Geometric-Imprints: A Significant Points Extraction Method for the Scan&Track Virtual Environment

Sudhanshu K Semwal[1,2] Jun Ohya[1]
[1]ATR Media Integration & Communications Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan 619-02
[2]Department of Computer Science, University of Colorado, Colorado Springs
Colorado Springs, CO, U.S.A. 80933-7150
semwal@redcloud.uccs.edu|ohya@mic.atr.co.jp

Abstract

We have implemented a new algorithm, called the *geometric-imprints* method for extracting key-features of a given contour. Series of points on the curve are used to find a geometric-imprint set of the contour. The algorithm is based upon the assumption that an articulated human and other figures have mostly cylindrical body-parts. The algorithm recursively partitions the contour based upon the cylindrical nature of human-silhouettes, and a starting point. Similar topological contours are expected to produce similar geometric-imprints. Thus the correspondence problem is somewhat simplified. Results of our implementation are presented.

Key words: Geometric Feature Identification, Tracking of articulated objects, 2D contours, Cylindrical Body Shapes.

Introduction

Virtual environments (VEs) are expected to understand gestures of participants as a first step towards *understanding* the participants. Understanding the intention of the participant is difficult as an inherent ambiguity exist. For example, a human-participant is perfectly capable of putting on an act, and their gestures may be understood by a VE to convey a completely opposite message than what the participant intended. In fact, given the same gestures it can be interpreted differently depending upon the context. For example, some gestures may carry a completely opposite meaning in different parts of the world. However, most of the time, participants wish to create meaningful tasks in a VE. Their body movements, gestures, facial expressions, and speech is purposeful, and actual intention is what it appears to be. In this case, VEs can be extremely useful. There has been much research effort in the area of understanding and recognition of facial expression, body movements, gestures and gaze of the participants [1, 2, 3]. Fundamental to all of this is the analysis and estimation of motion of the participant in a VE. For applications, where the participant does not wish to be tethered to the devices, usually video-images are analyzed. Krueger's work on analyzing images for 2D information and interaction based on the 2D contours has been well documented [4]. When multiple camera-images are analyzed for extrapolating the 3D information inherent in the scene, the well known correspondence problem is ever-present. Magnetic sensors, placed on several places on the body of the human participant, can be used for detecting the motion of the participant. The motion is then used to control a synthetic human-form or an avatar of the participant. Some solutions to drive an avatar using minimal sensors have been already developed [5, 6, 7, 8]. Interactive video-environments and wearable computers are also the emphasis in [9]. When we use encumbering devices on the participant, we actually avoid much of the correspondence problem. The correspondence problem is severe in camera-based applications because multiple camera images are analyzed and enough information has to be extracted from these images so that the position, motion, and other details of the participant can be determined. In addition, camera based systems also face the inherent ambiguity of 2D-camera projections. This can be illustrated by arranging fingers so that the shadow projected on the wall can, for example, look like a goat.

Related Work

Extraction of significant points, or key-features from an image, has been investigated in great detail [1, 2]. The idea behind significant point extraction is that a small collection of points may provide enough information about the intent or pose of the participant. Tracking the trajectory of an object is the focus of research in [10]. In Multiple Light Display, multiple beacons of light are used to analyze the motion of human participants [11]. Simple light displays placed on the participants are tracked. It is argued that twelve points are sufficient to give an impression of actual motion. This is interesting as it also may provide an answer to how many minimal significant points on the human-body are sufficient to understand the motion of the participant. The Virtual Kabuki system [12] uses an algorithm based on applying distance transformation for identifying significant points. The knee and elbow positions are estimated using genetic algorithms [12]. A variety of contour-extraction methods have been developed for gesture understanding [13]. The optical flow, using the intensity of certain areas of the image, is the basis in [14]. Immersive-video [15] application uses color averaging of multiple camera-images to determine the color of 3D grid-points. Snakes [16] use minimization and spline-

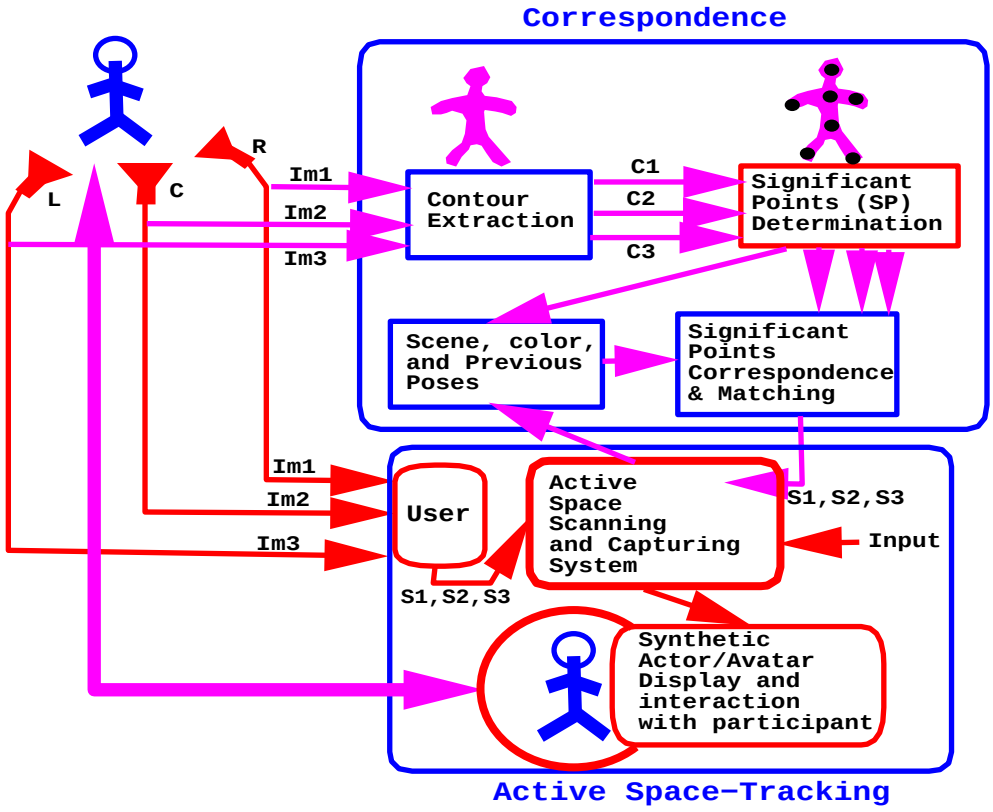


Figure 1: Block Diagram for the Scan&Track System

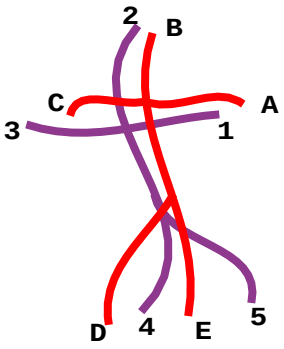
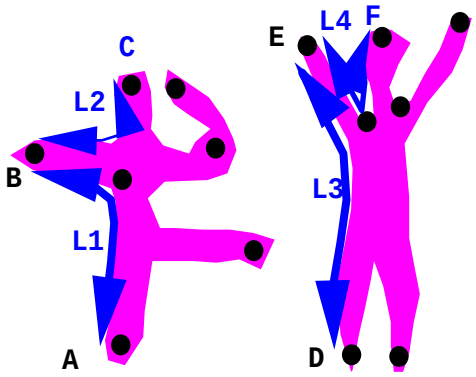


Figure 3: Correspondence Problem



(a) Dancing (b) Elation

Figure 2: Postures express emotions

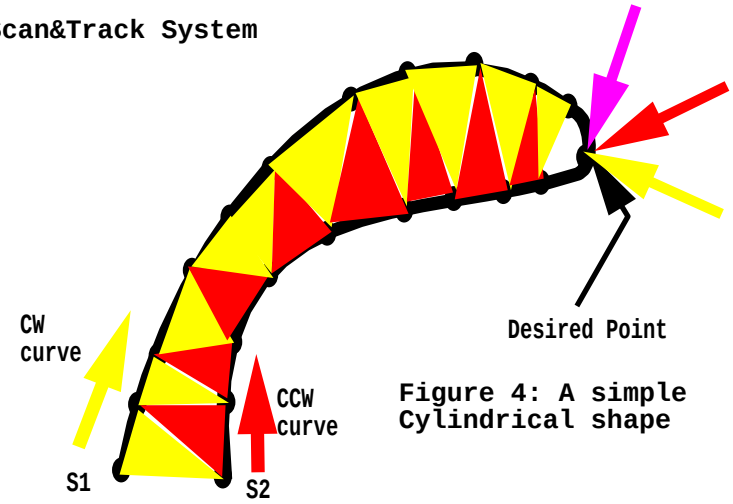


Figure 4: A simple Cylindrical shape

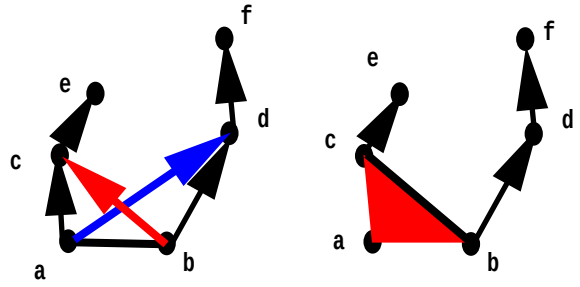


Figure 5: Next triangle selection based upon smaller area available

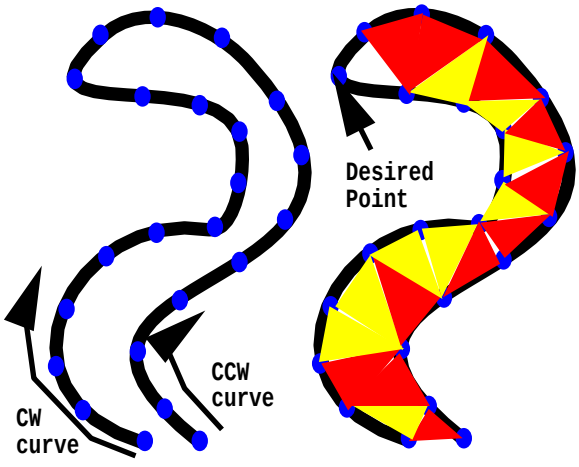


Figure 6: A modulating cylindrical shape

based contouring to lock onto an object of interest in the image. Blob models [9] are used in the Pfinder system to track 2D information.

The Scan&Track VE

We are developing an unencumbering VE, called the Scan&Track system, based upon the video image sequences from multiple cameras. The block diagram of the system is shown in Figure 1. There are two major components of the system: (a) Correspondence, and (b) Active Space Tracking. As shown in Figure 1, there are four sub-systems to the correspondence system: (i) contour extraction, (ii) significant points determination, (iii) significant points correspondence and matching, and, (iv) scene, color, and previous poses database. Let (S1, S2, S3) be a triplet representing a visible point S in 3D-space. As S1, S2, and S3 are the projection of the point S on the three camera images, the triplet (S1,S2,S3) is also called the image-imprint of S. If a 3D point is visible from multiple cameras, the location of the imprint of the 3D point in multiple camera-images can be used to estimate the 3D position of the point. We have implemented a mechanism, called active-space tracking, which allows us to determine the location of a point S, given its image-imprint (S1,S2,S3). The details of the active-space indexing mechanism are described elsewhere [17]. In the Correspondence system shown in Figure 1, the contours of the human-silhouette C1, C2, and C3 can be extracted from three images Im1, Im2, and Im3, respectively. These contours are used by the geometric-imprints algorithm for significant point determination. In the following sections, we describe the geometric-imprints algorithm.

The Geometric-Imprints Algorithm

Body postures express emotions and reveal our inner-self. For example, the contour-drawing in Figure 2a could be interpreted as a dance pose. Similarly, Figure 2b could mean that the person has just been victorious. The characteristics of both these poses can be captured by some key-points on the contour, as shown in Figure 2. Our geometric-imprints method is based on the observation that human body parts are mostly cylindrical in nature, especially those which are the basis of articulated motion. The geometric-imprints method extracts information about these cylindrical shapes from a contour. In case of Figure 2a, finger tips are the logical endings of the cylindrical human arm, and the bend at the elbow is quite obvious and should be detected. Similarly, we have circled some points on the the contour of Figure 2b to also indicate the geometric curve bending and the logical-ends of cylindrical body parts.

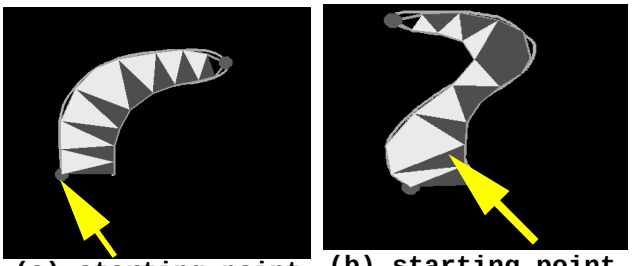
We wish to capture the points of the cylindrical endings of a 2D contour from the human-silhouette as shown in Figure 3. Some other points may be included depending upon the curvature of a given contour. We consider this set of points as *geometric-imprints* of the image. This is because these points and their topology facilitates determination of the pose of the participant. We can describe the pose of Figure 2a as follows: one hand is straight and the second hand is bent at the elbow. Notice that the elbow position of the second hand is not that necessary to determine the pose in 2a. To obtain this geometric information is the goal of the geometric-imprint algorithm. We conclude that geometric interpretation can be useful in estimating the pose of the participant. Our motivation for developing the geometric-imprints method was to also reduce the complexity of the correspondence problem. Consider 2D-contours extracted from multiple camera-images looking at a cylindrical object. The tip of the cylindrical object would project as a tip of a 2D-cylindrical curve in most of these 2D-contours. This has been shown as *desired point* in Figure 4. It is much easier to correspond extremities of curves across multiple camera images using the active-indexing mechanism [17], in comparison to finding these correspondences through only computer vision methods. Geometric-imprints are dependent primarily upon the analysis of contours of an image, thus our approach is different than other existing methods where we need to estimate the same key-feature of the image in every frame. For example, in many approaches, as well as our earlier efforts [12], the same joints are estimated in every frame. This is certainly useful when our display program is skeleton based where the joint information is needed for placing a synthetic human-actor in a desired pose. With this understanding, we now present the geometric-imprints method.

As we will explain later also, in the geometric-imprints algorithm, we take a different approach. As the participant assumes different positions, the shape of participant's silhouette on the camera-images would also change. Therefore we expect the geometric-imprint set to vary, from one frame to another, depending upon the shape of the curve in the camera-images from one moment to another.

Cylindrical Shape

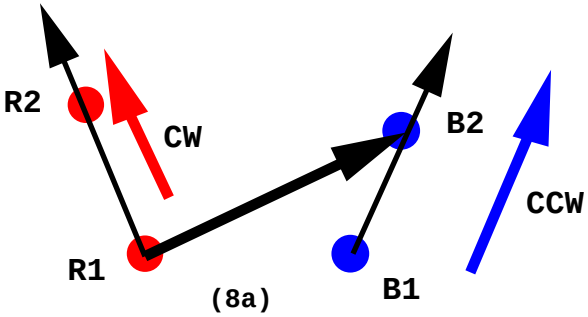
In our implementation, we first draw an arbitrary shape, and then specify a set of points on the curve in the clock wise direction. The geometric-imprint point is the tip of the cylindrical shape as shown in Figure 4. Then the beginning and the end points are selected as starting points S1 and S2 as shown in Figure 4. We move point S1 in the clock-wise (CW) and point S2 in the counter clock-wise (CCW) direction w.r.t. the curve. The algorithm we have implemented is based upon the triangulation of the area [18], but there is an important difference: our goal is not triangulation, instead we want to end up at the tip of the cylindrical shape so that that point can be identified as a geometric-imprint.

Let point c and d be two points in CW and CCW direction, respectively, as shown in Figure 5. Let line(a,b) be the starting point. The selection process is as follows: if distance(line(b,c)) is less that the distance(line(a,d)) then triangle(a,b,c) is colored and the new base is line(c,b) otherwise triangle(a,b,d) is colored with new base as line(a,d). The above is repeated until we have just one point left. In this case, as shown in Figure 5, the correct point is found. This simple algorithm is similar to that implemented for joining two curves on different CT-slices in [19]. This algorithm should also work for a modulating curve as shown in Figure 6. The algorithm works for simple cylindrical shapes because we always advance the CW-curve or the



(a) starting point (b) starting point

Figure 7: Cylindrical shapes.
(a) 19 points and 2 geometric points.
(b) 21 points and 2 geometric points.



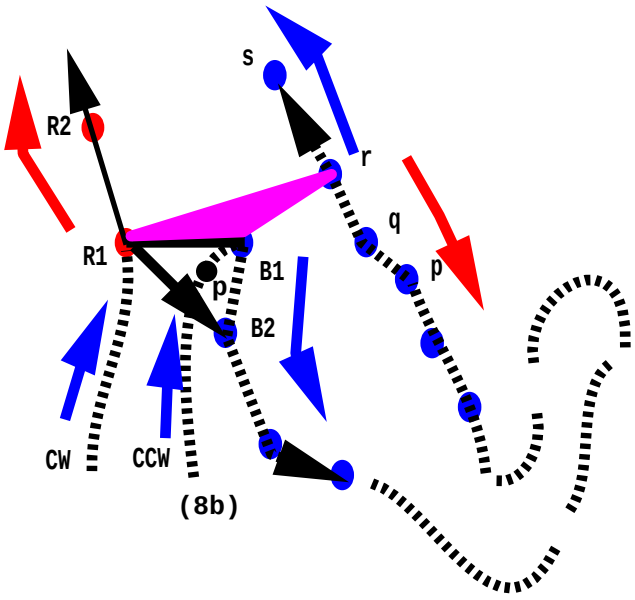
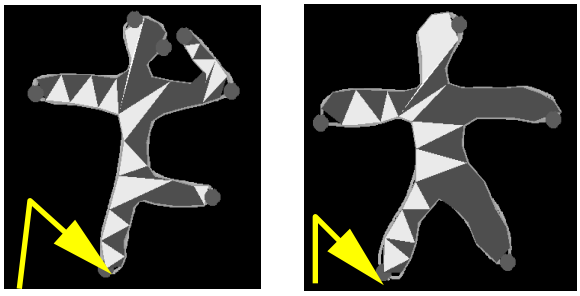
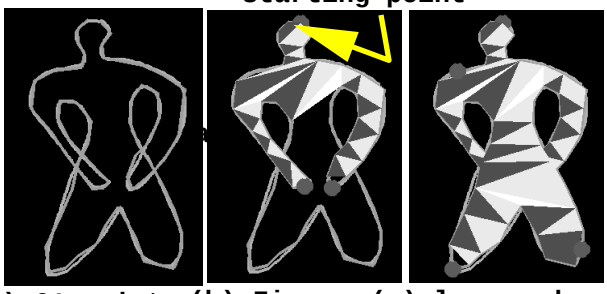


Figure 8: (a) Next vertex selection.
(b) Curve-Splitting



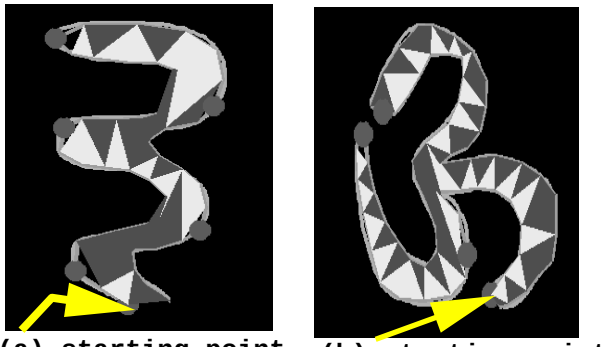
(a) starting point (b) starting point

Figure 9: (a) Spread out pose. 57 Points five geometric imprints.
(b) Dancing pose. 60 points, seven geometric imprints.



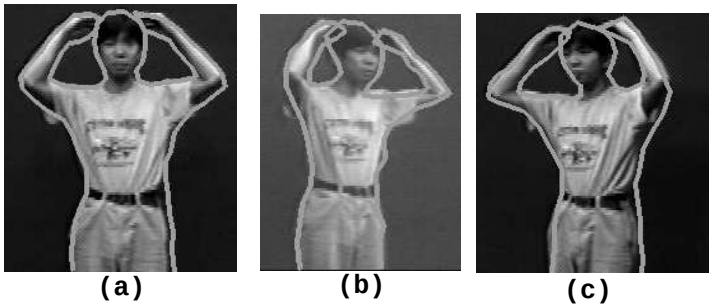
(a) 61 points (b) Finger tips (c) legs and shoulder

Figure 10: (a) Overlapping contours.
(b) Partial geometric-imprint of 3 points. (c) Remaining four geometric-imprint points



(a) starting point (b) starting point

Figure 11: Interesting Figures.
(a) Winding river, 45 points, Six geometric imprint points (b) Oohm sign, 67 points four geometric-imprint points.



(a) (b) (c)

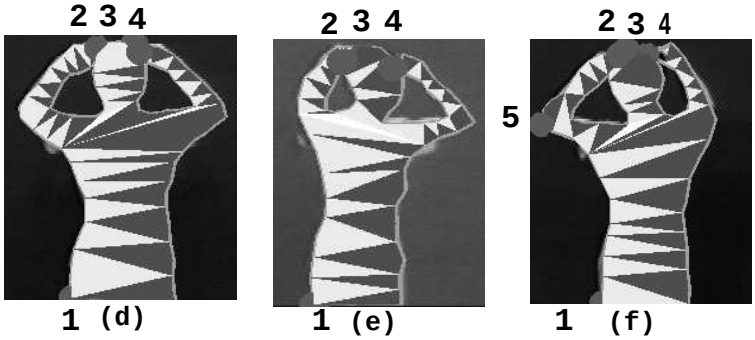


Figure 12: Geometric-imprint points for three camera image for the same pose. (a) 77 Points (b) 63 points (c) 68 points (d) four (e) Four (f) five geometric-imprint points.

CCW-curves so that they move together, thus when only one node remains it must be the extreme point for that cylindrical shape. The results of our implementation are shown in Figure 7a-b. A 2D-curve is drawn by the user, and multiple mouse-picks specify the points on this curve in clockwise order. Figure 7a shows the desired geometric-point of a cylinder with the yellow and red triangulation. In the following sections, we explain the geometric-imprint algorithm for an arbitrary curve specified by the user.

Arbitrary Curve, Dealing with Foldings, and Splitting Algorithm

The above algorithm works for simple curves, but will not generate a geometric-imprint for an arbitrary curve where folding of the curves occur. For this we will need the splitting algorithm. Figure 8 shows four points R1 and R2 on the CW-curve, and B1 and B2 on the CCW curve. For our discussion below, we will assume that the length (line(R1,B2)) is less than the length(line(B1,R2). In this case, we are considering line(R1,B2) to become the next base line as the recursion continues. The other situation, when length (line(R1,B2)) is greater than or equal to length(line(B1,R2) is dealt with in a similar manner in our implementation. The sign of the area of a triangle(A,B,C) is the sign of the cross product of AB with AC. There are two cases which we consider:

Case a: when R2 and B2 are moving along the same direction, the sign of the area is same, and so we simply move the base. So triangle(R1,B2,B1) is colored, and the new base is line(R1,B2). See Figure 8a.

Case b: when folding has occurred as shown in Figure 8b at point B1. In this case, folding has occurred for the CCW curve. We look for the first place where the area is positive again as the CCW winds around and comes at point r the area again has become positive. This condition is recognized and Tri(R1,B1,r) is colored as shown in Figure 8(b). The original curve is split into two curves. First curve with base line(R1,r), CW-1: R1$\rightarrow$R2$\rightarrow$... and CCW-1: r$\rightarrow$s$\rightarrow$ .... The second curve with the base line(r,B1), CCW-2: r$\rightarrow$q$\rightarrow$p$\rightarrow$ ... and CW-2: B1$\rightarrow$B2$\rightarrow$ ... In step 2, when there is a split we recurse for both the curves. Both curves are highlighted in Figure 8b.

In our implementation, folding is recognized by studying the area of the triangles and their sign changes. For example, folding is recognized as the sign of the area of the triangle(R1,B1,B2) is opposite to that of triangle(R1,B1,p). So we look for the triangle(R1,B1,r) which has the sign same as triangle(R1,B1,p). We note that point B1 is also a candidate for the geometric-imprint of the curve. When folding occurs we split the curve into two pieces and apply the above algorithm recursively to the two split curves.

As the recursion continues it is easy to see that the above implementation would terminate as the size of the curve reduces with each iteration. The number of splits depends upon the number of foldings in step 2 of the above algorithm. Let the number of splits be k. Since all the remaining nodes could be checked for finding vertex r in step 2, so step 2 in the worst case in O(N). So we have an upper bound on this algorithm of O(kN). We expect k to be small, unless the cameras are focussed on the hair of the participant. Note that the lower bound of the algorithm is O(N) where N is the number of specified points on the arbitrary curve. Any algorithm must look at each one of the points at least once.

Results

In our implementation, we draw an arbitrary curve on a window using the mouse, and specify an arbitrary number of points on the curve by using mouse picks. S1 is the starting point and S2 is the end point chosen on this curve. It is expected that the user will specify the points in a clock wise direction. The algorithm works on these selected set of points starting from two points S1 and S2. S1 is the starting point for the clock wise (CW) curve, and S2 is the starting point of the counter clock-wise (CCW) curve. The progress of implementation is shown by drawing the triangles in yellow color when the two points of the triangle are on the CW curve. The color is red when the two points of the triangle are on the CCW. Since S1 and S2 are considered topologically next to each other, only one of them, S1, is considered as a geometric-imprint point.

Results of our implementation are shown in Figures 9-12. Figure 9a shows simple human silhouettes similar to Figure 2a. We have the geometric-imprint identified. The algorithm will also work on a closed curve in Figure 10 for the case when the hands may overlap the body. Using the starting significant point at the head we have been able to identify six significant points in Figures 10b-c for the curve in Figure 10a. The implementation of this algorithm is dependent upon the selection of the starting point, as well as the user-specified points. Notice for example, that only one shoulder point in Figure 10c has been identified as the geometric-point. The implementation *correctly* identifies the other five cylindrical end-points on this complex curve in Figures 10b and 10c. Figure 11 show the results of our implementation on extreme folding cases. In these cases, the algorithm picks many points along the highly modulating curves to indicate extreme folding which is occurring in both the winding river example and the *Oohm* sign in Figure 11. In Figure 12, we have used the same pose and same staring point for the images captured from three cameras. Notice that all the significant points, marked one to five in Figure 12, are common in the three figures. One extra elbow-point is also a geometric-imprint point in Figure 12f.

In our implementation we mainly show the tips of cylindrical shapes, however, the places where folding occurs can also be geometric-imprint point as explained earlier. As shown in Figure 12, similar topological curves generate a similar geometric imprint set. Once the geometric imprint has been obtained it is much easier to find correspondence. Since there are only k geometric-imprint points out of N, it is much easier to match the geometric-imprints on these curves. In our future implementations, we plan to also use the matching information from *previous* frames to help us match the geometric-imprint of *present* camera-images.

Ratio-Theory and Correspondence

The idea behind the *ratio theory* can be best understood by imagining a tailor measurement. The extremities of

a contour can be related easily if we start with the dimensions of the participant as shown in Figure 2a and 2b. From one frame to another, the length of the body-extremities of the participant can never change. This idea has been implemented in our Virtual Kabuki system [12] from frame-to-frame. We now extend the same idea to correspond with the three images of the same pose. Let us assume, that the geometric-imprints of two poses have been obtained as shown in Figure 3. We now need to correspond point A with point 1, point B with point 2, and so on. Then the points are connected i.e. 1 to 2 to 3 and so on. In Figure 2, the curve length L1 between extremity A and B is similar to the curve length L2 between D and E, and so on. In particular, it can be said that the ratio L1/L2 would approximately equal L3/L4. So the correspondence problem reduces to matching and finding a best fit in a small search-space dependent upon the number of geometric-imprint points on curves from camera-images.

Conclusions and Future Research

In this paper, we have presented *the geometric-imprints* algorithm which finds the significant points based upon the cylindrical nature of projections of articulated figures, especially humans. We have presented an implementation of the geometric imprints algorithm which captures the extremities of a 2D-contour of an articulated figure. The 2D-curve can be arbitrary with multiple foldings. Results are shown and presented. The geometric-imprints algorithm is robust and finds similar geometric-imprints for 2D-contours from camera-images of the same pose. Thus, similar topological curves are expected to produce similar geometric imprints.

In future, we plan to automate the contour extraction process. In addition, we plan to develop matching algorithms which will solve the correspondence of geometric-imprints from different camera images for the same pose. This would allow us to identify a set of geometric imprint points. An imprint point triplet (S1,S2,S3) then would be used by the active-space tracking system as shown in Figure 1. This would allow us to join the correspondence and the active-space tracking components of the Scan&Track system towards unencumbering Virtual Environment.

Acknowledgments

We would like to thank Dr. R. Nakatsu, President of ATR Media Integration & Communications (MIC) Research Laboratories for making this research possible. Thanks are also due to Dr. K. Masse, Head of Department 2, for discussions on Pfinder method being used at his laboratory, and Dr. EJ Altman in Department 4 for stimulating discussions on a variety of topics. Special thanks are also due to the group members of Department 1 of the MIC laboratory: Mr. Fujimoto, A. Utsumi, M Yamada, K. Ebihara, T. Ohatsuka, S. Imura, H. Orainkyo, and Drs. T. Sakaguchi, I. Fermin, K. Sen Gupta for a variety of discussions and help.

References

[1] Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 1-384, October 14-16 1996,Killington, Vermont, USA, IEEE Computer Society Press (1996).

[2] Proceedings of the Second International Workshop on Automatic Face and Gesture Recognition, pp. 1-384, June 26-28, 1995, Zurich, Switzerland, IEEE Computer Society Press (1996).

[3] G.B. Newby *Gesture Recognition Based upon Statistical Similarity* Presence, 3(3), pp. 236-244 MIT Press (1994).

[4] M.W. Krueger, Artificial Reality II. Addison Wesley Publishing Company, Reading, MA. 1-277 (1991).

[5] D Geiger and TL Liu. *Recognizing Articulated Objects with Information Theoretic Methods*, Proceedings of International Conference on Automatic Face-and-gesture Recognition, Killington, Vermount, pp. 45-50, October 14-16 (1996).

[6] TK Capin, H Noser, D Thalmann, IS Pandzic, NM Thalmann. *Virtual Human Representation and Communication in VLNet*, IEEECG&A vol. 17, No. 2, pp. 42-53 (1997).

[7] SK Semwal, R Hightower, and S Stansfield. Closed form and Geometric Algorithms for Real-Time Control of an Avatar, Proceedings of IEEE VRAIS96, pp. 177-184 (1996).

[8] NI Badler, MJ Hollick, and JP Granieri. Real-Time Control of a Virtual Human using Minimal Sensors, *PRESENCE*, 2(1): 82-86, 1993.

[9] C Wren, A Azarbayejani, T Darrell, and A Pentland. *Pfinder: Real-Time Tracking of the Human Body*, Proceedings of International Conference on Automatic Face-and-gesture Recognition, Killington, Vermount, pp. 51-56, October 14-16 (1996).

[10] J Segen and SG Pingali. *A Camera-Based System for Tracking People in Real Time*, Proceedings of ICPR96, IEEE Computer Society, pp. 63-68, 1996.

[11] RF Rashid. Towards a system for the Interpretation of Moving Light Displays, *IEEE Transaction on PAMI*, vol.2, no.6, pp.574-581 (1980).

[12] J Ohya and F Kishino. Human Posture Estimation from Multiple Images using Genetic Algorithms. *Proceedings of 12th ICPR*, pp. 750-753 (1994).

[13] TS Huang and VI Pavlovic. *Hand Gesture Modeling, Analysis, and Synthesis*, Proceedings of International Workshop on Automatic Face-and-gesture Recognition, Zurich, Switzerland, pp. 73-79, June 26-28 (1995).

[14] K Mase. *Recognition of Facial Expression from Optical Flow*, IEICE Transactions, Vol. E 74, No. 10, pp. 3474-3483, October (1991).

[15] S Moezzi, A Katkere, DY Kuramura, and R Jain. Immersive Video, Proceedings *IEEE VRAIS 96*, IEEE Computer Society Press, Los Alamitos, pp. 17-24, Santa Clara, CA, (1996).

[16] M Kass, A Witkin, and D Terzopoulos. *Snakes: Active Contour Models*, International Journal of Computer Vision, 1, 4, 1987, 321-331.

[17] SK Semwal and J Ohya. *Scan&Track: An Active Space Indexing System for Unencumbering 3D Tracking in Virtual Environments*, Technical Report, ATR-MIC Laboratory, Hikari-dai, Seika-cho, Kyoto, 619-02, Japan.

[18] G Turk. *Re-tiling Polygonal Surfaces*, Proceedings of SIGGRAPH, vol. 26, No. 2, pp. 55-64, July (1992).

[19] Y Shirai. *Three-Dimensional Computer Vision*, Springer-Verlag, pp. 194-201, 1987.