

Robust, real-time people tracking in open environments using integrated stereo, color, and face detection.

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

Abstract

We present an approach to robust, real-time person tracking in crowded and/or unknown environments using multi-modal integration. We combine stereo, color, and face detection modules into a single robust system, and show an initial application for an interactive display where the user sees his face distorted into various comic poses in real-time. 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, and face pattern detection discriminates and localizes the face within the tracked body parts. We discuss the failure modes of these individual components, and report results with the complete system in trials with thousands of users.

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.

We have created a visual person tracking system which achieves robust performance through the integration of multiple visual processing modalities—stereo, color, and pattern. Each module can possibly track a user under optimal conditions, but have, in our experience, substantial failure

modes in unconstrained environments, as discussed below. We have, however, found that the failure modes between these particular modules are substantially independent, and by combining them in simple ways, we can build a system which robustly tracks users' faces in general conditions in real-time (in excess of 15Hz).

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 Person tracking modules

Three primary modules are used to track a user's head position: stereo depth estimation, flesh-hue color segmentation, and intensity-based face pattern classification. Depth is estimated from multiple fixed cameras and allows easy segmentation of a user in an open and unknown environment. 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 discriminate head regions from hands, legs, and other body parts. Knowledge of the location of the user's head in 3-D is then passed to the application; we show examples of a face distortion program in the following Sections. Figure 1 shows the output of the various vision processing modules on a scene with a single person present.

2.1 Silhouette extraction via 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

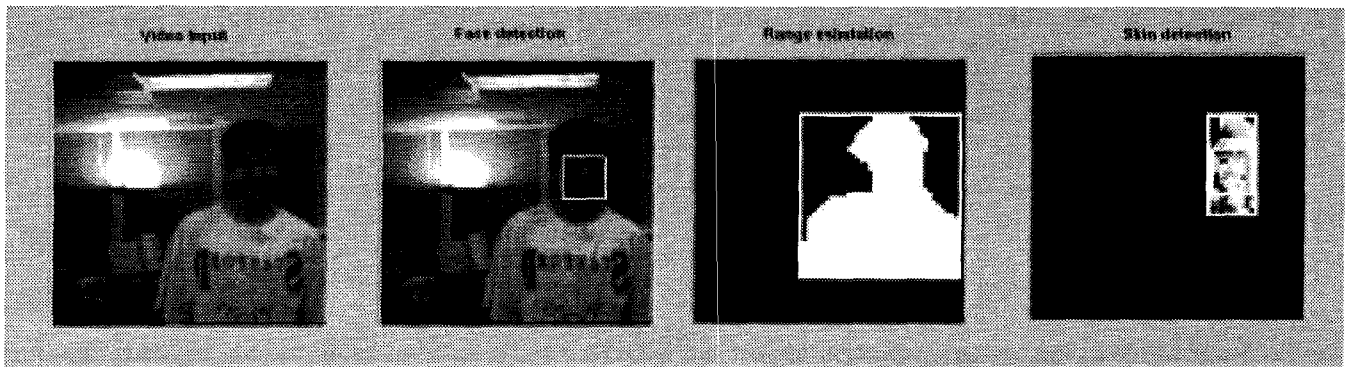


Figure 1. Output of vision processing modules: input image, face detection result, foreground region computed from range data, and skin hue classification score image.

correspondence techniques. A critical issue in determining stereo correspondence is that pixels from two cameras that correspond to the same scene element may differ due to both camera properties such as gain and bias, and to scene properties such as varying reflectance distributions resulting from slightly differing viewpoints. The census correspondence algorithm [8] overcomes these potential differences between images by taking a non-parametric approach to correspondence. The census algorithm 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.

The transform used – the census transform – maps each pixel P in an intensity image to a bit vector, where each bit represents the ordering between the intensity of P and that of a neighboring pixel. Thus a pixel at the top of an intensity peak would result in a homogeneous (all ones) bit vector indicating that its intensity is greater than those of its neighboring pixels. Two such census bit vectors in different images can be compared using the Hamming distance, i.e. by counting the number of bits that differ. The correspondence process – for each pixel in one image, finding the best match from within a fixed search window in the other image – is performed by minimizing locally summed Hamming distances. The displacement to the best match serves as the disparity result for a pixel.

We have implemented the census algorithm on a single PCI card, multi-FPGA reconfigurable computing engine [7]. 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 [2].

Given dense depth information, a silhouette of a user is found by selecting the nearest human-sized surface and then tracking that region until they disappear. Our segmentation and grouping technique proceeds in several stages of processing. We first smooth the raw range signal to reduce the effect of low confidence stereo disparities using a morphological closing operator. 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 thresholded gradient image (a binary quantity) with the smoothed range data. This creates regions of zero value in the image where abrupt transitions 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 similar depth and centroid location. 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. (Extending our system to simultaneously track and process several users is a planned topic of future work.) Specifically, we use the recovered connected component region marked as the target to be a boolean mask which we apply to images from the primary camera before passing them to the color

and face processing modules.¹

2.2 Skin detection via 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 with pixels corresponding to the red, green, and blue channels of the image, and is converted immediately 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 two-sided 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 $P(p = \textit{skin})/P(p = \textit{non-skin})$ is computed as a classification score. Our color representation is similar to that used in [1], 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.² 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

¹If 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 the above segmentation and grouping method, such as when the user is wearing clothes whose intensity values are saturating the camera and thus have no contrast on which to compute stereo correspondence. In practice this occurred in less than 2% of trials.

²See 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.

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, described below. 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 more than a specified threshold amount.

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 [4] 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 [3]. 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 faces of sufficient image size. However it alone was not sufficient for our application, as it was considerably slower than real-time when applied to the entire image, and offered poor performance at tracking faces when they were not in frontal pose, were too small, or had extreme expressions.³ In concert with skin color tracking and depth segmentation, however, the face detection module provides information essential for robust performance.

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

³We do note that we were able to obtain near-real time performance (10Hz) from the CMU face detector alone when we configured it in a tracking mode, where it looked in a window around the most recently found face. However the assumption of frontal pose, and thus this real-time tracking performance, would only hold for a small fraction of the time a user was interacting with the system. Our application, which encouraged users to move their head and make strange expressions, may be atypical in this regard.

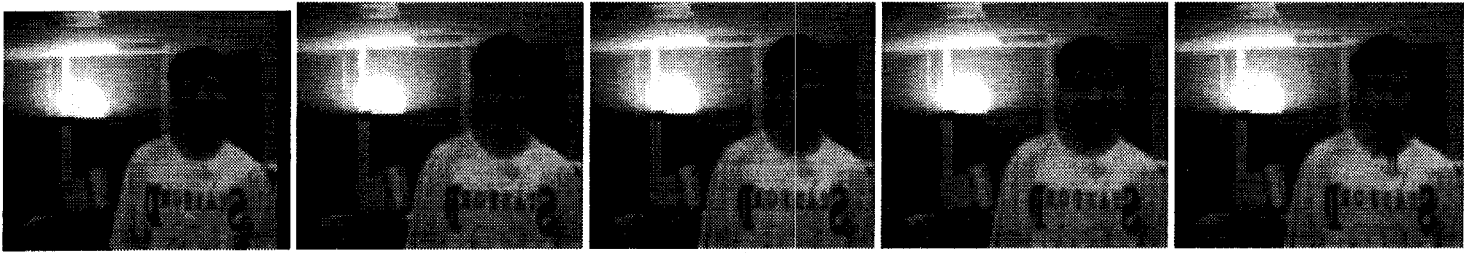


Figure 2. Distortion effects applied to user shown in Figure 1: spherical expansion, spherical shrinking, swirl, lateral expansion, and vertical sliding effect.

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.

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.

3 The Magic Morphin Mirror

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.

3.1 Imaging Geometry

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 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.

As described above, three primary modules are used to track a user's head: depth estimation, color segmentation, and pattern classification. The combination of these three techniques allows for robust performance despite unknown background, crowded conditions, and rapidly changing pose or expression of the user. With an estimate of the position of the user's head in 3-D, 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.

3.2 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



Figure 3. Typical scenes seen in the virtual mirror.

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.

3.3 Implementation Details

We implemented our system 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 moni-

tor 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 we encountered. By processing just the regions returned by the color and range modules, the face detector runs in excess of 15Hz.

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 2 shows the basic effects generated by our system.

4 Tracking Results

We demonstrated our system at SIGGRAPH'97 from Aug 3-8, 1997. An estimated 5000 people over 6 days used our system (approx. two new users per minute, over 42 hours of operation). The goal of the system was to identify the 3-D position and size of a user's head in the scene,



Figure 4. Input images sampled randomly from system's primary camera; effect bounding box is drawn where face was found. The last image is an incorrect result.

and apply a distortion effect in real-time only over the region of the image containing the users face. 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 3 shows typical images displayed on the virtual mirror. The system performed well with both single users and crowded conditions. This being SIGGRAPH, the background environment was also quite visually noisy, with many lighting effects being randomly projected (and cast upon our display!) throughout the time our system was in operation.

To quantitatively evaluate tracking performance we sampled the state of the tracking system approximately every 15 seconds over a period of 3.5 hours. At each sample point we collected images of the scene and the output of the range module, and recorded the position of the distortion effect (e.g., the tracked face position.) We then evaluated off-line the accuracy of the estimated face position. We labeled whether the effect position was correctly centered on the face of a user in the scene.

Overall, our results were good, in excess of 85% correct. We found that approximately 5% of our sampled images were false positive results, in that the effect was centered on a region that was not the face, or the size of the effect was not consistent with the real size of the face. A substantial fraction of these cases involved cases where our assumption that users were standing was violated; e.g. users stood below the screen and stuck their hands vertically into the workspace, or stuck their hands up while they leaned to one side. With a benign user population, we believe the false positive rate would be reduced by a factor of two. We are currently working on additional tests to detect the cases described above.

Approximately 10% of our sampled images had people who appeared to be in the system workspace⁴ but no effect was placed, formally a false positive. Almost half of these cases involved people wearing hats, where there was insufficient light on their face for the flesh color classifier to operate. To address these cases we are considering a default method which identifies head position from range alone in cases where that is feasible. The remaining cases involved user configurations where the face detector never returned a match and the user was not in a simple standing pose.

Given our requirement for real-time performance and the cluttered and noisy environment in which our system was tested, we are pleased with this level of performance. Nonetheless, this work is ongoing and we anticipate even greater performance levels as we gain experience with the system and refine both the components and our integration strategy. We are presently running experiments to evaluate the contribution each module makes to recognition perfor-

⁴As defined by thresholds on the maximum range distance and minimum color region size in the segmentation and grouping algorithm.

mance; to accomplish this we are extending the system to operate in batch mode with any subset of modules disabled.

5 Conclusion

We have demonstrated a system which can track and 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 has application for 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] Fleck, M., Forsyth, D., and Bregler, C., (1996) "Finding Naked People," European Conference on Computer Vision, Volume II, pp. 592-602. 1996.
- [2] Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M., "A Video-Rate Stereo Machine and Its New Applications", Computer Vision and Pattern Recognition Conference, San Francisco, CA, 1996.
- [3] Poggio, T., Sung, K.K., Example-based learning for view-based human face detection. Proceedings of the ARPA IU Workshop '94, II:843-850. 1994.
- [4] Rowley, H., Baluja, S., and Kanade, T., Neural Network-Based Face Detection, Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR-96, pp. 203-207, IEEE Computer Society Press. 1996.
- [5] Silicon Graphics Inc., The OpenGL Reference Manual, Addison-Wesley.
- [6] Silicon Graphics Inc., The OpenGL on SGI Systems Manual, Silicon Graphics Inc. Documentation.
- [7] Woodfill, J., and Von Herzen, B., Real-Time Stereo Vision on the PARTS Reconfigurable Computer. Proceedings IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, pp. 242-250, April 1997.
- [8] Zabih, R., and Woodfill, J., Non-parametric Local Transforms for Computing Visual Correspondence, Proceedings of the third European Conference on Computer Vision, Stockholm, pp. 151 - 158. May 1994.