Augmented Reality for Immersive Remote Collaboration

Dan Gelb   Anbumani Subramanian   Kar-Han Tan
HP Labs
{dan.gelb, anbumani, kar-han.tan}@hp.com

Abstract

Video conferencing systems are designed to deliver a collaboration experience that is as close as possible to actually meeting in person. Current systems, however, do a poor job of integrating video streams presenting the users with shared collaboration content. Real and virtual content are unnaturally separated, leading to problems with nonverbal communication and the overall conference experience. Methods of interacting with shared content are typically limited to pointing with a mouse, which is not a natural component of face-to-face human conversation. This paper presents a natural and intuitive method for sharing digital content within a meeting using augmented reality and computer vision. Real and virtual content is seamlessly integrated into the collaboration space. We develop new vision based methods for interacting with inserted digital content including target finding and gesture based control. These improvements let us deliver an immersive collaboration experience using natural gesture and object based interaction.

1. Introduction

The usage of video conferencing continues to increase, from low cost consumer products up to high-end enterprise solutions. In the consumer space improvements in computational power and available bandwidth mean that video conferencing is possible on an increasing number of platforms and devices. Video conferencing in the enterprise is being driven by the desire to reduce travel costs while still maintaining the experience of an in-person meeting. High-end products designed for full room installations, including systems such as Cisco's TelePresence and HP's Halo, rely on life-size display of remote participants in an attempt to create the illusion of meeting in the same physical space. When it comes to how shared digital content is displayed to meeting participants, all current systems fall short of simulating a face to face meeting. Some systems emphasize the shared content while minimizing the video of the users, making it difficult to perceive non-verbal cues such as facial expressions and reactions. Others relegate the digital content to a secondary display area that is not naturally integrated into the meeting. An example is shown in Figure 1, where collaboration content occupies a separate area above the video screens showing the remote users. Gesture and gaze signals related to collaboration content are not reproduced naturally in such systems. If a meeting participant attempts to point to an area of interest on the collaboration screen remote users will be unable to determine what they are indicating. Desktop conferencing systems suffer from similar issues with shared content. The layout of collaboration content and people video is often inconsistent from user to user, so even if a participant could point with their hand to indicate an area of interest different users would interpret the gesture differently. All current systems present cumbersome methods for interacting with the shared content, or prevent remote participants from interacting with the content entirely.

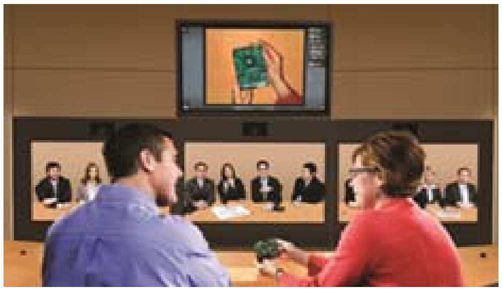

Figure 1: Collaboration using HP Halo Collaboration Studio. Shared media is shown above the people screens.

1.1. Augmented Reality

Augmented reality is the combination of real and virtual elements into an integrated image or video stream. Historically augmented reality was used with head-mounted displays capable of displaying digital content overlaid with a user's view of the real world. Head-mounted displays generally obstruct the view of the wearer's eyes, resulting in loss of eye contact which is critical for a good video conference experience. In spite of

978-1-61284-035-2/10/$26.00 ©2010 IEEE

this limitation head-mounted augmented reality has been applied to video conferencing using static images as avatars [3]. This was later extended to include live video of remote participants superimposed on the direct view of the real world at the locations of carefully designed augmented reality targets [4]. 3D reconstructions of remote users were superimposed in the work of Prince et al. [22]. If the AR targets are not found in the visual area then the remote participants are not displayed. Shared digital content is also not supported in these systems.

Early work on inserting digital content using augmented reality targets was presented in [23]. This work was in the context of a single user desktop. The possibility of using augmented reality without complex camera calibration was shown in [13]. Previous work on using AR targets in a video conferencing scenario was presented in [2]. In their work users must always hold up the AR target for collaboration content to be visible, and sometimes multiple targets must be held simultaneously. Requiring the users to continuously hold targets in the camera field of view can result in problems with fatigue. Collaborative interaction was limited due to latency issues.

1.2. Gesture Recognition

The idea of using a camera for user interaction has led to the development of software applications like CameraMouse [7] and Nouse [19]. In these applications, a user's face or facial feature (like nose) is tracked in webcamera images to move the mouse pointer. The objective of CameraMouse is to enable physically-challenged users to interact with a computer. However, for normal people, moving one's face or nose in front of a camera is not only cumbersome and tiring but is also an unnatural way for any interaction with the computer.

Freeman and Weissman built one of the frst systems for vision based hand-gesture recognition to control a television set [10]. In the past few years, there has been an increasing interest in using hand gestures for computer interactions. There are several survey papers on hand-gesture recognition [16, 21, 24, 26, 27, 29] which also serve as an indicator of the growing interest in this topic.

Hand gesture recognition from depth data has been reported recently. Although depth data can be estimated using a stereo rig for hand gesture recognition as presented in [8], the use of time-of-fight based infra-red depth sensors [28] has been reported for various applications. The use of human body gestures for gaming using depth sensing has been reported in [25]. This uses a combination of mean-shift tracking [9] and hidden Markov models (HMM) for gesture recognition. The influence of depth information for gesture recognition as depth silhouettes is proposed in [18]. The depth silhouettes used along with principal component analysis (PCA) and HMM are shown to perform better than PCA used with support vector machines (SVM). A laser-based camera producing low-resolution images at video rate is used to recognize hand poses in terms of finger poses and finger inter-relations in real-time [17]. Tracking pointing gestures using a stereo vision system that uses an FPGA-based dense depth mapping has been reported in [12]. The combination of depth and visual information as captured by a CSEM SwissRanger SR-2 camera has been used for view invariant gesture recognition using a probabilistic Edit distance classifier [11]. In [5], an articulated hand model is used to fit the depth data of a user's hand for estimating the hand pose.

Our present work uses a depth camera for image analysis. The depth data in images is used for segmenting the hand from an image. We use optical flow of the segmented hand regions in images and apply a rule-based approach to interpret the hand gestures. Our method does not depend either on hand models or machine learning methods to recognize a gesture.

2. Our Solution

2.1. Frame-based Insertion

Our first technique is inspired by a classic advertisement for the company HP [30], and involves tracking a simple rectangular frame in real time. While the HP ads required an artist to manually mark locations in every frame of the commercial, we use computer vision to locate a rectangular frame in a live video stream. Our system first performs a color similarity test on the video stream to identify regions matching our known target color. The algorithm must examine pixels within a relatively wide range of similar colors since the target’s brightness and even chrominance can change significantly due to the lighting conditions in the user’s location. The result is thresholded and median filtered to remove noise. After identifying pixels in the image that are close enough to the target values we connect neighboring candidate pixels into regions. The software then processes each connected set of pixels to look for regions that are ring shaped. We only want to use regions that are in the shape of our target, a connected ring matching our target color containing a non-target background in the center. This step is required to exclude areas of the scene that might happen to match our target color but that have different shapes. We compute the centroid of each region and then calculate the minimum distance from the centroid to pixels in the region. If the centroid overlaps pixels in the region, or is closer than a threshold the region cannot be our target and is excluded.

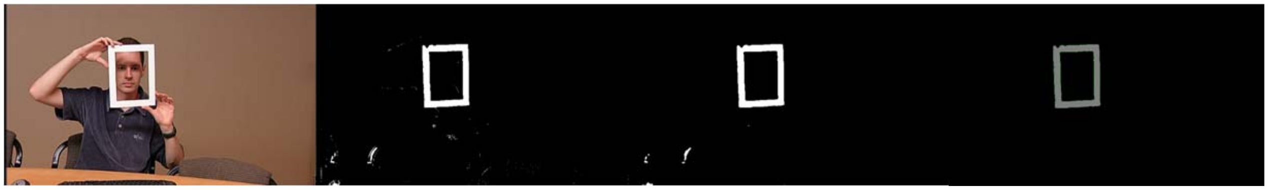

Figure 2: Target finding stages: Source, Color Match, Threshold/Filter, Region growing

For the identified ring-shaped regions we then compute interior corner locations to identify the inner edges of the frame target. The target identification stages are illustrated in Figure 2. We use image warping on the GPU to insert virtual digital content into the real frame, based on the identified location of the tracked frame in each video image. The inserted digital content can be media that the users are sharing, including presentation material such as slides, and even video sequences. The digital content is naturally integrated into the meeting.

As shown in Figure 3, a white rectangular frame is being held up and a video clip is inserted within the frame. This prototype demonstrates one of the significant advantages of using augmented reality in a remote collaboration system; the virtual digital content is tightly integrated into the real world video of the meeting participants. This allows the users to naturally interact with the digital content, such as by pointing at the virtual objects to indicate areas of interest. In traditional collaboration technologies, including even high-end telepresence systems, shared digital content is kept separate from the video streams containing the users, often on a separate screen entirely. As a result users cannot interact naturally with the digital content as there is not a consistent virtual space in which to indicate interest or interact with the objects.

We can also use the frame as a virtual capture device. Instead of using the frame as an indicator of where to display virtual content, the frame can be used to identify an area of interest to be captured. The actual capture event can be triggered when the system detects that the user has held the frame stationary around an object. Alternatively voice commands may be used to trigger capture events.

Our system can be configured to retain the inserted content at the last identified target location so that the user can position their content as desired keep it there without needing to hold the frame. The action of freezing the content location can also be triggered using keyboard or voice commands. The transparency of the inserted virtual content can also be adjusted to allow the user to be seen through the content. This enables pointing and gesturing at areas of interest to be directly and accurately perceived by remote meeting participants.

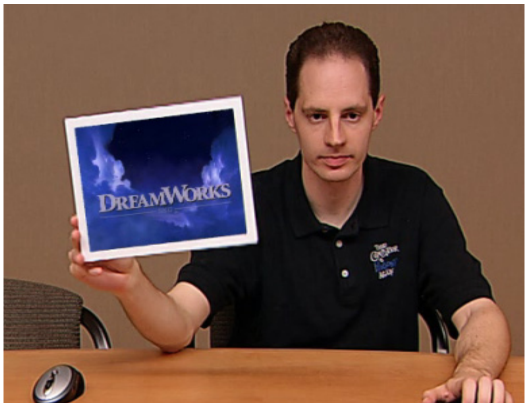

Figure 3: Frame based insertion

2.2. Gesture Recognition

Our second interface uses hand gestures that are identified based on an active depth camera which allows the users to control the augmented reality content based on those gestures. The gestures supported in our work are illustrated in Figure 4.

Hand gestures are recognized using a depth camera, which measures the time of flight of a pulsed IR source to determine a distance value for each pixel in its view. We currently use depth cameras such as the ZCam from 3DV Systems [1] (shown in Figure 5) and Canesta [6] for gesture recognition. These camera use active illumination for depth sensing - they emit modulated infra-red (IR) light and based on the time-of-flight principle, the reflected light is used to calculate depth (distance from camera) in a scene. The Zcam camera provides both RGB (VGA size) image and a grayscale depthmap (half-VGA size) image at 30 frames per second (fps). While depth cameras are currently limited to expensive, high-end devices, costs are coming down rapidly. The first mass-market device to feature an active 3D sensors will be the Xbox Kinect (Project Natal) video game controller from Microsoft [15]. The Project Natal camera also uses IR light as a way of determining distance to game players in front of the screen. Our system uses the ZCam camera to generate similar depth information. This data is combined with a regular RGB image to allow more information about the

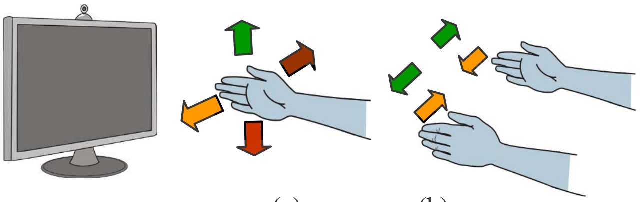

Figure 4: Illustration of camera based hand-gesture interaction. Vocabulary of gestures supported in our work: (a) move (left/right/up/down) (e) zoom (in/out – enlarge/shrink).

hand and shape to be captured than is feasible with just an RGB image alone.

The technical approach used for gesture recognition is shown in Figure 6. The depth image from the camera is used to find the threshold using Otsu's method [20] and generate a binary mask of the hand. This mask is used to segment the hand from the grayscale image, obtained from the color image. The segmented hand is then tracked over time using Lucas-Kanade method [14] to generate a flow vector. The flow vector is used to recognize if the hand-movement corresponds to a gesture supported by the system.

2.3. Interaction

Our system currently supports several gestures for adjusting the placement of digital content within the meeting video stream, and drawing annotations into the video. Gestures identified when the user moves one arm in front of the camera are interpreted as commands to move the collaboration content and, when the user moves both arms in front of the camera, the commands are used to resize the collaboration content. We detect gestures only within a controllable distance from the camera to avoid unintended interactions.

Single hand motions to the left/right or up/down move the object in the corresponding direction. We cause the detected motions to impart a velocity onto the object so that the user has the natural experience of pushing on the object. We apply frictional forces to smoothly dampen out the velocity over a short amount of time. Note that if a user moves their hand to the left they will see the object on the screen also move to the left. However on the remote users' displays they see the user from the perspective of the camera and as a result see them gesturing to the right hand side of the remote user's display. The motion of the object is kept consistent with each user's view of the displayed person streams. Maintaining a consistent presentation that matches the users' perceptions is critical for a natural experience. If a user must consciously think about what will happen when they make a gesture then intuitive interactions will not be possible.

Two handed gestures are interpreted as enlarge/shrink commands. If the user's hands are moving apart then the displayed content is enlarged. Shrinking happens similarly when the user's hands are detected moving together. In our testing most users find two handed gestures intuitive as they have a natural mapping to the two finger pinch/zoom touch gestures that exist on hand-held devices.

Our system also analyzes the depth data to identify pointing gestures based on the hand shape. We compute a pointing location based on the location of the closest finger to the depth camera. We currently paint into the video stream at the detected pointing location, using a different color for each user. We use alpha blending so drawings blend naturally into the video. We've found that users typically use drawing and mark-up to temporarily indicate areas of interest so we gradually reduce the opaqueness of the drawings so that they fade out over time. If permanent mark-ups are desired the alpha fading could be turned off using voice commands or other controls.

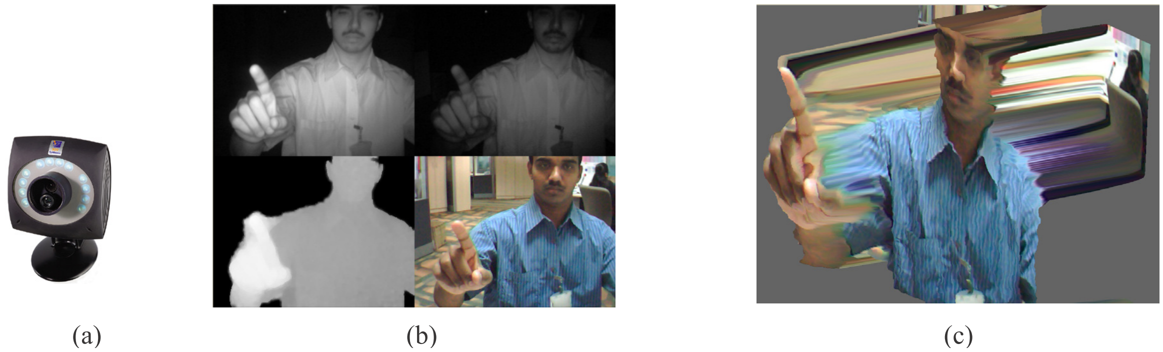

Figure 5: (a) ZCam from 3DV Systems. (b) Data output from ZCam - primary and secondary infrared images, a depthmap and a RGB color image. (c) Volumetric view of depth values fused with the RGB image.

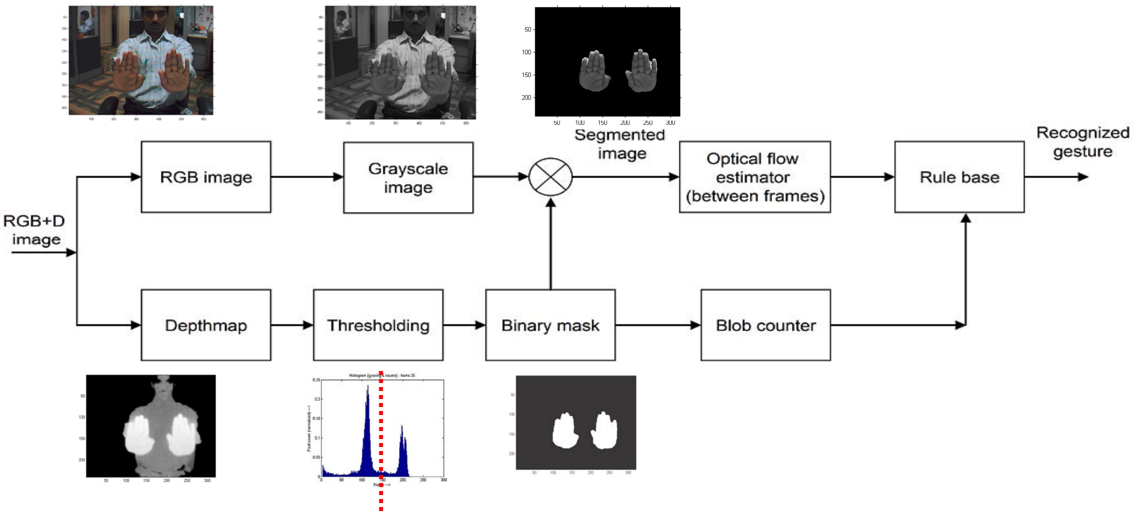

Figure 6: Our approach for hand gesture recognition.

3. Results

We have created two computer vision technologies for enabling new interaction methods that allow for natural insertion of shared digital content into distributed meetings. Each of our solutions work in real time at full video frame rates, at least 30 frames per second, which is required for reproducing natural motions. The detection and sensing algorithms are implemented in C++ and run on standard laptop and desktop CPUs. Our system is implemented in a custom dataflow software framework, similar to DirectShow or GStreamer. This enables different computational tasks, such as detection, video coding, and audio processing to be processed in parallel on the available compute cores to efficiently take advantage of the CPU parallelism present in current hardware. All compositing, warping, and blending is performed on the GPU. We have measured the end-to-end latency of our system at around 150 milliseconds, with some variation depending on the type of camera and display that is used, which is sufficient for remote collaboration given typical network latencies.

The majority of current desktop video conferencing systems give users the option of seeing live videos of themselves during the meeting. In systems such as these a user interacting with our technologies can directly see what the system is sensing and they receive immediate feedback. Systems such as in-room telepresence studios often attempt to create the illusion of all parties being in the same physical space, so they avoid showing a mirror-like view of the local participants. In configurations where the user does not see themselves we must provide some feedback to show the user what the vision algorithms are sensing. During frame-based tracking we can simply insert a virtual frame into the displayed video stream corresponding to where the frame is detected. During gesture interaction we use the captured depth information to display a ghost-like image of the user's own hand over the video, as shown in Figure 7. This is particularly important when pointing gestures are used.

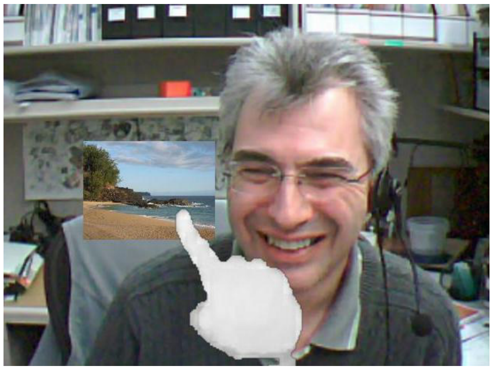

Figure 7: Visual feedback of hand location

During user testing we've found that even people who are inexperienced with gesture interfaces and computer vision quickly understand how to interact with the system.

4. Conclusion

We have presented a solution that uses real-time computer vision to enable intuitive interaction with digital content during live video conferencing. Users can control the system using simple objects and gestures without interrupting the normal meeting flow. Having shared media naturally integrated with the user displays also allows for much richer, more expressive interactions which significantly enhances the meeting experience. Looking ahead, we believe that augmented reality and computer vision techniques will make remote meetings more natural than the experience offered by today's video conferencing systems. In the future, it is possible that remote meetings may one day even surpass real meetings in effectiveness.

References

[1] 3DV Systems, http://www.3dvsystems.com

[2] I. Barakonyi, T. Fahmy, and D. Schmalstieg. Remote collaboration using Augmented Reality Videoconferencing. In *Proceedings of Graphics interface 2004*, pp. 89-96.

[3] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphett. A Wearable Spatial Conferencing Space. In *Proceedings of ISWC '98*, pp. 76-83.

[4] M. Billinghurst, A. Cheok, S. Prince, and H. Kato. Real world teleconferencing. *Computer Graphics and Applications, IEEE* , vol.22, no.6, pp. 11- 13, Nov/Dec 2002

[5] P. Breuer, C. Eckes, and S. Müller, "Hand gesture recognition with a novel ir time-of-flight range camera - a pilot study," Computer Vision/Computer Graphics Collaboration Techniques, pp. 247–260, 2007.

[6] Canesta Inc., http://www.canesta.com

[7] Camera mouse., http://www.cameramouse.org

[8] Y.-H. Chang, L.-W. Chan, J.-C. Ko, M.-S. Lee, J. Hsu, and Y.-P. Hung, "Qpalm: A gesture recognition system for remote control with list menu," in First IEEE International Conference on Ubi-Media Computing, Lanzhou, Jul./Aug. 2008, pp. 20–26.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, Hilton Head Island, SC, Jun. 2000, pp. 142–149.

[10] W. T. Freeman and C. D. Weissman, Television control by hand gestures, Mitsubishi Electric Research Laboratories, Tech. Rep. TR94-24, 1994. http://www.merl.com/papers/TR94-24/

[11] M. B. Holte, T. B. Moeslund, and P. Fihl, "Fusion of range and intensity information for view invariant gesture recognition," in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, Jun. 2008, pp. 1–7.

[12] Y. Jia, S. Li, and Y. Liu, "Tracking pointing gesture in 3d space for wearable visual interfaces," in Proc. of the Intl. Workshop on Human-Centered Multimedia. New York, NY, USA: ACM, 2007, pp. 23–30.

[13] K. Kutulakos and J. Vallino, "Calibration-Free Augmented Reality", Visualization and Computer Graphics, IEEE Transactions on , vol.4, no.1, pp.1-20, Jan-Mar 1998

[14] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereovision," in Proc. of the Image Understanding Workshop, 1981, pp. 121–130.

[15] Microsoft Xbox Kinect (Project Natal), http://www.xbox.com/en-US/live/projectnatal

[16] S. Mitra and T. Acharya, "Gesture recognition: A survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 37, no. 3, pp. 311–324, May 2007.

[17] Z. Mo and U. Neumann, "Real-time hand pose recognition using low-resolution depth images," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, 2006, pp. 1499–1505.

[18] R. Muñoz-Salinas, R. Medina-Carnicer, F. Madrid-Cuevas, and A. Carmona-Poyato, "Depth silhouettes for gesture recognition," Pattern Recognition Letters, vol. 29, no. 3, pp. 319–329, Feb. 2008.

[19] Nouse. http://ivim.ca/

[20] N. Otsu, "A threshold selection method from graylevel histogram," IEEE Transactions on System, Man, Cybernetics, vol. 19, no. 1, pp. 62–66, January 1978.

[21] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677–695, July 1997.

[22] S. Prince, A. D. Check, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, H. Kato. Real-time 3D interaction for augmented and virtual reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, pp. 238-238.

[23] H. Regenbrecht, G. Baratoff, and M. Wagner. A Tangible AR Desktop Environment. In *Computers & Graphics*, Vol. 25, No. 5, Elsevier Science, Oct. 2001, pp. 755-763

[24] T. Sowa, "The recognition and comprehension of hand gestures - a review and research agenda," Modeling Communication with Robots and Virtual Humans, pp. 38–56, 2008.

[25] Y. Wang, T. Yu, L. Shi, and Z. Li, "Using human body gestures as inputs for gaming via depth analysis," in Multimedia and Expo, 2008 IEEE International Conference on, Hannover, Jun./Apr. 2008, pp. 993–996.

[26] R. Watson, "A survey of gesture recognition techniques," Department of Computer Science, Trinity College, Dublin, Tech. Rep. TCD-CS-1993-11, 1993. http://www.tara.tcd.ie/handle/2262/12658

[27] Y. Wu and T. Huang, "Vision-based gesture recognition: A review," Gesture-Based Communication in Human-Computer Interaction, pp. 103–115, 1999.

[28] G. Yahav, G. J. Iddan, and D. Mandelboum, "3d imaging camera for gaming application," in Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on, Las Vegas, NV, Jan. 2007, pp. 1–2.

[29] M. Yeasin and S. Chaudhuri, "Visual understanding of dynamic hand gestures," Pattern Recognition, vol. 33, no. 11, pp. 1805–1817, November 2000.

[30] Youtube: http://www.youtube.com/watch?v=UirmvNktkBc