Implementing Bubblegrams: The Use of Haar-Like Features for Human-Robot Interaction

James E. Young, Ehud Sharlin, Jeffrey E. Boyd
Department of Computer Science
University of Calgary
Calgary, AB, T2N-1N4
Email: {jyoung;ehud;boyd}@cpsc.ucalgary.ca

***Abstract*— *Bubblegrams* - a human-robot interaction (HRI) technique - uses Mixed Reality (MR) to allow collocated humans and robots to interact directly by visually augmenting their shared physical environment. *Bubblegrams* uses interactive comic-like graphic balloons that appear above the robot to allow for interaction between humans and robots. A key technical challenge facing *Bubblegrams* is the detection of the location of the robot within the user's vision; the MR system needs this information to place the bubble. To solve this, we applied a vision algorithm based on Haar-like features to find and track the robot in real time. This paper introduces the *Bubblegrams* interface and details the vision algorithm used to detect and track the robot.**

I. Introduction

With the rapid advancement of robot technology, the need for effective human-robot interfaces is becoming clear and pressing [1]. As robots become increasingly capable, we can expect to find users sharing their everyday environments with robots in various ways [2], [3]. Research in Human-Robot Interaction (HRI) explores the various issues and problems surrounding interaction with robots and attempts to develop effective HRI interfaces [4].

Robots are a class of computers which are distinguished by their dynamic presence in the physical world. A robot, unlike the conventional computer which is primarily a digital entity, is both a physical and digital entity; a robot is simultaneously perceiving, functioning and interacting in both the digital and physical realms. Current human-robot interfaces often fail to integrate this duality and offer interaction which is restricted to either the physical or the virtual domain; interaction can be based on physical modalities such as speech or digital modalities such as remote control software tools. This separation can reduce levels of awareness [5] and ultimately hinder the quality of interaction between humans and robots [6].

One solution to this problem is to use MR as an interaction tool between humans and robots. MR is a technique which tracks components of the physical world and augments them with virtual data. This visual augmentation is commonly accomplished by projecting images onto the environment or by using a head-mounted display (HMD) to synthetically augment the vision of the wearer [7]. We believe that MR can solve many of the interaction problems mentioned above by allowing the robot to dissolve the border between physical and virtual interaction with humans; using MR, robots can superimpose digital information directly onto users' physical environments. At the same time, humans can interact with digital information directly, as if this information is an integral part of their physical interaction space.

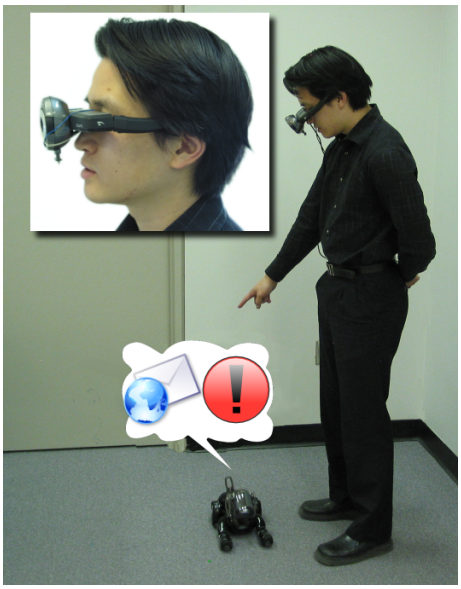

Fig. 1. A user and robot interact through a *Bubblegram*.

II. Bubblegrams

In this paper we present *Bubblegrams* - an MR interaction technique that combines physical and virtual interaction, allowing users to interact with robots simultaneously in the digital and physical realms [8]. *Bubblegrams* (see Figure 1) appear as visual cartoon-like bubbles floating above the robot. The user wears MR goggles, using displays and a camera, to view the *Bubblegrams* and can use *Bubblegrams* for direct access to the robot's status and functions. For example, in a home-environment a robot which completed a cleaning chore can present a smiley bubble above its head, showing

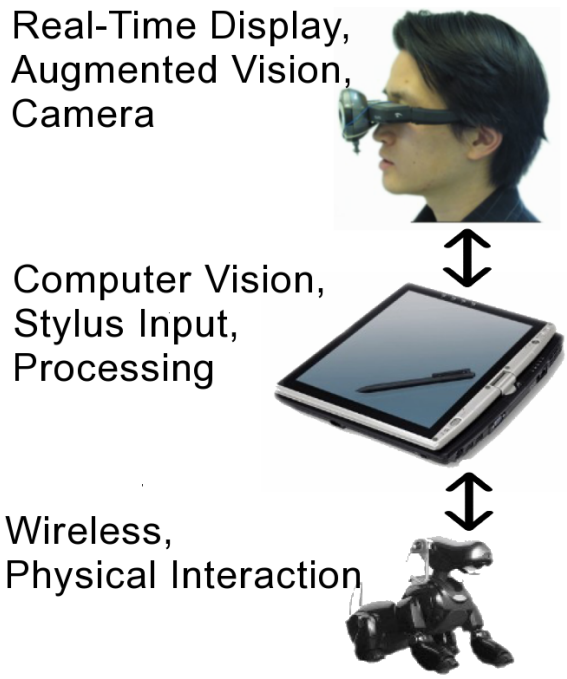

Fig. 2. The *Bubblegrams* architecture.

its satisfaction of fulfilling the task. It can also provide an interface allowing the user to direct the robot further ("keep cleaning", "come and play with me", etc.). In a search and rescue operation, a human can use the *Bubblegram* to control and send a robot into the next room or to call it back; this *Bubblegram* could display a video feed from the robot.

Our current implementation integrates an Icuiti HMD and webcam (as shown in Figure 2) as the MR interface, which is powered by a tablet PC (Toshiba Portege) to offer portability and wireless internet connectivity. In addition, the tablet PC's stylus interface can be used as one of many possible methods to interact with *Bubblegrams*. For the robot, we are using a Sony AIBO ERS-7 robot dog which generates *Bubblegrams* and conveys them to the user system through a wireless network connection. This connection is also the communication medium for the various interaction techniques.

For *Bubblegrams* to be effective we need to physically associate the location of the balloon with the robot whenever the robot is in the user's field of view; we need to track the robot in real-time through the user's MR vision channel. In this paper we present our vision technique for real-time detection and tracking of a Sony AIBO robot dog in a video sequence. Our technique, based on the Viola and Jones' "Rapid Object Detection" method [9], considers the specific problems facing robot detection in a *Bubblegrams* interaction session, achieving high detection success. In the coming sections we present an overview of research related to our efforts, we then describe our approach to the AIBO detection problem in streaming video, and detail our implementation and preliminary results.

III. Related Work

MR has been introduced recently as a means of combining digital information with the physical world for various applications such as interactive media (for example, the MagicBook project [7] and the ARTag system [10]), modelling volumetric data [11], [12], assisting with medical surgery [13] and as a computer supported cooperative work (CSCW) interaction theme and environment [14].

We can crudely classify MR techniques as either based on HMD or projective visualisation. Projective visualisation can be integrated seamlessly into a user's entire field of view allowing them to use their full natural vision capabilities. The downside, however, is that projectors are still less portable and flexible than HMDs, often being heavy and difficult to move, and require a projection surface and appropriate lighting. One can envision an MR environment based on projection techniques in a dedicated space that is designed and crafted specifically for the task. It is still difficult to implement projection-based MR in an environment which the robot and the user enter for the first time (for example, in a search and rescue operation). HMDs offer portability and flexibility since they are often lightweight and can be connected to a wearable computer. However, HMDs can constrict the user's vision due to a relatively low field-of-view, low resolution and possibly latency problems, potentially resulting in hand-eye coordination issues and motion sickness.

While MR has been used for various interaction applications, there has been a limited amount of work relating MR to human-robot interfaces. MR was suggested for tasks of controlling robots, both remotely and directly, increasing the human controller's awareness of the robots' environment and actions [15], [16]. For example, Milgram et. al.'s work in [15] uses MR with a stereographic display to provide a level of tele-presence to a human user controlling a remote robot. The MR elements here are used to augment the user's vision with various computer calculations and information.

Bubblegrams' uniqueness lies in it using MR not necessarily for controlling the robot but also as a collaborative shared medium that is used by both humans and robots to simultaneously interact in the digital and physical domains. We see *Bubblegrams* as a dynamic interface that is linked to the users and the robots rather than to the environment they share at a certain time. Following, we designed *Bubblegrams* with portability and flexibility in mind and decided to implement our prototype using HMD MR visualisation.

Our real-time object detection technique adapts an algorithm published in 2001 by Viola and Jones [9] and later expanded by Lienhart [17]. The technique uses identification and classification of template style features as its method of detection. A machine learning approach selects optimal template features, resulting in an overall effective and efficient object detection algorithm [9].

IV. Vision Algorithm

We use a feature-based approach to real-time object detection [9]. Using a set of sample images, machine learning,

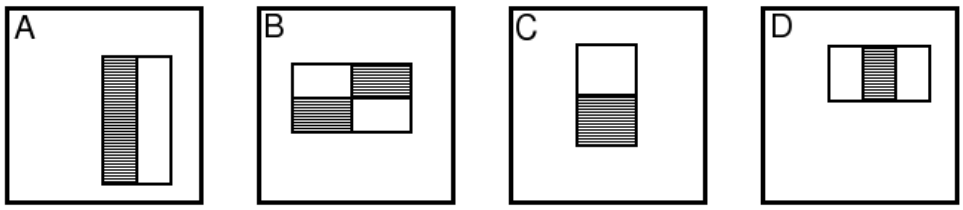

Fig. 3. Haar rectangle features shown relative to an enclosing window. The sum of pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (C). (D) shows a three-rectangle feature and (B) a four rectangle feature [9].

and a divide-and-conquer algorithm, this technique achieves effective object classification. The features used are called Haar-like features, which are rectangular and of varying size, subdivided into white and black regions (see Figure 3).

Using the Haar technique results in more features per image region than pixels. For example, a 24 x 24 window has 576 pixels but 45,396 features [9]; this is because the features encapsulate intensity-distribution domain data about a region. The value of a feature is calculated by subtracting the sum of the pixel intensities in the white regions from the sum of the pixel intensities in the black regions. The feature value, in combination with the feature type, is used as the basis for the feature matching. Figure 4 shows possible features and positions on an AIBO; these features identify the AIBO's darker body above the lighter background and legs, and the darker legs with lighter background in between.

The Haar-like feature detection system uses a cascade of classifiers for object detection (see Figure 5) where each classifier within the cascade is composed of one or more features. Classifiers which allow many false positives are placed at the beginning of the cascade, with the following classifiers being increasingly strict. This results in many image regions being discarded early in the process, while only promising regions are tested against the entire classifier cascade. The speed advantages of this cascade, in combination with a novel image representation technique called the *integral image*, are what enable the detection technique to work in real-time [9].

To build each classifier in the cascade, a training algorithm tests all features against the sample image set. The result is an optimum set of features for each classifier which best meets the pre-decided parameters of the classifier (such as target detection rate) [9]. The premise behind the training algorithm is that the resulting detection rate of the classifier cascade is approximately equal to the product of the detection rates of the individual classifiers. The same is true for the false positive rate. For example, if a cascade had six classifiers, and each classifier has a 50% false positive rate, then the false positive rate of the entire cascade is roughly 0.5^6 or 1.6%.

When training the algorithm, the user decides on the target detection and false positive rates for each classifier, and the number of classifiers in the cascade; the overall cascade approximate rates are calculated as explained. This training method has been shown to be extremely successful in doing real-time face detection with a high accuracy rate [9].

V. Detecting Robots

Applying the Haar-like feature detection technique to *Bubblegrams* is further complicated as robots can be both mobile and autonomous. This means that we can make very few assumptions about their orientation, location, physical shape, or environment. Robots can have dynamic and colourful displays which can change their appearance, and may be made out of a shiny material which may result in random specular lighting effects on their surface. Given that the Haar-like feature technique uses templates to match the shape and

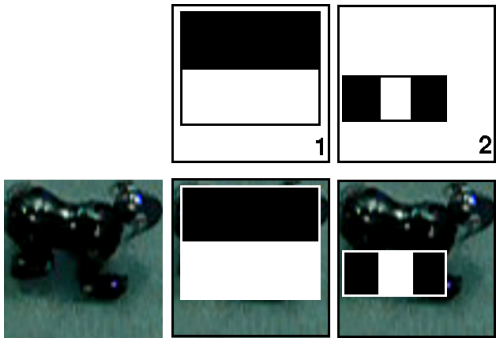

Fig. 4. Possible Haar-like features on the AIBO. Notice how the first feature (1) identifies a dark body over less dark legs and background, and the second feature (2) identifies dark legs with less-dark background in the middle.

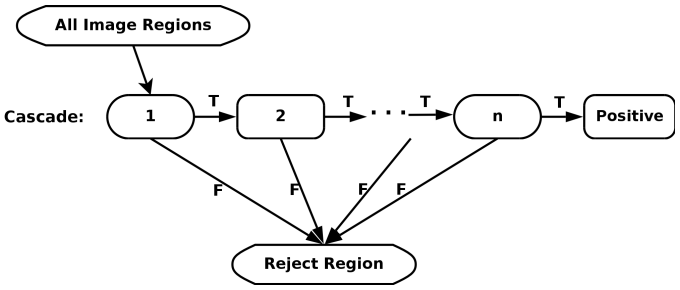

Fig. 5. A Haar-like classifier cascade. At each detector (numbered bubbles) image regions are either rejected (F) or pass (T) to the next step. Regions which pass the final detector are positive hits. Detectors are increasingly difficult, so many regions are quickly rejected earlier, while only promising regions pass through many detectors.

intensity distribution of objects, these issues have serious consequences on the effectiveness of the detector.

The approach that we use to apply the Viola and Jones technique is to divide the detection problem into cases and to add constraints to the robot and the settings. In doing this we target particular circumstances or poses which are much less complex than the general problem. Given the versatility of robots, there are many different possible cases; this number is drastically reduced by constraining the robot to certain task-related poses and environments. The result of this is a lower number of specific problems which are both practically approachable and are more suitable to the Haar-like feature technique than the more general problem. Although constraints are added, this approach allows the problem to be general enough to be effective in detecting robots in dynamic scenarios and environments such as is needed by *Bubblegrams*.

VI. AIBO Specifics

Detection of the AIBO robot dog is sensitive to the same complications detailed in Section V. For example, the AIBO can be sitting, standing on all fours or laying down, and can be facing the user, facing away from the user, or facing sideways. It can also have its head rotated or positioned up or down, can open its mouth and wag its tail, can display an assortment of lights, and can be situated on many surfaces.

In addition to characteristics of the Haar-like feature technique, typical *Bubblegram* interaction scenarios were considered for the division of the AIBO detection problem into multiple cases and the selection of constraints. The resulting cases limit robot shape-change, while allowing rotation only so that the AIBO can change direction and ignoring acceptable changes in scale and lighting conditions. The main constraint placed on the AIBO is that it will always use the same walking pose, whether it is walking or simply standing. While there is movement in the legs when the robot is walking, this eliminates major changes in shape associated with lying down, sitting, etc. The AIBO is also currently restricted from using its LED outputs in order to reduce the amount of change in appearance. We currently only consider the black ERS-7 AIBO on the same in-lab grey carpets, so that changes in contrast between the AIBO and its environment can be minimised.

In an attempt to isolate the different major views of the AIBO, the detection problem is divided into four cases: top, side, front, and back. These cases correspond to interaction using *Bubblegrams*, as a user can be facing an AIBO from any direction or may be looking down at it. While the AIBO is free to move its head for practicality reasons, the overall change in appearance caused by this is much smaller than the change caused by moving or rotating the entire body.

VII. Implementation

To realise the detection system, we used an implementation of the Haar-like feature detection technique included in the Intel Open Computer Vision library [18]. The main steps required when implementing the detector are: creating a database of training images, training and creating classifier cascades from the training images, applying the cascades to images of AIBOs, and extending the system to work on video streams. At each step there are implications of having multiple cases rather than the standard single Haar-like classifier.

A. Image Library

The training of the detection classifiers requires an image database consisting of both positive (with AIBO) and negative (without AIBO) image samples. To collect these samples we used video sequences of both the AIBO and the base environment; from these videos, we extracted more than 1300 positive and negative images. The strategy for negative samples used in this project was to use pictures of the environment where the AIBO will be working. The positive images were finally sorted into the four different classifier cases (top, front, back, side) discussed in Section VI. This separation into classifier cases is the key behind our detector, as it supports the wide range of robotic views required for interacting using *Bubblegrams*.

B. Training

The selection of the target false-positive rate and the target detection rate, as well as the number of classifiers in the cascade, are crucial training parameters. While it may seem reasonable to chose a very low target false-positive rate, lowering this rate increases the strictness of the classifier, forcing it to reject many likely matches; increasing the target detection rate will increase the false-negative rate, while decreasing the target detection rate will increase false-positive rate. The difference between these values is that increasing the false-positive rates emphasises the positive samples, while adding emphasis to the negative samples is done by lowering the target detection rate. Considering *Bubblegrams*, false negatives are preferred over false positives because the detector has many chances per second to find the AIBO, while a false positive could make the system start tracking a couch.

In order to emphasise false negatives, we selected a reasonably high target positive detection rate of 95% for the entire cascade, and an overall cascade target false positive rate of approximately 0.001%.

While the target rates discussed above focus on the correctness of the classifier cascade, the depth of the cascade affects the speed of the classifier. For example, if target rates do not change, a shorter cascade will generally be slower than a longer one. To meet the same overall detection rates, each classifier in the shorter cascade will have to be stricter than the classifiers in the longer cascade; each classifier must be tested against many image regions. However, a cascade which is too long will force promising image regions through a large number of classifiers, decreasing the overall speed of the cascade. Ideally, the cascade length should be selected somewhere between these two extremes in order to optimise speed. For our application we put special focus on this parameter as we have four classifiers running on each image. In combination with other parameters specified here, we received best results with cascades consisting of ten classifiers.

Fig. 6. Example of the Classifier Voting. On the left, various classifiers (using different colours) detect the AIBO is multiple locations. Voting is applied to these regions to find a common consensus. The resulting single region is shown on the right.

C. Detection of AIBO in a Single Frame

The Intel implementation of the Haar-like detector has two parameters that need to be set: *window increase rate*, and *minimum number of hits per window*. The *window increase rate* parameter determines the change in granularity between image scans and the *minimum number of hits per window* determines how many clustered finds are required to form a positive hit. Changing these parameters changes the balance between effectiveness and efficiency; we keep them as low as possible while maintaining our speed requirements.

A key point of our implementation is that we use four classifiers, AIBO top, front, side and back, to detect a single AIBO. Ideally, these classifiers would be mutually exclusive and only one classifier would detect at a time. However, given the variance of a *Bubblegrams* interaction session, there are times when multiple classifiers simultaneously detect the AIBO, either due to similarities between the cases or when a particular AIBO pose falls in between our defined cases. To handle this we have implemented a voting scheme where the positive hits from the various classifiers *vote* on the most likely positive hit. The image region with the most number of *votes* wins, and is selected as the most likely positive hit (see Figure 6). In fact, this technique was so successful that we increased the false positive rate of our classifiers slightly to provide more hits to be used in the *voting*. This shows that using multiple classifiers does not only allow for a wide range of AIBO poses, but offers a detection overlap which is used to increase accuracy.

D. AIBO in Streaming Video

Extending the AIBO detection system to a video sequence offers temporal history as extra information which can be used to improve detection performance and reliability.

We have implemented a tracking algorithm which makes assumptions based on the dynamics of the *Bubblegram* technique: there is a maximum speed at which the AIBO will move between frames and a maximum rate at which the AIBO can change in scale. These parameters are set considering that during *Bubblegrams* interaction, the human user is very likely to keep their attention and focus on the robot.

We also integrate a simple smoothing algorithm which, in the case where a tracked AIBO is lost, assumes that the AIBO does not move. If the AIBO is not re-found after a number of frames, the detector resorts to the single-frame algorithm presented in Section VII-C. This fits with the *Bubblegrams* scenario; while interacting with a robot it is very likely that the robot and user will not move a great deal.

The tracking algorithm proved to be very helpful for *Bubblegrams*. We found that without it, a periodic *false positive* would make the *Bubblegram* jump from the robot to another location (such as a chair), then jump back. When trying to interact with the *Bubblegram*, this made selection and navigation extremely difficult. The addition of the tracking algorithm greatly reduced this problem.

In order to maintain high detection rates for the video stream, the input images were sub-sampled to half-resolution and the detector granularity was decreased. The result was a large gain in speed with minimal loss in effectiveness.

VIII. Preliminary Evaluation

Overall our detection mechanism proved to be successful in its task of finding an AIBO in a *Bubblegram* video sequence. For our preliminary evaluation, we placed the AIBO in a lab environment and ran a random-walk program. A video of the AIBO was recorded in various lab settings and from multiple angles; the setting, camera angles and field of view, all matched the way the AIBO is seen during a *Bubblegrams* interaction session. Based on these sequences, we evaluated the system over a two minute video portion which consisted of both viewer and AIBO movement, varying distances, and busy backdrops. The overall behaviour of the algorithm consists of temporarily losing the AIBO when dramatic movements or changes occur, and then consistently locking-in on the AIBO when the *Bubblegram* interaction scene stabilises. In addition to this, we found that this implementation is resilient to occlusions and cluttered scenes.

We were pleased to find that during the mock interaction sessions in the video, where movement was minimal, the

detection rate was nearly 100% accurate. Overall in our video sequence tests, our system correctly detected the AIBO 79% of the time, with false positives 14% of the time, and no detection 7% of the time. Much of the false positive and no detection time was during motion where the AIBO was not entirely in view, and the images were blurred.

Finally, we did an informal comparison between the detector with multiple cases and tracking, and a single general case. The single classifier was jumpy in comparison with the improved detector and detected the AIBO much less consistently. This suggests the success of our approach in effectively utilising the Haar-like detection system with *Bubblegrams*.

IX. Future Work

The core future work for this project is to continue implementation of the various components of the *Bubblegrams* interface. This includes completion of a networking framework, the *Bubblegrams* graphics engine, and the integration of various interaction techniques. Currently we are working on the visual and flow design of several *Bubblegrams* interfaces for various tasks including household robots, search and rescue tasks and hospital robotic aids.

In terms of the vision algorithm presented in this paper, there are several improvements which we plan to pursue. The current image training set contains just over five hundred images and would be expanded to provide a more complete set. In addition, we plan to implement a more advanced tracking algorithm based on the Kalman filter [19].

X. Conclusions

In this paper we presented our vision algorithm for *Bubblegrams* - a MR-based human-robot interaction technique; we have shown how this detector can be used in demanding and dynamic Human-Robot Interaction scenarios 7. Furthermore, the effectiveness of the detector was significantly increased by constraining the problem and breaking it into cases. These cases not only allowed the specific targeting of various detectors, but gave an amount of *detector overlap* which was used to increase accuracy. Furthermore, temporal information was used from the video stream to further restrict and smooth out the detection, improving the quality of the real-time interaction.

References

[1] J. Scholtz, "Have robots, need interaction with humans!" in *ACM Interactions*. ACM Press, March/April 2005, vol. 12, pp. 13–14.

[2] H. Moravec, *Robot: Mere Machine to Transcendet Mind*. Oxford Press, 1998.

[3] D. Norman, *Emotional Design: Why We Love (or Hate) Everyday Things*. New York, USA: Basic Books, 2004.

[4] S. Kiesler and P. Hinds, "Introduction to This Special Issue on HRI," *Human-Computer Interaction*, vol. 19, no. 1/2, pp. 1–8, 2004.

[5] J. L. Drury, J. Scholtz, and H. A. Yanco, "Awareness in Human-Robot Interactions," in *Proc. SMC '03*. IEEE Press, 2003, pp. 912–918.

[6] B. Giesler, T. Salb, P. Steinhaus, and R. Dillmann, "Using augmented reality to interact with an autonomous mobile platform," in *Proc. ICRA '04*. IEEE Press, 2004, pp. 1009–1014.

[7] M. Billinghurst, H. Kato, and I. Poupyrev, "The MagicBook: Moving Seamlessly Between Reality and Virtuality," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 6–8, May/June 2001.

[8] J. E. Young and E. Sharlin, "Sharing Spaces with Robots: an Integrated Environment for Human-Robot Interaction," in *Proc. ISIE '06*. Microsoft Press, Apr. 2006, pp. 103–110.

[9] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proc. CVPR '01*. IEEE Computer Society, 2001, pp. 511–518.

[10] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. CVPR '05*, vol. 2. IEEE Press, 2005, pp. 590–596.

[11] H. Ishii, C. Ratti, B. Piper, Y. Wang, A. Biderman, and E. Ben-Joseph, "Bringing Clay and Sand into the Digital Design – Continuous Tangible User Interfaces," *BT Technology Journal*, vol. 22, no. 4, pp. 287–299, Oct. 2004.

[12] C. Ratti, Y. Wang, B. Piper, H. Ishii, and A. Biderman, "PHOXEL-SPACE: an Interface for Exploring Volumetric Data with Physical Voxels," in *Proc. of DIS '04*. ACM Press, Oct. 2004, pp. 289–296.

[13] W. Grimson, G. Ettinger, T. Kapur, M. Leventon, W. Wells, and R. Kikinis, "Utilizing Segmented MRI Data in Image-Guided Surgery," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 8, pp. 1367–1397, Feb. 1998.

[14] R. Ramesh, G. Welch, and H. Fuchs, "Spatially Augmented Reality," in *Proc. IWAR '98*. IEEE Press, Nov. 1998, pp. 64–71.

[15] P. Milgram, D. Drasic, and S. Zhai, "Applications of Augmented Reality in Human-Robot Communication," in *Proc. of IROS '93*. IEEE Press, 1993, pp. 1244–1249.

[16] J. Pretlove, "Augmenting Reality for Telerobotics: Unifying Real and Virtual Worlds," *Industrial Robot: An International Journal*, vol. 25, no. 6, pp. 401–407, Oct. 1998.

[17] R. LienHart and J. Maydt, "An Extended Set of Haar-Like Features for Rapid Object Detection," in *Proc. of ICIP '02*. IEEE Press, 2002, pp. 900–903.

[18] Intel, "Open Source Computer Vision Library," WWW, http://www.intel.com/technology/computing/opencv/, Vis. Jan 12, 2006, 2006.

[19] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering, the Transactions of the American Society of Mechanical Engineers, Series D*, vol. 83, no. 1, pp. 35–45, Mar. 1960.

Fig. 7. The prototype in use.