〔公開〕

ＴＲ－Ｍ－００２４

Scan & Track : An Active Space Indexing System for Unencumbered 3D Tracking in Virtual Environments

スダンシュ　クマール　セムワル　　大谷　淳
Sudhanshu Kumar Semwal　　Jun OHYA

１９９７．８．２２

ＡＴＲ知能映像通信研究所

Scan&Track: An Active Space Indexing System for Unencumbered 3D Tracking in Virtual Environments

Sudhanshu K Semwal[1] Jun Ohya

ATR Media Integration & Communications Research Laboratories,
ATR International, Kyoto, Japan 619-02
semwal@redcloud.uccs.edu|ohya@mic.atr.co.jp
[1]Invited reseracher from University of Colorado, Colorado Springs

ABSTRACT

We present a new method for unencumbered tracking of participants in a virtual environment using multiple cameras. During preprocessing, the system uses three cameras to record a planar-slice containing a simple pattern arranged on a regular grid. The planar slice is then physically moved at regular intervals and the corresponding camera-images are stored. In this manner, *active-space* or the 3D-space between these parallel slices is scanned. In this manner, we scan the active-space. The stored camera images are processed to create an interactive, active-space indexing mechanism which maps 3D points in active-space to the corresponding projections on the camera-images. The active-space indexing mechanism allows an operator to specify poses by specifying significant points on the camera-images of a participant. These significant points could also be automatically generated by analyzing the camera-images.

In this paper, we first discuss the Scan&Track system. Later, details of the active-space indexing method and results are presented. Our method avoids the complicated camera calibration operations, and is robust as the distortions due to camera projection are automatically avoided. In addition, the system is scalable, as active-indexing for the same 3D-space could be developed for both the low and the high-end systems. In addition, the active-space is also scalable. These qualities make the Scan&Track system ideal for future VR applications.

Key Words: 3D Motion Tracking, Multiple Cameras, Unencumbered VEs.

INTRODUCTION

Virtual environments pose severe restrictions on tracking algorithms due to the foremost requirement of real-time interaction. There have been several attempts to track human participants in a virtual environment. These can be broadly divided into two main categories: encumbering and non-encumbering [1, 2, 3, 4]. This is perhaps the most important choice in designing a virtual environment as it determines the style and quality of interaction of a participant. Encumbering virtual environments are based upon *when you come we will provide something for you to wear* philosophy. A variety of devices have been developed, for example acoustic [5], optical [6, 7], mechanical [8, 9, 10, 11], bio-controlled [12], and magnetic trackers [13, 14, 15, 16, 17, 18]. On the other hand, un-encumbering technology is based upon *come as you are and be yourself* philosophy [19]. Most of the camera-image based systems belong to this category. The degree of encumbrance also matters: outside-looking-in systems for optical tracking is less encumbering than the inside-looking-out systems because the user is required to wear lighter equipment on the head [1, 6]. Trackers have been compared and surveyed on the basis of resolution, accuracy, responsiveness, robustness, registration and sociability in [1].

In comparison to other position trackers, magnetic trackers have relatively low data rates, as the filtering required for the distortions in the emitted field, introduces lag. Mostly magnetic trackers have been used [20] in virtual environments as they are robust, relatively inexpensive, and have a reasonable working volume or active-space. Magnetic sensors allow multiple sensors to share the transmitters. In addition, multiple transmitters can share the same active-space. For example, in the 3Space FASTRAK system available from Polhemus [18], a transmitter shares four sensors, and there are upto four such transmitters. However, magnetic trackers can exhibit unacceptable tracking errors of upto 10 cms, if not calibrated properly [20]. This is especially true when we consider that invariably there are magnetic objects present in our surroundings. Magnetic trackers allow larger active-space in com-

parison to the mechanical trackers, but the person is restricted to remain within the range of transmitters.

Optical and camera-image based tracking has problems of occlusion [1]. In addition, optical trackers are encumbering as either four dedicated cameras are worn by the user on top of their HMD for the inside-out system, or a set of receiving beacons are placed on the user heads for the outside-in systems [1, 20]. In addition, optical tracking requires numerical optimization for camera calibrations [6, 7] with a large number of beacons placed on the wall. In the case of inside-out tracking, rows of LEDs (beacons) can be, in principle infinitely expanded to allow a large active-space where the participant can be tracked. Inside-out systems are suitable for multiple participants also, as each participant can have his/her own camera to see the LED-beacons. But these are highly encumbering, and suffer from occlusion.

The well known correspondence problem is ever-present when purely vision-based systems are used. As mentioned in [20], there are several compromises made to resolve the issue of correspondence and recognition. Even then, it is difficult to find a topological correspondence between the multiple contours obtained for the images of multiple cameras viewing the same scene. In the field of computer vision and photogrammetry, there is a wealth of research on motion tracking [21, 22, 23, 24].

In human-centered applications [25], a high-level of interaction is expected as the goal is to create virtual environments so that *humans live, science finds out, technology conforms* [25]. In addition, virtual environments are expected to be populated by both avatars or human forms which replicate the movement of participants, and virtual synthetic actors whose autonomous movement is directed by computers [13, 14, 15, 26, 27]. As human-centered applications [25] grow, there would be a need for tracking based on scalable technology. For the low-end systems, a lower resolution tracking might be satisfactory. For the high end tracking, higher resolutions might be desirable. The drawbacks of video-based systems have been described in [1] and [20]. In particular, the resolution and accuracy of vision and camera based systems is dependent upon the size of the pixel. It is difficult to differentiate between any two points if the size of the pixel is large and the projection of both points falls on the same pixel.

Tracking used in an application is directly related to the need and tracking methods available. For example, a doctor would prefer trackers to be more precise and accurate. Large working volumes, whether encumbering or not, are at best secondary choices for them. For human-centered applications, the choice of encumbering versus non-encumbering is critical as a large population is still hesitant about the technology, and therefore it is much more convenient if participants are not tethered by wires or gadgets. Even people comfortable with technology may not be happy with equipment which restricts their motion or makes them tired. Our motivation is to develop unencumbering virtual environments using multiple cameras.

RELATED WORK

Camera-based techniques are well suited for developing unencumbering applications. Inexpensive video-cameras are now readily available. This area is well studied, and Krueger's work [19, 28, 29] is well known for virtual environment interaction using 2D contours or silhouette of human participants, and interpreting them in a variety of ways. When 2D camera-images are used, there is no *depth* information available with the image. Much research has been done in the area of computer vision [21, 22, 23, 24], stereo-vision [21], object-recognition [22], and motion analysis [23, 30] to recover the depth information. Recently, there has been a growing interest in using successful 2D and 3D data structures for recovering the depth information.

Jain et. al. [31] combine the vision and graphics techniques for processing the video-streams offline, non-interactively, so that multiple participants can view the processed scene from different angles after preprocessing has been completed. In immersive video, the color-based discrimination is used between frames to identify pixels of interest, which are then projected to find the voxels in a 3D-grid space based on the color of the region (pixel) of interest. The marching cubes algorithm is then used to construct the iso-surface of interest. This method does not track the participants interactively in a virtual environment. Instead it processes the video-sequences from multiple cameras, and allows participants to view the processed data.

The Virtual Kabuki system [32, 33, 34, 35] uses thermal images from one camera and tracks the participant's planar motions in a 3D environment. Non-planar 3D motions are not tracked. Estimate of the joint positions are obtained in real-time by using the silhouettes obtained from thermal images and 2D-distance transformations. Other points such as the knee and the elbow of the hand are estimated using genetic algorithms [32]. Once extracted the motion is mapped on to a kabuki-actor [33]. The system is robust and tracks participants in real time. There are occasional problems

due to the lower thermal conductivity of clothes [32].

Multiple cameras have also been used effectively for security based systems. For example, visual surveillance, not tracking in VE, is the goal of the 3DIS (three dimensional intelligent space) system [36]. As explained in [37], there are several limitations to the 3DIS system. In particular, the 3DIS system can not estimate the orientation of the user's position, and therefore can not track the participant at all. In addition, there are no provisions for the system to detect any change in light intensity which can affect the camera-calibration.

Blob models [38] are used in the Pfinder system developed at the MIT Media Lab. The system uses one camera and works with one participant in the environment using a static scene, such as an office. A multiblob [38] model is created, for the person based upon the color information, for estimating the 2D contours and images. The estimation is based mainly on the color-changes of the pixels and classification based upon the color informations [38]. Interestingly, depth perception is limited to 2D, for example when the participant jumps it is considered a move backwards. So essentially, the system estimates only the 2D information.

Most of the camera-based VEs base their judgements upon the color information and estimate the 3D position of the participant by using the information available from multiple cameras. The process involves solving the contouring problem, correspondence problem, and significant point extraction [20]. In the following sections, we present our Scan&Track system. This is followed by implementation details of our new method of 3D tracking, called the *Active-Space-Indexing* method. This method and associated results are the primary focus of this paper.

THE SCAN&TRACK SYSTEM

We are developing an unencumbering VE, called the Scan&Track system, using the video image sequences from multiple cameras. This is an outside looking-in system [1]. 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. The correspondence system, which is *not* the focus of this paper, contains four subsystems: (i) contour extraction, (ii) significant points extraction, (iii) significant points correspondence and matching, and, (iv) scene, color, and previous poses database. We already have some experience in dealing with all the four issues [32, 34], especially for monoculor thermal images using one camera. We have also implemented a new significant point tracking algorithm explained elsewhere [39]. For this paper, we will assume that the correspondence system would be able to extract contours C1, C2, and C3, from three images Im1, Im2, and Im3, respectively. These contours would be used for estimating the 2D-location of significant points, in the images from the three cameras. Let S1, S2, and S3 be the projection of a 3D point S in all the three camera images respectively. We call the triplet (S1, S2, S3) as an imprint-set for point S. If a 3D point is visible from multiple cameras, then the location of the imprint of the 3D points in multiple camera-images can be used to estimate the 3D position of S.

The outcome of the matching algorithm is to provide an imprint-set, a triplet (S1,S2,S3), for every significant point S extracted from the images. The tracking method developed in this paper assumes that the triplets will be given to us by the correspondence sub-system. For this paper, from now on, we will assume that these points have been provided to the active-space tracking subsystem by the user, as shown in Figure 1. The active-space indexing mechanism uses the given triplet to estimate the position of point S. In the following sections, we explain the implementation details of the active-indexing mechanism for the Scan&Track system.

Camera calibration, Space-linearity and over-Constrained Systems

There have been many attempts to estimate the 3D motions of the participants with minimal number of points used for camera-calibration, for example, three points are sufficient as suggested by Ullman [7]. Recently, there have been many studies where five or more points are used for stereo-matching and creating novel views [28, 29]. Most of the systems which use a minimal number of points are usually under-constrained, in the sense that any error in camera-calibration may result in severe registration and accuracy errors [20]. In the Scan&Track system, we have gone the other way. We plan to use several points, during preprocessing, and create our system based on that, but then we do not require these points to be present for calibration or reference during the actual tracking.

Our motivation to use several points is to subdivide the active-space, so that only a few points are used locally to estimate the position during tracking, similar to piece-wise design for surfaces and contours [40]. In systems using a minimal number of points, a global calibration is used, and therefore it creates an under-constrained system [20]. The non-linearity and tracking errors are more profound due to the use of an under-constrained sys-

tem.

In our approach, by subdividing the 3D active-space into small, disjoint, voxels or 3D-cells, we make only a small number of points, which are the vertices of the 3D-voxel, be responsible for estimating a 3D position inside that 3D-voxel. In this way, only a few points determine the estimate during tracking. The major advantage of using a set of voxels is that the non-linearity due to camera calibration is minimal. In particular, we can assume that a linear motion inside the small, 3D-voxel space will also project linearly on a camera-image. Thus, effects of camera-distortion would be minimal. This assumption also allows the use of linear interpolation for estimating the exact position of a point during tracking.

Estimating depth using multiple cameras and slices

Consider the projections of two points in Figure 2. Depending upon the viewpoint and multiple cameras facing the two points P and Q, the projections of these two points change, particularly their relationship changes as we move from left to right, from image plane A to image plane B via planes L, C, and R in Figure 2. The spatial information between the two points is lost. Note that during tracking, we wish to actually recover the spatial information, which may change dynamically. If only these multiple images are available for recovering the information, as shown in Figure 2, then it is a difficult correspondence problem: how can we infer that points Q_A and Q_B are actually the same point Q in 3D space?

Next, we shall consider two points P and Q on a planar-slice and the same set of multiple cameras as shown in Figure 3. Note that the spatial relationship of projections of points P and Q, in all the planes in the same hemi-sphere w.r.t. the planer slice, remains same. Thus, it is much easier to deal with points in a planer-slice. In addition, as shown in Figure 4, zoom-in or zoom-out also does not change the relationship. In particular, w.r.t. to two lines L1 and L2 on a plane, arbitrary zooming of the camera or changing the orientation of the cameras in the same hemisphere, does not have an effect on this relationship. In particular, point Q remains to the right of projected-line PL1, and point P to the left in both the projected-images in the same hemisphere. Both points are above projected-line PL2 as expected. We can now extend the idea to many lines. A grid of lines partitions the slice into sixteen cells in Figure 5. Figure 5 also shows a camera image which is placed above the slice at an angle pointing downwards. There is a perspective deformation of the lines, yet the relationship of point P and Q is consistent with the planer slice, and the 2D cell-index of both points is same.

In Figure 3, camera images A and B are actually facing the point P from very different, and almost opposing angles, but both images are in the same hemisphere w.r.t. the slice-point P, so this discrimination is still possible. Since we plan to use several cameras, this gives us an important clue to place cameras; that cameras used for correspondence should be placed in the same hemisphere w.r.t. all the slices in the active-space.

Slices can be stacked for estimating the depth information. For simplicity, each slice, in Figure 6, has the 4 by 4 partitioning. Depending upon the complexity of scene, other partitioning methods are possible. We can recover the depth information by processing all the slices. Let (S1,S2,S3) be the 2D projection of a 3D point S on three image-planes as shown in Figure 7. Each of S1, S2, and S3 is the 2D pixel coordinate in the image of the three cameras. During preprocessing in our implementation, all three cameras are placed in such a way that they are in the same hemisphere w.r.t. the active-space slices. These active-space slices occupy space called an *active-space volume* where the participant can be tracked in the Scan&Track system.

Preprocessing to create Active-Space Indexing

We preprocess the slices of data available to us. One camera is placed such that the view-normal is perpendicular to the white-board used for our experiments. The other two cameras are to the left and right of the first camera as shown in Figure 7. The pattern we have chosen is a 12 by 12 grid pattern, where grid intersections are highlighted by small black circles. This is shown in Figure 8. Grid-pattern occupies a 55cm by 55cm space on a white-board. Each square of the grid is 5cm by 5cm. The white-board is physically moved and recorded by three cameras st the same time. Eight such recordings result in eight slices for every camera. The inter-slice spacing is 10 cm. Slice number 4, as viewed by the three cameras, is shown in Figure 8. The white-board and the grid-pattern on it are visible from all the three cameras. The active-space volume is a cube of 55cm by 55cm by 75cm. The active-space is large enough to track the face and upperbody of the participants for the two sets of experiments we have conducted so far. A larger active-space can be easily constructed by using a larger panel instead of the smaller white-board used for experiments.

Figure 8 shows slice number 4 used in our experiments. A slice contains images from all the three cameras. We have only shown the camera-

images in Figures 8-10 to save space. For creating active-space indexing mechanism, we specify a set of horizontal and vertical lines to cover the rows and columns of the grid-pattern as shown in Figure 9. During preprocessing, we do the following for every slice:

(1) For all the three camera-images, four corner points on the grid-pattern are picked using a mouse. These corner points define a 2D-extent of the projected grid-pattern.

(2) Again, by using a mouse, we pick the end-points of grid-lines on the pattern. These lines approximately correspond to the rows and columns of the grid pattern on the white board.

A red line is drawn on the image of the slice, or slice-image, to give feedback as the points are picked on the slice-image. First horizontal lines, then vertical lines are specified by picking the end-points. This process is shown in Figure 9. For every camera-image, a total of 12 horizontal lines and 12 vertical lines are specified. This corresponds to a 12 by 12 grid-pattern. So a total of 72 lines are specified for the three images as there are three camera-images in a slice-image. Figure 9 also show these red-lines after steps 1 and 2 as the user moves on the slice, picking points as specified above. To save space, only the portion form camera-images are shown in this figure also.

We show the intersections of horizontal lines and vertical lines by red and blue circles in Figure 10. Notice the expected similarity between Figures 8 and 10. The circles are drawn on top of camera-images and practically cover the grid-pattern. End points of a line specified in step 2 are by default intersection points on the grid. A blue circle is also an intersection and indicates the situation when an endpoint of a line does not fall on the edge of the 2D extent. Mouse-picks can be slightly off the mark. In practice, as can be seen by Figures 8 and 10, the lines cover the grid points well, and so using horizontal and vertical lines to find the location of grid-circles works well for our implementation.

We observe that a straight line's projection on a planar plane remains a straight line when projected on another planar plane. We have used this observation during our preprocessing to identify the grid-pattern by line-intersections. The camera images and lenses have some deformations, however, the deformations are negligible as can be seen in Figure 10. To process eight slices during preprocessing, it took approximately two hours. Note that the preprocessing is only performed once for a set of slices. In future, we plan to automate this process as grid-pattern can be recovered by simple image processing techniques also.

Finding a 2D-index during Tracking

For every slice we find a set of horizontal (H-lines set) and vertical lines (L-lines set) during preprocessing. During tracking, given the pixel coordinate of a 2D point on a slice, we can quickly find the grid-index using the horizontal and vertical lines. First, we check if the point is outside the area defined by four corner points of the 2D extent specified during step 1 of the preprocessing. If it is inside then we find the grid-index for the given pixel by searching the set of vertical lines for the x-index, and horizontal lines for the y-index. Since the lines are specified from left to right, we find two *consecutive* vertical lines p and p+1 such that the given point is on or to the right of line p and is on the left of line p+1. The x-index is then p. Similar algorithm is used to determine the y-index, q, by finding two *consecutive* horizontal lines such that the point is on or above line q, and below line q+1. Since we only use twelve horizontal and vertical lines each per slice, the grid index in our case would be between zero to twelve in both the horizontal and vertical directions. Since there are only fixed number of lines in our implementation, this operation is a constant time O(1) operation.

To estimate the location of a 3D point given its imprint-set, we have implemented the following algorithm:

Using Active-Space Indexing for 3D-Voxel Estimation

The active-space indexing mechanism finds the 3D location based on the imprint set triplet. The triplet in our implementation is provided by the user by using mouse-picks on the respective camera-images. For example, if tip of the nose is visible in all the three camera images for our experiment, then the operator can click on the tip of the nose in all the three images to obtain a triplet (S1,S2,S3) for the nose.

Given a triplet (S1,S2,S3) corresponding to a 3D-point, we perform the following for *every* slice:

(1) For an imprint-point, use the vertical lines, collected for the corresponding camera-image during preprocessing, to find horizontal index (x-index). This can be determined by checking all the vertical lines as described in the previous section.

(2) Perform the same with the set of horizontal lines for the camera-image. This time we test for a point to be above or below a set of horizontal lines as explained in the previous section.

(3)Perform the above for each of the triplet specified for every camera-image on the slice. For left camera-image, let the 2D index be denoted by I1 with x and y indices to be $I1_x$ and $I1_y$, respec-

tively. Similarly the indices for the middle and right camera-image would be denoted by I2 and I3, respectively.

For every triplet (S1,S2,S3), we collect I1, I2, and I3 points for every slice as shown in Figure 11. Figure 11 shows the basic concept which is used to find the 3D-voxel. We assume that three cameras are being used, and we want to determine the 3D location of point P. We have shown three consecutive slices k, k+1 and k+2 in Figure 11. The Active Space Indexing Method identifies grid-indices to be I1, I2, I3. It is possible that I1,I2, and I3 are within range for more than one slices as shown in Figure 11.

For discussion, we assume that point P is between slice k and k+1 as shown in Figure 11. The three grid indices I1, I2, and I3 are also shown for the three slices in Figure 11. These three indices define a triangles on every slice. Simple ray-optics suggests that the area would be decreasing as the rays converge at point P and then start increasing as the rays diverge. Since we have only eight slices in our experiments, the simple linear search algorithm to determine the slice with the minimum area is constant, O(1) time. A binary search on the slices would be more appropriate if large number of slices are present to find the two slices k and k+1 where the area of the slices decreases. Let k be the left slice with the area A_L, and k+1 the right slice, with area A_R as the rays diverge. We determine that slice k is the nearest slice to the point P of interest, so the point's 3D cell index would be (i, j, k). Here i is the average of x-indices of I1, I2, and I3 for slice k. Similarly j is the average of y-indices of I1, I2, and I3 for slice k. Since the 3D cell-index of a point in active-space can be determined using this method, we call this method the *Active-Space Indexing Method.*

We have shown all the active-space points on all the eight slices for the three cameras together in Figure 12. They represent the 3D-grid active-space defnied by eight-slices and the grid-pattern on these slices.

RESULTS AND DISCUSSIONS

We use the slice with three camera images of the participant, and manually provide the triplet S1, S2, and S3 in our implementation as shown in Figure 13a-c. We start with the image of the participant, then specify a series of triplets. Each triplet is of the form (S1,S2,S3) and denoted as the image-imprint of the 3D point, we then use the active indexing method described above and determine the 3D voxel for a given triplet. We can specify as many triplets as desired. The user specifies these triplets in such a way that they correspond to the participant's features, e.g. tip of the nose, arm-pits etc.

Figure 13d-e shows the result of our implementation using six imprint-points. Corresponding 3D voxels are also shown by visualizing them as a connected skeleton figure. We have highlighted the corresponding points in the image in Figure 13c and 13d. Another, slightly different set of points is shown in Figure 13e. Figure 13f shows that it is possible to mimic the participant's pose by using a in-house synthetic actor developed in our laboratory.

We have also tested our active-space indexing method by using two different camera settings, thus creating a different set of slices and indexing mechanism, and have obtained similar results.

The technique works because there is enough shift in the projection of the points due to the camera positioning that the depth discrimination is possible. Unless the three camera angles are identical or the grid is very wide, it is highly unlikely that more than two slices have the same area formed by I1, I2, and I3. In fairness to our method, this would mean that the resolution of the sliced active space needs to be finer, or camera angles need to change. It can be concluded that, given three imprint-points, a unique 3D cell-index can be found.

Since there are only eight slices and twelve lines in the horizontal and the vertical direction, we have used a linear searching technique in our implementation. This works well and is a constant O(1) time operation. As the number of slices and number of lines increase in future applications, we can apply binary search on the slices, and a fast grid indexing on the lines by using binary search based on the area of the triangle on the slices. Finally, to further increase the performance, we can apply a suitable *regular* 2D-grid, where x and y interval are same, on camera-images of every slice. This grid-intervals would depend upon the minimal separation between lines. Since access into a regular grid, is constant time operation this method would be preferred when a large number of lines and slices have been used to thinly slice the active-space.

The 3D location of point P can also be further refined by using a variety of linear interpolation methods based upon the area A_L and A_R of the triangle as shown in Figure 11b. For example, we can select the middle point of the triangles on two slices, call them M_L and M_R, then the 3D location of the point P is equal to $(A_L{*}M_R + A_R{*}M_L)/(A_L + A_R)$. Notice that larger the area of the triangle, farther is the point P from the slice. In the event that the area of triangles, on two consecutive slices, is same, then the above method would give a point in the middle of the line joining the two points M_L and M_R. Other interpolations are also possible. For

example, since the lines of the area-triangles are on parallel slices, we can apply the law of similar triangles. As shown in Figure 11, let M_A and l be the middle point, and the length of the line between points I_1 and I_2 of area A_L, respectively. Similarly, let M_B and q be the middle point and the length of the line between points I_1 and I_2 for area A_R. Then the location of point P = $(l^*M_B + q^*M_A)/(l+q)$. Other methods of interpolations, for example, the line joining the two I_1 points on slices k and k+1, respectively, also passes through P. So there are several ways to confirm our estimate of the 3D-location when three imprint-points are given.

Analytic Evaluation of the Scan&Track System

As is true of any vision-based system, the discrimination capability of our system is limited by the physical-size of the pixel in the camera-images. However, the system allows the use of multiple cameras, and thus multiple indices and a more accurate prediction of depth. Once we have determined a 3D grid index value and an approximate location, then we could automatically use another set of cameras for a much closer and precise look at the 3D-space near the 3D-point. In other words, the active indexing method can be applied twice or more. The first time, it would be used to find a high-grained index, and later we would find a more precise index by using those cameras which have the most detailed view of the slice and nearby areas. The information about more detailed view would be recorded during preprocessing, as the camera arrangements do not change during active-tracking. We are suggesting the use of local and global cameras. Note that one camera could be local for one point, and could be listed as global for some other point in 3D space. Active indexing is extendible and amenable to hardware implementation; special purpose chips can be designed to calculate left/right of the line calculations for a large number of points. In addition, different slice-patterns could be available for low and high-end users for automatic preprocessing.

We must answer one obvious question: how would we deal with the situation when the camera is placed with such an acute angle that all the slices project as lines. In this situation, grid-lines may be barely visible for a very acute camera angle, and slice will project as a line when the camera view-normal is parallel to the slices. We note that it is a highly unlikely situation as cameras are *placed by us*, and we can ensure that none of the slices are parallel to the view-normal of any camera.

The second, related, question is: can we still track all of the active space with the restriction that cameras can not be placed at the horizon. The answer is that all the details of that region can be obtained by using cameras slightly away from the horizon, closely zooming on the area of interest. In addition, if for some reason the camera can not be moved, then the orientation of the slices can also be changed during preprocessing to create a new active-space indexing mechanism. It is this freedom to change the orientation of cameras and/or slices which makes the Scan&Track system versatile.

We now provide an analytical discussion of the Scan&Track system on the basis of six comparison criteria used in [1].

Resolution: pertains to how close points can be and still be determined to be distinct. The active indexing method can be used to identify the grid-indices extremely fast. In addition, linear-interpolation of the slice-areas and their locations can be used to further discriminate points. Since the Scan&Track system is expected to use linear interpolation to find the exact location of a point, it is precise and would have a high resolution.

Accuracy: The accuracy of the system is related to the correspondence of the points. In other words, the calculation is based on the active-space indexing mechanism which is fast and precise, especially when the linear interpolation methods would be used. Once the triplet (S1,S2,S3) is given it takes constant time as the number of points on the grid-pattern and number of slices are fixed. Since multiple cameras are being used we expect that the occlusion problem to be not as severe in comparison to when one camera is used. When the complete system has been implemented, we expect that the area, where greater accuracy is needed, will be zoomed in by several cameras, during both preprocessing and tracking, so that the active-space indexing mechanism would be as accurate as desired. Note that cameras remain in the same orientation for the duration of preprocessing and 3D-tracking in the Scan&Track system. In future, we plan to also automate preprocessing to create faster and precise scan of the 3D active-space.

Responsiveness: As mentioned, the active-indexing mechanism is extremely fast, constant time, and implementable in hardware for a fixed number of slices and grid-points. The responsiveness would also depend upon the time it takes to calculate imprint-points from the camera-images. This needs to be further investigated. It is believed that correspondence sub-system (Figure 1) will be the key to real-time (at least 30 frames per second) interaction and should be be implemented in hardware.

Robustness: The active-indexing system is extremely robust as the active-indexing mechanism can always find a 3D-cell for given imprint-points within active-space.

Registration: The registration is related to the *swimming* problem encountered in virtual environments. Virtual objects, when placed on a tracked-position, tend to swim because of the slight variations in estimating the 3D position of the tracked-position. This leads to the well known swimming effect which is very evident in systems using magnetic trackers. When we use multiple cameras, the same corresponding points on the image will produce the same result, because the camera and slices remain in the same position for the duration of tracking. We expect stationary participants to remain stationary. However, although rare, voltage fluctuations can account for a change in size of the image, and can create swimming effect. In addition, mechanical vibrations, for example, constructions in the virtual environments could cause the camera-images to jitter. In fairness, we do not know any tracking device where voltage fluctuations and mechanical vibrations would not have an effect on tracking.

Sociability: Our motivation to use video-based tracking was to create an unencumbered system. The Scan&Track system is an outside-in system as the cameras are expected to be mounted on the wall, and out of the way of the participants. For this reason, we expect the Scan&Track system to be a good choice for human-centered applications.

Planer slices can be arbitrarily large or small, and cameras can be zoomed in and out depending upon the user's wish. This makes the Scan&Track system useful for distributed VEs. Multiple user's can track the same 3D space based upon their own choice of slices and dot patterns. New cameras can be used to add more active-spaces as desired. The Scan&Track system can be used for personal active-space in front of the user's screen by placing three cameras at the top of the monitor. The preprocessing algorithm is simple, and can be easily automated.

CONCLUSIONS AND FUTURE RESEARCH

In the Scan&Track system, we have provided a framework for unencumbered tracking based upon multiple video sequences. We implemented a new algorithm for 3D-tracking called the *active-space indexing* method. Given the imprint-set of a point, the active-space indexing method determines the 3D cell index of that point in constant time. We also presented the results of our implementation. Linear interpolation could be used to estimate the exact position of a point when the imprint-set of a point is given. The simple preprocessing method for creating an active-index can also be easily automated using image processing techniques.

The process of automating the correspondence part of the system remains. Although we had similar experiences in this regard, and a new significant point algorithm has been already implemented, much work still needs to be done.

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 the pFinder method being used in his laboratory. Special thanks to Mr. Fujimoto for help with the Figure 13f. Thanks are also due to other group members of Department 1 of the MIC laboratory: Mr. 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] K. Meyer, H.L. Applewhite, and F.A. Biocca. A Survey of Position Trackers. *Presence*, 1(2), p. 173-200, MIT Press (Spring 1992).

[2] D. J. Sturman and D. Zeltzer. A Survey of Glove-based Input. *IEEE CG&A*, p. 30-39 (January 1994).

[3] T.H. Speeter. Transforming Human Hand Motion for Tele-manipulation. *PRESENCE* 1(1), p. 63-79 (Winter 1992).

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

[5] Various Pamphlets from CrystalEyes and Virtual Technologies, CA (1993).

[6] S Gottschalk and JF Hughes, Autocalibration for Virtual Environments Tracking Hardware, Proceedings of *SIGGRAPH 1993*, pp. 65-71.

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

[8] I.E. Sutherland, A head-mounted three dimensional display *ACM Joint Computer Conference* 33(1), p. 757-764 (1968).

[9] F.P. Brooks, M.J. Ouh-Young, J.J. Batter, and P.J. Kilpatrick, Project GROPE – Haptic Displays for Scientific Visualization. Proceedings of *SIGGRAPH 1990* 24(4), p. 177-185 (August 1990).

[10] G. Burdea, J. Zhuang, E. Roskos, D. Silver, and N. Langrana, A Portable Dextrous Master with Force Feedback. *Presence* 1(1), p. 18-28 (Winter 1992).

[11] H. Iwata, Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator, Proceedings of *SIGGRAPH 1990* 24(4), p. 165-170 (August 1990).

[12] Biosignal Processing and Biocontroller: Technology Overview, Video presentation by Biocontrol Systems, Inc., 430 Cowper Street, Palo Alto, CA 94301, USA.

[13] 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).

[14] S Stansfield, N Miner, D Shawver, and D Rogers. An Application of Shared Virtual Reality to Situational Training, *VRAIS 1995*, 156-161, 1995.

[15] S Stansfiled, D Shawver, D Rogers, and R. Hightower. Mission Visualization for Planning and Training, *IEEE CG&A*, 15(5):12-14, September 1995.

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

[17] JP Granieri, J Crabtree, and NI Badler. Production and Playback of Human Figure Motion for 3D Virtual Environments, *IEEE VRAIS 1995*, 127-135 1995.

[18] 3SPACE FASTRAK user's manual, Revision F, Polhemus, A Kaiser Aerospace and Electronics Company, PO Box 560, Colchester, Vermount, 05446.

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

[20] A State, G Hirota, DT Chen, WF Garrett, MA Livingston, Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, Proceedings of *SIGGRAPH 1996*, pp. 429-438.

[21] O Faugeras. *Three-Dimensional Computer Vision: A geometric Viewpoint.* MIT Press, Cambridge, Massachussets, 1996.

[22] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, Massachussets, 1990.

[23] Stephen Maybank. *Theory of Reconstruction from Image Motion*, Springer-Verlag, 1993.

[24] J Serra. *Image Analysis and Mathematical Morphology*, Academic Press, vol. 1 and 2, 1988.

[25] N Talbert. Toward Human-Centered Systems. *IEEECG&A*, 17(4):21-28, July/August 1997.

[26] C. Cruz-Neira, D.J. Sandlin, and T.A. DeFanti, Surround-Screen Projection-based Virtual Reality: The Design and Implementation of the CAVE. *Computer Graphics Annual Conference Series*, ACM SIGGRAPH, New York, p. 135-142, 1993.

[27] MR Macedonia, MJ Zyda, DR Pratt, DP Brutzman, and PT Barham. Exploiting Reality with Multicast Groups, *IEEE CG&A*, 15(5):38-47, September 1995.

[28] 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).

[29] 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).

[30] 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.

[31] 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).

[32] S Iwasawa, K Ebihara, J Ohya, S Morishima. Real-Time Estimation of Human Body Posture from Monocular Thermal Images, *International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, pp 15-20 (1997).

[33] Moving Beyond the Wires, New York Times, pp.D5, September 30, 1996.

[34] J Ohya, K Ebihara, J Kurumisawa, and R Nakatsu. *Virtual Kabuki Theater: Towards the Realization of Human Metamorphosis Systems, Proceedings of 5th IEEE International Workshop pn Robot and Human Communications*, pp. 416-421 (1996).

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

[36] Three Dimensional Intelligent Space, 3DIS video presentation, Colorado Springs, CO, USA.

[37] Sudhanshu Kumar Semwal, *A Proposal to Apply Virtual Reality for the Mobility Training of the Blind*, IEEE communications Conference, Ocho Rios, Jamaica, pp. 24-29, (August 1995).

[38] C Wren, A Azarbayejani, T Darrell, A Pentland. Pfinder: Real-Time Tracking of the Human Body, *International Conference on Automatic Face and Gesture Recognition, 1996*, pp. 51-56, 1996.

[39] SK Semwal and J Ohya. Geometric-Imprints: An Optimal significant Points Extraction Method for the Scan&Track Virtual Environment, Technical Report, ATR MIC lab. Seika-cho, Kyoto, Japan. Also manuscript submitted for review to the Face and Gesture Recognition Conference, Nara, Japan, April 1998.

[40] G Farin. *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego, USA, Third Edition (1992).

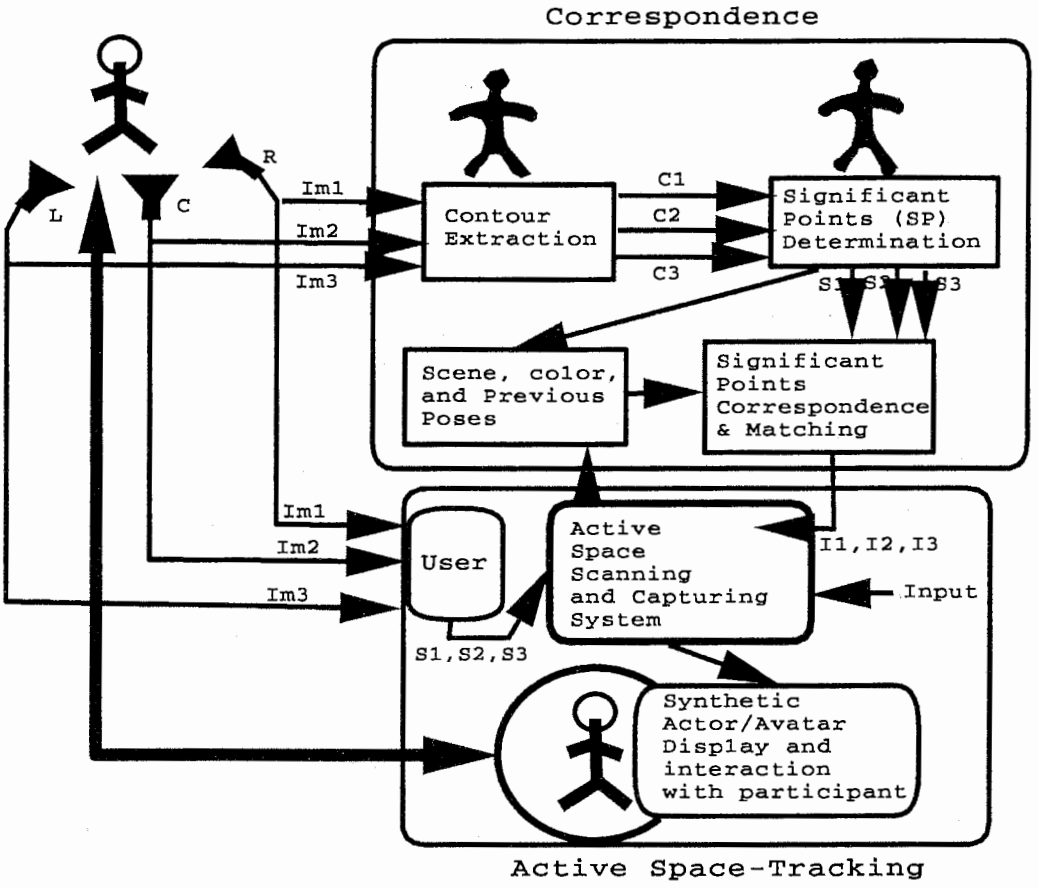


Figure 1: Block Diagram for the Scan&Track System

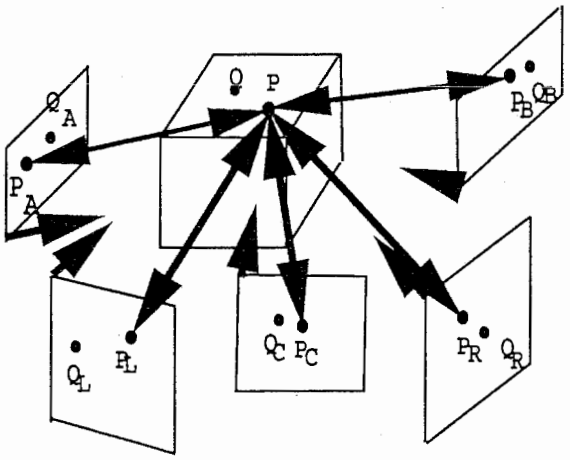


Figure 2: Relation ship of point P and Q changes depending upon the view as cameras move around it.

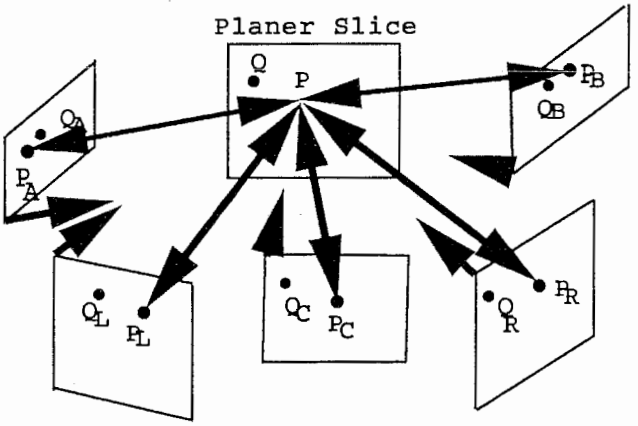


Figure 3: The relationship of the point on a slice remains same for a variety of planar views in the same hemisphere related to the slice.

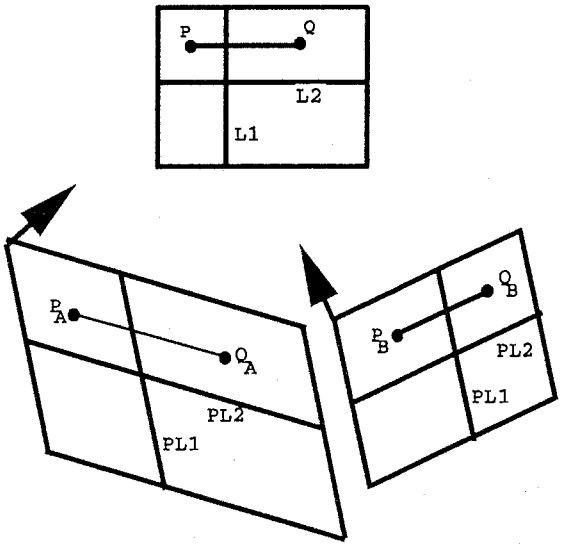


Figure 4: Projection of Two points on two planes

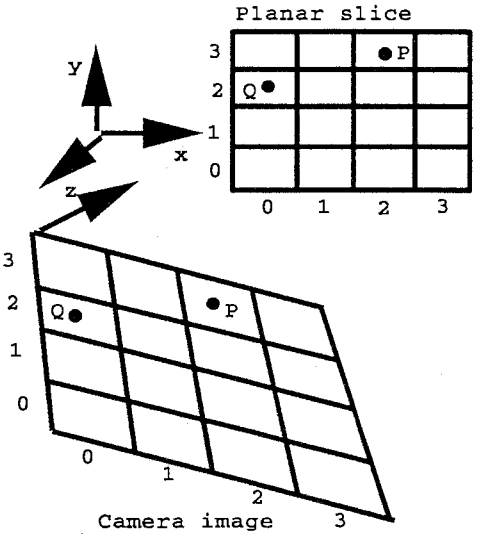


Figure 5: Camera image of two points P and Q, cell lines. Camera is looking downward and is tilted a bit from the top.

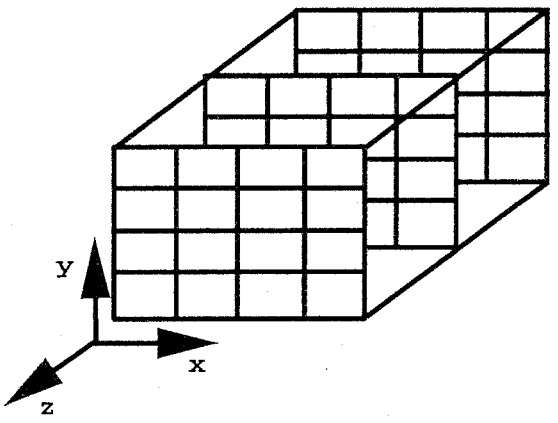
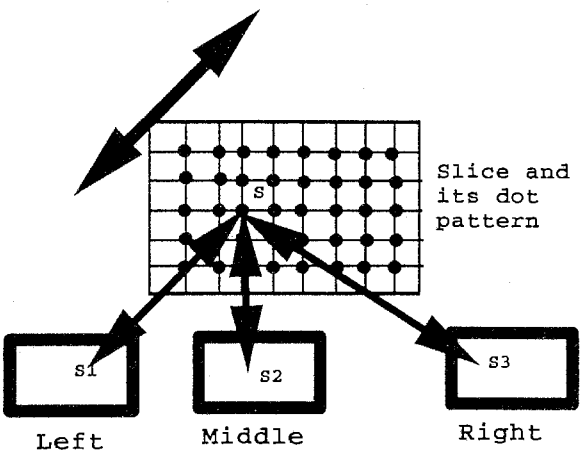
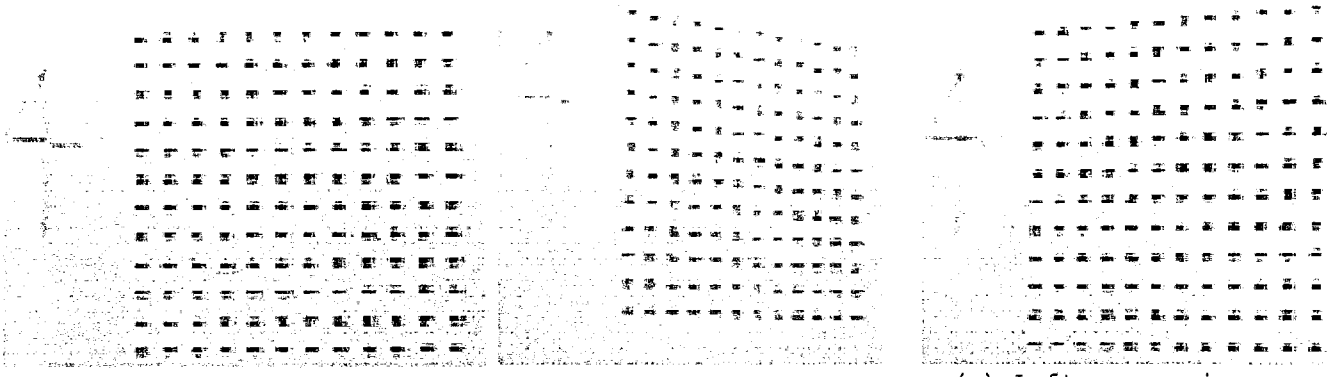


Figure 6: A set of 3 planer slice



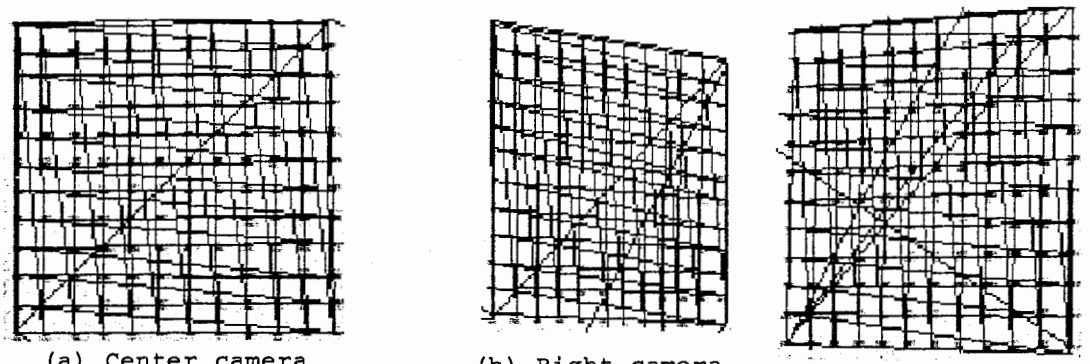
Active space creation

Figure 7: Imprint-set (S1,S2,S3) for point S. S1, S2, and S3 are 2D points on the respective camera-images.



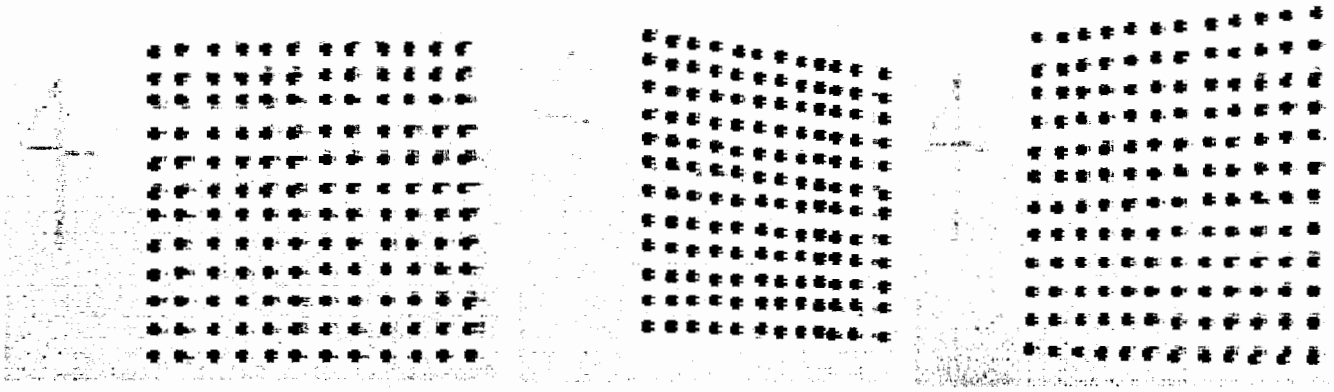
(a) Center camera-image (b) Right camera-image (c) Left camera-image

Figure 8: Camera-images from Slice 4. There are eight slices available



(a) Center camera (b) Right camera (c) Left camera

Figure 9: Red lines are the mouse-pattern created as the lines covering the grid-pattern are specified during Preprocessing.



(a) Center camera (b) Right camera (c) Left camera

Figure 10: Line intersection cover the grid-patterns well.

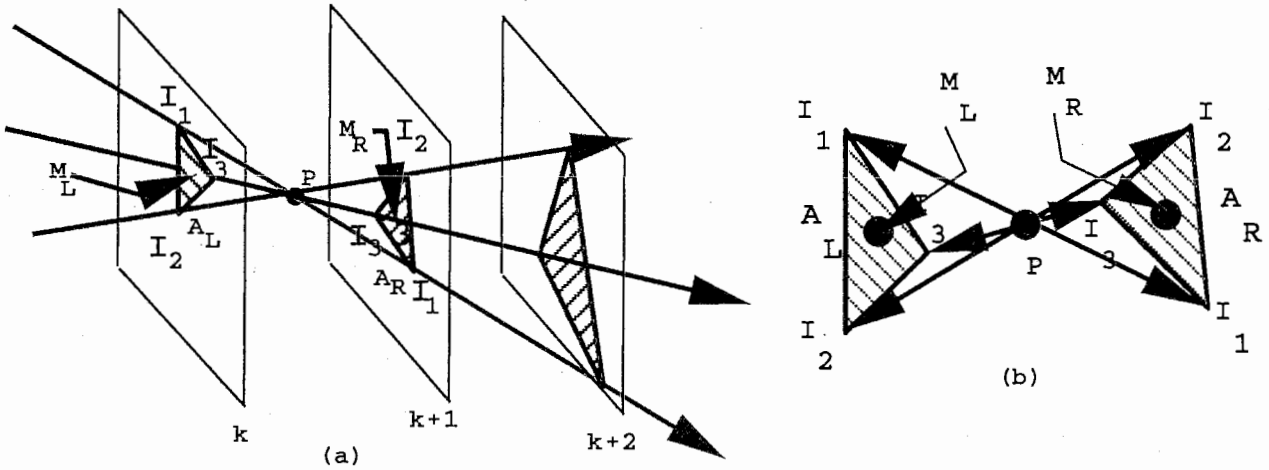


Figure 11. (a) Estimating the active-index of P. (b) Linear Interpoloation

(a) Center camera (b) Right camera (c) Left camera

Figure 12: Active-space points for all the three images

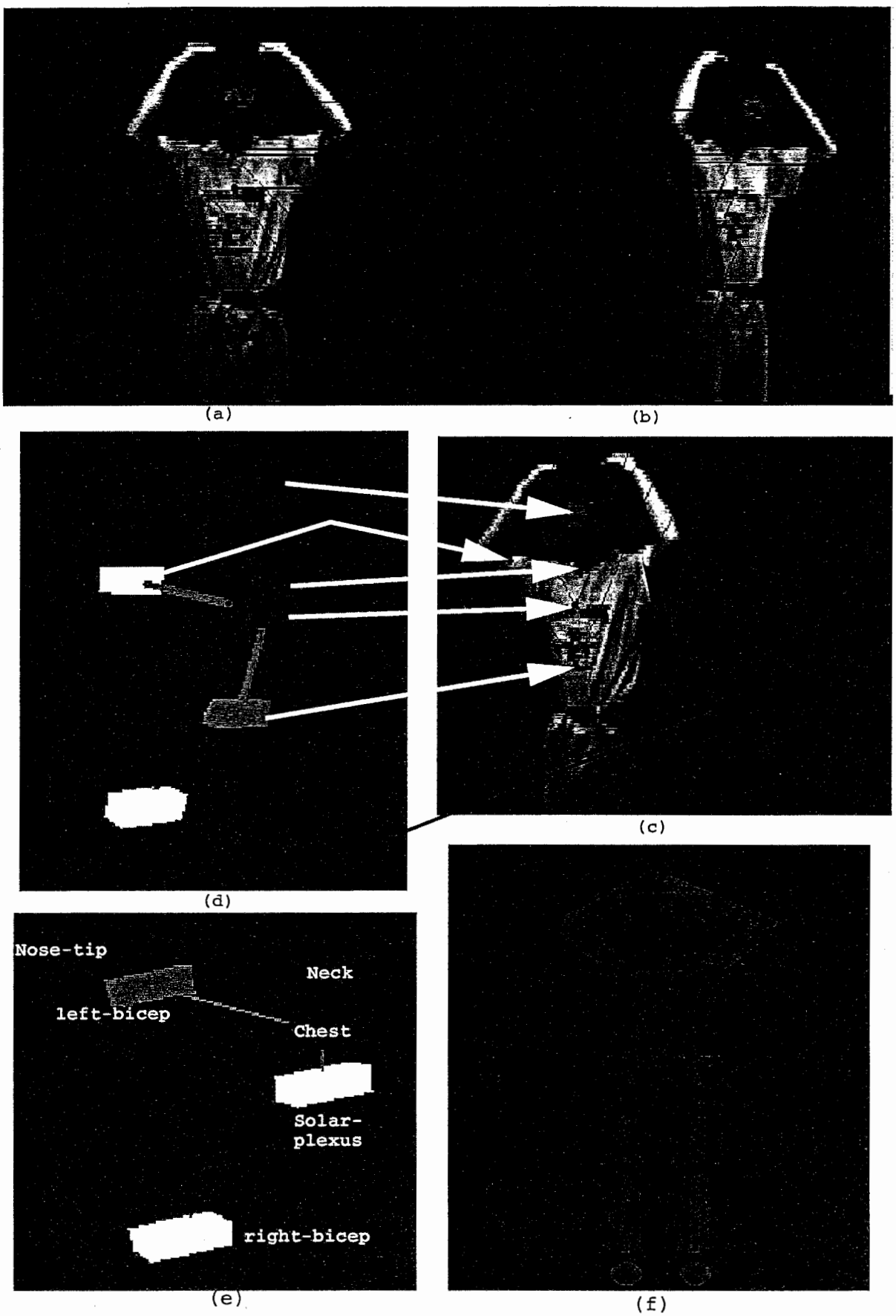


Figure 13: a-c: Selecting six image-imprints on the middle. left and right camera images. (d) associated 3D-cells connected by simple skeleton. (e) Another skeleton representing a different set of six points. (f) A simple synthetic actor mimicing the pose by the participant.