Using Haar-like Feature Classifiers for Hand Tracking in Tabletop Augmented Reality

Bruno Fernandes, Joaquin Fernández
Univesitat Politècnica de Catalunya
brunofer@gmail.com, jfernandez@ege.upc.edu

Abstract

We propose in this paper a hand interaction approach to Augmented Reality Tabletop applications. We detect the user's hands using haar-like feature classifiers and correlate its positions with the fixed markers on the table. This gives the user the possibility to move, rotate and resize the virtual objects located over the table with their bare hands.

1. Introduction

Augmented reality (AR) techniques have been applied to many application areas; however, there is still research that needs to be conducted on the best way to interact with AR content. As Buchmann et al. [1] affirms, Augmented Reality interaction techniques need to be as intuitive as possible to be accepted by end users and may also need to be customized for different application needs. Since hands are our main means of interaction with objects in real life, it would be natural for AR interfaces to allow free hand interaction with virtual objects. This would not only enable natural and intuitive interaction with virtual objects, but it would also help to ease the transition when interacting with real and virtual objects at the same time.

Interactive tabletop systems aim to provide a large and natural interface for supporting direct manipulation of visual content for human-computer interactions. In this paper we present a system that tracks the 2D position of the user's hands on a table surface, allowing the user to move, rotate and resize the virtual objects over this surface.

In creating an AR interface that allows users to manipulate 3D virtual objects over a tabletop surface there are a number of problems that need to be overcome [2]. From a technical point of view it is necessary to consider tracking, registration accuracy and robustness of the system. From a usability viewpoint we need to create a natural and intuitive interface and address the problem of virtual objects occluding real objects.

Our implementation is based on a computer vision tracking system that processes the video stream of a single usb camera. The tracking of hands by a webcam not only provides more flexible, natural and intuitive interaction possibilities, but also offers an economic and practical way of interaction. Users don't need any other equipment or device for interaction.

Vision-based hand tracking is an important problem in the field of human-computer interaction. Song et al. [3] list many constraints of some vision-based techniques used to track hands: methods based on color segmentation [4, 5] need users to wear colored gloves for efficient detection; methods based on background image subtraction require a fixed camera for efficient segmentation [6, 7]; contour based methods [8] work only on restricted backgrounds; infrared segmentation based methods [9, 10] require expensive infrared cameras; correlation-based methods [11, 12] require an explicit setup stage before the tracking starts; and the blob-model based methods [13] imposes restrictions on the maximum speed of hand movements.

The approach we've chosen is to use statistical models (classifiers). These models can be obtained by analyzing a set of training images, and then be used to detect the hands. Statistical model-based training takes multiple image samples of the desired object (hands in our case) and multiple images that do not contain hands (so called "negative" images). Different features are extracted from the training samples and distinctive features that can classify the hands are selected [14, 15]. The Haar-like features (so called because they are computed similarly to the coefficients of Haar wavelet transforms) were used combined with the AdaBoost algorithm [16] to extract the features characteristics of the hands.

The use of classifiers makes hand tracking possible independently of the complexity of the background and the illumination conditions. Images of hands exposed to different lighting conditions can be included in the training images set to make sure the system will perform strongly under these illumination conditions.

The paper is organized as follows. In section 2 we present some previously proposed interaction approaches to AR tabletop applications. In section 3 we give an overview of our system and explain the hand tracking approach. Finally, conclusions are drawn and future plans discussed.

2. Related work

Interaction with AR content is a very active area of research. In the past, researchers have explored a variety of interaction methods to AR tabletop applications. We present some of them in this section.

The most traditional approach for interacting with AR content is the use of vision tracked fiducial markers as interaction devices [17, 18]. This approach enables the use of hand-held tangible interfaces for inspecting and manipulating the augmentations [19, 20, 21]. For example, a fiducial marker can be attached to a paddle-like hand-held device enabling users to select, move and rotate virtual objects [22, 23, 24, 25]. In the “VOMAR” [2] application, the paddle is the main interaction device, its position and orientation is tracked using the ARToolKit. The paddle allows the user to make gestures to interact with the virtual objects. The "Move the couch where?" [26] application, allow users to select and arrange virtual furniture in an AR living room design application. Paddle motions are mapped to gesture based commands, such as tilting the paddle to place a virtual model in the scene and hitting a model to delete it.

Some researchers have also placed makers directly over the user’s hands or fingertips in order to detect its pose and allow simple manipulations of the virtual objects [1, 27].

Lee et al. [28] suggests an occlusion based interaction approach, in which visual occlusion of physical markers are used to provide intuitive two dimensional interaction in tangible AR environments.

Instead of markers, some approaches [29, 30, 31, 32, 33] use trackable objects like, for example, small bricks as an interaction device to the tabletop Augmented Reality. These objects normally are optically or magnetically tracked.

Many researchers have studied and used glove-based devices to measure hand location [34]. In general, glove-based devices can measure hand position with high accuracy and speed, but they aren’t suitable for some applications because the cables connected to them restrict hand motion [35].

Some systems use retro-reflective spheres that can be tracked by infrared cameras [33, 36, 37]. Dorfmüller-Ulhaas et al. [38] proposes a system to track the user’s index finger via retro-reflective markers and use its tip as a cursor. The select command is triggered by bending the finger to indicate a grab or grab-and-hold (drag) operation.

The “SociaDesk” [39] project, uses a combination of marker tracking and infrared hand tracking to enable the user interaction with software-based tools at a projection table. The “Perceptive Workbench” system [40] uses an infrared light source mounted on the ceiling. When the user stands in front of the workbench and extends an arm over the surface, the hand casts a shadow on the desk’s surface, which can be easily distinguished by a set of cameras. Sato et al. “Augmented Desk Interface” [10] makes use of infrared camera images for reliable detection of user's hands and uses template matching strategy for finding fingertips. This method allows the user to simultaneously manipulate both physical and electronically projected objects on a desk with natural hand gestures.

A common approach these days is to use touch sensitive surfaces to detect where the user hands are touching the table [34, 41], but Song et al. [3] arguments that barehanded interfaces enjoy higher flexibility and more natural interaction than tangible interfaces. Furthermore, tabletop touch interfaces requires special hardware that is still expensive and not readily accessible to everyone.

To allow users to interact with virtual content using natural hand gestures and without any marker or glove Lee et al. [42] proposed a hand tracking approach where it is possible to detect the five fingertips of the hand and calculate the 6DOF camera pose relative to the hand. This method allows the users to interact with the virtual models with the same functionality as provided by the fiducial paddle. In this case, there is no need to use a specially marked object, users can select and rotate the model with their own bare hand.

Unlike Lee et al. approach, our system can recognize different hand postures and needs no calibration prior to usage, the user just needs to print the array of markers on A3 (or bigger) sheet of paper and place it over any planar surface to be able to use the system.

3. System overview

We have implemented our tabletop system using ARTag [18] to track an array of markers over the table. In order to create the feature classifiers and detect the user's hands at run time we used the OpenCV library [15].

We have established a general coordinate system for the planar tabletop AR environment. When the hand is detected we convert its coordinates from the 2D camera image to the 3D tabletop coordinate system. This can be calculated by using the inverse of the OpenGL modelview[1] matrix. The modelview matrix is automatically calculated by ARTag when it recognizes the markers over the table. We also need to check if the hand is over the region defined by the markers, this can be done by tracing a ray back from the hand in direction of the table to see if it intercepts the desired region. We consider that the hand is always very close to the table.

We tested our system in a laptop with a 2.2 GHz Core 2 Duo processor and got 15 frames per second when performing the hand recognition at an image with 320x240 pixels resolution. If we change the camera capture image size to 640x480 pixels we get 8 frames per second. We prefer to display a 640x480 image but run the image recognition over a 320x240 image. We still get 15 fps with this approach.

Our system accepts three interaction methods, each method is characterized by a combination of some hand postures that the users should employ to interact with the system. The postures employed in each interaction method are shown in Figures 1, 2 and 3.

Interaction Method 1:

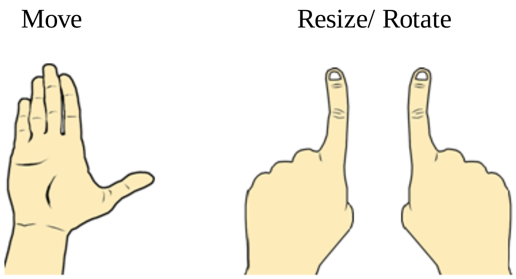

Figure 1. Interaction method 1.

Interaction Method 2:

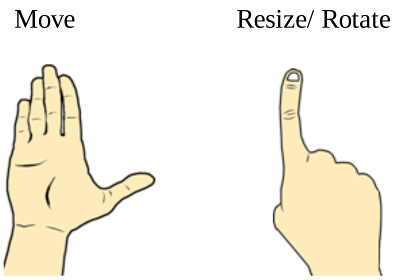

Figure 2. Interaction method 2.

Interaction Method 3:

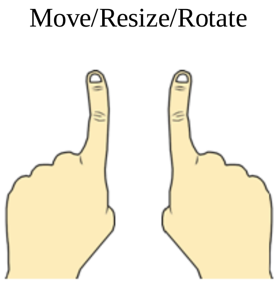

Figure 3. Interaction method 3.

For example, considering the interaction method 1, the "palm up" posture is employed to change the object position over the table, as seen in Figure 4. To eliminate the occlusion problem that happens when the virtual object incorrectly overlays the user's hand, the user will hold the object from below in order to move it.

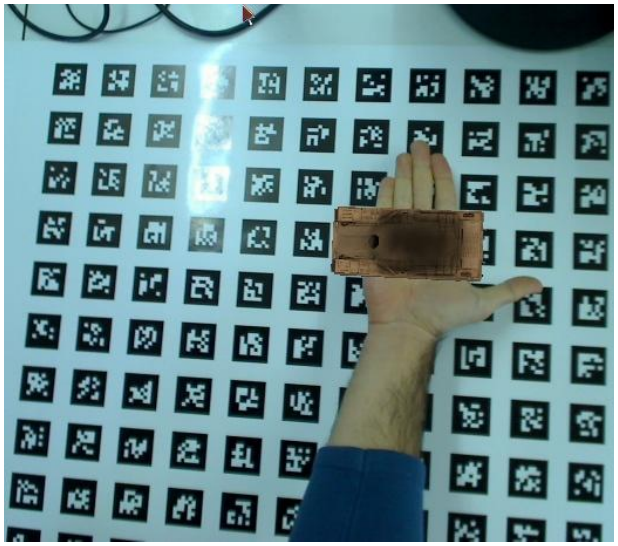

Figure 4. User "holds" a virtual object from below in order to move it.

[1] The modelview matrix is the transform between the tabletop coordinate system and the camera view.

Considering the interaction method 1 again, in order to resize and rotate the object, the user should employ both hands with the index finger extended, as shown in Figure 5. The object will be affected by the movement of the hands, when the distance between the hands increase the object size will increase accordingly and when this distance decreases the object size will decrease. To rotate the object the user will move its hands as if each hand were holding one side of the object.

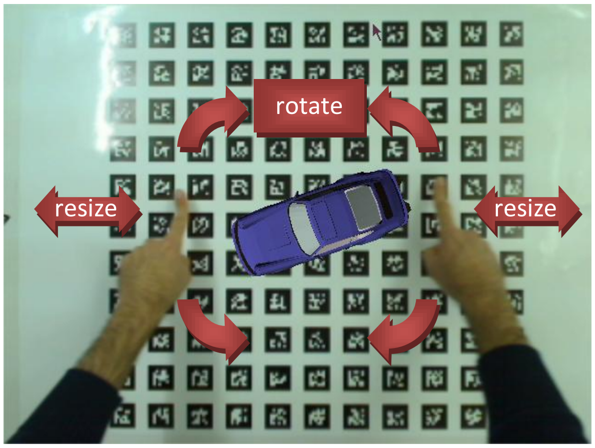

Figure 5. User applies both hands in order to rotate and resize the virtual object.

3.1. Hand Tracking

Viola et al. [43] proposes a methodology combining Haar-like features and the AdaBoost algorithm to detect faces. We have used this same methodology to detect the user's hands. In one image sub-window, the total number of haar-like features is very large, far larger than the number of pixels. In order to ensure a fast classification, the learning process must exclude the large majority of the available features, and focus on a small set of critical features. A variant of AdaBoost [16] is used for selecting those features.

The algorithm proposed by Viola et al. uses simple features reminiscent of Haar basis functions which have been used by Papageorgiou et al. [44]. More specifically, they use three kinds of features. The value of a *two-rectangle feature* is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent, see Figure 6. A *three-rectangle feature* computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a *four-rectangle feature* computes the difference between diagonal pairs of rectangles.

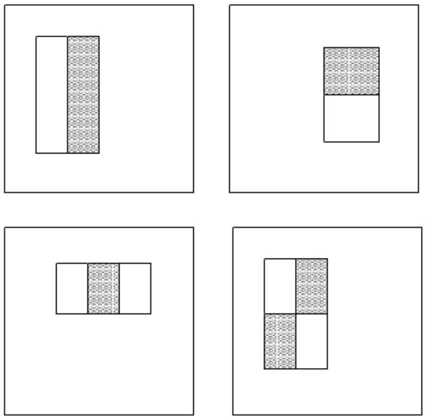

Figure 6. Example rectangle features shown relative to the enclosing detection window. [43]

A cascade of classifiers is a degenerated decision tree where at each stage a classifier is trained to detect almost all objects of interest while rejecting a certain fraction of the non-object patterns [45]**.** Each stage in the cascade reduces the false positive rate and decreases the detection rate. A target is selected for the minimum reduction in false positives and the maximum decrease in detection. Each stage is trained by adding features until the target detection and false positives rates are met (these rates are determined by testing the detector on a validation set). Stages are added until the overall target for false positive and detection rate is met.

Each stage was trained using the Discrete Adaboost algorithm [16]. Discrete Adaboost is a powerful machine learning algorithm, it can learn a strong classifier based on a (large) set of weak classifiers by re-weighting the training samples. Weak classifiers are only required to be slightly better than chance, and thus can be very simple and computationally inexpensive. Many of them smartly combined result in a strong classifier. The weak classifiers are all classifiers which uses one feature in combination with a simple binary thresholding decision. At each round of boosting, the feature-based classifier that best classifies the weighted training samples is added. With increasing stage number, the number of weak classifiers which are needed to achieve the desired false alarm rate at the given hit rate increases.

We have generated three hand classifiers, one for the palm of the hand, another one for the right hand with the index finger extended and the last one to detect both hands with the index finger extended. For all classifiers we used image samples from the hands of 12 different persons.

To generate the palm classifier we used 3000 positive images of palms of hands under different light conditions like indirect sunlight, high artificial light and low artificial light (see Figure 7), and 7500 negative images (images that do not contain hands). We didn't use images of the hands under direct sun light because we consider it is very unlikely that our system will be used under this circumstance. If we had included the direct sun light images we would be unnecessarily increasing the false positive rate of the classifier and to keep this rate at an acceptable level we would need to set a slightly lower detection rate for it. The sample hand images had the resolution of 15 x 28 pixels. The training process took approximately 4 days in a computer with a Core 2 Duo 1.8 GHz processor and 1 GB of RAM. It was completed in 15 training stages.

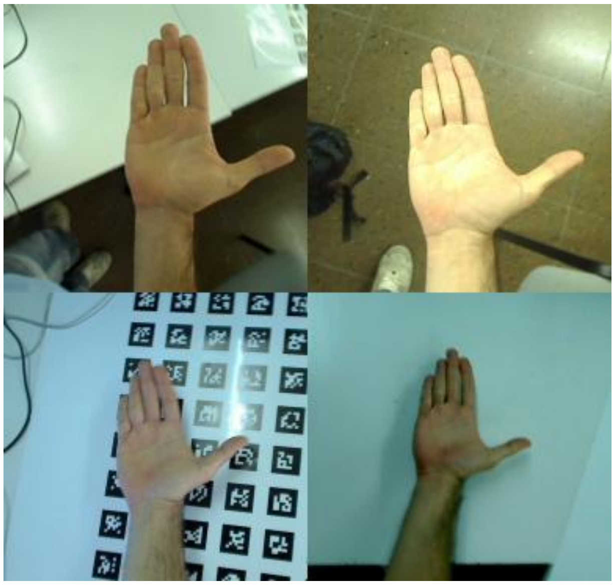

Figure 7. Hand image samples under different light conditions, used for classifier training.

To create the classifier of the back of the right hand with the index finger extended we used 2480 positive images of the hand under different light conditions and 7500 negative images. The sample images had the resolution of 16 x 27 pixels. The training process took approximately 5 days and it was completed in 19 training stages. To create the third classifier, the one for both hands with the index finger extended, we included images of both left and right hands in the positive images set and followed the same procedure.

The desired hand postures, for all classifiers, were captured subjected to small rotations of a maximum of 15 degrees. To detect the hands subjected to bigger rotations it is recommended to create different classifiers [46]. This fact brings a limitation to our system, it can detect the user hands in one orientation only, with a 15 degrees in-plane rotation tolerance. To detect the hands at many different orientations we would need to search for many classifiers at the same time which is possible but very computationally expensive. We would not get an interactive experience with today's mainstream computer's processing power.

It is important to notice that although the hand postures should always be aligned to the camera sight, the camera can move and rotate freely over the table (such as when attached to an HMD), as long as it is always looking, from the top, down to the table. In this case, the user must be aware to always employ the hands correctly aligned to the camera view.

3.2. Classifier's Performance. We have calculated the detection rate of our classifiers for hand images of persons that were and weren't included in the training images set, see Table 1.

Table 1. Detection rate of our classifiers.

Classifier	Included in training set	Not included in training set
Palm	93%	63%
Both hands - index finger	91%	83%
Right hand - index finger	92%	74%

As seen in Table 1, the detection rate is smaller for the hands of persons that did not "model" for the creation of the classifiers, but this fact doesn't make them unable to use the system. As the system does not need to recognize the user's hands at every frame, this smaller detection rate doesn't influence that much and the people that didn't have their hands taken pictures for the creation of the classifiers can also use the system.

4. Conclusion and future work

In this paper we have presented a prototype tabletop AR interface that supports hand interaction. Our intention was to create a simple and natural way of allowing users to move, rotate and resize the virtual objects over a table. The human hand is a ubiquitous input device for all kinds of real-world applications, so hand recognition should be one of the most natural and intuitive ways to interact with virtual objects in an AR environment.

In order to perform the hand recognition, we have integrated OpenCV's haar-like feature detection into ARTag. We have chosen to use haar-like feature classifiers because this method gives an acceptable performance and is robust when dealing with lighting and background variations. Infrared based methods for hand detection are meant to improve the robustness under poor lighting conditions, but they can fail in the presence of sunlight and need special cameras. The use of markers attached to the user's hands can provide reliable detection but they present obstacles to natural interaction similar to glove-based devices. Methods based on color segmentation require the users to wear colored gloves for efficient detection. Background subtraction methods can fail when a non-stationary camera is used [47]. Our proposed method was designed to be effective even in such challenging situations.

Compared to previous approaches of bare hand interaction in Augmented Reality systems, our approach is less sensitive to changes in illumination, can recognize different hand postures and needs no calibration prior to usage, the user just needs to print the array of markers on an A3 (or bigger) sheet of paper and place it over any planar surface to be able to use the system. Another advantage of our approach is that other skin color regions can appear in the view without interfering in the tracking, provided they are not a hand in one of the recognizable postures (or a similar posture).

Our system has two limitations, the first one is the fact that the users can only interact two-dimensionally with the virtual objects, but we believe this is acceptable in a tabletop environment. The second limitation is the fact that the hand postures can only be detected at one orientation, with a small tolerance of a 15 degrees rotation. This second limitation is due to performance issues.

As future work, we are planning to perform a formal evaluation of the system from the user's perspective. We plan to compare our three proposed interaction methods among them and also against the traditional method of "paddle with marker" interaction.

5. References

[1] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "FingARtips: gesture based direct manipulation in Augmented Reality," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* Singapore: ACM, 2004.

[2] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment," in *Proceedings. IEEE and ACM International Symposium on Augmented Reality. (ISAR 2000).* 2000, pp. 111-119.

[3] P. Song, S. Winkler, S. Gilani, and Z. Zhou, "Vision-Based Projected Tabletop Interface for Finger Interactions," in *Human–Computer Interaction*, 2007, pp. 49-58.

[4] C. C. Lien and C. L. Huang, "Model-based articulated hand motion tracking for gesture recognition," *Image and vision computing*, vol. 16, pp. 121-134, 1998.

[5] R. Y. Wang and J. Popovic, "Real-time hand-tracking with a color glove," in *ACM SIGGRAPH 2009 papers* New Orleans, Louisiana: ACM, 2009.

[6] Y. Guo, B. Yang, Y. Ming, and A. Men, "An Effective Background Subtraction under the Mixture of Multiple Varying Illuminations," in *Computer Modeling and Simulation, 2010. ICCMS '10. Second International Conference on*, pp. 202-206.

[7] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1-4.

[8] V. Gemignani, M. Paterni, A. Benassi, and M. Demi, "Real time contour tracking with a new edge detector," *Real-Time Imaging*, vol. 10, pp. 103-116, 2004.

[9] J. M. Rehg and T. Kanade, "DigitEyes : vision-based human hand tracking," School of Computer Science, Carnegie Mellon University, 1993.

[10] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition.*, 2000, pp. 462-467.

[11] J. L. Crowley, F. Berard, and J. Coutaz, "Finger Tracking as an Input Device for Augmented Reality," in *International Workshop on Face and Gesture Recognition*, Zurich, Switzerland, 1995.

[12] S. Wong, "Advanced Correlation Tracking of Objects in Cluttered Imagery," in *SPIE 2005, Proceeding of the International Society for Optical Engineering: Acquistion, Tracking and Pointing XIX*, 2005.

[13] I. Laptev and T. Lindeberg, "Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features," in *Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision*: Springer-Verlag, 2001.

[14] G. Monteiro, P. Peixoto, and U. Nunes, "Vision-Based Pedestrian Detection Using Haar-Like Features," in *Encontro Científico Guimarães*, 2006.

[15] G. Bradski and A. Kaehler, *Learning OpenCV : Computer vision with the OpenCV library*: O'Reilly Media, 2008.

[16] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," in *International Conference on Machine Learning*, 1996, pp. 148-156.

[17] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*, 1999, pp. 85-94.

[18] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 590-596 vol. 2.

[19] I. Poupyrev, D. S. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani, "Developing a generic augmented-reality interface," *Computer*, vol. 35, pp. 44-50, 2002.

[20] H. Seichter, A. Dong, A. v. Moere, and J. S. Gero, "Augmented Reality and Tangible User Interfaces in Collaborative Urban Design," in *CAAD futures* Sydney, Australia: Springer, 2007.

[21] J. Rekimoto and M. Saitoh, "Augmented surfaces: a spatially continuous work space for hybrid computing environments," in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* Pittsburgh, Pennsylvania, United States: ACM, 1999.

[22] M. Fjeld and B. M. Voegtli, "Augmented Chemistry: An Interactive Educational Workbench," in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*: IEEE Computer Society, 2002.

[23] R. Grasset, A. Dunser, and M. Billinghurst, "The design of a mixed-reality book: Is it still a real book?," in *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, 2008, pp. 99-102.

[24] Z. Zhou, A. D. Cheok, T. Chan, J. H. Pan, and Y. Li, "Interactive Entertainment Systems Using Tangible Cubes," in *IE2004, Australian Workshop on Interactive Entertainment* Sydney, 2004.

[25] T. V. Do and J.-W. Lee, "3DARModeler: a 3D Modeling System in Augmented Reality Environment," *International Journal of Computer Systems Science and Engineering*, 2008.

[26] S. Irawati, S. Green, M. Billinghurst, A. Duenser, and K. Heedong, ""Move the couch where?" : developing an augmented reality multimodal interface," in *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, 2006, pp. 183-186.

[27] M. Dias, J. Jorge, J. Carvalho, P. Santos, and J. Luzio, "Usability evaluation of tangible user interfaces for augmented reality," in *Augmented Reality Toolkit Workshop, 2003. IEEE International*, 2003, pp. 54-61.

[28] G. A. Lee, M. Billinghurst, and G. J. Kim, "Occlusion based interaction methods for tangible augmented reality environments," in *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry* Singapore: ACM, 2004.

[29] J. Patten, H. Ishii, J. Hines, and G. Pangaro, "Sensetable: a wireless object tracking platform for tangible user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems* Seattle, Washington, United States: ACM, 2001.

[30] M. Rauterberg, M. Bichsel, U. Leonhardt, and M. Meier, "BUILD-IT: a computer vision-based interaction technique of a planning tool for construction and design," in *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*: Chapman \&amp; Hall, Ltd., 1997.

[31] S. Minatani, I. Kitahara, Y. Kameda, and Y. Ohta, "Face-to-Face Tabletop Remote Collaboration in Mixed Reality," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, 2007, pp. 43-46.

[32] S. Balcisoy, M. Kallman, R. Torre, P. Fua, and D. Thalman, "Interaction techniques with virtual humans in mixed environments," in *Biomedical Imaging, 2002. 5th IEEE EMBS International Summer School on*, 2002, p. 6 pp.

[33] C. Sandor and G. Klinker, "A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality," *Personal Ubiquitous Comput.*, vol. 9, pp. 169-185, 2005.

[34] H. Benko and S. Feiner, "Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection," in *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, 2007.

[35] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *Computer Graphics and Applications, IEEE*, vol. 22, pp. 64-71, 2002.

[36] A. MacWilliams, C. Sandor, M. Wagner, M. Bauer, G. Klinker, and B. Bruegge, "Herding Sheep: Live System Development for Distributed Augmented Reality," in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*: IEEE Computer Society, 2003.

[37] J. Looser, M. Billinghurst, R. Grasset, and A. Cockburn, "An evaluation of virtual lenses for object selection in augmented reality," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques*

in Australia and Southeast Asia Perth, Australia: ACM, 2007.

[38] K. Dorfmuller-Ulhaas and D. Schmalstieg, "Finger tracking for interaction in augmented environments," in *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, 2001, pp. 55-64.

[39] C. McDonald, G. Roth, and S. Marsh, "Red-handed: collaborative gesture interaction with a projection table," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, 2004, pp. 773-778.

[40] T. Starner, B. Leibe, D. Minnen, T. Westyn, A. Hurst, and J. Weeks, "The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks," *Machine Vision and Applications*, vol. 14, pp. 59-71, 2003.

[41] S. Na, M. Billinghurst, and W. Woo, "TMAR: Extension of a Tabletop Interface Using Mobile Augmented Reality," in *Transactions on Edutainment I*, 2008, pp. 96-106.

[42] T. Lee and T. Hollerer, "Hybrid Feature Tracking and User Interaction for Markerless Augmented Reality," in *Virtual Reality Conference, 2008. VR '08. IEEE*, 2008, pp. 145-152.

[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. I-511-I-518 vol.1.

[44] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 555-562.

[45] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, pp. I-900-I-903 vol.1.

[46] M. Kolsch and M. Turk, "Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*: IEEE Computer Society, 2004.

[47] M. Sivabalakrishnan and D. Manjula, "Adaptive Background subtraction in Dynamic Environments Using Fuzzy Logic," *International Journal on Computer Science and Engineering*, vol. 2, pp. 270-273, 2010.