The Use of Harr-like Features in Bubblegrams: a Mixed Reality Human-Robot Interaction Technique

James E. Young, Ehud Sharlin, Jeffrey E. Boyd
Department of Computer Science, University of Calgary, Alberta, Canada
{jyoung;ehud;boyd}@cpsc.ucalgary.ca

Abstract: We present the application of a vision algorithm based on Harr-like features in *Bubblegrams* - a new mixed reality-based human-robot interaction (HRI) technique. *Bubblegrams* allows humans and robots working on collocated synchronous tasks to interact directly by visually augmenting their shared physical environment. *Bubblegrams* uses comics-like interactive graphic balloons or bubbles that appear above the robot's body and allow intuitive interaction with the robot. Users wear light-weight mixed reality goggles that integrate displays and a camera, allowing the user to view and interact with the physical environment as well as with the virtual *Bubblegrams* interface linked to the robot's body. In order to efficiently link *Bubblegrams* in real-time to the physical robot we implemented a vision algorithm based on Harr-like features which is the main topic of this paper. This paper briefly details the design of the *Bubblegrams* interface, the hardware and software we use for the current prototype, and the full details of the vision algorithm.
Keywords: Computer Vision, Mixed Reality, Human-Robot Interaction, Harr-like features

1 Introduction

With the latest rapid advancement of robotic technology, the need for effective human-robot interfaces is becoming clear and pressing [18]. As robots become increasingly capable and intelligent, we can expect to find users sharing their everyday environments with robots in various ways [13, 14]. Human-robot interaction (HRI), a new sub-domain of human-computer interaction (HCI), is an attempt to understand the various issues and problems surrounding interaction with robots and to develop effective interfaces between them and humans [10].

Robots can be viewed as 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 a 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; for example, interaction can be based on physical modalities such as speech-based interaction or digital modalities such as remote control software tools. This separation of interaction spaces can reduce the level of the HRI awareness [2] and ultimately hinder the quality of the resulting interaction between humans and robots [4].

One solution to this problem is the use of mixed reality (MR) as an interaction paradigm between humans and robots. MR is a technique which tracks components of the physical world and augments them with virtual digital entities. Visual augmentation of the physical world is commonly accomplished by projecting images onto the user's environment or by using a head-mounted display (HMD) to synthetically augment the vision of the wearer [1]. We believe that MR can solve many of the interaction problems mentioned above by allowing the robot to dissolve the borderline separating the physical and virtual modalities it uses when interacting with humans. Using MR techniques, robots can superimpose digital information directly onto users' physical environment. At the same time, humans can interact with digital information intuitively, as if this information is an integral part of their physical interaction space.

2 Bubblegrams

In this paper we present *Bubblegrams* - an MR-based interaction technique that combines physical

and virtual interaction spaces and allows users to interact with robots simultaneously in the digital and physical realms. *Bubblegrams* appears as visual cartoon-like balloons or bubbles above the robot's head or body. In order to view and interact with *Bubblegrams* the user wears mixed reality goggles which combine miniature displays and a web camera (see Figure 1). *Bubblegrams* can be used by the human for direct access to the robot's status and functions. For example, in a home-environment application a robot that just completed a cleaning chore can present a smiley bubble above its head, showing its satisfaction of fulfilling the task as well as allowing the user to choose the robot's future course of action (for example, "keep cleaning", "come and play with me" etc.). In a search and rescue operation a fire-fighter can send a robot ahead into the next room keeping line-of-sight connection with the robot's *Bubblegrams* which display a video feed from the robot thermal imager as well as an interface that enables the fire-fighter to send the robot further into the room or to call it back.

Our current implementation of the MR goggles integrates an Icuiti HMD[6] and webcam (as shown in Figure 1) as the mixed reality visual interface. This interface is powered by a tablet PC (Toshiba Portege, Pentium Centrino 1.7GHz) which offers both portability and wireless internet connectivity to the system. In addition, the tablet PC can be used as one of many possible methods to interact with *Bubblegrams*, for example by using the stylus. For the robotic element, we are using a Sony AIBO robot dog (Black, model ERS-7). The AIBO, which utilises the wireless network, generates *Bubblegrams* and conveys them to the user system through the network connection. This network connection is then also used as the communication medium for the various interaction techniques.

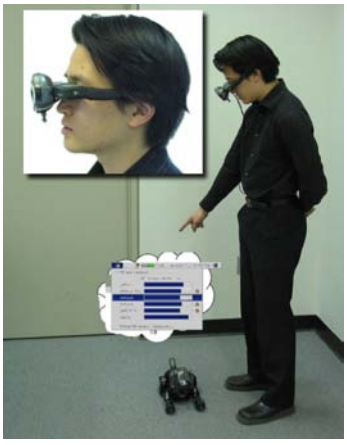

Figure 1: A user and robot team interacting with a *Bubblegram*.

For *Bubblegrams* to be effective we need to physically associate the interactive balloon with the physical robot, in the visual field of the user. For this we need to efficiently track the robot in real-time through the user's MR goggles vision channel. In this paper we focus on the use of a particular object detection technique for solving the problem of real-time detection and tracking of a Sony AIBO robot dog in a video sequence. The technique, based on the Viola and Jones "Rapid Object Detection Using a Boosted Cascade of Simple Features" [19] paper, uses machine learning to develop robust and flexible classifiers for detecting objects in still images. The technique uses novel image representations to obtain very fast detection speeds; our reported algorithm based on the implementation of a single classifier obtained a rate of more than 40 frames per second using an Intel Pentium 4 3.4GHz PC, analysing a 320 pixels by 200 pixels video, with very high detection success. In the coming sections we briefly present an overview of research related to our efforts, we then describe our approach to the problem of locating an AIBO in streaming video, and detail our algorithm implementation and preliminary results. Other aspects of the *Bubblegrams* system are not presented in this paper.

3 Related Work

Mixed reality (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 [1] and the ARTag system[3]), modelling volumetric data [8, 17], assisting with medical surgery [5] and as a computer supported cooperative work (CSCW) interaction theme and environment [16].

We can crudely classify MR techniques as either based on head-mounted-display (HMD) visualisation or projective visualisation. Projective visualisation can be integrated seamlessly into a users' 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, being often heavy and difficult to move, and require a projection surface and appropriate lighting. One can envision an MR envi-

ronment based on projection techniques in a dedicated space that is designed and crafted especially for the task, but it is still difficult to implement a 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). HMD visualisation offers portability and flexibility since modern HMDs are 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, all potentially resulting in hand-eye coordination issues and possibly motion sickness.

While mixed reality has been used for various interaction applications, there has been a limited amount of work using mixed reality for human-robot interaction (HRI). Mixed Reality was suggested for tasks of controlling robots, both remotely and directly [12, 15]. This work uses mixed reality to increase the human controller's awareness of the robots' environment and actions. For example, Milgram et. al.'s work in [12] uses mixed reality with a stereographic display to provide a level of tele-presence to a human user controlling a remote robot. The mixed reality 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.

We based *Bubblegrams*' real-time object detection technique on a previous algorithm published in 2001 by Viola and Jones [19] and later expanded by Lienhart [11]. The technique uses identification and classification of template style features as its method of detection. A machine learning approach is employed to select optimal template features, resulting in an overall effective and efficient object detection algorithm.

4 Vision Algorithm

Bubblegrams uses a feature-based approach to real-time object detection[19]. Using a data set of sample images, the technique uses machine learning and a divide-and-conquer algorithm for effective and efficient object classification. The features used are called Harr-like features, based on Harr basis functions, which are spatial rectangular features of varying size, subdivided into white and black regions (see Figure 2).

Using the Haar feature detection technique results in more features per image region than pixels. For example, a 24x24 window has 576 pixels but 45,396 features[19]; 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 3 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, or series, of classifiers for object detection (see Figure 4) where each classifier within the cascade is composed of one or more features. Simple classifiers which allow many false positives are placed at the beginning of the cascade, with the following classifiers being increasingly complex and strict. This results in most image regions being dis-

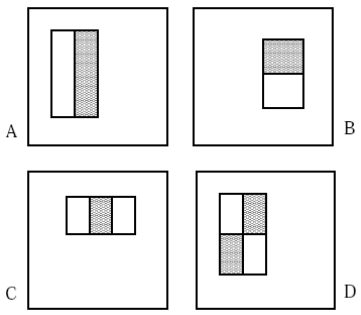

Figure 2: Example rectangle features shown relative to the enclosing window. The sum of pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). (C) shows a three-rectangle feature and (D) a four rectangle feature[19].

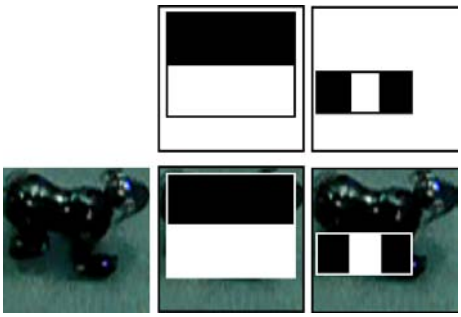

Figure 3: Possible Harr-like features on the AIBO. Notice how the first feature finds a dark body over less dark legs and background, and the second feature finds dark legs with less-dark background in the middle.

carded early in the detection 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[19].

To build each classifier in the cascade, computer training is used to test the entire feature set against the set of sample images. The result is an optimum (as explained in [19]) configuration of features for each classifier, a configuration which best meets the pre-decided parameters of the classifier. 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 resulting overall approximate rates are easily calculated. This training method has been shown to be extremely successful in doing real-time face detection with a high accuracy rate[19].

5 Problem and Approach

The problem of detecting a particular robot is further complicated by the fact that robots are often both mobile and autonomous. This means that we can make very few assumptions about their orientation, location, physical shape, or environment. Robots often 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.

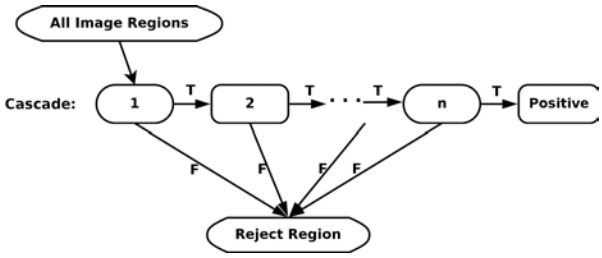

Figure 4: A Haar-like classifier cascade.

The approach that we use to simplify detection is to break the problem into cases and to add constraints to the robot and situation. By dividing the problem into cases we create specific detection problems which target particular circumstances or poses and are much less complex than the general problem. However, given the versatility of robots, there are many different possible cases. By constraining the robot to certain task-related poses and environments, the number of realistic cases to be considered can be drastically reduced, resulting in a lower number of specific problems which we can practically approach.

6 AIBO Specifics

Detection of the AIBO robot dog is sensitive to the same complications detailed in Section 5.

Detecting the Sony AIBO robot dog is no exception to the complications mentioned in the previous section. 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 kinds of surface.

When breaking the AIBO detection problem into multiple cases and adding constraints, we considered the strengths and weaknesses of the detector used. As such, we generally limited shape-change and rotation within a particular case, but ignored reasonable changes in scale and lighting conditions. In addition to this, the parameters used in the training process were selected to try and achieve a bal-

ance between cascade depth and classifier complexity, fine tuning the balance between efficiency and effectiveness.

In our current prototype 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 while 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. Similarly, only a black model ERS-7 AIBO is being used, so there is no need to consider other models or colours. Lastly, the AIBO is always on the same flooring (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 broken into four cases: top, side, front, and back. While the AIBO is still free to move its head for practicality reasons, the overall change in appearance caused by moving the head is much smaller than the change caused by moving or rotating the entire body.

7 Implementation

To realise the detection system, we used an implementation of the Haar-like feature detection technique included in the freely available Intel Open Computer Vision library (Intel OpenCV)[7]. The main steps required when implementing this 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.

7.1 Image Library

The training of the detection classifiers requires a complete image database consisting of both positive (images with an AIBO) and negative (images without an AIBO) samples. To collect these samples we used a video camera to capture 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. These images were finally sorted into the four different classifier cases discussed in Section 6.

7.2 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. Therefore, a balance must be found which has few false positives while reliably detecting the AIBO. 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 changing these values is that increasing the false positive rates adds emphasis to the positive image samples, while lowering the target detection rate adds emphasis to the negative image samples.

In order to put emphases on the correctness of the negative image samples, 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 and reliability of the classifier cascade, altering the depth of the cascade affects the speed of the classifier. Changing the number of classifiers in the cascade changes the distribution of the cascade. For example, assuming that target rates do not change, a shorter cascade will generally be slower than a longer cascade. To meet the same overall detection rates, each classifier in the shorter cascade will have to meet stricter requirements than the classifiers in the longer cascade. Also, the complex and slow classifiers in the shorter cascades must be tested against many image regions.

However, a cascade which is too long will force promising or positive 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 training, and in combination with other parameters specified here, we received best results with cascades consisting of ten classifiers.

7.3 Detection of AIBO in a Single Frame

Our implementation of the AIBO detector utilises Intel's OpenCV library[7] for basic object detection. The library provides a nearly automatic implemen-

Figure 5: Example of voting scheme. The image on the left represents the results from the various classifiers: red=Top, green=Back, right/left=Yellow, front=Blue. The image on the right shows the resulting match after voting.

tation given a particular classifier cascade, with only two parameters which need to be set: *window increase rate*, and *minimum number of hits per window*. Note that the detection algorithm searches the image space at varying scales; the *window increase rate* parameter determines the change in scale between searches. The *minimum number of hits per window* parameter is related to the detection process; while searching the image, the detector marks positive hits. Next, closely overlapping regions are combined to form a positive hit. The *minimum number of hits per window* parameter determines the number of overlapping hits required to form a positive hit.

Altering the *window increase rate* parameter changes the balance between effectiveness and efficiency; increasing this rate increases speed while finding fewer hits, and decreasing this rate does the opposite. For our implementation, we keep this parameter as low as possible while maintaining our speed requirements.

A unique point of our implementation is that we have four classifiers, AIBO top, front, side and back, attempting to detect a single AIBO. Ideally, these classifiers would be mutually exclusive and only one classifier would detect at a time. However, realistically there are many 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, and assuming there is only one AIBO in the scene, 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 5). In fact, this technique was so successful that we increased the false positive rate of our classifiers to provide more hits to be used in the *voting*.

7.4 AIBO in Streaming Video

Extending the AIBO detection system to a video sequence was largely similar to finding an AIBO in an image. However, a video stream offers temporal history as an extra dimension of information which can be used to improve detection performance and reliability.

We have implemented a simple tracking algorithm which makes the following assumptions based on the dynamics of the task and the MR system: there is a maximum speed at which the AIBO can move between frames, and there is a maximum rate at which the AIBO can change in scale. These assumptions test the location and size of a matching region against the last known AIBO match. We also integrate a simple memory mechanism which keeps the last detected match in case no AIBO was found. This technique utilises a timeout so that if no AIBO is found for a period of time (currently 0.5 seconds) then the detector resorts to the single-frame algorithm presented in Section 7.3, until an AIBO is found.

Processing speed is crucial for detection in a video. Using the Section 7.3 algorithm, an Intel

Pentium Centrino 1.7GHz could only detect at a frame per second on a 640 by 480 image. Profiling the program revealed that 92% of the runtime was being spent in the Harr-like feature detection, and our solution was to lower the detector quality to increase the detector speed.

Following, two changes were made: the input images are scaled to half resolution (320,240) before running the detector, and the *window increase rate* (discussed in Section 7.3) was increased to 20%. With these two changes, speed was increased to seven frames per second with no quality loss observed.

8 Preliminary Evaluation

Overall our system proved to be successful in its task of finding an AIBO in a video sequence. For our preliminary evaluation, we placed the AIBO in our lab environment and ran a random-walk program. A video stream 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 will be 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. We were pleased to find that during the mock interaction sessions in the video, where movement was minimal, the detection rate was nearly 100%. Overall in the video sequence, 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.

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 interaction scene stabilises.

In addition to this, we found that this implementation is fairly resilient to occlusions and difficult situations, as shown in Figure 6.

9 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, a *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.

Figure 6: Screenshots of successful detections in difficult scenes.

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 [9].

10 Conclusions

In this paper we presented *Bubblegrams* - a mixed reality-based human-robot interaction technique which allows users and robots to intuitively share physical and virtual information and functions in a collocated synchronous task domain. *Bubblegrams* uses a mixed reality interface to link an interactive cartoon-like balloon with the robot, allowing the user direct access to the robot's condition, parameters and functions as long as the robot is in the user's field of view.

The paper details our current *Bubblegrams* hardware setup based on an HMD mixed reality interface and the implementation of a tracking algorithm for dynamically locating the robot, linking it with *Bubblegrams* whenever the robot is in the user's field of view. The tracking algorithm and the results of a preliminary study of its effectiveness are detailed.

References

[1] Billinghurst, M., Kato, H., and Poupyrev, I. The MagicBook: Moving Seamlessly Between Reality and Virtuality. *IEEE Comput. Graph. Appl. 21*, 3 (2001), 6–8.

[2] Drury, J., Scholtz, J., and Yanco, H. Awareness in human-robot interactions. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics* (October 2003).

[3] Fiala, M. ARTag, a Fiducial Marker System Using Digital Techniques. In *CVPR '05: Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (DC, USA, 2005), IEEE Computer Society, pp. 590–596.

[4] Giesler, B., Salb, T., Steinhaus, P., and Dillmann, R. Using Augmented Reality to Interact with an Autonomous Mobile Platform. In *Proceedings ICRA '04. IEEE Int. Conf. Robotics and Automation* (2004).

[5] Grimson, W., Ettinger, G., Kapur, T., Leventon, M., Wells, W., and Kikinis, R. Utilizing Segmented MRI Data in Image-Guided Surgery. *International Journal of Pattern Recognition and Artificial Intelligence 11*, 8 (February 1998), 1367–1397.

[6] Icuiti. Hmd dv920. WWW, http://www.icuiti.com, Visited Feb 9, 2006., 2006.

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

[8] Ishii, H., Ratti, C., Piper, B., Wang, Y., Biderman, A., and Ben-Joseph, E. Bringing Clay and Sand into the Digital Design – Continuous Tangible User Interfaces. *BT Technology Journal 22*, 4 (Oct. 2004), 287–299.

[9] Kalman, R. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, the Transactions of the American Society of Mechanical Engineers, Series D 83*, 1 (1960), 35–45.

[10] Kiesler, S., and Hinds, P. Introduction to the special issue on human-robot interaction. *Special Issue of Human-Computer Interaction 19*, 1,2 (2004), 1–8.

[11] LienHard, R., and Maydt, J. An extended set of haar-like features for rapid object detection. In *Proceedings of the IEEE International Conference on Image Processing 2002* (2002).

[12] Milgram, P., Drasic, D., and Zhai, S. Applications of Augmented Reality in Human-Robot Communication. In *proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1993)* (1993), pp. 1244–1249.

[13] Moravec, H. *Robot: Mere Machine to Transcendent Mind*. Oxford University Press, Oxford, UK, 1999.

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

[15] Pretlove, J. Augmenting Reality for Telerobotics: Unifying Real and Virtual Worlds. *Industrial Robot: An International Journal 25* (1998), 401–407.

[16] Ramesh, R., Welch, G., and Fuchs, H. Spatially Augmented Reality. In *First IEEE Workshop on Augmented Reality (IWAR 98)* (San Francisco, CA, November 1 1998).

[17] Ratti, C., Wang, Y., Piper, B., Ishii, H., and Biderman, A. PHOXEL-SPACE: an Interface for Exploring Volumetric Data with Physical Voxels. In *Proceedings of Designing Interactive Systems (DIS 2004)* (4 Oct. 2004).

[18] Scholtz, J. Have robots, need interaction with humans! In *ACM Interactions* (March-April 2005), pp. 13–14.

[19] Viola, P., and Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *CVPR '05: Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (2001).