot is a closed left hand turned with the palm's external edge facing the camera. The current system's algorithm would consider both scenarios the same, causing the user to accidentally pick up the object or drop it unintentionally. While the issues discussed in Figures 4 and 5 could be addressed by an additional camera in a different angle, the extra computing power needed would be drastic.

Figure 5. Problematic orientations (from left to right): Open left hand vs. closed left hand.



There is yet another limitation of the vision-based camera system; when more than one user were engaged in the virtual space through the same Kinect camera, some user's virtual hand might engage the object when it should not. This is due to how the camera captures every hand's physical location and converts it into the virtual space. One user's hand might be "in front of" the object, while the other's hand was "behind" the object. In real life the hand behind would not be able to touch the object. However, the current implementation would still allow the hand to interact with the object. This would not happen in Kinect's application in the Xbox game console, as participants were either engaging in different items, or the joints' spatial coordinates were utilized to determine the specific gesture patterns instead.

Conclusion and Future Research

In conclusion, we presented in this paper the need justification, design consideration, and issues of a gestured-based system for 3D object manipulation in the virtual collaboration environment based on the Microsoft Kinect V2 system. The advanced vision-based camera enabled the user to manipulate the object in the virtual workspace without additional hardware. While the functionality of the current implementation is limited, it presented an effort toward a natural interface that echoes our experience in the physical world. As more and more companies engage in the development of vision-based sensory systems, a gesture system could drastically reduce the effort of learning and utilization, especially for the upcoming virtual reality, augmented reality, and mixed reality applications.

To better understand whether the proposed gesture-driven natural interface is able to enhance the user experience, further studies will be necessary. Firstly, an experimental study is needed to evaluate the user's task performance between using conventional mice and through the Kinect system for object manipulation. The intuitiveness of the system can also be determined by the user's time needed to "master" the interface and the results of a self-reported survey. Secondly, a carefully designed study is essential to understand whether the addition of visible hand gestures significantly improve the communication between users at different locations and reduce the time to complete assigned tasks. Last but not the least, while Microsoft claimed that the Kinect V2 system was able to support up to six users simultaneously, it is unknown whether the proposed system will be able to handle the amount of spatial data and provide meaningful yet simple collaboration sessions when handling the "control" of the object among more than one set of hands. Further development of the system might be necessary.

Reference

Armstrong, V., Barnes, S., Sutherland, R., Curran, S., Mills, S., & Thompson, I. (2005). Collaborative research methodology for investigating teaching and learning: the use of interactive whiteboard technology. Educational Review, 57(4), 457–469.

Carmel, E., Dubinsky, Y., & Espinosa, A. (2009). Follow the sun software development: New perspectives, conceptual foundation, and exploratory field study. In System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on (pp. 1–9). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/4755341/

Developing with Kinect. (n.d.). Retrieved July 11, 2017, from https://developer.microsoft.com/en-us/windows/kinect/develop

Dutta, T. (2012). Evaluation of the KinectTM sensor for 3-D kinematic measurement in the workplace. Applied Ergonomics, 43(4), 645–649.

Gabel, M., Gilad-Bachrach, R., Renshaw, E., & Schuster, A. (2012). Full body gait analysis with Kinect. In Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE (pp. 1964–1967). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/6346340/

Gallo, L., De Pietro, G., Coronato, A., & Marra, I. (2008). Toward a natural interface to virtual medical imaging environments. In Proceedings of the working conference on Advanced visual interfaces (pp. 429–432). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1385651

Gallo, L., Placitelli, A. P., & Ciampi, M. (2011). Controller-free exploration of medical image data: Experiencing the Kinect. In Computer-based medical systems (CBMS), 2011 24th international symposium on (pp. 1–6). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/5999138/

Moeslund, T. B., Störring, M., & Granum, E. (2001). A natural interface to a virtual environment through computer vision-estimated pointing gestures. In International Gesture Workshop (pp. 59–63). Springer. Retrieved from http://link.springer.com/chapter/10.1007/3-540-47873-6_6

Moscovich, T., & Hughes, J. F. (2008). Indirect mappings of multi-touch input using one and two hands. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 1275–1284). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1357254

Nielson, G. M., & Olsen Jr, D. R. (1987). Direct manipulation techniques for 3D objects using 2D locator devices. In Proceedings of the 1986 workshop on Interactive 3D graphics (pp. 175–182). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=319134

Phillips, C. B., Badler, N. I., & Granieri, J. (1992). Automatic viewing control for 3D direct manipulation. In Proceedings of the 1992 symposium on Interactive 3D graphics (pp. 71–74). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=147167

Rekimoto, J. (1998). A multiple device approach for supporting whiteboard-based interactions. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 344–351). ACM Press/Addison-Wesley Publishing Co. Retrieved from http://dl.acm.org/citation.cfm?id=274692

Sanna, A., Lamberti, F., Paravati, G., & Manuri, F. (2013). A Kinect-based natural interface for quadrotor control. Entertainment Computing, 4(3), 179–186.