FEATURED WAND FOR 3D INTERACTION

Feng Guo†, Don Kimber‡, Eleanor Rieffel‡

‡FX Palo Alto Laboratory, {kimber, rieffel}@fxpal.com
†Arizona State University, Tempe, feng.guo@asu.edu

ABSTRACT

Our featured wand, automatically tracked by video cameras, provides an inexpensive and natural way for users to input 3D data or interact with devices such as large displays. The wand supports six degrees of freedom for manipulation of 3D applications like Google Earth. Our system uses a 'line scan' to estimate the wand pose tracking which simplifies processing. Several applications are demonstrated.

1. INTRODUCTION

Gesture interfaces support a large vocabulary of basic, intuitive interactions. The design of our passive wand includes features that support robust tracking by one or more cameras for use as a gesture interface. Our wands are simple, light, and uninstrumented, so they are inexpensive and do not need battery power or electronic parts that may break. Our purely vision based system, unlike predecessors, can specify a full 6 degrees of freedom (DoF).

Two types of gesture interfaces are in common use: touch screens and instrumented gadgets. Screen based interactions are restricted to 2D interactions. Even when 2D interaction supports a wide enough vocabulary for interaction, gesture interfaces may be preferable; for example, getting close enough to touch large screens means losing the overall perspective. Interface devices like gloves which require the user to wear additional hardware are unpopular, relatively expensive, and subject to failure. Computer vision based tracking of hand gestures is not sufficiently robust for most applications. We are interested in simple, static, inexpensive objects that support robust gesture interfaces through computer vision techniques in which all of the intelligence is in the camera system.

Cao and Balakrishnan's VisionWand[1] shows that such a system can be effective and intuitive in its interaction. Our techniques improve on their work by making the wand easier to detect and track, by increasing the precision of the tracking, and by enabling determination of the twist of the wand, thereby supporting another dimension for a gesture interface to use. Our wand can be used as a pointing device, suitable for controlling positions in a 3D modeling tool, whereas the VisionWand is a gestural device, not a pointing device. These improvements come through a combination of additional features on the wand and in the tracking system. Three applications demonstrate effective use of our wand system as an interface device.

Figure 1 shows a wand with different colored region at each end that are easy to track coarsely for initialization. The black and white area with spiral markings enables determination of the twist of the wand about its axis. Wands may be constructed easily by printing a pattern and wrapping it around a cylinder. Our wand's features enable higher precision tracking and 6 DoF pose estimation.

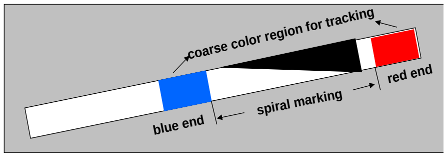

Fig. 1. Typical Feature Wand

In spite of decades of work on tracking, how to obtain robust and real time performance within a multi-camera framework remains an important problem. Our system takes advantage of color-based mean shift tracking, motion analysis, calibration information for each camera, and local image analysis. Figure 2 gives an overview of our approach. Initialization of the 3D wand tracking system uses motion history, color histogram and calibration information. A modified CAMSHIFT algorithm then tracks the wand's two ends independently using color (in HSV color space). At each frame, tracking is verified by local color detection and 3D triangulation. If a tracker registers as lost, the system recovers the missed tracker or re-initialize the system. After coarsely locating the wand ends, feature points on the wand are extracted by line scanning to obtain the full 6 DoF pose which is then fed to the application.

2. RELATED WORK

Cao and Balakrishnan also use stereo cameras to track color wands for large display interaction [1]. Their tracking works

This research was fully supported by FX Palo Alto Laboratory (FXPAL) where Feng Guo worked as an intern during the summer of 2006. The authors would like to thank our colleagues for useful discussions.

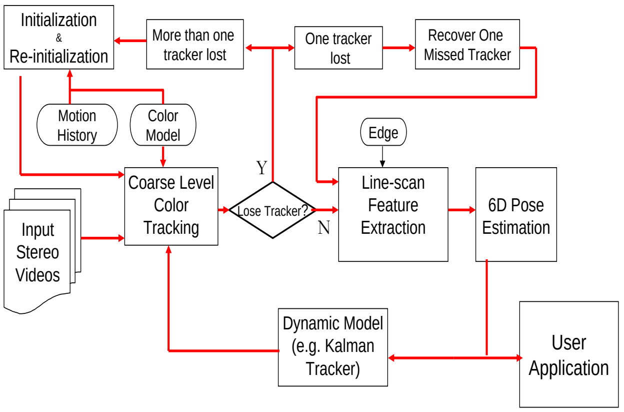

Fig. 2. Overview of feature wand tracking system.

differently: they detect wand body color, fit to a line and then determine the wand ends. The main differences between our work and theirs are: 1) the VisionWand provides only two rotation angles, no twist. 2) we track wand ends directly without using the body, so when the body is occluded we can still reconstruct the wand pose. 3) We use two colors not three since we don't use the body color; fewer colors are more easily distinguished from the background.

Other computer vision based gesture interfaces track laser pointers, spheres, or hands. Laser pointers only provide point location as replacement for mouse pointing [2]. Interface control is not straightforward with a sphere, though 6 DoF are obtained [3]. Hand tracking is not robust or precise [4]. Wilson and Shafer's XWand[5] is an expensive, highly electronic device. Magic Wand [6] also uses magnetic sensor-based tracking and can only detect the pointing direction. The most common interfaces are point and click devices interacting with a screen [7], which are limited to 2 dimensions of the screen.

3. SYSTEM DESCRIPTION

Standard stereo vision techniques yield the wand's pose. The Camera Calibration Toolbox for Matlab is used to calibrate the cameras. The location of the wand's ends are calculated by triangulation. Given tracking results from two cameras, the inter-ray distance between the two rays to the center of the object is calculated. In the ideal case the two rays intersect, so in practice the inter-ray distance should be small. Using this assumption, the tracking results are checked for consistency.

3.1. Wand Blob Tracking

Each color blob is tracked independently by the CAMSHIFT algorithm [8], a technique based on the mean shift algorithm[9]. We model the target using the H channel in the HSV color space to obtain one dimensional H-channel histograms for the predefined wand end colors. Following initialization and the successful location of the wand's ends, the algorithm switches into tracking mode. During tracking, the color distribution of each pixel in the H channel is computed. The mean shift iteration applied in the two view images separately finds the best target candidate. Figure 3 gives an example in which the red and blue ends are correctly tracked in two camera views.

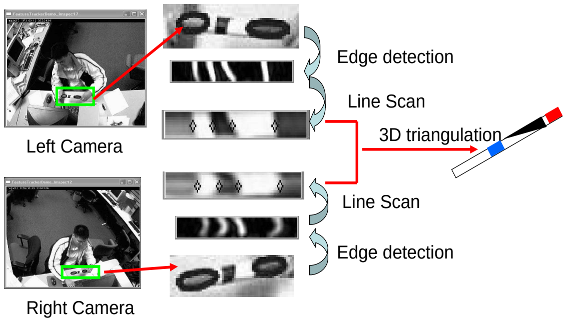

Fig. 3. Example frames of the tracking system in stereo cameras.

3.2. Initialization

For initialization, we use a color model and motion analysis to find regions of interest. For each camera view, motion history is estimated independently to rapidly determine where movement has occurred using an algorithm of Davis and Bobick [10]. A foreground silhouette is obtained through subtraction between two consecutive frames instead of background subtraction. As the wand moves, the most recent foreground silhouette is copied as the highest value in the motion history image which reduces the disturbance from similar background colors. The result is called the "motion history image" (MHI). MHI pixel values that fall below a threshold are set to zero. We search the MHI to find the initial candidates.

3.3. Re-initialization

Sometimes due to fast movement and low frame rate or confusion with the background, initial tracking results poorly estimate the location of an object in current frame. Inconsistency is reported whenever the inter-ray distance of the two-camera rays for one wand end is larger than a threshold. If for several frames this inconsistency happens or the color of the wand ends is not detected, we re-initialize the tracking system.

3.4. Recovery of One Missed Tracker

If a single region is lost in one camera due to faulty tracking or occlusion, its location can be recovered using geometry. In Figure 4 the right camera's view of blob B is occluded, so no image point b_r is reported. Because the other trackers found the object, the 3D location of the point $\overline{A}$(red object) can be calculated. The location $\overline{B}$ (blue object) for other end of the wand is the intersection of the ray through b_l in left camera view with the sphere of radius L centered at $\overline{A} = (x_A, y_A, z_A)$, where L is the wand's length.

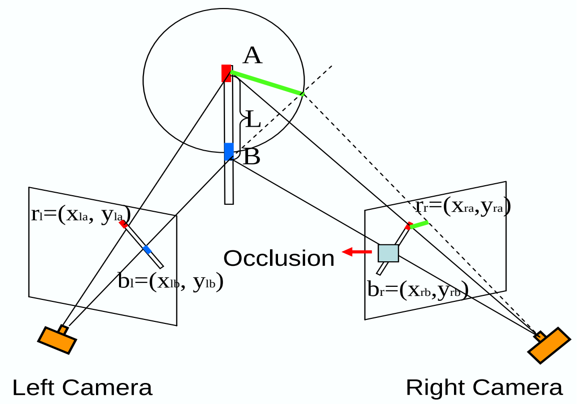
A
L
B
r$_l$=(x$_{la}$, y$_{la}$)
b$_l$=(x$_{lb}$, y$_{lb}$)
Occlusion
r$_r$=(x$_{ra}$,y$_{ra}$)
b$_r$=(x$_{rb}$,y$_{rb}$)
Left Camera
Right Camera

Fig. 4. Recover occluded object from two cameras.

Given the left image point location $b_l = (x_{lb}, y_{lb})$, assume the depth value of $\overline{B}$ in left camera coordinate system is z_{lB}. The location $\overline{B}$ is obtained by solving equations (1) and (2)

$$R_{Lcam}\left[z_{lB}\begin{pmatrix} x_{lb} \\ y_{lb} \\ 1 \end{pmatrix} + T_{Lcam}\right] = \overline{B} \qquad (1)$$

$$|\overline{A} - \overline{B}| = L \qquad (2)$$

where R_{Lcam} and T_{Lcam} are rotation and translation matrices from the left camera's coordinate system to global coordinate system. There are two possible solutions for the location of $\overline{B}$; we choose the location which is closer to previous tracking results. The location of the object in the right camera image $b_r = (x_{rb}, y_{rb})$ can be calculated by projection:

$$\begin{pmatrix} x_{rb} \\ y_{rb} \\ 1 \end{pmatrix} = f_p(\overline{B}) \qquad (3)$$

3.5. Line Scanning

Given coarse tracking results, the images of the wand's two ends are used to calculate the 3D positions of two ends. The position of the two ends define 5 degrees of freedom for the wand pose. To determine the rotation of the wand along its axis, further analysis of wand features is needed. We use the feature points not only to calculate this "twist" angle, but also to obtain a more accurate estimate for the wand pose.

The feature points we use come from the intersection of scan line along the center of the wand with the spiral pattern and the black lines marking the edge of the color regions (Figure 5). Using the location of the wand ends, a sub-image which includes only the wand is extracted, fitted to a rectangle, and transformed to a gray scale image. A one direction Sobel edge detector is applied to obtain the edges along the perpendicular direction. From the edge image, a 1D signal is generated by pixel projection to the horizonal by summing the pixel values in the orthogonal line segment. We know the number of edges K, so the K largest peaks are taken to be the location of the edges. The location of feature points in the original image are obtained by back projection. From the two camera views, the 3D locations of the feature points are obtained. Because of the view angle difference, the corresponding feature points in two views are not the same. Using knowledge of the cameras' poses, we can obtain the twist and an improved estimation of the wand pose.

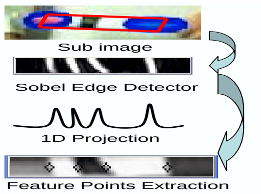

Fig. 5. Line scanning example

3.6. Pose Estimation

The pose of the wand is characterized by 6 DoF: a position vector $\vec{w} = (w_x, w_y, w_z)$ for one endpoint of the wand, two degrees for the direction the wand is pointing, and one degree for the twist. Instead of using two angles, it is convenient to specify the direction using a unit vector $\vec{n} = (n_x, n_y, n_z)$. Given the feature points in the images, the location of features $\vec{p}_0 = (x_0, y_0, z_0)$, $\vec{p}_1 = (x_1, y_1, z_1)$, $\vec{p}_2 = (x_2, y_2, z_2)$ are easily computed along with the direction of the wand $\vec{n} = eigen(\vec{p}_0, \vec{p}_1, \vec{p}_2)$. (See Figure 6.)

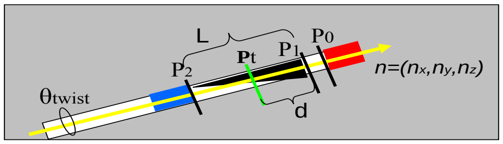

Fig. 6. Wand pose estimation from feature points.

In each camera view, the intersection of the line scan with the spiral defines two 3D points, p_l and p_t (Figure 6). The ratio of the distance $d = |p_1 - p_t|$ to the length L of the spiral band determines the amount of twist with respect to the camera's coordinate system: $\theta^{perceived}_{twist} = 2\pi\frac{d}{L}$. Let θ^{cam}_{twist} be the perceived twist for an unrotated wand with respect to the camera's view. The twist with respect to the global coordinate system is given by $\theta_{twist} = \theta^{perceived}_{twist} - \theta^{cam}_{twist}$.

3.7. System Specifics

We used AXIS 206 network cameras with 640×480 video images and a Pentium 4 3GHz desktop, with 1GBytes of memory, and processing rate of 9 frames/sec. Our implementation is not optimized; its speed could be substantially improved.

4. APPLICATIONS

We investigated wand based control of several 3D applications including navigation in a 3D building model, construction of simple elements in 3D models, and controlling Google Earth. Some common elements of the interfaces are control of virtual motion by movements of the wand from a neutral position, and control of view direction by pointing (Figure 7).

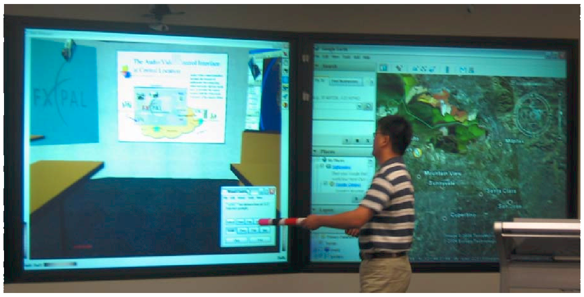

Fig. 7. Demo of the applications. Left screen shows the building navigation, right screen shows the Google Earth control.

For Google Earth control, the wand is started in a neutral position pointing forward. Tilting up or down controls the inclination of the viewpoint, and rotating it left or right controls heading. Raising or lowering controls elevation, while lateral motion controls panning. Google Earth does not yet have an open API for full control, so we implemented a server which maps viewpoint change requests to mouse events. This approach has limitations; for example it does not allow simultaneous control of pitch and heading. Nevertheless, this mapping provides wand control to a wide variety of applications.

We also use the wand to control virtual navigatation though a 3D model of our research center, using a viewer implemented with Open Inventor. The control scheme is similar to that described above for Google Earth, except that the primary navigation mode is ‘walking’ in which the height is constrained. In addition to navigating, the user can enter modes for creating 3D model elements. The wand can be used as a 3D stylus, and its trajectory captured as a wireframe shape, or the user may select points to define polylines or rectangles, which may be colored or texture mapped. A twist motion could be used to turn the stylus on and off. In this way the user creates simple models, either imaginary objects or representatives of real objects by tracing over the physical objects. For example, the user can model a work area by tracing the edges of computer displays, and then displaying a stream of screen captures from computers the user specifies. Once the virtual computer displays have been defined in this manner, touching a display with the wand, or a gesture of jabbing towards it, enables wand control of the application running on the associated computer. A gesture such as moving the wand up and back can be used to turn off the interaction.

5. CONCLUSION

Our featured wands and computer vision tracking system provide a full 6 degrees of freedom for use as an interface device in applications including 3D modeling, and provides more accurate estimates of the wand pose than previous work. It has been used to control multiple applications. We plan to design sets of gestures geared to certain applications and to evaluate their effectiveness with user studies.

6. REFERENCES

[1] X. Cao and R. Balakrishnan, “Visionwand: interaction techniques for large displays using a passive wand tracked in 3D,” in *Proceedings of the ACM symposium on User interface software and technology*, 2003, pp. 173–182.

[2] J. Oh and W. Stuerzlinger, “Laser pointers as collaborative pointing devices,” in *Graphics Interface*, 2002, pp. 141–149.

[3] D.Bradley and G.Roth, “Six degree-of-freedom sphere tracking for 3D interaction,” in *Advances in Computer Entertainment Technology*, 2005, pp. 19–26.

[4] Y.Sato and M.Saito, “Real-time input of 3D pose and gestures of a user’s hand and its applications for HCI,” in *IEEE Virtual Reality*, 2001.

[5] A.Wilson and S.Shafer, “XWand: UI for intelligent spaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003, pp. 545–552.

[6] J.Ciger, M.Gutierrez, F.Vexo, and D.Thalmann, “The magic wand,” in *Proceedings of the 19th spring conference on Computer graphics*, 2003, pp. 119–124.

[7] S.Elrod, R.Bruce, and R.Gold etc., “Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 599–607.

[8] Gary R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” *Intel Technology Journal*, vol. Q2, pp. 15, 1998.

[9] D. Comaniciu and P. Meer, “Robust analysis of feature spaces: Color image segmentation,” in *Proceedings Computer Vision and Pattern Recognition*, 1997, pp. 750–755.

[10] J. Davis and A. Bobick, “The representation and recognition of action using temporal templates,” in *Proceedings Computer Vision and Pattern Recognition*, 1997, pp. 928–934.