Sketches on natural interactions with virtual scenes

Andréa Lins e Lins Souza, Ricardo Marroquim
Programa de Engenharia de Sistemas e Computação
UFRJ - Universidade Federal do Rio de Janeiro
Email: {andrealins, marroquim}@cos.ufrj.br

Luiz Velho
IMPA - Instituto de Matemática Pura e Aplicada
Email: lvelho@impa.br

***Abstract*—In this work we present some early ideas about natural 3D interaction using augmented reality. Our main goal is to explore approaches to allow interacting with 3D objects in an intuitive and clear way, and freeing the user from interacting with 3D scenes using cumbersome and many times counterintuitive 2D paradigms. We sketch our initial thoughts as well as some early implementations using recent low-cost hardware, such as the *Microsoft Kinect* and the *Leap Motion*. We present ideas that are device-independent, since we are more interested in the interaction paradigm than the actual technological devices. In addition, even though we are still on an early implementation stage, we describe our first prototypes for the physical realization of these ideas.**

***Keywords*-3D interaction; gesture-based manipulation;**

I. Introduction

Human interaction with virtual scenes has greatly evolve during the last years. One of the first reported real-virtual interactions was during the sixties [11]. Hitherto, large steps were taken in advancing this field, and in particular the entertainment industry has recently walked side-by-side with this progress by introducing new low-cost devices to allow for full or partial immersion into the virtual world.

According to Bowman *et al.* ([12], [9]), interaction techniques with 3D virtual worlds may be divided in three main categories: 1) navigation; 2) selection and manipulation; 3) system control. The first allows the user to move around the virtual world. The second allows for modifications of the virtual world, as well as virtual objects manipulation. Finally, system control refers to task of changing interaction modes.

In this context, our proposal is to explore new natural interactions ways with 3D scenes. We start by slowly moving away from the 2D manipulation and viewing paradigm, to a fully immersive system. In this work we describe a few experiments we are building to test some new interactions ideas using recent low-cost 3D devices.

A. Related work

Nowadays there are a few devices that work with 3D gestures of voice commands. A well known example is the *Microsoft Kinect*. This type of interfaces frees the user from physical contact with the device. Even though many devices were born for a specific use, such as gaming for example, through available SDKs it is possible to access its raw data and use it for other purposes. The Kinect is now a common device in many research areas, such as Computer Vision and HCI [3].

A few recent works use the *Kinect* device to create interaction systems with virtual worlds. We briefly describe only a few.

Held *et al.* [5] present a system capable of authoring 3D animations by tracking physical objects, such as toys or puppets. Their virtual objects are, however, replicas of the original ones. Thus, a pre-processing step is necessary where the object is digitizes using the Kinect. Afterwards, the real object is tracked during the creation of the animation using an *ICP* algorithm.

Another 3D animation approach is presented by Gupta *et al.* ([7]). Their *MotionMontage* system also uses a Kinect to record object movement, and then combines a series of video takes to produce the animation. Furthermore, it allows for multiple objects moving at the same time.

Barnes *et al.* [6] created a system based on 2D videos to produce animations from paper characters. As an initial step, a database is created with the characters images so they can be rendered in real-time when the system is tracking the paper pieces. The system is limited to 2D transformations, i.e., 2D rotations, translations, and a scale (zoom), and some special effects, such as illumination changes.

II. Interaction sketches

In the next sections we describe some 3D interaction ideas that are currently being implemented and tested.

A. Object and camera manipulation

Our first experiment is a virtual manipulation of objects and camera control using physical proxies. The idea is to use the minimum amount of hardware devices, in this case only a Kinect, and compensate for the lack of immersion with physical proxies, i.e., physical objects that are used to manipulate and control the virtual scenes.

1) Object manipulation: Even though manipulating a 3D object using 2D interfaces has become a common task, it is usually far from intuitive and efficient. Placing the object at a required position and orientation using handlers and paradigms such as arcballs, is still far from a natural 3D interaction. On the other hand, full virtual interactions lacks the tactile sensations. We advocate the use of simple 3D proxies to allow for a intuitive object manipulation.

Using a simple 3D box, for example, its rotation and position can be tracked and the transformations applied to a virtual object. Since physical manipulation is a natural task, one can easily manipulate a simple object while still focusing on the 2D display output.

Our first prototype to test this interaction idea, uses a Kinect to track a box being manipulated. To retrieve the transformations, we mark a few points on the box and track them using an ICP algorithm. Actually, since we know the box's dimensions beforehand, we only need to mark four points, and the remaining are automatically computed. Usually we use at least fourteen points: the eight corners plus the center of each face.

At each frame, the ICP algorithm compares our marked point set against the point cloud returned by the Kinect, searching for correspondences. The algorithm works under the assumption of a rigid transformation (rotation and translation) between the original set and their correspondences, which is exactly what we need to manipulate our virtual object. Figure 1 illustrates the idea with three different moments of an interaction.

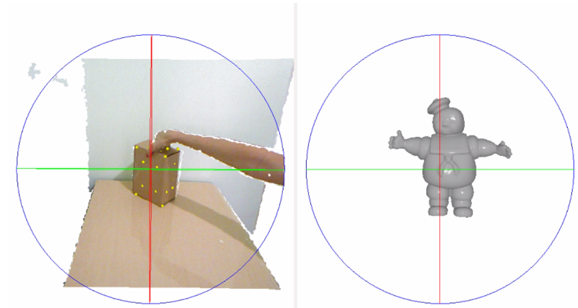
In addition, we employed two simple modifications to the ICP algorithm to render it more efficient. Since we expect smooth movements and to accelerate the algorithm, we only search for each point's correspondence around a small window in the depth image. Furthermore, normals are also computed for the initial set and transformed accordingly after each frame, and used to discard non-visible points during the ICP iterations.

Note that, different from some similar works, we use a physical proxy, so there is no need to scan the real object. In fact, there is no need to exist the physical object at all, it could be any 3D model. Actually, MotionMontage also does not need a physical replica, but has very limited camera control and interaction, since their main goal is to produce simple animations. We will expand on this idea in the following section.

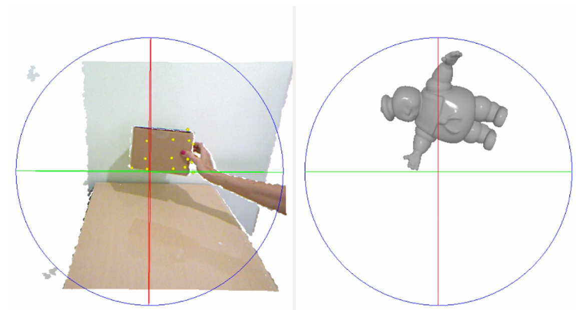
2) Camera manipulation: In our first experiment, the camera (or viewpoint) is fixed. To allow for more freedom during interaction, it is necessary to introduce intuitive ways to manipulate the virtual camera. In this simple interaction scenario, tracking the user's head or eyes is not useful, since he/she will be looking at the 2D display, and not the physical proxy.

Continuing with the idea that hand movement is intuitive enough to be carried out without looking or actually making a mental effort, the virtual camera can be manipulated using a finger.

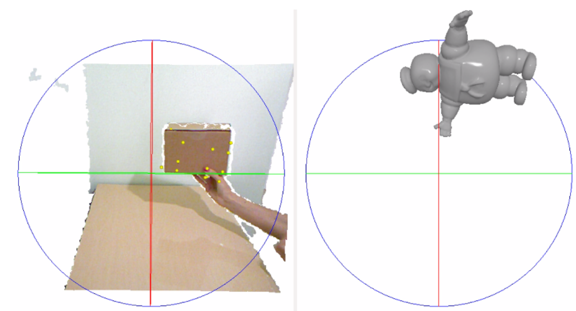
To track the finger we use 2D image processing techniques. To make the interaction more robust, we use a colored thimble to facilitate the finger tip tracking. We employ a series of 2D filters, starting with a color segmentation (using the thimble's color), and apply a CCL algorithm to find the centroid. See Figure 2 for an illustration of the interaction. For efficiency sake, all filtering operations are implemented using GPU programming. Note that at this point, we have not yet precisely retrieved the camera's direction. We will also incorporate depth information to make the tracker more robust in the future.



(a) Initial position



(b) Rotation



(c) Translation

Fig. 1. Object manipulation using a physical proxy (cardboard box). The ICP algorithm tracks the box and the transformation is applied to the virtual character on the right. Note that due to an early stage implementation, the ICP still has some lag, so the yellow points are not exactly over the box's corners at the frame capture moment.

B. Virtual finger puppeteering

Once the camera tracker is fully implemented, we can sketch even more interesting interactions, such as virtual puppeteering. A very common way to describe an action scene, is by using two fingers on a table, to represent the movement of a character, as shown in Figure 3. One can easily describe a character walking, running, jumping and even simulate gait, ex., to describe a drunk movement, someone tiptoeing, or even to describe the famous moon walk. The finger puppeteering can be translated to virtual characters for immediate 3D animation authoring. The essence of the idea is a minimalist motion capture system, to quickly animate characters, or describe movements, such as animations, a dance or a performance.

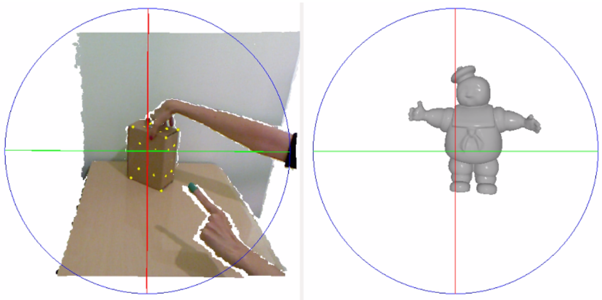


Fig. 2. Camera movement by tracking a finger. The camera direction is still not very precise in the version of the implementation.

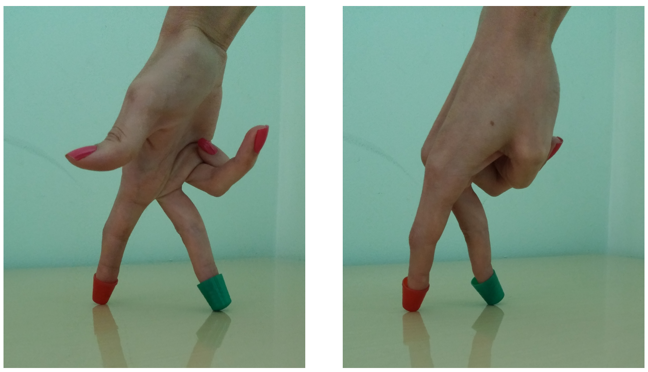


Fig. 3. Finger puppeteering. The thimbles capture the feet movement of a walk. By tracking more features, such as the finger joints, knees could be represented as well.

C. Finger tracking with Leap Motion

We recently started making new experiments using the Leap Motion for hand and finger tracking. Even though it is a much more precise and efficient device for hand tracking than the Kinect, the approach is somewhat different. With the Kinect we are interested in a more tactile-based approach when manipulating generic 3D objects, such as the one described in Section II-A1. On the other hand, for camera or general finger movement, the Leap Motion becomes a valuable asset, since it can provide precise finger position and direction.

D. Immersive scenes

At the moment, we are also starting to move on to more immersive scenes. The idea is to move away from the 2D display and let the user interact directly in 3D, in a more What-You-See-Is-What-You-Get approach. Of course, this also moves away from tactile-based ideas, since what one sees is exactly what it is being manipulated, and not a physical proxy. Handling and manipulating virtual objects as one would do in the physical world can be cumbersome without physical feedback, and generic haptic feedback systems are still not available.

To overcome the lack of physical contact, we are turning to gesture based interactions. For example, it would be difficult to exactly grab a virtual object, but a simple *grab* gesture would many times be simpler and more efficient. As an example, we will describe a first sketch on a virtual immersive animation and camera control system.

The scene would be projected on a table, and the user would move around it using 3D glasses for augmented reality, ie., projecting the virtual scenario on the physical table. A Leap Motion could be fixed on head mount, to move around with the user while tracking the hands. A virtual character could be selected with a simple gestures, and finger pupperteering would indicate the actions the character would employ during the next time frame. Another character could be selected and his movements also described during the same time frame. Finally, the animation of the characters could be played back while the camera path would be described using hand gestures for composing the viewing frame, for example, as film directors usually frame shots by forming a rectangle with their fingers in front of the eyes to simulate the viewfinder (Figure 4).

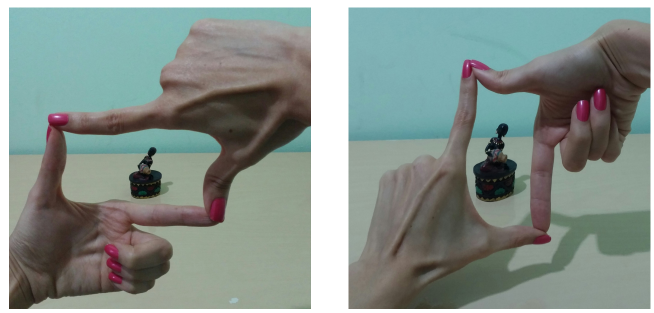


Fig. 4. Finger framing. Camera tracking can be activated with a framing gesture. A head mounted Leap Motion is able to capture the hands in this position.

For more generic controls, such as switching to play back mode, or setting the desired time interval for the animation, a separate hand tracking system could be fixed at place around the table.

III. Conclusion and Discussion

In this paper, we present or work in progress about 3D interactions with virtual scenes and objects. We are interested in researching new paradigms that can be implemented independently of the devices. We sketched some interaction ideas, such as physical proxies and finger pupperteering, and presented an idea for an animation and camera control authoring system. Even though we are still implementing our first prototypes, we also lay out some proposals for using recent low-cost devices to realize our ideas.

Since this work is on its early stages, we have still have a long road before fully realizing our proposed ideas, however, we genuinely believe that good ideas will outlast the hardware devices, and that 3D interactions will become a common reality in a near future, so the search for new paradigms that convey natural interactions system is of great importance.

Of course, the more devices we add to help achieving a complete interaction, the more specific becomes the system, and consequently, it is capable of reaching a smaller audience.

From the simple object manipulation, where only an RGBD camera is necessary, there is a big leap to a system that requires a projector, a Leap Motion, and 3D glasses, and of course, a good setup and calibration steps. But again, we believe that with good interaction ideas, the necessary devices will be certainly available soon enough.

ACKNOWLEDGMENT

The authors would like to thank Djalma Lucio (VISGRAF Lab - IMPA) for his valuable support during this work. The authors also thank the two anonymous reviewers for their comments.

REFERENCES

[1] S. M. O'Connell, "A gpu implementation of connected component labeling," *Illinois State University*, 2009.

[2] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," Shanghai, China, May 9-13 2011.

[3] L. Cruz, D. Lucio, and L. Velho, "Kinect and rgbd images: Challenges and applications," in *Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, ser. SIBGRAPI-T '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 36–49. [Online]. Available: http://dx.doi.org/10.1109/SIBGRAPI-T.2012.13

[4] M. Bailey and S. Cunningham, *Graphics Shaders: Theory and Practice*. Natick, MA, USA: A. K. Peters, Ltd., 2009.

[5] R. Held, A. Gupta, B. Curless, and M. Agrawala, "3d puppetry: A kinect-based interface for 3d animation," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12. New York, NY, USA: ACM, 2012, pp. 423–434. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380170

[6] C. Barnes, D. E. Jacobs, J. Sanders, D. B. Goldman, S. Rusinkiewicz, A. Finkelstein, and M. Agrawala, "Video puppetry: A performative interface for cutout animation," in *ACM SIGGRAPH Asia 2008 Papers*, ser. SIGGRAPH Asia '08. New York, NY, USA: ACM, 2008, pp. 124:1–124:9. [Online]. Available: http://doi.acm.org/10.1145/1457515.1409077

[7] A. Gupta, M. Agrawala, B. Curless, and M. Cohen, "Motionmontage: A system to annotate and combine motion takes for 3d animations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2017–2026. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557218

[8] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011.

[9] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[10] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992. [Online]. Available: http://dx.doi.org/10.1109/34.121791

[11] I. E. Sutherland, "The ultimate display," in *Proceedings of the IFIP Congress*, 1965, pp. 506–508.

[12] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, "An introduction to 3-d user interface design," *Presence: Teleoper. Virtual Environ.*, vol. 10, no. 1, pp. 96–108, Feb. 2001. [Online]. Available: http://dx.doi.org/10.1162/105474601750182342