Machine Vision and Applications Manuscript No.
(will be inserted by hand later)

Head Tracking Using Stereo

Daniel B. Russakoff and Martin Herman

National Institute of Standards and Technology
100 Bureau Drive, Stop 8940
Gaithersburg, MD 20899

Received February 15, 2001

Abstract. Head tracking is an important primitive for smart environments and perceptual user interfaces where the poses and movements of body parts need to be determined. Most previous solutions to this problem are based on intensity images and, as a result, suffer from a host of problems including sensitivity to background clutter and lighting variations. Our approach avoids these pitfalls by using stereo depth data together with a simple human torso model to create a head tracking system that is both fast and robust. We use stereo data[1] to derive a depth model of the background which is then employed to provide accurate foreground segmentation. We then use directed local edge detectors on the foreground to find occluding edges which are used as features to fit to a torso model. Once we have the model parameters, the location and orientation of the head can be easily estimated. A useful side effect from using stereo data is the ability to track head movement through a room in three dimensions. Experimental results on real image sequences are given.

Key words: Security / Surveillance / Human Motion, Human-Computer Interaction

1 Introduction

Human head tracking has been an area of active research in computer vision for several years. The term *head tracking*, however, means different things to different people. Some think of it as the problem of figuring out not only the location of the head, but also the 3D pose and sometimes even the complex facial expressions. Much research has been devoted to this difficult problem [26], [27], [28], [6], [29], [30], [31], [32], [33], and [5]. We will, however, focus our efforts on a simplified version of this problem– the determination of 3D position and 2D orientation of the head in a sequence of images. Head position and orientation are important parameters for a variety of applications including virtual reality and teleprescence [1], [16], [17], [18], [19], [20],[21], augmented reality [22], [23], face recognition [2], voice recognition [24], audio equalization zone steering [25], and perceptual user interfaces [15].

2 Background

There has been very little work on head tracking using stereo. Until recently, systems which produce real-time stereo depth data have been unavailable on the commercial market. As a result, most approaches to this problem have relied exclusively on intensity images and, as a result, must use color cues and intensity edges for face/head detection and tracking. In [4], they use color histograms and intensity gradients together with a second-order motion model and a local search to track skin-colored elliptical face blobs. Similar methods are used by other researchers to track skin-colored blobs with various additions. In [34], they also add in a hypothesis-tree model to explicitly handle occlusions. [35] adds a blink detection module to augment the power of the color module, [36] uses a best-fit ellipse energy function to more accurately classify skin-colored regions as faces, [37] uses acoustical information to constrain the color information, and [38] adds mouth shape features to their color module. More recently systems that rely exclusively on color models have been pursued by [39], [41], and [40].

Not all methods track blobs of skin-color, though. [42] and [15] track the contour of the head and shoulders using image intensity gradients. [43], [9], [44], [45], and [7] use combinations of skin color models, template models, and various other cues to help detect and track heads. A less typical approach is applied in [46] where they use variable numbers of wavelets to track intensity-based face templates.

Correspondence to: Daniel B. Russakoff

[1] Commercial equipment and materials are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

All of these approaches, by virtue of their reliance on intensity and color information, are notoriously sensitive to environmental factors that affect intensity values such as changes in illumination or background clutter. Recent work in adaptive mixture modelling [48] may mitigate these effects somewhat, but they still cannot effectively handle self-shadowing and situations where the foreground and background are similar. In addition, they cannot, in general, handle the full range of possible skin tones. Stereo depth calculations, however, do not encounter these problems. This observation, coupled with recent advances in stereo hardware which allow us to gather depth data in real time [10], suggests a new approach to the head tracking problem.

There has been very little work done on head tracking using stereo. Some systems ([11],[8]) use stereo but still rely heavily on skin-tone pixel extraction, an intensity-based measure subject to all of the aforementioned problems. One approach ([14]) uses only stereo data but their complicated models prevent tracking of rapid movements and require the user to move slowly. In addition, this algorithm also requires a manual initialization step.

3 Basic Idea and Motivation

Our approach is to use stereo data to perform a more accurate foreground segmentation. We then use depth and intensity information to fit a simple torso model to the foreground, looking specifically for the occluding edges of the shoulders. Because the model is so simple, we can perform the fit to each frame separately without having to use traditional tracking techniques to limit the search space. This means that our tracker will not get confused as easily by rapid movements or temporary occlusions. Another important benefit of using stereo depth data is that, once we find the head in the depth image, if we know the cameras' focal lengths and baselines, we can easily determine its position in 3D world coordinates.

We hope to use this information in several ways. For one, this is an important complement to a smart room where knowing the 3D position of a user's head is a way to steer a microphone array to listen in that direction. We would also like to use it as a first step in the bootstrapping of a more complicated articulated motion tracking system that can perform human gesture recognition. In addition, work is underway to use this system to provide real-time 3D head coordinates as an input to a face recognition algorithm similar to the one in [2].

The paper is organized as follows. Section 4 will discuss the algorithm for segmentation in detail. Section 5 will describe our torso model and its parameters, while Section 6 will discuss how we acquire that model in each frame. Section 7 presents our head localization algorithm. Our results and conclusions will be presented in Sections 8 and 9, respectively.

4 Segmentation

Our system begins with a segmentation of the human figure in the foreground of our image sequence. Conventional approaches using intensity images [7] create a Gaussian model of the intensity over a certain interval of time of each pixel in the background and then determine whether a pixel is part of the foreground based on its distance from the background in the chosen color space. This method has two important limitations. First, it is extremely sensitive to variations in lighting conditions. For example, if the lighting suddenly changes, the background model is no longer valid and the resulting segmentation is incorrect. Similarly, the effects of shadows are very difficult to handle. If the foreground figure casts a shadow, the darkened region could differ enough from the background to be classified as foreground. In addition, if the foreground figure happens to be similar in color to the background, it will be classified as background.

4.1 Using Depth Data

The use of stereo eliminates the aforementioned problems. With depth images, we proceed as before, modeling each background pixel as a Gaussian with a mean μ and a standard deviation σ. This time, however, we build the model with depth instead of intensity values. Closely following the work of [12], once we build a depth model of the background, we can identify the foreground as any region where the depth is sufficiently closer to the camera than the background. This is much more physically intuitive than the intensity segmentation and more accurate as well. Because the nature of the stereo correlation calculation makes it insensitive to color, shadows, or lighting variations, we do not have to worry about the previous problems.

The segmentation, however, is not quite this simple. Stereo matching is extremely sensitive to image texture. In our case, the correlation-based stereo system has a great deal of difficulty operating in regions where there is little texture. For example, consider a blank wall. A stereo system attempting to correlate pixels in such an untextured region will have a difficult time finding the correct matches as all pixels look alike. The result is an area of incorrect matches yielding disparities more or less randomly distributed throughout a range dependent on the size of the correlation window. This noise, depicted in Figure 1 is neither Gaussian nor white, making it very difficult to model. Unfortunately, this adversely affects our segmentation as we cannot effectively model the background in regions without adequate texture. We can, however, identify those background pixels that are unreliable with a simple test of our model's standard deviation: if $\sigma_{ij} > \beta$ where β is a user-defined threshold (we used $\beta = 2$). To combat this problem in untextured regions, we've devised our own segmentation scheme, closely related to [12] but with an important additional validation step.

Fig. 1. TOP: Sample input image from camera. White square highlights region with little texture. MIDDLE: Disparity image from the Digiclops^{TM} Stereo System from Point Grey Research, Inc. Note noise in the highlighted region. BOTTOM LEFT: Reconstruction of physical surface based on disparity values in highlighted region (negative z axis trends away from camera). BOTTOM RIGHT: Reconstruction of physical surface based on region centered inside of the human figure in the middle.

4.2 Surface Validation Segmentation

Once we've modeled the background, we face the problem of picking out the foreground in a new disparity[2] image DI. In the case where a pixel of the foreground DI_{ij} is in front of a reliable background pixel ($\sigma_{ij} < beta$), we've demonstrated that the segmentation is simple. All we must say is that a pixel is part of the foreground if the disparity value at that point is in front of the mean background by more than a standard deviation. When we are dealing with an unreliable background pixel ($\sigma_{ij} \geq beta$), things get much more complicated. In these cases, since μ_{ij} is not a reliable representation of depth, we cannot know based only on the value of DI_{ij} whether that pixel is in front of the background or not. Here we make an important assumption: the foreground figure must consist of a smooth blob of pixels with similar disparity values. In other words, it should be distinguishable from untextured background in that its disparity values suggest a surface that is smooth and realistic (Figure 1, (bottom right)), not noisy and spiked like the one in Figure 1 (bottom left). Assuming we can identify all regions in an image that can be considered smooth physical surfaces, segmentation of the foreground is as simple as identifying all such surface regions that occur in front of the background. To find these surfaces, we use a modified connected components algorithm as follows:

1. consider DI to be a graph G where every vertex G_{ij} corresponds to a pixel DI_{ij}
2. for each vertex G_{ij} connect it to its four neighbors if and only if $\sum_{n \in N_{ij}} |G_{ij} - G_n| < t$ where N_{ij} is the 4-neighborhood of vertex G_{ij} and t is a user-defined threshold. Since neighboring disparity values in untextured regions differ by large amounts, there is a great deal of latitude in choosing this threshold.
3. connected components larger than a nominal size are accepted as surfaces and all other pixels are ignored. The size cutoff is also relatively easy to choose and is based on the assumption that the human figure will take up at least 10% of the image.

4.3 Segmentation Algorithm

Thus, our segmentation algorithm is:

1. using 20-30 images, model the background using a Gaussian $[\mu, \sigma]$ for each pixel
2. based on the values for standard deviation, determine unreliable background models by looking for pixels with $\sigma > \beta$ where β is a user-defined threshold (we used $\beta = 2$)
3. for each subsequent disparity image DI, calculate the areas considered to be physical surfaces
4. a surface pixel DI^s_{ij} is classified as foreground if
   a) the background is reliable at $[i, j]$ and $DI^s_{ij} > \mu_{ij} + \sigma_{ij}$
   b) the background is unreliable at $[i, j]$
5. finally, to eliminate all remaining noise, we run a binary connected components algorithm and extract the largest component.

[2] In the rest of the paper, we refer to disparity images obtained using Digiclops; these are the depth images used in our experiments. Digiclops is a commercial stereo system from Point Grey Research. It uses 3 cameras to do multiple-baseline stereo disparity calculation. Calibration is done in the factory and, according to Point Grey, the input images are rectified to fit an ideal stereo camera model within 0.06 pixels. The system also has a "calibration retention system" which makes it robust to shocks and vibrations. Noise is also an issue with these systems; however, the image transmission in Digiclops is completely digital (via an IEEE 1394 interface), which removes the problems of frame grabber jitter and analog-to-digital conversion noise. That said, Digiclops suffers from the same limitations every other stereo system has. It does not perform well in areas without much image texture, it does not handle specular reflections, and it cannot handle occlusions.

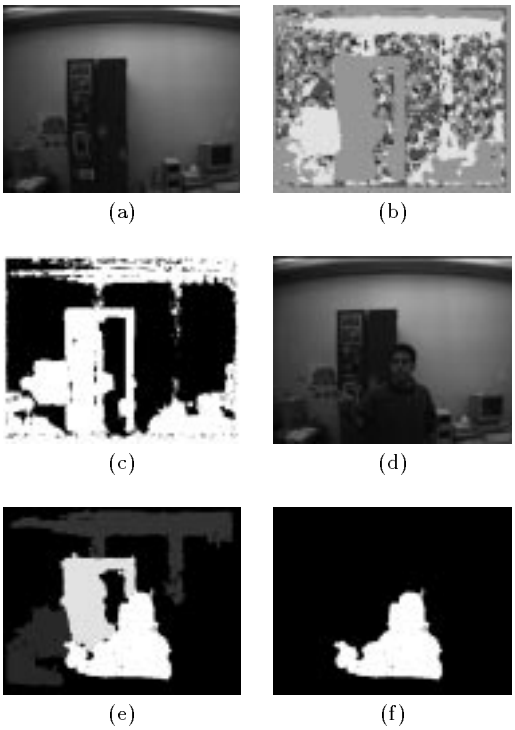


Fig. 2. Illustration of segmentation: (a) One of the images in the input sequence of background images. (b) Disparity map output from Digiclops for this image. (c) Illustration of reliable (white) background pixels based on standard deviation of background model. (Steps 1 and 2 of our algorithm) (d) New, test image for foreground segmentation. (e) Results from surface validation (Step 3). Surfaces shown (all regions except black) passed the validation requirement. (f) Final result of segmentation (Steps 4 and 5).

We've already looked at the first case in step 4, but the second deserves a bit of explanation. When we classify the background as unreliable, we are implicitly assuming that it is located in an untextured region. This is a safe assumption as it is generally only in those regions that the disparity value would fluctuate so much from frame to frame. As a result, the data in this region do not correspond to a physical surface and would not aggregate into a connected component large enough to classify as a surface. So, if we see a surface pixel where we expect to find an unreliable background pixel, we know that it must be part of the new foreground. Figure 2 illustrates these concepts.

5 Torso Model

Once we have an accurate segmentation, we look to fit a simple torso model to the foreground figure. The model is based on the assumption that the figure is upright or leaning slightly to one side, a reasonable assumption for our domain of interest. Given this, we notice that the occluding edges of the shoulders, heretofore referred to as the *mantle*, are strong cues that vary very little with respect to the motion of an upright figure. Thus our torso model consists only of five parameters, two for the straight line that captures the general lean of the figure and three for the quadratic that traces the outline of the mantle. Figure 3 shows the model and its application to

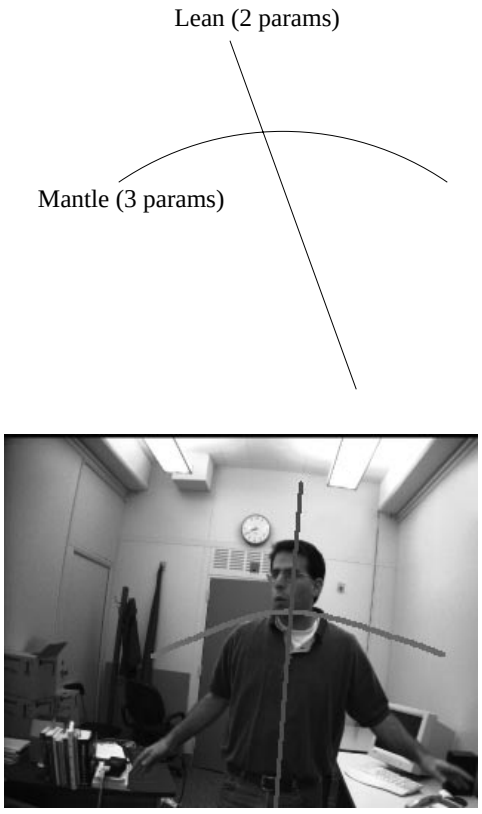


Fig. 3. TOP: Illustration of simple torso model. BOTTOM: Application of torso model to image.

a real image. We loosely interpret the intersection of the lean and the mantle as being the neck point. This will be useful later as we're looking to localize the head.

6 Model Acquisition

Because we have such a simple model, it is relatively easy to acquire. Given the segmented foreground figure as a binary image, we take advantage of the depth information stereo gives us to help us extract the lean in the following way:

1. For each row i of the binary image, calculate the median of the column values of the foreground pixels. We will call this value the *horizontal median*. We use medians instead of means because they are less sensitive to outliers in the foreground figure caused by waving arms (Figure 4).
   In addition, we use a second level of outlier rejection based on the perceived 3D position of the figure. To do this, we need a rough approximation of the 'center' (C_{fg}) of the figure in 3 dimensions. If we can make a reasonable guess about this point, we can ignore foreground pixels that are too far away horizontally from it (e.g. those belonging to waving arms). Since we're using stereo and have access to depth data, we can define the term 'too far away' in world space

Fig. 4. Illustration of the horizontal mean values (pluses) vs. horizontal median values (squares). Notice how much more the means are affected by the waving arm.

and not in image space. This allows us to handle figures at any depth and maintain scale independence without having to resort to messy multi-resolution calculations. We make our guess about C_{fg} using the assumption that the majority of the lower area of the figure is usually evenly distributed around the center of the figure, a reasonable assumption for our domain of interest. This area encompasses a figure's legs and lower to mid-torso region and its horizontal medians are very seldom disturbed by waving arms. We can get a value for the image coordinates of C_{fg} by finding the centroid of the horizontal medians calculated on the lowest 33% of the foreground figure. Once we have C_{fg} in image coordinates, we can use our depth data and the known camera parameters to project this point into 3D world coordinates. (Figure 5).

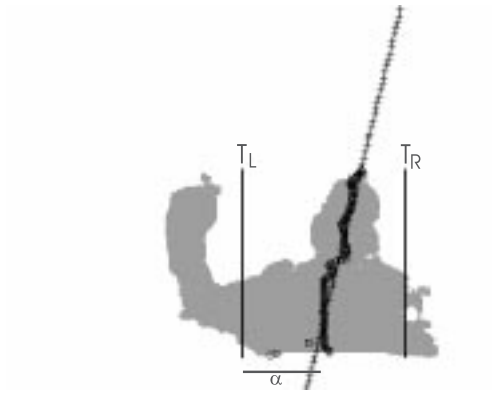


Fig. 5. Example of thresholds (depicted as vertical bars) in action. The dark circles represent the recalculated horizontal medians and the pluses represent the line that best fits them.

2. Perform outlier rejection in the following manner:
   - Set pixel threshold columns to the left (T_l) and right (T_r) of C_{fg} by calculating the distance in pixels represented by world displacements of α centimeters to the left and right of C_{fg}, respectively. The idea here is that α should approximate the half-width of a standard human figure
   - Using the thresholds, reject foreground pixel values as outliers if their column component does not fall within the left and right thresholds. Recalculate the horizontal medians based on this new information.
3. using a Singular Value Decomposition line fit, find the best-fit line to the adjusted horizontal median values for each row. That line is the lean.

In our experiments, we used $\alpha = 25$ centimeters though there is a fair amount of latitude in this choice. We used SVD line fitting instead other, less expensive approaches because of its numerical robustness. Specifically, its stability makes it a good deal less sensitive to perturbations of the data [49]. Figure 5 depicts an example of the lean acquisition.

Once we have the body lean, we can start acquisition of the mantle. As we mentioned before, the strongest cues are the occluding edges of the shoulders on either side of the head. To find these we employ directed local edge detectors similar to those used in [13]. We orient these detectors perpendicular to the lean and look for edges by thresholding the image gradient along slices perpendicular to the detector's orientation (parallel to the lean). Figure 6 offers an illustrated example of such an edge detector. To place these detectors in the best position, we use the depth data of the points along the lean to give us an estimate of how far the figure is from the camera. Based on this information, we can determine where, in image coordinates, to place the edge detectors. More specifically, once we decide where in world coordinates we'd like to place the edge detectors, using our camera parameters we can project these points back into image coordinates. For example, if the figure is close to the camera, we're going to look for edges along a much longer line than if it is further away. Figure 7 shows an illustration of these concepts. We use the assumption that the typical head is .2 meters wide and that the typical mantle is .4 meters across.

We place a series of these local edge detectors up and down the image perpendicular to the body lean and keep a running tally of how many potential edge points we find. After searching the length of the body lean line, we select the pair of trackers that yield the most edge points and, using least squares, fit a quadratic to those points. That quadratic is the mantle and represents the final three parameters of our model.

7 Head Localization

Once we've acquired a model, we calculate the intersection of the mantle and the lean, which we interpret as the neck. We then look radially out from the neck at points in the foreground that are:

- 'above' the mantle
- within a reasonable distance (.2 meters) in world coordinates from the neck (again, we can do this because we are working with 3D data)

After identifying such points, we calculate their centroid and make the assumption that, regardless of tilt, this point will represent the center of the head. We can now

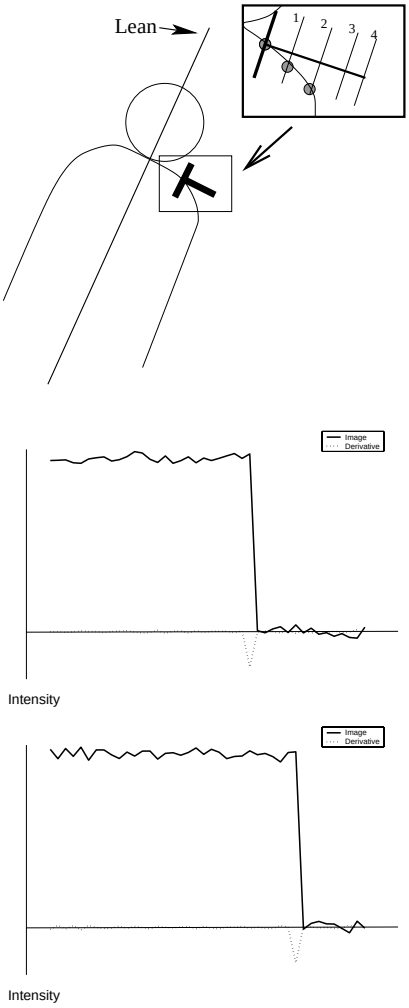


Fig. 6. TOP: Example of a directed local edge detector. The detail of the figure shows examples of four of the slices along which the image gradient is calculated. The gray dots represent occluding edge points found by the detector. MIDDLE: graph of 1D slice of the image intensity together with its gradient along slice 1 of the edge detector above. BOTTOM: same as above except with slice 2 of the edge detector.

determine the orientation of the head simply by calculating the angle made by the line containing the head's centroid and the neck point. We assume that the distance between those two points is half of the height of the head and can easily draw a box around it (Figure 7). Also, since we're using stereo, once we know the centroid of the head region we can easily figure out its position in 3D.

8 Results

Figures 8-12 illustrate the results of our tracking scheme. They were all recorded using the same values for the aforementioned user-defined thresholds and parameters for each experiment. The choice of these parameters was easy for this particular domain and the experiments show that a single choice can handle different people and different motions in this domain. The sequences feature a

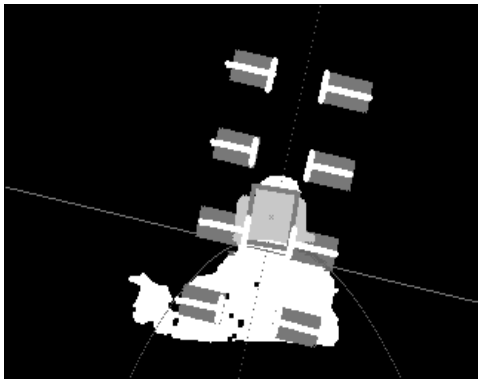


Fig. 7. This image shows the lean, the mantle, and a few examples of the placement of the local edge detectors (gray rectangles). Also, the light grey area above the mantle line represents points classified as being part of the head. The 'x' represents the centroid (in image coordinates) of those head points.

variety of skin tones, cluttered backgrounds and rapid head movements that would be likely to confuse a tracker that relied on accurate predictions based on past motion. Plotted on each image is our acquired torso model as well as the orientation of the head. Figure 8 shows the tracker's ability to track changes in head orientation. Similarly, Figure 9 illustrates the tracker's work on a figure approaching the camera and then moving away. Since we can adjust our algorithm using our knowledge of the depth of the figure, we maintain scale independence without any significant complications. Both figures also demonstrate our tracker's ability to work without any assumptions based on skin tone. The figure's dark skin is something that would confuse many of the trackers that rely on the identification of "skin colored" pixels.

Figures 10 and 11 show the tracker's ability to work in the presence of waving arms and image clutter. Figure 10 also shows one of the failure modes of the system. When the assumption that the figure more or less faces the camera (a reasonable assumption for our domain of interest) is violated, the shoulder cues are not always strong enough to lead us to the correct configuration of the model. Fortunately, since our next step is entirely independent of the previous one, we are not confused for long and reacquire the figure soon after.

Figures 12 illustrates two failure modes of the tracker. At the top, the figure comes too close to the edge of the field of view so that our torso model cannot be acquired, and at the bottom, the figure's arms occlude the head and shoulders, obscuring our most important cues. In both cases, a simple tracker could easily get thrown off and have a difficult time finding the target again. In our case, however, regardless of where the figure is, we simply reacquire our torso model as soon as it becomes available again.

Figure 13 shows one of the important side effects from using stereo. Since we are using stereo and we know the cameras' intrinsic and extrinsic parameters, once we find where the head is located in image coordinates, we can easily turn that into a 3D point. As a result, we can track the movement of the head throughout a room in 3D.

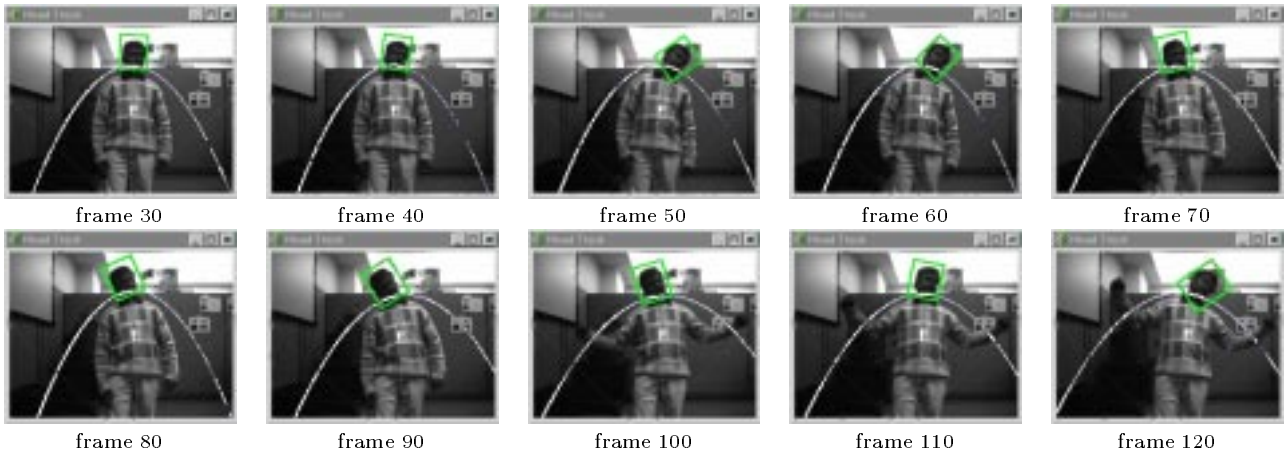


Fig. 8. Results from a sequence of rapid head movements. The images were acquired at about 2 Hz so this sequence lasts about 45 seconds.

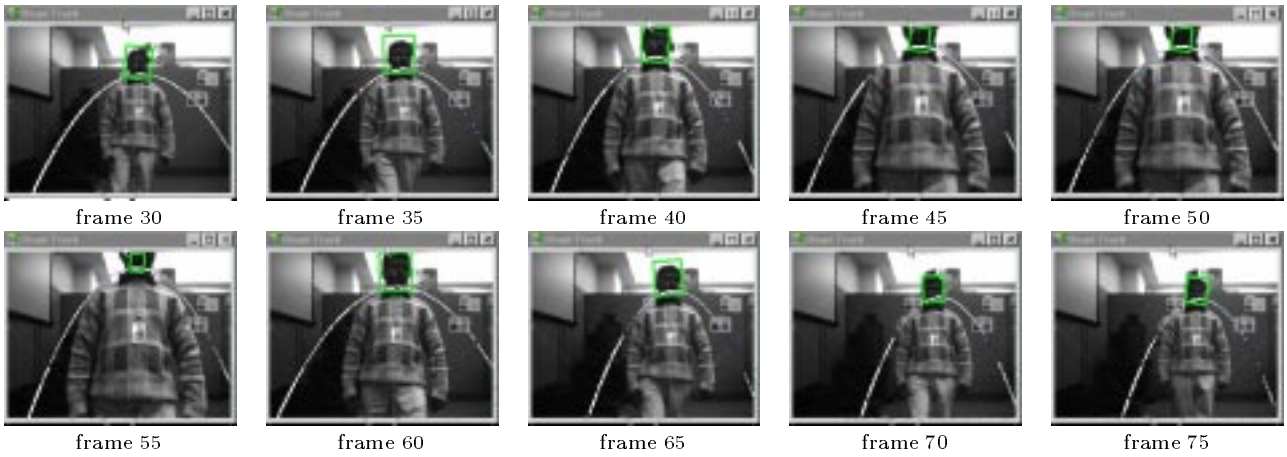


Fig. 9. Results from a figure approaching and then walking away from the camera. The images were acquired at about 2 Hz so this sequence lasts about 22 seconds.

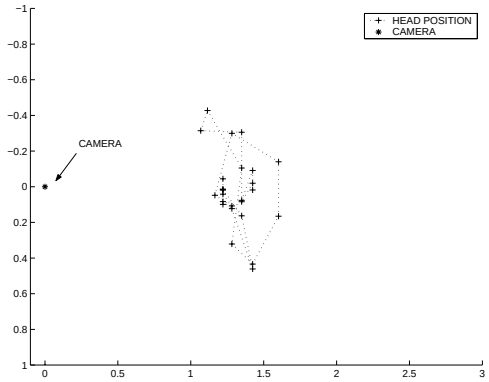


Fig. 13. Bird's eye view of head movement through room in sequence from Figure 11. Both axes are in meters.

8.1 Performance

This system is run using a resolution of 320x240 pixels and the processing time per frame is approximately one second on a dual Pentium II 350 MHz.

9 Conclusion

What we have shown is a new approach to head tracking taking advantage of stereo depth data as well as the segmentation accuracy real-time stereo affords. We've created a simple torso model that is quick to acquire and does not require accurate predictions between frames to work. As a result, we can ignore the common assumption of small interframe motions as well as the problems generated by occlusions. We use this system to track heads in 3D throughout a room.

9.1 Future Work

As mentioned earlier, work is underway to use the results of this algorithm as input to a steerable phased array of microphones in order to achieve more accurate voice recognition without the use of user-mounted microphones. Also, we hope to use the 3D position of the head as input to a face recognition algorithm or to bootstrap a more complicated articulated motion tracker.

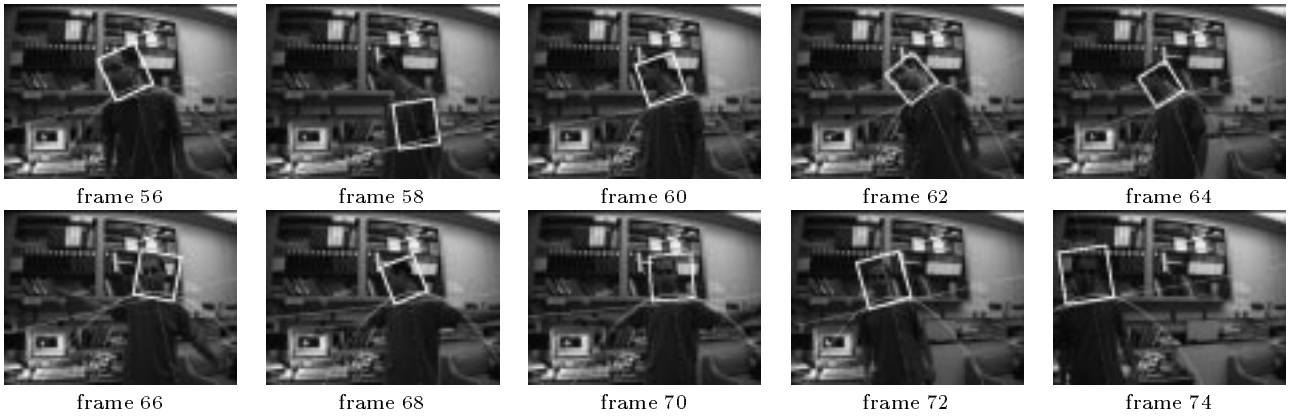


Fig. 10. Results from another rapid sequence of head and arm movements with a cluttered background. Notice the failure of the tracker (frame 58) when the figure is turned too much to the side reducing the strength of the shoulder cues. The timing is the same as in Figure 9

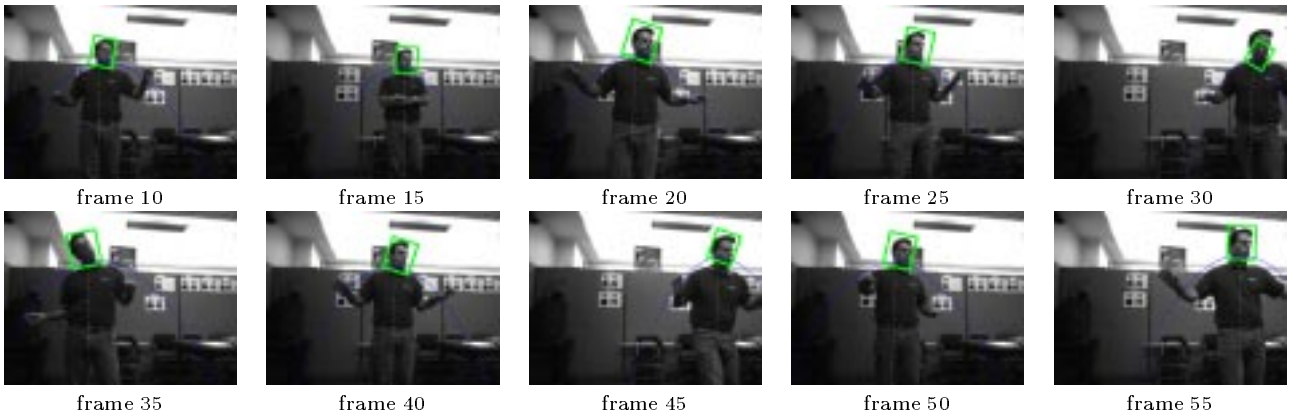


Fig. 11. More results from rapid sequence of head and arm movements with a different figure. The timing is same as in Figure 9

As for extensions of the algorithm itself, a relatively easy one would be to track multiple heads in an image. This is a simple matter of identifying all of the blobs in the foreground and acquiring a torso model for each. Another extension under consideration is to add a very simple prediction step that would reduce the computation time to acquire a model, but not sacrifice the robustness of our 'one image at a time' system. We could also integrate this prediction step as a separate module. In this way we might have the model acquisition and prediction act independently and then use a comparison function to decide which solution makes the most sense. This could potentially eliminate failure modes in which one of these approaches fails but the other doesn't.

Other interesting extensions could be used to handle more difficult failure modes. For example, if a figure is holding a large object that is occluding its shoulders, or if the figure is occluded by another person, our algorithm will fail. We would need to be able to either recognize that situation and handle it gracefully or, potentially, perform a more intelligent segmentation of the foreground into layers, recognize the boundaries between them and ignore all pixels except those actually belonging to the figure. This segmentation would obviously require extensive work, especially to make it recognize such complicated and smoothly-varying boundaries in real time. However, advances in image segmentation techniques suggest that it might not be out of reach [47].

Acknowledgments

The authors would like the thank Michael Benharrosh and Vince Stanford for many fruitful discussions. Many thanks also go to Mani Muniyandi for his help with the Results section.

References

1. T. Darrell, B. Blumberg, S. Daniel, B. Rhodes, P. Maes, and A. Pentland, "Alive: Dreams and Illusions," *ACM SIGGraph, Computer Graphics Visual Proceedings*, July, 1995.
2. T. Darrell, B. Moghaddam, and A. Pentland, "Active Face Tracking and Pose Estimation in an Interactive Room," *IEEE Conference on Computer Vision and Pattern Recognition*, June, 1996.
3. S. Birchfield, "An Elliptical Head Tracker," *31st Asilomar Conference on Signals, Systems, and Computers*, pp. 1710-1714, November, 1997
4. S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 232-237, June, 1998.

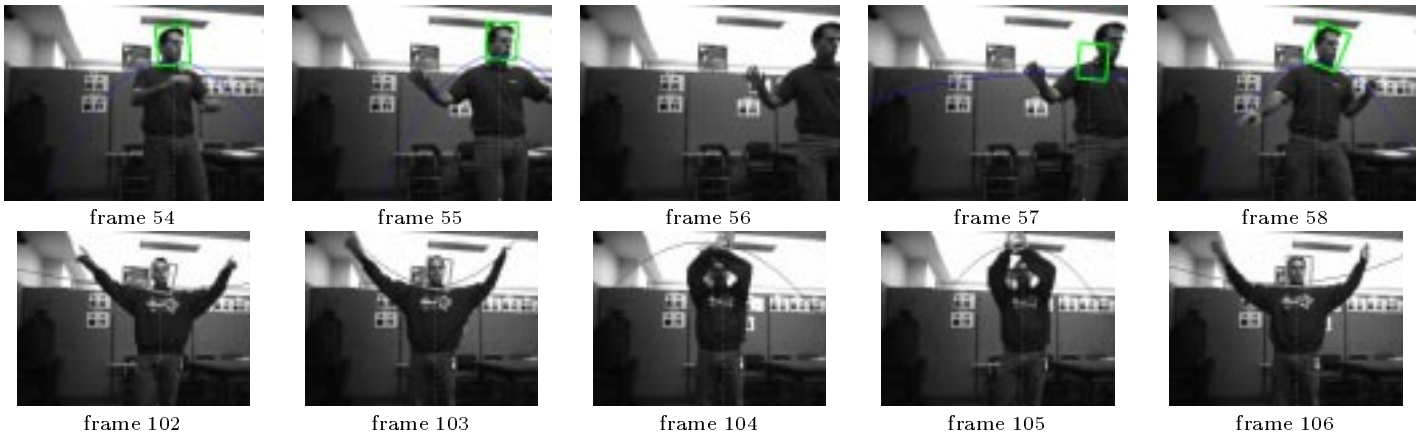
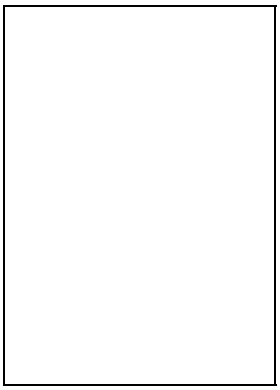


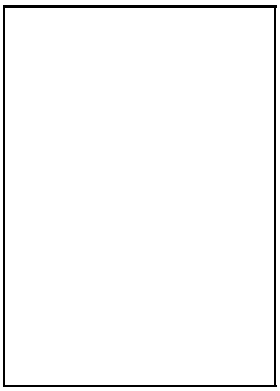
Fig. 12. TOP: Failure and reacquisition when figure moves out of field of view. BOTTOM: Failure and reacquisition when figure's head and shoulders are occluded. The timing is the same as in Figure 9

5. M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 322-336, April, 2000.
6. S. Basu, I. Essa, and A. Pentland, "Motion Regularization for Model-based Head Tracking," *Proceedings of the International Conference on Pattern Recognition*, August, 1996.
7. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 780-785, July, 1997.
8. A. Azarbayejani, C. Wren, and A. Pentland, "Real-time 3-D Tracking of the Human Body," *Proceedings of IMAGE'COM 96*, May, 1996.
9. S. Niyogi and W. Freeman, "Example-Based Head Tracking," *IEEE 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, October, 1996.
10. K. Konolige, "Small Vision Systems: Hardware and Implementation," *Eighth Interational Symposium on Robotics Research*, October, 1997.
11. T. Darrell, G. Gordon, J. Woodfill, and M. Harville, "Integrated person tracking using stereo, color, and pattern detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 601-609, June, 1998.
12. C. Eveland, K. Konolidge, and R.C. Bolles, "Background Modeling for Segmentation of Video-Rate Stereo Sequences," *IEEE Conference on Computer Vision and Pattern Recognition*, June, 1998.
13. J. Rehg and T. Kanade, "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking," *Proceedings of the European Conference on Computer Vision*, pp. 35-46, May, 1994.
14. N. Jojic, M. Turk, and T. Huang, "Tracking Self-Occluded Articulated Objects in Dense Disparity Maps," *International Conference on Computer Vision*, pp. 123-130, September, 1999.
15. M. Turk, "Visual Interaction With Lifelike Characters," *IEEE 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, October, 1996.
16. I. Essa, S. Basu, T. Darrell, and A. Pentland, "Modeling, Tracking and Interactive Animation of Faces and Head Using Input from Video," *Proceedings of Computer Animation*, pp. 68-79, June, 1996.
17. E.E. Saad, T.P. Caudell, and D.C. Wunsch, II, "Predictive Head Tracking for Virtual Reality," *International Joint Conference on Neural Networks*, pp. 3922-3936, July, 1999.
18. H.A. Sowizral and M.F. Deering, "The Java 3D API and Virtual Reality," *IEEE Computer Graphics and Applications*, pp. 12-15, May, 1999.
19. E. Wegman, "Affordable Environments for 3D Collaborative Data Visualization," *Computing in Science and Engineering*, pp. 68-72, Nov, 2000.
20. Y. Yanagida, T. Maeda, and S. Tachi, "A Method of Constructing a Telexistence Visual System Using Fixed Screens," *IEEE Proceedings of Virtual Reality*, pp. 117-124, March, 2000.
21. O. Bimber, L.M. Encamacao, and D. Schmalstieg, "Real Mirrors Reflecting Virtual Worlds," *IEEE Proceedings of Virtual Reality*, pp. 21-28, March, 2000.
22. D. Kim, S.W. Richards, and T.P. Caudell, "An Optical Tracker for Augmented Reality and Wearable Computers," *Proc. of IEEE Virtual Reality Annual International Symposium*, pp. 146-150, March, 1997.
23. S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A Touring Maching: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," *International Symposium on Wearable Computers*, pp. 74-81, October, 1997.
24. H. Mizoguchi, T. Shigehara, M. Yokoyama, and T. Mishima, "Virtual Wireless Microphone– A Novel Application of Real-time Visual Tracking and Sound Signal Processing," *Proceedings of the 37th SICE Annual Conference*, pp. 999-1004, July, 1998.
25. W.G. Gardner, "Head Tracked 3-D Audio Using Loudspeakers," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 19-22, October, 1997.
26. A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually Controlled Graphics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 602-605, June, 1993.
27. T.S. Jebara and A. Pentland, "Parametrized Structure from Motion for 3D Adaptative Feedback Tracking of Faces," *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
28. T. Horprasert, Y. Yacoob, and L.S. Davis, "Computing 3-D Head Orientation from a Monocular Image Sequence," *Proceedings of the International Conference on Face and Gesture Recognition*, 1996.
29. D. DeCarlo and D. Metaxas, "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
30. H. Li, P. Rovainen, and R. Forcheimer, "3-D Motion Estimation in Model-Based Facial Image Coding," *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, pp. 545-555, June, 1993.
31. D. Terzopoulos and L. Waters, "Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 569-579, June, 1993.
32. I.A. Essa and A.P. Pentland, "Coding Analysis, Interpretation, and Recognition of Facial Expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 757-763, July, 1997.
33. M.J. Black and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Model of Image Motion," *International Journal of Computer Vision*, pp. 23-48, 25(1) 1997.
34. P. Fieguth and D. Terzopoulos, "Color-based Tracking of Heads and Other Mobile Objects at Video Frame Rates," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21-27, 1997.
35. J. Crowley and F. Berard, "Multi-modal Tracking of Faces for Video Communication," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 640-645, 1997.
36. K. Sobottka and I. Pitas, "Segmentation and Tracking of Faces in Color Images," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 236-241, 1996.
37. M. Collobert, R. Feraud, G. Le Tourneur, O. Bernier, J.E. Viallet, Y. Mahieux, and D. Collobert, "LISTEN: A System for Locating and Tracking Individual Speakers," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 283-288, 1996.
38. N. Oliver, A. Pentland, and F. Berard, "LAFTER: Lips and Face Real-Time Tracker," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 123-130, 1997.
39. L. Jordao, M. Perrone, J. Costeira, and J. Santos-Victor, "Active Face and Feature Tracking," *Proceedings of the International Conference on Image Analysis and Processing*, pp. 572-576, 1999.
40. K. Yachi, T. Wada, and T. Matsuyama, "Human Head Tracking Using Adaptive Appearance Models with a Fixed-Viewpoint Pan-Tilt-Zoom Camera," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 150-155, 2000.
41. R.J. Qian, M.I. Sezan, and K.E. Matthews, "A Robust Real-Time Face Tracking Algorithm," *Proceedings of the International Conference on Image Processing*, pp. 131-135, 1998.
42. M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proceedings of the European Conference on Computer Vision*, pp. 343-356, 1996.
43. Y. Raja, S.J. McKenna, and S. Gong, "Tracking and Segmenting People in Varying Lighting Conditions Using Colour," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 228-233, 1998.
44. J. Triesch and C. von der Malsburg, "Self-Organized Integration of Adaptive Visual Cues for Face Tracking," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 128-134, 2000.
45. F. J. Huang and T. Chen, "Tracking of Multiple Faces for Human-Computer Interfaces and Virtual Environments," *IEEE International Conference on Multimedia and Expo*, pp. 1563-1566, 2000.
46. V. Kruger and G. Sommer, "Affine Real-Time Face Tracking Using a Wavelet Network," *International Conference on Computer Vision*, pp. 141-148, 1999.
47. P. Felzenszwalb and D. Huttenlocher, "Image Segmentation Using Local Variation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 98-104, June, 1998.
48. C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, November, 1999.
49. M. Heath, *Scientific Computing: An Introductory Survey*, McGraw-Hill, New York, 1997.



Daniel Russakoff received an AB in Geophysics from Harvard University in 1996 and an MS in Computer Science from Stanford University in 1999. He spent a year working in the Smartspaces Laboratory at the National Institute of Standards and Technology before returning to Stanford to pursue a PhD in Computer Science. His research interests include stereo, tracking, and medical image registration.



Martin Herman received a Ph.D. in computer science from the University of Maryland. He is currently Chief of the Information Access Division, National Institute of Standards and Technology (NIST). He is responsible for the overall program in research, measurements, testing, and standards in information access technologies at NIST, including speech processing and human language technology, multimedia information retrieval, image recognition, visualization and usability testing, and smart spaces. Previously, he was Group Leader of the Perception Systems Group at NIST, and held a faculty appointment at Carnegie Mellon University. He has performed research in computer vision, smart spaces, robotics, and automated manufacturing, and has published over 75 papers in these areas. He was Program Co-Chair of the IEEE Workshop on Applications of Computer Vision, 1992, and was a Guest Associate Editor for the IEEE Transactions on Systems, Man, and Cybernetics Special Issue on Unmanned Vehicle and Intelligent Robotic Systems, 1990.

This article was processed by the author using the LaTeX style file *cljour2* from Springer-Verlag.