A Leap-supported, hybrid AR interface approach

Holger Regenbrecht **Jonny Collins** **Simon Hoermann**
University of Otago, Information Science, P.O. Box 56, 9054 Dunedin, New Zealand
holger.regenbrecht@otago.ac.nz

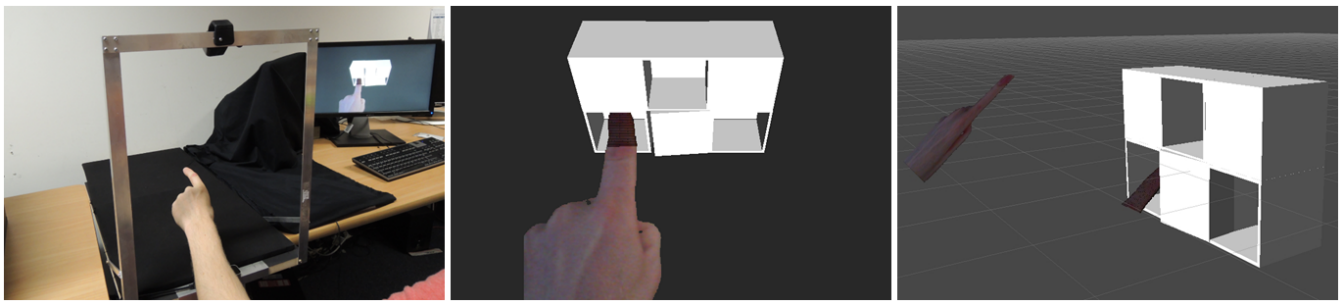

Figure 1. VoxelAR: (left) Prototype system in use, (center) Screenshot of user's perspective, (right) 3D perspective screenshot (non-users view) with partial 2D (video) and 3D (voxel) rendering and interaction.

ABSTRACT

We present a novel interface approach which combines 2D video-based AR with a partial voxel model allowing for more convincing interactions with 3D objects and worlds. It enables users in a hand-controlled interface (a) to interact with a virtual environment (VE) and at the same time (b) to allow for correct mutual occlusions between interacting fingers and the VE. A Leap motion controller is used to track the users' fingers and a webcam overlay allows for an augmented view.

Our "VoxelAR" concept can be applied in modified ways to any video-see through AR system - we demonstrate our approach in a physical rehabilitation application scenario. Our prototype implementation and our work-in-progress findings are presented.

Author Keywords

Augmented Reality, Occlusion, Interaction, Physical and Motor Rehabilitation

ACM Classification Keywords

H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities;

INTRODUCTION

Many Augmented Reality applications allow or would like to allow for a hand-based interaction with the augmented content. Pointing at objects, grabbing and moving items or gestural interaction to control the environment or the system state are amongst the obvious examples.

We are developing application prototypes for physical rehabilitation, in particular for stroke patients. Often, these users have difficulties moving and controlling their upper limbs, including hand and finger movements. Repetitive exercise is normally used to mitigate those movement deficits; however it is often difficult to motivate to exercise frequently and to a desired quality. To overcome this we incorporate casual game components and changing exercise content in our systems [1,2].

In addition to physical exercise immediate feedback about progress might lead to neuroplastic change and with this to improvements in motor function [3]. If we can fool the user's perception of the progress of the therapies, i.e. we positively influence his/her progress perception, we can be even more effective and efficient in the treatment. In particular, a convincing visual feedback about progress is important.

Because of the nature of our scenario we try to avoid instrumenting the user's hands or fingers with e.g. tracking components or data gloves. Tools, like pointing devices or dynamometers can be used with very mild impairments, but users with more severe conditions are unable to hold and operate them; either because the users are not strong enough, or experience pain or are fatigued too fast in their current stage of progress.

Uninstrumented hand and finger interaction with augmented environments faces two major challenges: (1) The mutual visual occlusion between virtual and real elements has to be of convincing quality [4]. In particular the correct occlusions happening between the user's fingers and the virtual objects in the augmented environment should be as correct as possible. (2) The user's fingers should be able to physically interact with virtual and real objects in an almost seamless way. Here, the ability to touch real objects is normally given (if the reality is not fully or partially diminished), a physically correct collision detection and processing with virtual objects is key.

We are presenting a solution to those two problems in the context of our physical rehabilitation scenario which can be generalized to other AR application areas which require manual interaction. We developed a hybrid computer graphics solution where parts of the user's fingers are still rendered in 2D (video see-through) and the interacting parts (mainly upper part of the fingers) are converted into voxels. We are using a single camera - the depth estimation of the finger positions (and orientations) is provided by a Leap motion controller (www.leapmotion.com). We can show that both major problems (mutual occlusion and physical interaction) can be addressed in a convincing manner.

Our work contributes to the body of knowledge in the field of novel AR interfaces, which are needed in particular with stationary and mobile systems focusing on manual interaction where mouse and keyboard interfaces are not suitable.

In the remainder of this paper we discuss relevant related work in the field, present our conceptual approach, describe the implementation of our prototype and present our early findings. We conclude with limitations and future work.

RELATED WORK

The visual aspects of Augmented Reality allow for a classification into (a) optical see-through (OST), (b) video see-through (VST) and (c) spatial AR (SAR) systems. If there is no data available on the true geometry of the real scene the problem of occlusions between real and virtual elements is usually addressed with compromise solutions. In particular for the visualization of hands and fingers (unknown geometry,

OZCHI'13, November 25–29, 2013, Adelaide, SA, Australia.

unknown positions and poses) OST and SAR systems often use a partial transparent rendering approach - to varying degrees the opacity of the virtual content is controlled (or given by the device) to achieve a ghosting effect (e.g. [5]). This allows for the visualization of both the virtual and the real content, but it is far away from expected occlusions happening in reality.

For known geometries (normally not the hands or fingers) sometimes so called "phantom models" are used: real objects are taken into account while rendering but they are not actually rendered, just their occluding characteristics are used. In VST systems this is normally done by filling the graphics pipeline's Z buffer apriori (e.g. [6])

Also, in VST systems, if the real world geometries are unknown, the way the rendering happens solves the problem at hand inherently: virtual objects always occlude the real world. This is useful for e.g. the superimposition of virtual landmarks over real scenery at far distances. For hand interaction this approach doesn't work: the hands are too close and would often, or sometimes be partially occluded by virtual content. Here, the opposite approach can be taken with VST. If the foreground objects (here hands) can be visually separated from the background then they can be rendered in a way that the real foreground objects always occlude the virtual content. This approach is useful if the virtual objects are out of reach of the user.

If depth sensing technology can be used, e.g. fast calculations of disparity maps from stereo cameras [7] or Kinect sensors [8], then a depth pixel cloud can be used to render occlusions correctly. This requires that the depth sensor operates from the same (or close) viewpoint as the users actual view or a scene reconstruction (outside-in) is calculated and a sensing range and spatial resolution and temporal resolution, accuracy and precision is achievable. Often, the devices used are either not operating in the required range (e.g. Kinect for egocentric view within reach) or require expensive computations (e.g. stereo pair matching with a decent resolution and speed). In most cases they are error-prone which result in visual artifacts intolerable for certain applications, like clinical interventions.

To track the pose (position and orientation) of a user's hand and fingers or tools held in the hand a number of technical approaches exist targeting different application scenarios: Data gloves in combination with 6DOF trackers attached to them can determine the rigid pose of the hand in space as well as the flexion and pose of each finger. They inherently require instrumentation and often individual calibration. Because of the impairments of our users (and the discomfort to don a glove in other application scenarios) this is not an option here. In addition, data gloves delivering an accuracy and precision required for a convincing visual AR overlay are highly expensive. PinchGloves [9] and similar devices would allow for controlling the AR environment but are not suitable for hand overlay (missing data).

Hand reconstruction from multiple views [10] can be used for correct overlay, mutual occlusion and interaction, but requires a high degree of instrumentation, calibration and computation.

Real interaction tools or props can be used to interact with virtual content but do not solve the occlusion problem between hand and fingers and virtual content. In addition they are unsuitable for our scenario because of the user's impairments.

More recently, a new technology was made available which can track a user's hand and fingers within reach (designed for the space in front of a desktop monitor) - a Leap motion controller. While the principle of the underlying technology is not made public the device functionalities exactly address what we have in mind for our system: tracking of hand pose, fingertip positions and finger poses in a spatial range fitting or rehabilitation scenario requirements. If combined with VST AR technology the Leap controller could support our two main objectives: (1) solve the mutual occlusion problem between fingers and virtual content and (2) allow for interactions with virtual objects.

We have been provided with Leap controllers as part of the Leap developer program, integrated a Leap controller into our hardware and software environment and tested its reliability and perceived accuracy and precision. While the overall quality of the device is impressive it is neither reliable (drop-outs in recognition) nor precise (variations from true poses) enough to serve as the sole underlying tracking technology. However, when combined with a visual approach the results look very promising.

We use the Leap motion controller to estimate the user's finger positions and vectors. This depth information is then combined with 2D finger pixel data to achieve the desired results.

In the following section we describe our "VoxelAR" concept which is a hybrid combination of 2D and voxel graphics and geometry which uses coarse depth estimations of fingers.

With this solution we are directly addressing perceptual augmentation issues as described by Kruijff, Swan, and Feiner [11] and significantly extending earlier work in the field [12-14] trying to overcome occlusion and interaction issues in augmented environments.

VOXELAR CONCEPT

Our application scenario affords certain requirements which can also be found in other application areas like desktop AR applications:

Affordability and Feasibility: Customizable off-the shelf (COTS) components in combination with powerful, but standard computing technology allow for a later dissemination of the prototype system in (clinical and experimental) practice. The instrumentation of the user should be kept to a minimum and the system should fit into an office or clinics environment.

Interaction Space: As shown with our earlier studies in upper limb rehabilitation, the interaction volume can be limited to a space of about 300x300x300 mm3 in front of the user (within immediate reach). The background can either be controlled (e.g. black or green cloth) or can be assumed to be more or less known. Even if we consider a dynamic background it won't change significantly like in for instance urban outdoor scenes.

Camera Pose: We are using a fixed camera position in relation to the interaction space. Hence, the external (and also internal) camera parameters are known. The user does not have to wear any tracking targets or gear nor do we have to install other tracking sensors. Even if this is a rather special AR setup, our concept can be applied to any scenario with a well computed camera pose e.g. by appropriate high fidelity tracking.

Interface Fidelity: The visual rendering of the hands and the scene including and in particular the mutual occlusions as well as the interaction quality (speed, lag, resolution) should be of a quality suitable for experimental and clinical studies involving patients. I.e. the users have to be convinced that the displayed hands are their own and that the environment has a degree of realness so that an effective treatment can happen. Earlier studies have shown that this can be achieved with COTS components and tailored software and procedures.

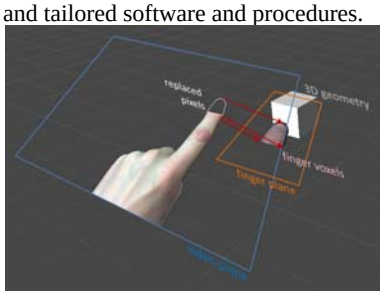

Figure 2. Non-user's 3D view with annotations.

The VoxelAR concept is based on Augmented Reflection Technology (ART) [3] and extends this concept towards hybrid rendering and interaction. While the original ART system only allows for 2D interaction and video-over-virtual visualization we provide a solution which delivers 3D interaction capabilities

and correct real/virtual visualizations. Figure 2 illustrates the principle approach.

The goal is to generate finger voxels which can be used for occlusion and interaction. We are faced with two challenges here: (1) how to determine the position, in particular depth of each voxel and (2) how to process a large number of voxels on currently available standard hardware?

The depth could be determined by computer vision (like stereo matching) approaches but is computationally intensive. Or a Kinect sensor could be used, but can't be applied in the range while maintaining a reasonably high spatial resolution. The number of voxels can be processed with high-end, parallel or distributed computing, or lower resolutions could be used.

We are tackling both challenges with (1) a just coarse depth estimation per finger (and not per pixel) using the Leap motion controller and (2) by just generating a limited number of voxels - the ones who are most important for occlusion and interaction. This number can be scaled according to required fidelity and available computing resources.

The user's camera is capturing the interaction space in front of a defined background. This camera is statically positioned in relation to the interaction space. The camera's video stream is subtracted against the background by either using a simple color and brightness threshold model per pixel or by applying a CodeBook background segmentation (OpenCV) depending on the nature of the background (very well controlled and static or less controlled and dynamic). The simple threshold model usually delivers superior background segmentation results. Because the hand is normally the only object acting in the foreground, a well-defined hand segmentation is achieved. All non-foreground pixels in the video frames are set to transparent.

If the fingers are interacting with virtual objects in the scene (3D geometry) at least the upper parts of the fingers should be turned into voxels. For reasons of computational performance only a certain, pre-defined number of voxels is generated. Each image in the foreground separated video (video plane) is scanned from top to bottom and from left to right. Because we only have foreground pixels in the image we can safely assume that all first pixels we found belong to fingers, separated by alpha pixel spaces. For each of the finger pixels found we store the pixel information (finger number, 2D position, color (RGB)). Those pixels found to belong to fingers are removed from the video image (alpha set to transparent). If the pre-defined maximum number of finger pixels (and therefore to be generated voxels) is reached the scanning stops and the next image is processed.

The resulting sequence of images is now rendered into a video plane sitting in front of the virtual scene, i.e. foreground pixels occlude the virtual scene, alpha pixels don't.

The Leap controller is positioned in our system in a way that it detects finger positions from below and is located in a fixed relation to the interaction space (and user's camera). It delivers positional data for the finger tips as well as a vector from that finger position towards the palm of the user's hand. We use this position and vector to form a plane for each finger in virtual space. This plane is invisible to the user and only serves the purpose of a depth estimation for the finger voxels. For each finger pixel identified, a virtual ray is cast from the 2D pixel position on the video plane towards the appropriate finger plane. The intersection of this ray with the finger plane determines the position of the finger voxel in space. Out of a pool of pre-defined (and pre-generated) voxels the next voxel will be translated to this new position, scaled accordingly and colored with the RGB pixel information stored during the scanning process.

Once all (available) voxels are in place mutual occlusions and physical interaction with the 3D geometry happen inherently. While the 2D finger pixels are still always in front of the 3D geometry the important interactive parts of the fingers (upper part) interacting "naturally" with the virtual world augmented to the scene.

SETUP

While our plan is to integrate VoxelAR into our existing ART system for the purpose of this study we built an extra piece of supporting hardware (see figure 1 left).

Instead of putting his or her hand inside a black box with a curtain the user is reaching through an aluminum frame. The hand is captured by a camera from above in front of a black background. The leap motion controller is firmly fixed to a defined position below. The entire frame construction is solidly built - interaction space (above black board), camera and Leap motion controller always maintain the same geometrical relation to each other.

The camera (Microsoft Lifecam HD-5000) and the Leap motion controller (0.7.7) are connected via USB 2.0 connections to a standard PC. The output is visualized on a standard monitor (1680x1050, 60Hz) sitting next to the VoxelAR frame.

The video stream of the camera is processed by an OpenCV library component built into our Unity 3D (ver 3.0) graphics scenegraph environment. We are capturing and processing images with a resolution of 640X480 and an average framerate of 30 frames per second. The Leap controller data are received and processed by an external application operating as a (localhost) socket connection client to our Unity 3D server application. A script starts the whole application including the Leap Socket client.

To test the quality of our implementation we are using a physical rehabilitation finger pointing task: The user has to point at different virtual objects with his/her index finger while some of the objects are movable and others are not. They also define different levels of reachability for the user. Here, a number of virtual cubes is used, half of them with an open front (static) and others with a closed front (figure 3).

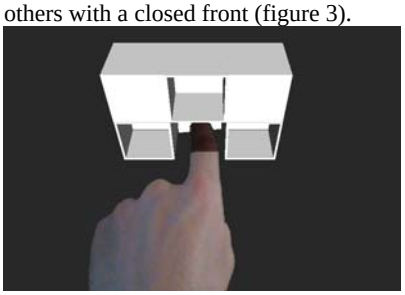

Figure 3. Pointing gesture in physical rehabilitation scenario pushing out one of the cubes.

For this pointing scenario we are using 2000 pre-generated voxels which gives us a reasonable framerate of about 15 scene updates per second. There is no noticeable latency when interacting with the virtual cubes. The rendering quality is sufficient for this scenario. Figure 5 shows the hybrid rendering of the 2D video stream with 2000 voxels. The color matching between the 2D and the 3D spaces is surprisingly accurate, however there are still some gaps between the voxels probably due to scaling errors and aspect ratio mismatches between pixels and rendered voxels.

Figure 4. Rendering on index finger with 2000 voxels.

The number of voxels is sufficient for this scenario too: only a single finger is used for the interaction with the virtual environment and allows for a "deep enough" mutual occlusion between objects and finger. In different scenarios, where more

fingers will be used the voxels are distributed over all interacting fingers.

The number of voxels, the desired occlusion quality, the achievable framerate, the number of fingers involved in the virtual objects interaction and the needed computational power have to be traded against each other. We tested different voxel numbers applied to our system and found that voxel numbers below 500 are only of limited use but delivering very high framerates while a desired number of many thousand voxels makes our system too slow. Figure 5 shows our measures of framerates for different voxel numbers.

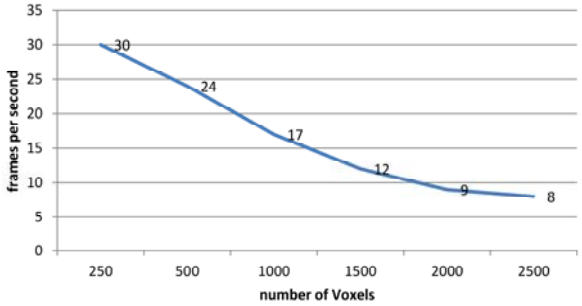

Figure 5. Frame-rate with increasing number of voxels.

CONCLUSION AND FUTURE WORK

We presented a novel concept and prototypical implementation of a Leap-supported, hybrid AR interface approach which allows for correct mutual occlusions and interactions in a finger-based interface. Even if our context and scenario is specific (physical rehabilitation) this approach can be applied to many other application scenarios. For instance, if the user's camera perspective is tracked accurately in space, the same technique can be used in head-worn or handheld display settings where manual interaction is key.

Currently the main limiting factor is the number of voxels which can be processed with interactive framerates. This might be solved by assigning more computational power, by using a more efficient scenegraph API or by optimizing the current algorithms. For instance, if very high visual quality of finger rendering is of lesser concern, only fringe pixels of the finger get assigned voxels, the rest in-between is rendered using stretched boxes using an average skin color. This can significantly lower the number of voxels needed. Also the size of the voxels might vary, either overall (coarser voxel size) or partially (fine-grained in some areas, more coarse in others). Well-known techniques in computer graphics could be adapted here (octrees, marching cubes, etc.).

With an increasing fidelity and reliability of the Leap motion controller (or other depth and finger sensing devices) larger parts of the fingers up to the whole hand might be replaced by voxels.

At the moment we are evaluating our system with healthy volunteers. In the near future we are integrating VoxelAR into our Augmented Reflection Technology system and applying it in a rehabilitation context. Additional scenarios are going to be developed, e.g. a grabbing task for post stroke treatments. We are going to report on those clinical findings in due course.

ACKNOWLEDGMENTS

We'd like to thank all people who helped to test and improve VoxelAR. Thanks to Leap Motion for letting us be a part of the motion controller developer program.

REFERENCES

[1] Regenbrecht, H., McGregor, G., Ott, C., Hoermann, S., Schubert, T., Hale, L., Hoermann, J., Dixon, B., & Franz, E. (2011). Out of reach? - A novel AR interface approach for motor rehabilitation. Proceedings of The 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2011), Oct. 26 - 29, 2011, Basel, Switzerland, 219 - 228.

[2] Hoermann, S., Hale, L., Winser, S., & Regenbrecht, H. (2012). Augmented Reflection Technology for Stroke Rehabilitation - A clinical feasibility study. Proceedings of the 9th International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT 2012), Laval, France, September 10-12, 2012.

[3] Regenbrecht, H., Franz, E., McGregor, G., Dixon, B., & Hoermann, S. (2011). Beyond the looking glass: Fooling the brain with the Augmented Mirror Box. Presence: Teleoperators and virtual environments 20(6), MIT Press, Cambridge/MA, USA, 559-576.

[4] A. B. Sekuler and S. E. Palmer. Perception of partly occluded objects: a microgenetic analysis. Journal of Experimental Psychology: General, 121(1):95–111, 1992.

[5] Zollmann, S. Kalkofen, D., Mendez, E. Reitmeyer, G. (2010).Image-based Ghostings for Single Layer Occlusions in Augmented Reality. Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on , 19-26.

[6] Sanches, S. R R; Tokunaga, D.M.; Silva, V.F.; Sementille, A.C.; Tori, R., "Mutual occlusion between real and virtual elements in Augmented Reality based on fiducial markers," Applications of Computer Vision (WACV), 2012 IEEE Workshop on , vol., no., pp.49,54, 9-11 Jan. 2012; doi: 10.1109/WACV.2012.6163037

[7] M. M. Wloka and B. G. Anderson. Resolving occlusion in augmented reality. In Proceedings of the 1995 symposium on Interactive 3D graphics, I3D '95, pages 5–12, New York, NY, USA, 1995. ACM.

[8] Newcombe, Richard A.; Davison, Andrew J.; Izadi, S.; Kohli, P.; Hilliges, Otmar; Shotton, J.; Molyneaux, David; Hodges, Steve; Kim, David; Fitzgibbon, A., "KinectFusion: Real-time dense surface mapping and tracking," Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on , pp.127-136, 26-29 Oct. 2011

[9] Piekarski, W.; Thomas, B.H., "Tinmith-Metro: new outdoor techniques for creating city models with an augmented reality wearable computer," Wearable Computers, 2001. Proceedings. Fifth International Symposium on , pp.31,38, 2001

[10] John, C., Schwanecke, U. & Regenbrecht, H. (2009). Real-time Volumetric Reconstruction and Tracking of Hands in a Desktop Environment. Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, September 2009, Munster/Germany, Lecture Notes in Computer Science, 2009, Volume 5702/2009, 1053-1060, DOI: 10.1007/978-3-642-03767-2_128.

[11] Kruijff, E., Swan, J.E. II, and Feiner, S. (2010). Perceptual Issues in Augmented Reality Revisited. IEEE International Symposium on Mixed and Augmented Reality 2010, Science and Technolgy Proceedings, 13 -16 October, Seoul, Korea

[12] Fischer, J., Regenbrecht, H. & Baratoff, G. (2003). Detecting Dynamic Occlusion in front of Static Backgrounds for AR Scenes. Proceedings of ACM Eurographics Workshop on Virtual Environments 2003., Zurich, Switzerland.

[13] Berger, M.: Resolving occlusion in augmented reality : a contour based approach without 3d reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition. (1997)

[14] Kim, H., Yang, S., Sohn, K.: 3d reconstruction of stereo images for interaction between real and virtual worlds. In: ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality. (2003)