Fast Stereo-Based Head Tracking for Interactive Environments

Louis-Philippe Morency Ali Rahimi Neal Checka Trevor Darrell
MIT AI Lab
Cambridge, MA, 02139, USA
{lmorency, rahimi, nealc, trevor} @ai.mit.edu

Abstract

We present a robust implementation of stereo-based head tracking designed for interactive environments with uncontrolled lighting. We integrate fast face detection and drift reduction algorithms with a gradient-based stereo rigid motion tracking technique. Our system can automatically segment and track a user's head under large rotation and illumination variations. Precision and usability of our approach are compared with previous tracking methods for cursor control and target selection in both desktop and interactive room environments.

1. Introduction

Head pose or gaze is a potentially powerful and intuitive pointing cue if it can be obtained accurately and non-invasively. In interactive environments, like public kiosks or airplane cockpits, head pose estimation can be used for direct pointing when hands and/or feet are otherwise engaged or as complementary information when desired action has many input parameters. In addition, this technology can be important as a hands-free mouse substitute for users with disabilities or for control of gaming environments.

When interacting directly with users, robustness and efficiency are key requirements for a successful system. Interactive environments often include dynamic video projection across multiple large screens, and thus have illumination levels which can change spontaneously. A head tracking system for such environments must be able to handle variations of illumination and large head rotations. In addition, the system should be fast enough to maintain transparent interaction with the user.

In this paper, we evaluate a head pose tracker designed for interactive environments. Our system is accurate, fast, and automatically initialized. We rely on a online drift reduction algorithm based on Rahimi *et al.*[17] and a rigid stereo motion tracking technique[10] which can deal with large rotations of the user's head. Our intensity- and depth-based technique is relatively insensitive to illumination variation. Since it is based on real-time 3-D observation, it can be more accurate than previous approaches that presumed approximate models.

The performance of our system was evaluated on a shape tracing task and a selection task. We compared our tracker performance with published reports and side-by-side implementations of two other systems. We evaluated tracing accuracy with small and large head rotations and with different levels of lighting variation. We also compared the performance of our tracker with that of a head-mounted inertial sensor.

In the following section, we review related work on head tracking. We then describe the components of our tracking system, followed by our experimental paradigm and interaction task. We evaluate the spatial accuracy and temporal resolution of our system, compare it to previously reported systems, and conclude with a discussion of these results.

2. Related Work

Several authors have recently proposed face tracking for pointer or scrolling control and have reported successful user studies [19, 15]. In contrast to eye gaze [23], users seem to be able to maintain fine motor control of head gaze at or below the level needed to make fine pointing gestures[1]. However, performance of the systems reported to date has been relatively coarse and many systems required users to manually initialize or reset tracking. They are generally unable to accurately track large rotations under rapid illumination variation (but see [16]), which are common in interactive environments (and airplane/automotive cockpits).

Many techniques have been proposed for tracking a user's head based on passive visual observation. To be useful for interactive environments, tracking performance must be accurate enough to localize a desired region, robust enough to ignore illumination and scene variation, and fast enough to serve as an interactive controller. Examples of 2-D approaches to face tracking include color-based [22], template-based [15] and eigenface-based [9] techniques.

[1] Involuntary microsaccades are known to limit the accuracy of eye-gaze based tracking[14].

Techniques using 3-D models have greater potential for accurate tracking but require knowledge of the shape of the face. Early work presumed simple shape models (e.g., planar[2], cylindrical[16], or ellipsoidal[1]). Tracking can also be performed with a 3-D face texture mesh [18] or 3-D face feature mesh [21].

Very accurate shape models are possible using the active appearance model methodology [5], such as was applied to 3-D head data in [3]. However, tracking 3-D active appearance models with monocular intensity images is currently a time-consuming process, and requires that the trained model be general enough to include the class of tracked users.

In contrast to these head tracking systems, our system is robust to strong illumination changes, automatically initializes without user intervention, and can re-initialize automatically if tracking is lost (which is rare). In addition, it can track head pose under large rotations and does not suffer from drift.

3. Stereo-Motion Head Tracking

Our system has three main components. Its core is an algorithm for instantaneous depth and brightness gradient tracking [10], combined with two other modules for initialization, and stabilization/error-correction. For initialization we use a fast face detection scheme to detect when a user is in a frontal pose, using the system reported in [20]. To minimize the accumulation of error when tracking in a closed environment, we rely on a scheme which can perform tracking relative to multiple base frames [17].

The following subsections describe the initialization and basic tracking algorithm which recovers the rotation and translation of an object between two time steps t and s, given images I_t and I_s. The last subsection explains how to use multiple base frames to reduce drift.

3.1. Initialization with Face Detection

When it first comes online, the tracker scans the image for regions which it identifies as a face using the face detector of [20]. As soon a face has been consistently located near the same area for several frames, the tracker switches to tracking mode. The face detector is sensitive only to completely frontal heads, making it possible for the tracker to assume that the initial rotation of the head is aligned with the coordinate system. The face detector provides the tracker an initial region of interest, which is updated by the tracker as the subject moves around. Since depth information is readily available from the stereo camera, the initial pose parameters of the head can be fully determined by 2D region of the face with the depth from stereo processing.

When we observe erratic translations or rotations from the tracker, the tracker automatically reinitializes by reverting to face detection mode until a new target is found. This occurs when there is occlusion or rapid appearance changes.

3.2. Finding Pose Change Between Two Frames

Our system uses stereo cameras by applying the traditional Brightness Change Constraint Equation (BCCE) [11] jointly with the Depth Change Constraint Equation (DCCE) of [10] on range and intensity imagery.

To recover the motion between two frames, the BCCE finds motion parameters which minimize the appearance difference between the two frames in a least-squares sense:

$$\begin{aligned} \delta^* &= \arg\min_{\delta} \epsilon_{BCCE}(\delta) \\ \epsilon_{BCCE} &= \sum_x \|I_t(x) - I_{t+1}(x + u(x;\delta)\|^2 \end{aligned} \quad (1)$$

where $u(x;\delta)$ is the image flow at pixel x, parameterized by the details of a particular motion model. In the case of 3D rigid motion under a perspective camera, the image flow becomes:

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} (\delta_\omega \times X + \delta_\Delta), \quad (2)$$

where X is the world coordinate of the image point x, δ_ω is the infinitesimal rotation of the object, δ_Δ is its infinitesimal translation, and f is the focal length of the camera[4].

The DCCE of [10] uses the same functional form as equation (1) to constrain changes in depth. But since under rotation, depth is not preserved, the DCCE includes an adjustment term:

$$\epsilon_{DCCE} = \sum_x \| Z_t(x) - Z_{t+1}(x + u(x;\delta)) + V_z(x;\delta)\|^2,$$

where V_z is the flow towards the Z direction induced by δ. Note that the DCCE is robust to lighting changes since lighting does not affect the depth map. We combine the BCCE and DCCE into one function optimization function with a weighted sum:

$$\delta^* = \arg\min_{\delta} \epsilon_{BCCE}(\delta) + \lambda \epsilon_{DCCE}(\delta),$$

The only unknown variables are the pose parameters, since Z is available from the depth maps. For an approximate way to optimize this function, see [10], where one iteration of Newton-Raphson is shown to be adequate for tracking.

3.3. Reducing Drift

Given a routine for computing the pose difference δ_s^t between frames I_s and I_t, there are two common strategies for estimating the pose ξ_t of frame I_t relative to the pose of frame I_0. One approach is to maintain the pose difference between adjacent frames I_s and I_{s+1}, for $s = 0..t-1$, and

to accumulate these measurements to obtain the pose difference between frames I_t and I_0. But since each pose change measurement is noisy, the accumulation of these measurements becomes noisier with time, resulting in unbounded drift. A common alternative is to compute the pose difference between I_t and I_0 directly. But this limits the allowable range of motion between two frames, since most tracking algorithms (including the one described in the previous section) assume that the motion between the two frames is very small.

To address the issue of drift in parametric tracking, we compute the pose change between I_t and several base frames. These measurements can then be combined to yield a more robust and drift-reduced pose measurement. When the trajectory of the target crosses itself, pose differences can be computed with respect to early frames which have not been corrupted by drift. Trackers employing this technique do not suffer from the unbounded drift observed in other differential trackers.

In [17], a graphical model is used to represent the the true poses ξ_t as hidden variables and the measured pose changes δ_s^t between frames I_s and I_t as observations. Unfortunately, the inference algorithm proposed is batch, requiring that pairwise pose changes be computed for the entire sequence before drift reduction can be applied.

We use a simple online algorithm to determine the pose of a frame I_t. Our algorithm first identifies the k frames from the past which most resemble I_t in appearance. The similarity measure we use is the sum of squared differences:

$$d_s^t = \sum_x \sum_y \|I_s(x,y) - I_t(x,y)\|^2. \qquad (3)$$

Since the frames from the past have suffered less drift, the algorithm discounts the similarity measure of newer frames, biasing the choice of base frame toward the past.

Once the candidate base frames have been identified, the pose change between each base frame I_s to I_t is computed using the algorithm described in the previous section. The final pose assigned to frame I_t is the average pose of the two base frames, weighted by the similarity measure of equation (3):

$$\xi_t = \frac{\sum_i (\xi_{si} + \delta_{si}^t)/d_{si}^t}{\sum_i 1/d_{si}^t}.$$

As an alternative, we are investing an algorithm for performing online inference on the graphical model of [17].

4. Experiments and Results

To evaluate the system we performed two series of experiments, the first in a desktop screen environment and the second in an interactive room with large projection screens. In the following subsection, we describe the tracking systems used in this user study. We then present the experimental setups and results for both experiments.

4.1. Tracking Techniques

We compared side-by-side the stereo motion tracker of section 3 with a 2D tracker based on normalized cross-correlation, and a head-mounted inertial rotation sensor. The following sections describe each tracker in more detail.

4.1.1 Stereo-Motion Tracker

The stereo-motion head tracker is a standalone system which takes video-rate intensity and range imagery from a stereo camera such as the SRI Small Vision System [6] camera and locates and tracks heads in real-time. The SRI camera software produces 320x240 pixel resolution intensity and range images at 15 fps. The tracker runs on a 1.5 Ghz Pentium 4 running a Windows operating system, and takes advantage of Intel's SIMD architecture through the Intel Performance Library. This tracker uses the rigid motion stereo algorithm described above, together with face detection and drift reduction (with 2 past frames).

As in [19], we use the tracked head position to infer a point of intersection of a "face ray" with the control or display surface, and use this to set a pointer target.

4.1.2 Inertial Rotation Sensor

We used evaluted tracking in comparision to an intertial rotation sensor, using InterSense's $\mathrm{Intertrax}^2$ [12]. The manufacturer reports that it is able to measure changes in rotations in three axes with 0.02 degrees of precision. Our software samples the tracker at about 100 samples per second, though the tracker is reported to have a 256 Hz internal sampling rate; it was attached to the test PC via USB. The documentation for the tracker reports zero jitter, which after some experimentation, we concluded was the result of a hysteretic filter. Based on a patent filed by the manufacturer, the inertial sensor may combine various sources of inertial measurement such as the earth's magnetic and gravitational fields[7].

In contrast to the vision-based tracker of section 3 which automatically tracks after detecting a face, the Intertrax tracker must be manually initialized to provide it with a reference frame. The head-mounted tracker is equipped with a reset button which must be pushed before the user begins each experiment in order to define the initial coordinate system and to reset the accumulated drift.

4.1.3 Normalized Cross-Correlation Tracker

We evaluated 2D tracking techniques to explore the importance of stereo observations for robust real-time tracking. We used a side-by-side implementation of 2D normalized correlation tracking similar to that proposed in [15]. (We also compared published reports of other 2D trackers, as reported below.) The normalized cross-correlation tracker works in two phases, similar to the stereo tracker described above: first, a face detector [20] locates a face and reports

Figure 1. Brightness change during the lighting variation experiment. Left: lamp on. Right: lamp off.

a region of interest which represents the bounding box of a face. Second, the correlation tracker takes a snapshot of the resulting region, scales its magnitude to 1, and uses it as the template in its tracking phase[8].

Once a template is acquired, for each new image, the correlation tracker scans a 70 by 30 pixel region around the location where the face was originally found. For each candidate location (u, v), it computes the similarity measure:

$$\epsilon(u,v) = \sum_x \sum_y \|\tilde{I}_t(x+u, y+v) - \tilde{T}(x,y)\|^2, \quad (4)$$

where $\tilde{T}$ is the magnitude-normalized face template acquired during detection and $\tilde{I}_t$ is the magnitude-normalized current image.

The correlation tracker reports the value of (u, v) which minimizes (4). Typically, this displacement would be scaled by constants in u and v and used as the location of the pointer on the screen. However, because the domain of ϵ is integers and the resolution of the camera is low, the approach is insensitive to small motion. As such, the pointer's precision suffers.

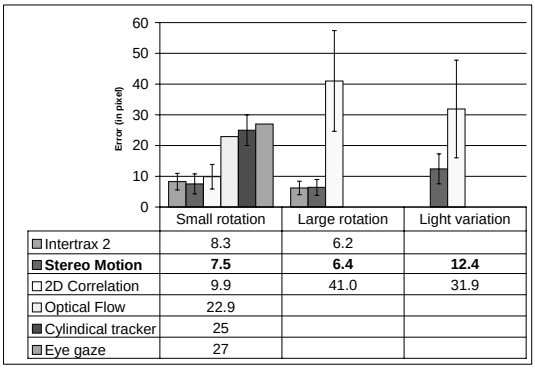
Instead, we resolve the motion (u, v) to sub-pixel resolutions, by approximating the D by D pixel neighborhood around the minimum of ϵ by a second order polynomial $\hat{\epsilon}$. Then instead of reporting the minimum of $\epsilon(u, v)$, the correlation tracker reports the minimum of $\hat{\epsilon}$.

4.2. Desktop Experiment

The desktop experiment involved 8 experiments per subject. Each subject tested the three tracking techniques described in section 4.1. Each of the trackers was tested in small-screen and wide-screen mode. The former allows the user to trace the rectangle using small head motions. The latter simulates a larger screen which requires larger head rotations to navigate. In addition, the correlation tracker and the stereo motion tracker were tested in the small-screen mode under abruptly varying lighting conditions (see figure 1).

As shown in figure 2, users sat about 50 cm away from a typical 17" screen, subtended a horizontal angle of about

Figure 2. A user during the desktop experiment. The SRI stereo camera is placed just over the screen and the user is wearing the Intertrax^2 device on his head.



	Small rotation	Large rotation	Light variation
Intertrax 2	8.3	6.2	
Stereo Motion	**7.5**	**6.4**	**12.4**
2D Correlation	9.9	41.0	31.9
Optical Flow	22.9		
Cylindical tracker	25		
Eye gaze	27		

Figure 3. Comparison of average error on tracing task of the desktop experiment. The error bars in the histogram reprensent the standard deviation between user results.

30 degrees and a vertical angle of about 20 degrees. The screen displayed a black background and a white rectangular path drawn in the middle. The task was to use head pose to move a 2D pointer around the screen to trace the rectangular path as accurately as possible. Users were allowed to take as much time as they liked, as long as they were able to complete the path eventually. Thus, we suggest that the dominant feature under observation is the tracker's accuracy in mapping the user's head to a 2D location.

4.2.1 Results

The first three rows of figure 3 compares the accuracy of the stereo motion tracker with the 2D normalized cross-correlation tracker and the Intertrax^2 tracker. The histogram shows the average error and standard deviation of 4 subjects. The average error is computed as the average distance in pixels between every point on the cursor trajectory and the closest point on the given rectangular path. The three last rows of the same figure compares our results with some published system: an optical flow tracker[13], cylindrical tracker[16], and an eye gaze tracker[23].

Figure 4 shows typical pointer trajectories for each scenario. It took an average of 50 seconds to trace each rectangle.

In a desktop environment, small rotations are sufficient to drive a cursor, since the angle subtended by the screen tends to be small. This situation serves as a baseline where all three trackers can be compared under moderate conditions. Under the small rotation scenario, all trackers showed similar deviation from the given trajectory, with an average deviation of 7.5 pixels for the stereo motion tracker, 9.8 pixels for the normalized cross-correlation tracker, and 8.3 pixels for the inertial tracker. Note that the drift of the inertial sensor becomes significant during the last quarter of its trajectory (figure 4), forcing subjects to compensate for its error with exaggerated movements.

Navigating a pointer on a wide screen (multiple monitors, projection screens, cockpits) requires larger head rotations. As expected, the correlation tracker loses track of the subject during rotations beyond 20 degrees, because the tracker is initialized on the appearance of the frontal face only. It incurred an average error of 41.0 pixels. The stereo motion tracker, however, successfully tracks the head as it undergoes large rotations, with an average error of 6.4 pixels. The $\mathrm{Intertrax}^2$ tracker shows an average error of 6.2 pixels. Note that due to the accumulated drift of the inertial sensor, typical users had difficulty controlling the cursor in the last portion of the trajectory.

Under varying lighting conditions (the light was modulated at about 1/2 Hz), the normalized cross-correlation tracker lost track of the target regardless of the degree of rotation, yielding an average error of 31.9 pixels as opposed to its 9.9 pixels under unvarying lighting. The stereo motion tracker did suffer slightly, averaging an error rate of 12.4 pixels as opposed to its initial error of 7.5 pixels under normal lighting conditions. This is only a factor 1.6 increase in average error, compared to the correlation tracker's factor of 3.2 loss of performance.

4.2.2 Discussion

The inertial rotation sensor $\mathrm{Intertrax}^2$ is accurate for a short period of time, but it accumulates noticeable drift. Approximately after 1 minute of use of the tracker, subjects were often forced to contort their bodies significantly in order to compensate for the drift.

The normalized cross-correlation tracker appears to be suitable for situations involving small head rotations and minimal illumination changes.

The stereo motion tracker is robust to lighting variations because it largely relies on depth information, which is unaffected by the illumination changes. In addition, it can track arbitrarily large transformations without suffering from drift due to the drift reduction algorithm described in section 3.3.

Figure 5. Setup for the room experiment. The SRI stereo camera is placed on the table.

	Average error (in pixel)	Standard deviation (in pixel)
Small rotation	6.3	0.4
Large rotation	6.1	0.6
Light variation	11.5	3.1

Table 1. Experimental results of the stereo-based tracker inside the interactive room.

4.3. Interactive Room Experiment

As shown in figure 5, the second experiment was run in an interactive room with large projection screens. Users were sitting about 1.8 meters away from a 2.1m x 1.5m projection screen, subtended a horizontal angle of about 100 degrees and a vertical angle of about 80 degrees. Subject were asked to perform two tasks: the tracing task described in section 4.2 and a selection task where the user must reach different colored squares without touching the red squares. A short interview was performed following the experiment to obtain feedback from the subject about the usability of these head trackers.

4.3.1 Results and Discussion

With more then 90 degrees of rotation to reach both sides of the screens, the limitations of the normalized cross-correlation tracker appeared clearly. Subjects could not use the tracker without unnaturally translating their heads over long distances to move the cursor correctly.

The stereo-based tracker was successful on both the tracing task and the selection task. Table 1 presents the average errors and standard deviation for the tracing task of 3 subjects.

The interviews after the second experiment showed that users doesn't like a linear mapping between the head pose and the cursor position. For slow movement of the head, the ratio cursor distance by head movement should be smaller to give more precision on small selections. For fast movement of the head, the ratio should larger to give more speed on large displacement. These observations corroborate Kjeldson results[15].

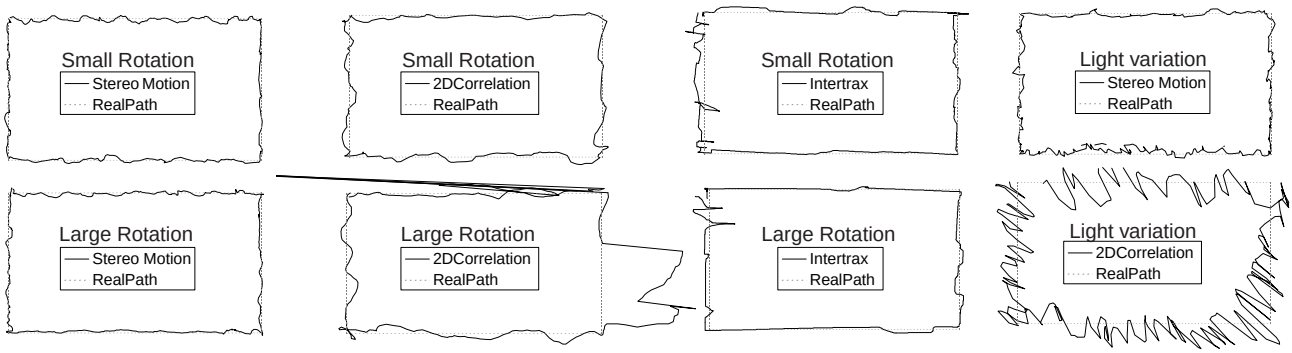


Figure 4. Typical trajectories for all three trackers when users perform small rotations (first row), large rotations (second row) and under light variation (last column). The trajectory starts from the upper left corner of the rectangle and ends in the same location.

5. Conclusion

The stereo head tracking system presented here requires no manual initialization, does not drift, and has been shown to be accurate at driving cursors and selecting objects. Performance of this tracker was compared against that of a head-mounted inertial sensor and a simple tracker based on normalized cross-correlation. The latter tracker was prone to lighting changes, and the former experienced drift over time. The stereo system was insensitive to these conditions, and was found usable by naive users. We believe this tracking system will be an important module in designing perceptual interfaces for intelligent environments, cockpit applications, and for disabled users who are not able to use traditional interfaces.

References

[1] S. Basu, I.A. Essa, and A.P. Pentland. Motion regularization for model-based head tracking. In *ICPR96*, page C8A.3, 1996.

[2] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV95*, pages 374–381, 1995.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH99*, pages 187–194, 1999.

[4] A.R. Bruss and B.K.P Horn. Passive navigation. In *Computer Graphics and Image Processing*, volume 21, pages 3–20, 1983.

[5] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *PAMI*, 23(6):681–684, June 2001.

[6] Videre Design. *MEGA-D Megapixel Digital Stereo Head*. http://www.ai.sri.com/ konolige/svs/, 2000.

[7] Eric M. Foxlin. Inertial orientation tracker apparatus having automatic drift compensation for tracking human head and other similarly sized body. US Patent 5,645,077, US Patent and Trademark Office, Jun 1994.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, Massachusetts, 1992.

[9] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[10] M. Harville, A. Rahimi, T. Darrell, G.G. Gordon, and J. Woodfill. 3d pose tracking with linear depth and brightness constraints. In *ICCV99*, pages 206–213, 1999.

[11] B.K.P. Horn and B.G. Schunck. Determining optical flow. *AI*, 17:185–203, 1981.

[12] InterSense Inc. *Intertrax 2*. http://www.intersense.com.

[13] Mouse Vision Inc. *Visual Mouse*. http://www.mousevision.com.

[14] R.J.K Jacob. *Eye tracking in advanced interface design*, pages 258–288. Oxford University Press, 1995.

[15] R. Kjeldsen. Head gestures for computer control. In *Proc. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 62–67, 2001.

[16] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of textured-mapped 3d models. *PAMI*, 22(4):322–336, April 2000.

[17] A. Rahimi, L.P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *ICCV01*, volume 1, pages 315–322, 2001.

[18] A. Schodl, A. Haro, and I. Essa. Head tracking using a textured polygonal model. In *PUI98*, 1998.

[19] K. Toyama. Look,ma - no hands!hands-free cursor control with real-time 3d face tracking. In *PUI98*, 1998.

[20] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[21] L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *PAMI*, 19(7):775–779, July 1997.

[22] C.R. Wren, A. Azarbayejani, T.J. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *PAMI*, 19(7):780–785, July 1997.

[23] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *CHI99*, pages 246–253, 1999.