

# A Virtual Mirror Interface using Real-time Robust Face Tracking

T. Darrell, G. Gordon, J. Woodfill, M. Harville  
Interval Research Corp.  
1801C Page Mill Road  
Palo Alto CA 94304  
trevor,gggordon,woodfill,harville@interval.com

## Abstract

We describe a virtual mirror interface which can react to people using robust, real-time face tracking. Our display can directly combine a user's face with various graphical effects, performed only on the face region in the image. We have demonstrated our system in crowded environments with open and moving backgrounds. Robust performance is achieved using multi-modal integration, combining stereo, color, and grey-scale pattern matching modules into a single real-time system. Stereo processing is used to isolate the figure of a user from other objects and people in the background. Skin-hue classification identifies and tracks likely body parts within the foreground region. Face pattern detection discriminates and localizes the face within the tracked body parts. We show an initial application of the mirror where the user sees his or her face distorted into various comic poses. Qualitatively, users of the system felt the display "knew" where their face was, and provided entertaining imagery. We discuss the failure modes of the individual components, and quantitatively analyze the face localization performance of the complete system with thousands of users in recent trials.

## 1 Introduction

The creation of displays or environments which passively observe and react to people is an exciting challenge for computer vision. Faces and bodies are central to human communication and yet machines have been largely blind to their presence in real-time, unconstrained environments.

To date, research in computer vision for person tracking has largely focused on exploiting a single visual processing technique to locate and track features of a user in front of the display. These systems have often been non-robust to real-world conditions and fail in complicated, unpredictable visual environments and/or where no prior information about the user population is available. The integration of the three

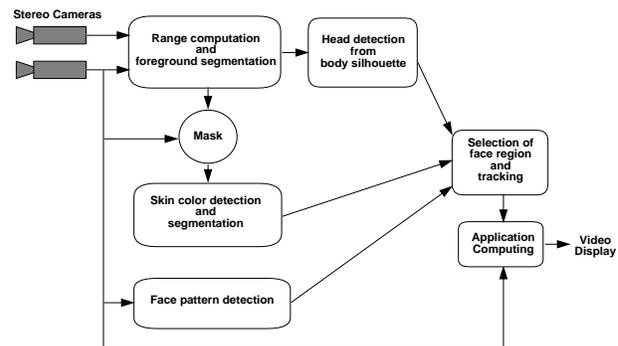


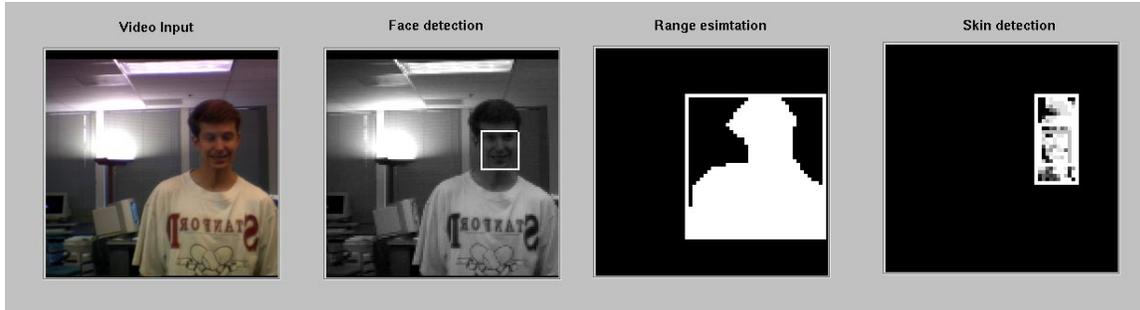
Figure 1. Vision processing module overview.

cues in our system is unique, but for related work see [8] which approaches the tracking task using color and motion.

We have created a visual person tracking system which achieves robust real-time performance through the integration of multiple visual processing modalities—stereo, color, and pattern. In the following section we describe each of these component modules as well as the framework within which they are integrated. We will then describe an initial application of our tracking system, an interactive video mirror. Finally we will show the results of our system as deployed with naive users, and analyze both the qualitative success of the application and the quantitative performance of the tracking algorithm.

## 2 Multi-modal head tracking

We use three primary vision modules to track a user's head position: depth estimation, color segmentation, and intensity pattern classification. Figure 1 shows an overview of how the modules are integrated. Depth information is estimated from multiple fixed cameras and allows easy segmentation of the user from other people and background objects. An intensity-invariant color classifier detects regions of flesh tone on the user and is used to identify likely body part regions. Finally a face detection module is used to



**Figure 2. Vision processing results: input image, face detection result, foreground region computed from range data, and skin hue classification score image.**

discriminate head regions from hands, legs, and other body parts. Knowledge of the location of the user's head in 3-D is passed to the application. Figure 2 shows the output of the various vision processing modules on a scene with a single person present. Our system is designed to function successfully when one or two modules are unavailable or unreliable, as described below.

## 2.1 Real-time dense stereo processing

Video from a pair of cameras is used to estimate the distance of people or other objects in the world using stereo correspondence techniques. The census correspondence algorithm [11] determines similarity between image regions, not based on inter-image intensity comparisons, but rather based on inter-image comparison of intra-image intensity ordering information. The census algorithm involves two steps: first the input images are transformed so that each pixel represents its local image structure; second, the elements of these transformed images are put into correspondence, producing a disparity image.

We have implemented the census algorithm on a single PCI card, multi-FPGA reconfigurable computing engine [12]. This stereo system is capable of computing 24 stereo disparities on 320 by 240 images at 42 frames per second, or approximately 77 million pixel-disparities per second. These processing speeds compare favorably with other real-time stereo implementations such as [4].

Given dense depth information, a silhouette of a user is found by selecting the nearest human-sized surface and then tracking that region until it disappears. Our segmentation and grouping technique proceeds in several stages of processing. We first smooth the raw range signal using a morphological closing operator to reduce the effect of low confidence stereo disparities. We then compute the response of a gradient operator on the smoothed range data. We threshold the gradient response above a critical value, and multiply the inverse of this thresholded gradient image (a binary quantity) with the smoothed range data. This creates regions of zero value in the image where abrupt tran-

sitions occur, such as between people. We finally apply a connected-components grouping analysis to this separated range image, marking contiguous regions with distinct integer labels.

This processing is repeated with each new set of video frames obtained from the video cameras. After a new set of regions is obtained, it is compared to the set obtained for the previous frame. Temporal correspondences are established between regions through time on the basis of true surface area and proximity. We mark a particular region as the target person and follow it until it leaves a defined workspace area; we then select a new target by choosing the nearest depth region.

This depth information is used to isolate figure from ground, so that the color and face detection modules described below are not confused by clutter from background content or other users who are not currently being tracked. (We are also currently extending our system to simultaneously track and process several users.) Specifically, we use the recovered connected component target region as a boolean mask which we apply to images from the primary camera before passing them to the color and pattern matching modules.<sup>1</sup> It is possible to estimate head location using just the peaks of a person's silhouette computed from a range region [10]; when color and face information are not available, we use this estimate to determine head position. In all modes of the system, we also use range to constrain estimated face size; if the estimated real size of a face is not within one standard deviation of average head size, we use the projected average size to set the the head size (but not position).

<sup>1</sup>If the range segmentation system returns no connected component region, the mask defaults to be a fixed range filter. In this mode the mask image is set to 1 wherever in the range image values are within a defined interval, usually the region of space immediately in front of the display. This allows the system to still perform under conditions which are difficult for our segmentation method, such as when the user is wearing clothes which show no contrast on which to compute stereo correspondence. In practice this occurred very infrequently.

## 2.2 Flesh-hue classification

Within the foreground depth region of a particular user, it is useful to mark regions that correspond to skin color. We use a classification strategy which matches skin hue but is largely invariant to intensity or saturation, as this is robust to different lighting conditions and absolute amount of skin pigment in a particular person.

We apply color segmentation processing to images obtained from the primary camera. Each image is initially represented in terms of the red, green, and blue channels of the image. It is converted directly into a “log color-opponent” space similar to that used by the human visual system. This space can directly represent the approximate hue of skin color, as well as its log intensity value. Specifically,  $(R, G, B)$  tuples are converted into tuples of the form  $(\log(G), \log(R) - \log(G), \log(B) - (\log(R) + \log(G))/2)$ . We use a classifier with a Gaussian probability model; mean and full covariance are estimated from training examples for a “skin” class and a “non-skin” class. When a new pixel  $p$  is presented for classification, the likelihood ratio  $P(p = \text{skin})/P(p = \text{non} - \text{skin})$  is computed as a classification score. Our color representation is similar to that used in [3], but we estimate our classification criteria from examples rather than apply hand-tuned parameters.

For computational efficiency at run-time, we precompute a lookup table over all input values, quantizing the classification score (skin similarity value) into 8 bits and the input color channel values to 6, 7, or 8 bits. This corresponds to a lookup table which ranges between 256K and 16MB of size. This is stored as a texture map, if texture mapping hardware supports the ability to apply “Pixel Textures”, in which each pixel of an input image is rendered with uniform color but with texture coordinates set according to the pixel’s RGB value.<sup>2</sup> Otherwise a traditional lookup table operation is performed on input images with the main CPU.

After the lookup table has been applied, segmentation and grouping analysis are performed on the classification score image. The same algorithm as described above for range image processing is used, except that the definition of the target region is handled differently. The target region (the face) is defined based on the results of the face detector module, or via range shape analysis, described below. (Should color be the only module available, we fall back to simply choosing the highest color region whose size agrees with the proportion of an upright face.) Connected-component regions are tracked from frame to frame as in the range case, with the additional constraint that a size constancy requirement is enforced: temporal correspondence is not permitted between regions if their real size changes

<sup>2</sup>See the SGI Developer Notes, July ’97. The use of texture mapping hardware for color classification can offer dramatic speed and lookup table size advantages due to built-in interpolation; however at present the Pixel Texture feature is only supported on the SGI Octane series.

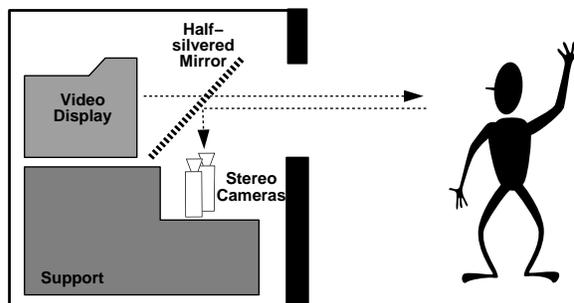


Figure 3. Display and viewing geometry: cameras and video display share optical axis through a half-silvered mirror.

more than a specified threshold amount. Overall, this algorithm allows us to infrequently identify the face region using the face detection or range shape criteria, but still track it through all frames using color information.

## 2.3 Face pattern discrimination

Stereo and color processing provide signals as to the location and shape of the foreground user’s body and hands, faces, and other skin tone regions (clothing or bags are also possible sources of false positives). To distinguish head from hands and other body parts, and to localize the face within a region containing the head, we use pattern recognition methods which directly model the statistical appearance of faces.

We based our implementation of this module on the CMU face detector [9] library. This library implements a neural network which models the appearance of frontal faces in a scene, and is similar to the pattern recognition approach described in [7]. Both methods are trained on a structured set of examples of faces and non-faces. We use the face detector to identify which flesh color region contains the face, and the relative offset of the face within that region. Our experience showed the CMU face detector to be remarkably robust across different imaging and lighting conditions at identifying frontal view faces.

In the simplest cases, the face detector identifies which flesh color regions correspond to the head, and which to other body parts. When a face is detected to overlap with a skin color region, we mark that region as the “target”, and record the relative offset of the face detection result within the bounding box of the color region. The target label and relative offset persist as long as the region is tracked as in Section 2.2. Thus if the face detector cannot find the face in a subsequent frame, the system will still identify the target color region, unless it has left the scene, become occluded, or violated the size change constraint imposed on color region tracking.



**Figure 4. Distortion effects applied to user shown in Figure 2: spherical expansion, spherical shrinking, swirl, lateral expansion, and vertical sliding effect.**

When a color region does change size dramatically, we perform an additional test to see if two regions in fact performed a split or merge relative to the previous frame. If this has occurred (we simply compare the bounding box of the new region to the bounding boxes of the previous region), we attempt to maintain the face detection target label and subregion position information despite the merge or split. In this case we make the assumption that the face did not actually move, compute the absolute screen coordinates of the face subregion in the previous frame, and re-evaluate which region it is in in the current frame. We also update the subregion coordinates relative to the newly identified target region. The assumption of stationary face is not ideal, but it works in many cases where users are intermittently touching their face with their hands.

If there was no face pattern detected corresponding to a likely skin region, we optionally check to see if a face region can be inferred from the overall configuration of the skin color regions and near range regions. We test each color region to see if corresponds to a peak in the selected range silhouette. If so, we label this region to be the target. If not, we use the estimate of the head computed from the range silhouette alone.

### 3 A Virtual Mirror Display

Our initial application of our integrated, multi-modal visual person tracking framework is to create an interactive visual experience. We have created a virtual mirror which distorts and exaggerates the facial expression of people observing the device.

We create a virtual mirror by placing cameras so that they share the same optical axis as a video display, using a half-silvered mirror to merge the two optical paths. Since we are using stereo processing, we use multiple cameras to observe the user: the primary color camera is mounted in the center of the imaging frame and an additional camera is mounted off-axis. The cameras view the user through a right-angle half-silvered mirror, so that the user can view a video monitor while also looking straight into (but not seeing) the cameras. Video from the primary camera is displayed on the monitor, after the various distortion effects

described below, so as to create a virtual mirror effect. Figure 3 shows the display and viewing geometry of our apparatus. With an estimate of the position of the user’s head in 3-D from the tracking system, graphics techniques to distort and/or morph the shape or apparent material properties of the user’s face can be applied; this creates a novel and entertaining interactive visual experience.

Interactive facial distortion has been explored before on static imagery (such as the software imaging tool “Kai’s Power Goo” by Metatools, Inc.). Performing the effect in video is qualitatively different from still image processing in terms of the entertaining quality of the device. The live image of one’s face evokes a quality of being connected and disconnected at the same time; by distorting that face in real-time, we create a self-referential experience with an image that is clearly neither oneself, nor is it entirely synthetic or autonomous. Users seem to find the effect entertaining and interesting, and are willing to make quite exaggerated expressions to see how they appear in distorted form.

#### 3.1 Graphics Processing

Video texture mapping techniques[5] are used to implement the distortion of the user’s face. For this discussion we assume that texture and position coordinates are both normalized to be over  $[0,1]$ . We define a vertex to be in “canonical coordinates” when position and texture coordinates are identical. To construct our display, a background rectangle is set to cover the display (from 0,0 to 1,1) in canonical coordinates. This alone creates a display which is equivalent to a non-distorted, pass-through, video window. To perform face distortions, a smaller mesh is defined over the region of the user’s head. Within the external contour of the head region, vertices are placed optionally at the contour boundary as well as at evenly sampled interior points. Initially all vertices are placed in canonical coordinates, and set to have neutral base color.

Color distortions may be effected by manipulating the base color of each vertex. Shape distortions are applied in one of two modes: parametric or physically-based. In the parametric mode distortions are performed by adding a deformation vector to each vertex position, expressed as a

weighted sum of fixed basis deformations. In our application these bases are constructed so as to keep the borders of the distortion region in approximately canonical coordinates, so that there will be no apparent seams to the video effect. In the physically-based mode forces can be applied to each vertex and position changes are computed using an approximation to an elastic surface; a vertex can be "pulled" in a given direction, and the entire mesh will deform as if it were a rubber sheet.

The weight parameters associated with parametric basis deformations vary over time, and can be expressed as a function of several relevant variables describing the state of the user: the distance of the user to the screen; their position on the floor in front of the display, or their overall body pose. In addition the weight parameters can vary randomly, or according to a script or external control. Forces for the physically-based model can be input either with an external interface, randomly, or directly in the image as the user's face touches other objects or body parts.

### 3.2 Implementation Details

We implemented our system for SIGGRAPH'97 using three computer systems (one PC, two SGI O2), a large NTSC video monitor, stereo video cameras, a dedicated stereo computation PC board, and an optical half-mirror. The monitor, mirror, and cameras are arranged such that the camera and monitor share the same optical axis: the user can stare into the camera and display simultaneously, but sees only the monitor output. Depth estimates are computed on the stereo PC board based on input from the stereo cameras, which is sent over a network from the PC to the first SGI at approx. 20Hz for 128x128 range maps. On this SGI color video is digitized at 640x480 and used as a texture source for the distortion effect. Skin color lookup and connected components analysis is performed at 20Hz at 128x128 resolution.

The color segmentation classifier was trained across various lighting conditions at the demonstration site by taking images of a reference color sample grid, as well as images of people and background scenes.

A second SGI O2 performed face detection routines: at 128x128 resolution it takes approximately 0.8 seconds to find all faces in a typical scene.

The output image is constructed by applying the acquired video as a texture source for the background rectangle and the face mesh. The full system, including all vision and graphics processing, runs at approximately 12Hz.

For this demonstration four parametric deformations and one physically-based distortion were implemented: a spherical expansion, spherical shrinking, swirl, and a lateral expansion were defined as bases, and a vertical sliding effect implemented using simulated physics. Figure 4 shows the basic effects generated by our system.

Modules Enabled			Misses	False Positives	Correct	Error Rate
Color	Range	Pattern				
✓	✓	✓	0	11	321	3.3%
✓	✓		0	12	320	3.6%
	✓	✓	0	14	318	4.2%
	✓		0	18	314	5.4%
✓		✓	3	39	290	12.7%
✓			3	49	280	15.7%
		✓	259	2	71	78.6%

**Table 1. Face detection and localization results using different combination of input modules, ordered by increasing error rate. Results from the pattern module alone are not representative of previous analyses - see end of Section 4.**

## 4 Face Tracking Results

The goal of the visual tracking portion of our system is to identify the 3-D position and size of a user's head in the scene, so that distortion and other effects can be applied. We have analyzed the performance of our system, both with on-line tests with thousands of users and with off-line tests where we could quantitatively analyze the performance of the system with different modules enabled or disabled.

Our system was first demonstrated at SIGGRAPH'97 from Aug 3-8, 1997 [1], with an estimated 5000 people over 6 days experiencing our system (approx. two new users per minute, over 42 hours of operation). Qualitatively, the system was a complete success. Our tracking results were sufficient to localize video distortion effects that were interesting and fun for people to use. Figure 5 shows typical images displayed on the virtual mirror. The system performed well with both single users and crowded conditions; in an initial report of the system [2] we estimated that correct head placement occurred in 85-95% of cases we surveyed. In this version of the system the only module that was allowed to fail was the face detector.

As described above we can now produce estimates of face location using any combination of modules. In addition to improving our overall performance, this allows us to analyze the system under different failure conditions, and assess the value of our integration strategy. We stored range and color images at sparse intervals from our system during the SIGGRAPH show; we analyzed performance off-line on those which contained images of people. (Performance of our system on blank scenes is in excess of 99% correct—no effect is generated.)

Table 1 summarizes the results we found. A correct match was defined when the corners of estimated face region was sufficiently close to manually entered ground truth (within  $\frac{1}{4}$  of the face size.) Overall, when all modules were functioning we achieved an error rate of 3.3%. When any one module was allowed to fail, error was still less than



Figure 5. Typical scenes seen in the virtual mirror.

13%. In terms of individual performance, the range module was best (5.4% error), followed by the color module (15.7%), and the face module (78.6%).

We draw two main conclusions from these data; first, that range data is a powerful cue to localizing heads in complex scenes, as is flesh color detection. Second, integration is useful: in every case, the addition of modules improved the system performance significantly. The addition of color or face detection to the range system reduced error by approximately 30%.

We note that the solo performance of the pattern based face detector presented here show significantly more errors than in previous analyses [9]. Several issues impacted these results. First, the CMU system was trained with primarily upright, frontal view faces, without extreme facial expressions. These expectations match a broad range of application scenarios. However, the distortions presented to our users encouraged them to make large head rotations and greatly exaggerated facial expressions. These poses, largely atypical of the training material, and were therefore not detected as probable face patterns. A second relevant factor was image resolution. Informal inspection of detection results indicated that many errors occurred when the face was less than 13 pixels across. While we cannot remedy this in the current dataset we anticipate that performance will improve when higher-resolution imagery are used as input to the face detector. Overall, however, in concert with skin color tracking and depth segmentation, the face detection module provides information essential for robust performance.

## 5 Summary

We have demonstrated a system which can respond to a user's face in real-time using completely passive and non-invasive techniques. Robust performance is achieved through the integration of three key modules: depth estimation to eliminate background effects, skin color classification for fast tracking, and face detection to discriminate

the face from other body parts. Our system would be valuable for applications in interactive entertainment, telepresence/virtual environments, and intelligent kiosks which respond selectively according to the presence and pose of a user. We hope these and related techniques can eventually balance the I/O bandwidth between typical users and computer systems, so that they can control complicated virtual graphics objects and agents directly with their own expression.

## References

- [1] Darrell, T., Gordon, G., Woodfill, W., Baker, H., "A Magic Morphin Mirror", SIGGRAPH '97 Visual Proceedings, ACM Press. 1997.
- [2] Darrell, T., Gordon, G., Woodfill, W., Baker, H., Harville M., "Robust, real-time people tracking in open environments using integrated stereo, color, and face detection", Workshop on Visual Surveillance, International Conf. on Computer Vision, 1998.
- [3] Fleck, M., Forsyth, D., and Bregler, C., "Finding Naked People", European Conf. on Computer Vision, Vol II, pp. 592-602. 1996.
- [4] Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M., "A Video-Rate Stereo Machine and Its New Applications", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 196-202. 1996.
- [5] *The OpenGL Reference Manual*, Addison-Wesley.
- [6] *The OpenGL on SGI Systems Manual*, Silicon Graphics Inc.
- [7] Poggio, T., Sung, K.K., "Example-based learning for view-based human face detection". Proc. of the ARPA Image Understanding Workshop, II:843-850. 1994.
- [8] Rehg, J., Loughlin, M., and Waters, K., "Vision for a Smart Kiosk", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 690-696. 1997.
- [9] Rowley, H., Baluja, S., and Kanade, T., "Neural Network-Based Face Detection", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 203-207. 1996.
- [10] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P., "Pfinder: Real-time Tracking of the Human Body", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:7, July 1997.
- [11] Zabih, R., and Woodfill, J., "Non-parametric Local Transforms for Computing Visual Correspondence", Proc. of the third European Conf. on Computer Vision, Stockholm, pp. 151 - 158. May 1994.
- [12] Woodfill, J., and Von Herzen, B., "Real-Time Stereo Vision on the PARTS Reconfigurable Computer", Proc. IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, CA, pp. 242-250, April 1997.