International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:8, 2008

# 3D Star Skeleton for Fast Human Posture Representation

Sungkuk Chun, Kwangjin Hong, and Keechul Jung

***Abstract*—** In this paper, we propose an improved 3D star skeleton technique, which is a suitable skeletonization for human posture representation and reflects the 3D information of human posture. Moreover, the proposed technique is simple and then can be performed in real-time. The existing skeleton construction techniques, such as distance transformation, Voronoi diagram, and thinning, focus on the precision of skeleton information. Therefore, those techniques are not applicable to real-time posture recognition since they are computationally expensive and highly susceptible to noise of boundary. Although a 2D star skeleton was proposed to complement these problems, it also has some limitations to describe the 3D information of the posture. To represent human posture effectively, the constructed skeleton should consider the 3D information of posture. The proposed 3D star skeleton contains 3D data of human, and focuses on human action and posture recognition. Our 3D star skeleton uses the 8 projection maps which have 2D silhouette information and depth data of human surface. And the extremal points can be extracted as the features of 3D star skeleton, without searching whole boundary of object. Therefore, on execution time, our 3D star skeleton is faster than the "greedy" 3D star skeleton using the whole boundary points on the surface. Moreover, our method can offer more accurate skeleton of posture than the existing star skeleton since the 3D data for the object is concerned. Additionally, we make a codebook, a collection of representative 3D star skeletons about 7 postures, to recognize what posture of constructed skeleton is.

***Keywords*—**computer vision, gesture recognition, skeletonization, human posture representation.

## I. Introduction

HUMAN action and posture recognition is a significant part on human centered interface which is coming up to the recent issues nowadays. In posture recognition applications, while the 3D representation of posture is invaluable, many of them require "compact" representation ways of 3D models. One of such representation is a line-like or stick-like 1D representation, which is sometimes referred to as a "skeletal representation" or "curve-skeleton". This is different from the skeletal-surface representation (medial surface), which is a higher dimensional structure [1]. The skeletal representation captures the essential topology of the underlying object in an easy to understand and very compact form.

There are three existing methods for constructing a skeleton, such as distance transformation [2]–[5], Voronoi diagram [6]–[8], and thinning [9]–[12]. Distance transformation is to detect the ridges (local extremes) as skeletal points through the distance from boundary points. This process is executed in linear time and fulfils the geometrical requirements. On the other hand, it does not guarantee the topological correctness. For Voronoi diagram skeletonization, the boundary between each cluster, which is produced by image pixel classification, is extracted as a skeleton. However, it requires an expensive computation, especially for a large and complex object. Skeletonization through thinning is an iterative object reduction technique. Even though this method can preserve the shape of object and produces one pixel/voxel size as a skeleton, the topological correctness cannot be guaranteed if an object is disconnected and a cavity exists in the object.

The existing skeleton construction methods as mentioned above can generate an accurate skeleton. However, those are not applicable to real-time posture recognition since they are computationally expensive and highly susceptible to noise of boundary. To complement these, a 2D star skeleton was proposed [13], [14].

The star skeleton is the technique for human action and posture recognition. The star skeleton, which is "star" fashion, is defined as jointing gross extremities of boundary to its centriod. It is a simple and real-time technique due to using only boundary points. However, since this star skeleton exists in 2D space, the precision of execution result is readily influenced by the yaw and the pitch of target object. That means, there are some limitations to describe a 3D human posture, because the star skeleton is constructed using only 2D information about the real human posture in the 3D space. One way to overcome these problems is to construct 3D star skeleton, called "greedy" 3D star skeleton. This greedy star skeleton, based on 2D star skeleton, is constructed by using the all boundary points on the object surface. Therefore, it is too expensive to extract extremal points by reason of the searching all points on the surface of object. Table I shows the various kinds of computer vision-based skeletonizations and their advantage and disadvantage respectively.

S. Chun is with the School of Media, Soongsil University, Seoul, KOREA (e-mail:k612051@ ssu.ac.kr).

K. Hong is with the Department of Media, Graduate school of Soongsil University, Seoul, KOREA (e-mail: hongmsz@ ssu.ac.kr).

K. Jung is with the Department of Media, Graduate school of Soongsil University, Seoul, KOREA (corresponding author to provide phone: 82-2-812-7520; fax: 82-2-822-3622; e-mail: kcjung@ ssu.ac.kr).

**This work was supported by the 'Seoul R and BD Program (10581cooperateOrg93112)'**

TABLE I
SKELETONIZATIONS, ADVANTAGE AND DISADVANTAGE

| Purpose | Algorithm for skeletonization | Author[paper] | Advantage | Disadvantage |
|---|---|---|---|---|
| Accuracy | Distance transformation | Gunilla Borgefors[2]–[4]<br>Frank Y.Shih et al.[5] | Execution time<br>( in arbitrary dimension ) | Correctness |
| | Voronoi diagram | Franz Aurenhammer[6]<br>Jonathan W. Brandt et al.[7]<br>Kenneth E. Hoff III et al.[8] | Correctness | Execution time |
| | Thinning | Kalman Palagyi et al.[9], [10], [11]<br>Ta-Chih Lee et al.[12] | Correctness | Execution time |
| Posture recognition | Star skeleton | H. Fujiyoshi et al.[13]<br>Hsuan-Sheng Chen et al.[14] | Execution time | Limitation for accurate human posture representation |
| | Greedy 3D star skeleton | | Correctness | Execution time |

In this paper, we propose an improved 3D star skeleton. The proposed 3D star skeleton is based on the 3D information of the human posture, and the purpose is to recognize human action and posture, like the existing 2D star skeleton. Our 3D star skeleton has the distinct points as follows; (1) 3D star skeleton considers 3D information of posture through using 8 projection maps which contain 3D data of human, and (2) our skeletonization is real-time method, since it is constructed by using only the boundary points of posture in the projection map without applying the whole points on the surface. Therefore the skeleton is constructed quickly.

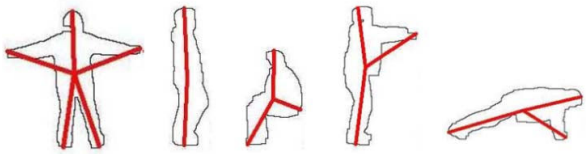
To construct 3D star skeleton, we first calculate the distances between boundary points and the centoid at each projection map. Secondly, Smoothing filter and low pass filter are adapted to the distance function in order to reduce noise on the distance function. And then, the probable candidates as the features of 3D star skeleton can be detected by finding zero crossings in the smoothed distance function. We transform the candidates of each projection map into one particular coordinate using reflection, translation and rotation matrices. To define the extremities as features of final 3D star skeleton, we classify the scattered candidates into 5 groups and obtain the means of each group as extremities. Consequently, the proposed 3D star skeleton is constructed by connecting the centroid with the extremities.

The paper is organized as follows. The section 2 introduces the concept of the existing 2D star skeleton. In section 3, we propose the method to construct the improved 3D star skeleton. Then we show the experimental results in section 4. Finally, conclusion is presented in section 5.

## II. STAR SKELETON

Star skeleton is a technique for human action and posture recognition. Star skeleton, which is a "star" fashion, is a kind of representative features to describe a human posture. The features consist of the several vectors which are the distance from the extremities of human contour to its centroid. Since the star skeleton does not need a lot of pixel computations, it is a simple, real-time and robust technique.

The basis of the star skeleton is to connect the extremities of human contour with its centroid. To find the extremities, each distance from boundary point to the centroid is calculated through boundary tracking in a clockwise or counter-clockwise order. In distance function, the extremities are located at local maxima. Noise reduction should be applied to the distance function by using a smoothing filter or low pass filter, since the distance function of human contour has noises. Consequently, the final extremities are detected by finding local maxima in smoothed distance function. Fig. 1 shows the result of star skeleton.



(a) (b) (c) (d) (e)

Fig. 1 The result of star skeleton [14]

While the existing star skeleton is simple, there are some limitations for 3D human posture recognition. For example, we cannot distinguish whether the human posture is "T pose" or "stand" in case of Fig. 1(b).

## III. PROPOSED METHOD

### A. System Overview

The proposed system consists of 4 parts, including projection map generation, candidates extraction, coordinate transformation, and clustering candidates and the 3D star skeleton construction.

1. Projection map generation : creating 8 projection maps which contain the 3D information of human posture
2. Candidates extraction : finding extremities of posture boundary points in 8 projection maps
3. Coordinate transformation : transforming all candidates of each projection map into one particular coordinate system
4. Clustering candidates and $a$ the 3D star skeleton construction : classifying the transformed candidates and constructing the 3D star skeleton

Projection map can be obtained from the memory buffers when the model is rendered. Projection map contains the 2D silhouette information and the depth information of target

model. In order to apply the 3D data related with the posture to the skeleton, we use the 8 projection maps. Fig. 2(a) shows the process of projection map generation.

In the candidates extraction, we respectively extract the 5 extremities at the each projection map. At each projection map, through 2D and depth information of posture we get the distance function, including the distances from each boundary point to the centroid. And then noise reduction is executed. As the last phase of this part, we detect the candidates, which can be the features of 3D star skeleton, by finding zero crossing points of the distance function. The process is displayed in Fig. 2(b).

After the process of candidates extraction, transformation is implemented to convert the candidates into the new position in one particular coordinate. In this phase, all candidates, which exist in different coordinate system, are transformed into the identical coordinate. For coordinate transformation, we use reflection, translation and rotation matrices (Fig. 2(c)).

To classify the candidates scattered in the 3D space, we use K-Means Clustering algorithms. Through this process, all candidates are clustered into K groups. We define K as 5. The mean of each group represents the extremity as the feature of proposed 3D star skeleton. And then the new centroid is defined as the mean of all candidates. Finally, the proposed 3D star skeleton is constructed by connecting the each mean with the new defined centroid. Clustering candidates and 3D star skeleton construction are shown in Fig. 2(d).

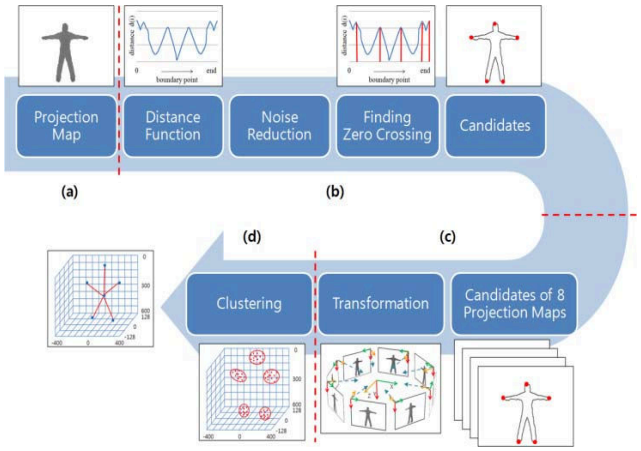



Fig. 2 System flow of the 3D star skeleton: (a) projection map generation, (b) candidates extraction, (c) coordinate transformation, (d) clustering and 3D star skeleton construction

### B. Projection Map

The definition of projection map is to project the 3D world onto the 2D image plane orthographically. And a pixel value on the projection map means the distance from the projection map to the surface of projected object, depth information. Projection map generally contains 3D information of the target object, such as nearest boundary, furthest boundary, and thickness of the object. As the input data, we use the 8 projection maps to construct our 3D star skeleton. The used each projection maps has a different silhouette of human posture according to where the projection map is generated, and each pixel value means a distance from a view point to a voxel of model surface, depth information. Through these properties of projection map, the proposed star skeleton can diminish the loss of 3D human posture information.

The projection map can be generated by rendering the target object. Consider the example in Fig. 3. Given a vector perpendicular to the projection plane, we can find hitpoints: on the front-most surface. We can compute the projection map as follows. First, we set a virtual camera to be able to view the 3D object. The object from viewpoint is then projected onto the camera's view plane (or projection plane). Rasterization, which is the process of converting geometric primitives into pixels, determines the viewing direction and its hitpoint. In rendering the objects front-most surface, the hitpont, on the front-most surface along the viewing direction, is easily extracted for each pixel and saved in a buffer.

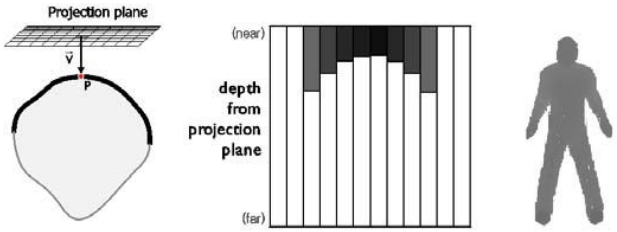



Fig. 3 Projection map generation using depth map: (a) distance from a projective plane to front-most of an object, (b) nearest boundary projection map stores the depth values of front-most surface, (c) generated projection map

### C. Candidates Extraction

The concept of candidates extraction is to detect gross extremities of posture at 8 projection maps respectively. We call the extremity of boundary in each projection map "candidate". Candidate is likely to be a final extremity of 3D star skeleton as a feature.

We use the 8 projection maps, which are the images offered from 8 views. We can get the depth and position information of posture from these projection maps. Since the existing star skeleton is generated by the extremities which are from 2D boundary points, there are some limitations to describe a 3D human posture. To overcome these limitations, we reduce the 3D information losses of human posture through using the 3D boundary, in which depth and x, y coordinate data are included, of posture in the projection map.

To find the candidates, the distances from each boundary point to the centroid are obtained through boundary trace in clockwise or counter-clockwise order. Since the distance function has noises of boundary, noise reduction process is executed through using a smoothing filter or low pass filter. Consequently, candidates are detected by finding zero crossings in the smoothed distance function.

Since, we find 5 candidates at one projection maps, so totally we have 40 candidates that are generated from the 8 projection maps. However, since some projection map cannot

produce 5 candidates according to the posture, the candiate not to be created is defined as a zero vector. Fig. 4 shows the process to find candidates. And an algorithm for the candidate extraction is explained in Fig. 5.

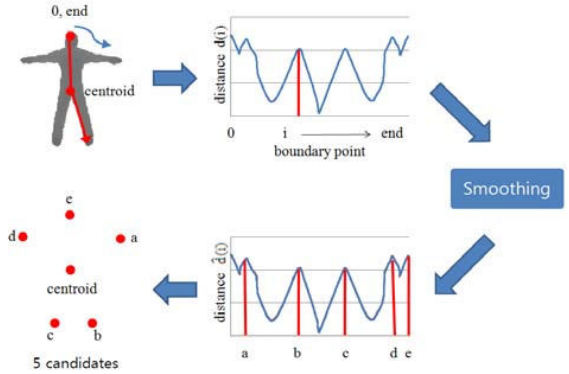



Fig. 4 Distances from the extremities of boundary to the centroid are presented by the distance function. Then the distance function is smoothed and candidates are extracted by finding zero crossings of the difference function.

**Candidates extraction process**

**Input :** Boundary of posture in projection map

**Output :** Candidates

1. Compute the centroid of the posture boundary in a projection map $(x_c, y_c, z_c)$.

$$x_c = \frac{1}{N_b}\sum_{i=1}^{N_b} x_i \quad (1)$$

$$y_c = \frac{1}{N_b}\sum_{i=1}^{N_b} y_i \quad (2)$$

$$z_c = \frac{1}{N_b}\sum_{i=1}^{N_b} z_i \quad (3)$$

where $N_b$ is the number of boundary points, and $(x_i, y_i, z_i)$ is a boundary point of the projection map.

2. Obtain the distances $d_i$ from each boundary point $(x_i, y_i, z_i)$ to the centroid $(x_c, y_c, z_c)$

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2} \quad (3)$$

These are declared as a one dimensional discrete function $d(i) = d_i$.

3. Reduce the noises of the distance function $d(i)$ by using a smoothing filter or low pass. $\hat{d}(i)$ is the smoothed distance function.

4. Take local maxima in $\hat{d}(i)$ as the candidates which are likely to be the extremities of 3D star skeleton. Local maxima which are more than $T_{peak}$ are detected by finding zero crossings of the difference function. And the detected local maxima are far from other local maxima more than $T_{adj}$.

$$\delta(i) = \hat{d}(i) - \hat{d}(i-1) \quad (5)$$

if $\delta(i) < 0$, $\hat{d}(i) > T_{peak}$

and $\sqrt{(x_i - x_m)^2 + (y_i - y_m)^2 + (z_i - z_m)^2} > T_{adj}$,

then i-th boundary point is one of local maxima.

where $(x_m, y_m, z_m)$ is the one of local maxima on m-th boundary point.

Fig. 5 Algorithm for candidates extraction

### D. Coordinate Transformation

In transformation phase, the candidates of each projection map are transformed into one particular coordinate. The coordinate systems of the extracted candidates are not identical. All candidates exist in their own projection map coordinate system. To construct 3D star skeleton, the candidates should be in the same coordinate. Fig. 6 shows the whole system to generate 8 projection maps.

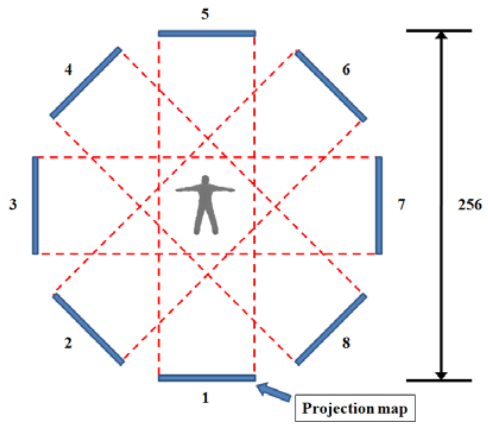



Fig. 6 Projection map generation system

To transform all candidates into one particular coordinate, we do 3 phases; reflection, translation, rotation process. The first process is to fit the two opposite projection map coordinates to one coordinate. For example, in case of Fig. 6, the coordinates of 5, 6, 7, 8 projection map are fitted respectively into the coordinates of 1, 2, 3, 4 projection map. We use the reflection matrix to transform. The applied reflection matrix is;

$$M_{reflection} = \begin{pmatrix} -1 & 0 & 0 & W \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 255 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where W is the width of projection map and 255 is the maximal value of z in the projection map. Fig. 7 shows the reflection result as an example.


Fig. 7 Result of reflection matrix

To rotate the candidates to one particular coordinate, the axes of all candidates should be translated to one identical axis. We use the translation matrix to move the axis. The next matrix performs the axis shift operation.

$$M_{translation} = \begin{pmatrix} 1 & 0 & 0 & -W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 256/2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For more details, the axis shift operation is illustrated in Fig. 8.

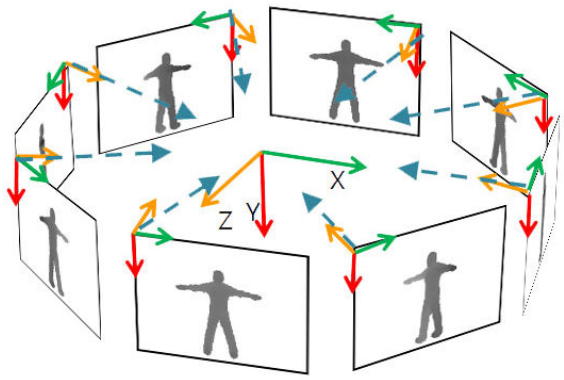



Fig. 8 Axis shift operation. Each axis of the projection maps is translated to one particular axis along each dotted arrow.

The third process is to rotate the 4 couples of projection map coordinate, which are fitted at reflection process, into one particular coordinate. We use the rotation matrix. The used rotation matrix is;

$$M_{rotation} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $\theta$ is a variable which is changed according to the projection map.

Fig. 9 shows $\theta$ of the respective projection maps.

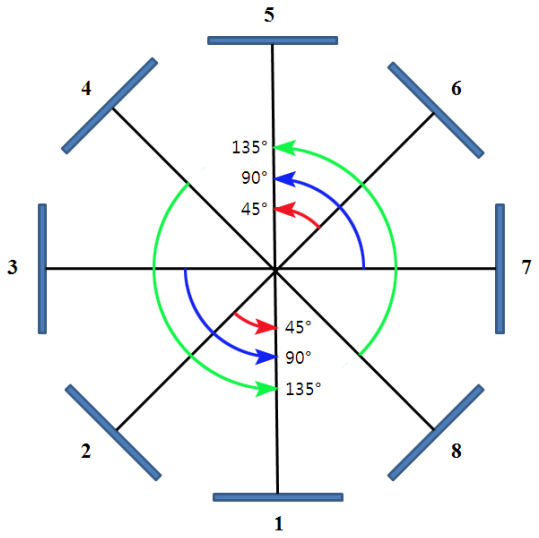



Fig. 9 Coordinate rotation of each projection map

The 4 couples of the projection map are rotated according to each own angle. The coordinates of 6, 7, 8 projection maps should be circulated based on the angle of opposite projection map coordinate.

*E. Clustering Candidates and the 3D Star Skeleton Construction*

To determine the extremities as features of proposed 3D star skeleton, the transformed candidates are classified. The transformed candidates are scattered in one particular coordinate. After transformation process, the transformed candidate indicating the some part of posture locates near others generated from the same part, not exist at one specific position together due to the thickness of human body. For example, even if the candidates are generated from left foot, some candidates locate at a toe, and other candidates are created at the heel according to the projection map. Therefore, all candidates should be classified into the several groups. And each mean of the clusters become the extremities of star skeleton.

K-Means Clustering algorithm is used to classify. Through this process, all candidates are divided into K groups. K is defined as 5. And the centroid is obtained from the average on the centroids of all projection maps. Finally, the 3D star skeleton is constructed by connecting the extremities with the centriod. The features of 3D star skeleton are generated by calculating the distances from extremities to the centroid of 3D star skeleton. Clustering and constructing the 3D star skeleton are illustrated in Fig. 10.

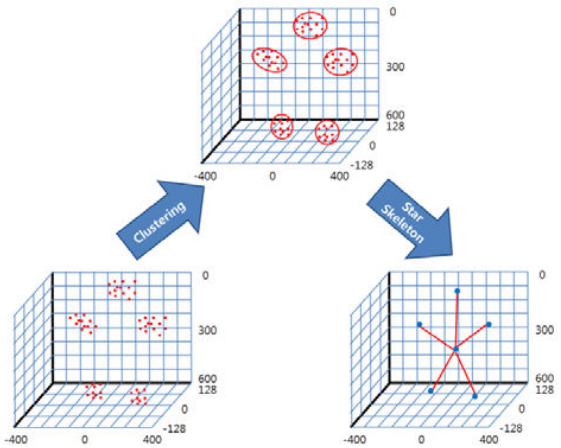


Fig. 10 Process of clustering and constructing three dimensional star skeleton

Additionally, K-Means Clustering algorithm is modified for fitting to our method. In some cases, the candidates might not locate at the landmark points, such as feet or hands. Such points would lead to error in clustering process. For example, one knee point can screw up the foot center if it is included in that cluster. To reduce the error candidates, noises are filtered from the cluster by using standard deviation. After clustering process, if some candidate is far from the mean of its cluster, it is removed and the mean of that cluster is recalculated.

## IV. Experimental Result

To evaluate the efficiency of our 3D star skeleton, we performed our system using 7 human postures. The proposed system has been examined in order to compare with the existing star skeleton technique in 3 cases: (1) The number of operation times (2) The representation result of each posture (3) Correlation between the rotation of human and the number of projection maps. In the number of operation times, we would like to present the advantage about the execution time of proposed system, comparing with the greedy 3D star skeleton and the 2D star skeleton. In the representation result of each posture, we show the contrast between the 2D star skeleton and the improved 3D star skeleton. In correlation between the rotation of human and the number of projection maps, we display the reason that our system is robust to the rotation of human.

### *A. The Number of Operation Times*

The distinguishing point between the greedy 3D star skeleton and our 3D star skeleton begins from the definition of used boundary. The former considers all voxels on the surface as the boundary points. On the other hand, the latter uses only the boundary points, containing depth information, of human posture silhouette in the 8 projection maps. That means our system is implemented more quickly than the greedy 3D star skeleton. It is quite right that our technique is more effective than the greedy way, since the number of used boundary points is smaller. To make more clearly, we prove the advantage of ours using a formula. The following formula shows the number of operation times.

$$N_o = \sum_{i=0}^{N_b} P(b_i) \tag{6}$$

where $N_o$ is the number of operation times to construct star skeleton. $N_b$ is the number of boundary points. $P(b_i)$ is the number of operation times at the boundary point $b_i$. That means the number of times to utilize $b_i$ for the 3D star skeleton construction.

Especially, when boundary tracking for the distance function and clustering are implemented, the time complexities for the number of operation times at one pixel are $O(n^2)$ and $O(n^2)$ respectively. Therefore, the difference of execution time between two star skeleton techniques becomes more conspicuous according to the increase of boundary points. Consequently, on the number of operation times, the proposed 3D star skeleton is more efficient than the greedy 3D star skeleton. The following relationship formula shows the non-linear relation of two techniques about the number of operation times.

$$\sum_{i=1}^{E_b} P(b_i) > \frac{E_b}{P_b} \times \sum_{i=1}^{P_b} P(b_i) \tag{7}$$

where $E_b$ is the number of all pixels, as boundary points, on the object surface. $P_b$ is the number of boundary points which exist in the 8 projection maps. $b_i$ is a boundary point.

On the other hand, our star skeleton is inferior to the 2D star skeleton. As we mentioned above, the number of operation times and processing time are dependent on the number of used boundary points. To construct the skeleton, while the 2D star skeleton use only 1 image, the proposed skeleton is generated using 8 projection maps. Therefore, our technique requires more time for skeleton construction than the existing method. Table Ⅱ shows the number of required boundary points and processing time on the 2D skeleton and the 3D skeleton for "T pose" respectively.

TABLE Ⅱ
PROCESSING TIME AND THE NUMBER OF USED BOUNDARY POINTS FOR 2D AND 3D STAR SKELETON (T POSE)

| | 2D Star Skeleton | 3D Star Skeleton |
|---|---|---|
| The number of boundary points | 1880 | 9880 |
| Processing time(ms) | 81 | 734 |

Through Table Ⅱ we could get the information that the processing time of our technique is more expensive than that of the existing 2D star skeleton. The processing time was increased linearly according to the number of used boundary points.

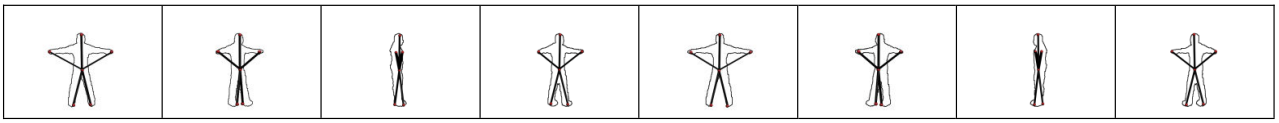
Finally, while our skeleton is more effective than greedy

star skeleton, it needs more time to construct skeleton than the 2D star skeleton. However, our skeleton technique is superior to the 2D star skeleton about How to represent the posture efficiently. It is mentioned in the next section.

### *B. The Representation Result of Each Posture*

The proposed 3D star skeleton has been evaluated on the representation result of human posture. To test the performance efficiency, we used the 7 types of posture: "T pose", "right hand up", "left hand up", "sit", "stand", "both hands up", "band". As mentioned before, we used the 8 projection maps to construct skeleton. To show the result 3D skeleton on this paper effectively, we describe the constructed 3D star skeleton through 8 viewpoints located around the skeleton.

The extremities as the features of star skeleton are extracted from each mean of the clusters. The candidates come out of 8 projection maps. In normal cases, the 5 candidates are generated in one projection map. However, in abnormal cases, the candidates may be obtained less than 5 from each projection map. For generalization, we considered the candidate not to be generated as a zero vector. The results of 7 human postures using our star skeleton are displayed in Fig. 11. We also put the results of the existing star skeleton about 8 rotated "T pose" postures, in order to compare with our technique. The 8 2D star skeletons in Fig. 12 show the different shapes according to human rotation, although being generated from the same "T pose".



(a) T pose

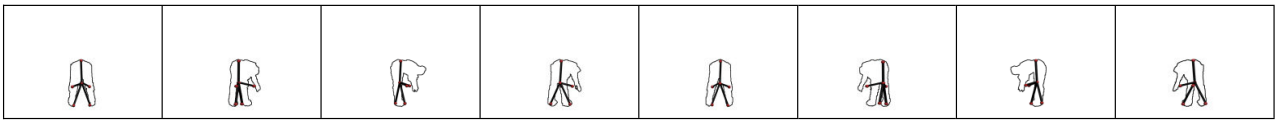
(b) right hand up

(c) left hand up

(d) sit

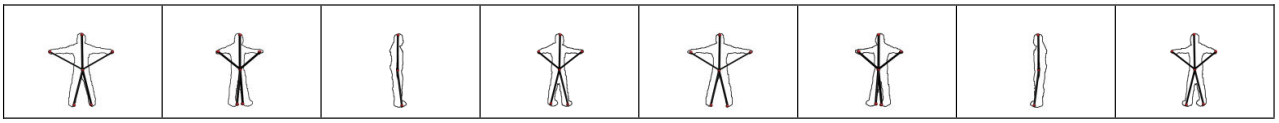
(e) stand

(f) both hands up



(g) band

Fig. 11 3D star skeleton results for 7 human postures



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12 2D star skeleton results for 8 differently rotated "T shape" postures

In cases of Fig. 12(a), (b), (d), (e), (f) and (h), human posture can be described as "T pose" by the 2D star skeleton. That is, the extremities can be the valid features to represent human posture in some cases. However, since the existing star skeleton is constructed in 2D space, the precision of execution result is readily influenced by the yaw and pitch of human posture. For example, in case of (c) and (g), the constructed star skeleton is not enough to describe "T pose".

On the other hand, the proposed method generates the 3D skeleton using the boundary points of 8 projection maps, which has 3D information of posture. Therefore, the constructed star skeleton can represent the human posture more exactly than the existing star skeleton does.

In cases of "left hand up", "right hand up" and "stand", our method drew the different skeletons from intuitive thinking. The existing skeletonizations focus on the accurate skeleton itself about where the parts of human body locate. However, the proposed technique deals with how to detect the feature for human action and posture recognition. In other words, the aim of our method is to extract the distinguishing features, which describe a human posture, more rapidly and effectively than the existing skeletonizations. For example, "left hand up" pose is described through the features extracted at the raised left arm, such as the end of finger and the elbow of left arm. In case of "stand", that human body is perpendicular to the ground is considered as more important feature than the position of both hands. Therefore, our skeleton for recognition is different from the perception of human.

### *C. Correlation between the Rotation of Human and the Number of Projection Maps*

As we mentioned before, our skeleton uses the 8 projection maps. Even though the computation time increases according to the number of projection maps, the more projection maps can make the system to be strong to the rotation of human. In other words, the more projection maps ensure the more accurate skeleton, but the system may be more expensive computationally due to the increasing number of operation times. In order to evaluate the advantage of our system on rotation, we implemented our system by using the various number of projection maps. Table Ⅲ shows the each processing time on using the different number of projection maps.

TABLE Ⅲ
PROCESSING TIME ON USING THE VARIOUS NUMBER OF PROJECTION MAPS (T POSE)

| The number of used projection maps | Processing time |
|---|---|
| 1 projection map | 94 ms |
| 2 projection maps | 187 ms |
| 4 projection maps | 375 ms |
| 8 projection maps | 734 ms |

The generation time for the 3D star skeleton linearly goes up according to the number of used projection maps.

To test about the correlation between the human rotation and the number of projection map, we rotated the human in $0^\circ, 45^\circ, 90^\circ$. The reason the human rotated in $45^\circ$ per each test is that the silhouette image of posture in projection map is varied apparently by rotation in $45^\circ$ per each test. Fig. 13 displays the constructed 3D star skeleton of "T pose" when 2, 4 and 8 projection maps are applied respectively.

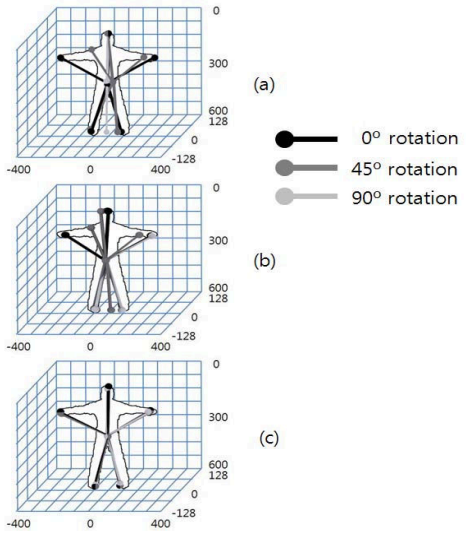



Fig. 13 3D star skeletons generated according to the each number of projection maps: (a) 2 projection maps, (b) 4 projection maps, (c) 8 projection maps

Moreover, we made a codebook, collection of the 7 representative 3D star skeletons for 7 postures, in order to evaluate the constructed star skeleton. In the codebook, the vector, called a codeword, as representative 3D star skeleton for each posture is contained. The vector has 2 components, the distances from each extremity to the centroid and the orientation of extremities. For recognition of what the posture of constructed skeleton is, we compared the skeleton with each codeword in the codebook. We use Euclidian distance for comparison. Among all codewords, as a result, the posture of the codeword having a minimal distance with the constructed skeleton is assigned as the posture of skeleton.

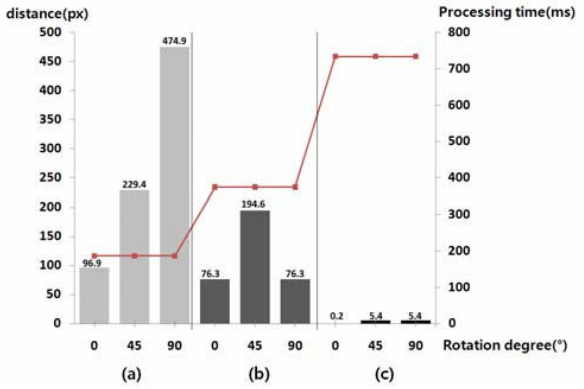



Fig. 14 Processing time and the distance between "T pose" codeword and 3D star skeletons using the various number of projection maps: (a) 2 projection maps, (b) 4 projection maps, (c) 8 projection maps

Fig. 14 shows the processing time and the distance between "T pose" 3D star skeletons using 2, 4, 8 projection maps and "T pose" codeword.

Table Ⅳ displays the distances between "T pose" 3D star skeleton and codeword of each posture defined in the codebook, according to rotation degrees and the number of projection maps.

In case of $0^\circ$ rotation, the star skeletons have the minimal distance at "T pose" codeword regardless of the number of projection maps. Therefore, the skeleton is recognized as "T pose" posture. However, the precision of recognition was susceptible to the number of projection map when the human rotated.

In case of 2 projection maps (Fig. 14(a)), the skeletons, constructed from the postures rotated in $45^\circ$ and $90^\circ$, show the a lot of difference with "T pose" codeword. On the recognition (Table Ⅳ), moreover, the skeleton using $90^\circ$ posture is matched to "stand" codeword. And consequently it is not recognized as "T pose", since the distance with the codeword is large. Even though the skeleton with the posture rotated in $45^\circ$ is recognized as "T pose", the recognition result is not guaranteed due to the several reasons; the gap with "T pose" codeword is large and similar to "stand" codeword. Therefore, there is much probable the skeleton is not recognized as "T pose" if the posture type similar to "T pose", like "left hand to the left" or "right hand to the right", is included in the codebook. In other word, if the codebook is expanded, it may be not possible to distinguish "T pose" from "stand", "left hand to the left" or "right hand to the right".

In case of 4 projection maps (Fig. 14(b)), our method constructs the skeleton of $90^\circ$ rotation posture similar to $0^\circ$ rotation "T pose" skeleton. And the recognition result is also correct, matching with "T pose" codeword (Table Ⅳ). However, the skeleton with $45^\circ$ rotation posture has a wrong recognition result, since it is most similar with "stand" codeword.

On the other hand, our star skeleton, which is generated by using 8 projection maps, guarantees the precision of recognition result, regardless of the rotation of human. In Table Ⅳ, our star skeleton has small distance and is highly similar with "T pose" codeword. And on the side of recognition our technique is recognized as "T pose", having the apparent differences with other codewords. Therefore, our method is an effective way to represent human posture since it is robust to the rotation of human, despite of some losses on computation time.

## V. Conclusion

In this paper, we have presented the improved 3D star skeleton. The proposed system is optimized to human posture representation. To construct the proposed star skeleton, we use the 8 projection maps, which include the depth and silhouette information of human posture. Through the projection maps, 3D data for human posture representation is reflected to our 3D star skeleton. Moreover, our system is a simple and real-time method due to using not all voxels on the object surface but only the boundary points of silhouette in the projection maps. The proposed skeleton is constructed by using the initial candidates, transformation matrix and clustering operation. The improvement of performance was proved through the number of operation times, the representation results of 7 postures and correlation between the rotation of human and the number of projection maps.

For the future work, we will research on 3 parts. First, we will define the reasonable number of extremities (K value in clustering method) which compose 3D star skeleton. In this paper, we defined K as 5 since the existing star skeleton also used 5 extremities based on the limbs and head. It is the influential part to define the number of extremities, for the reason that the recognition rate and the computation time are closely related to the number of extremities. Therefore, the number of extremities should be determined clearly. Secondly, we will compare our method with other skeletonizations more in details, through converting the existing skeletonizations to the 3D skeleton techniques. Then we will be able to evaluate the efficiency of our method exactly. Third, we will study on human action recognition based on the improved 3D star skeleton. Additionally, we will apply HMM, Hidden Markov Model, to our system for human action recognition.

TABLE Ⅳ
DISTANCES BETWEEN "T POSE" 3D STAR SKELETON AND EACH CODEWORD ACCORDING TO ROTATION DEGREES AND THE NUMBER OF PROJECTION MAPS (PX)

| Rotation degree(°) | 0 | | | 45 | | | 90 | | |
|---|---|---|---|---|---|---|---|---|---|
| Codeword posture \ The number of projection map | 2 | 4 | 8 | 2 | 4 | 8 | 2 | 4 | 8 |
| T pose | **96.9** | **76.3** | **0.2** | **229.4** | 194.6 | **5.4** | 474.9 | **76.3** | **5.4** |
| right hand up | 523.0 | 503.3 | 463.4 | 482.4 | 359.8 | 264.4 | 607.5 | 503.5 | 461.4 |
| left hand up | 538.7 | 530.2 | 478.8 | 472.9 | 423.7 | 479.7 | 615.3 | 530.3 | 479.7 |
| sit | 563.3 | 558.7 | 518.0 | 443.6 | 399.2 | 516.5 | 416.5 | 558.0 | 515.5 |
| stand | 387.2 | 359.7 | 306.7 | 291.0 | **180.3** | 306.4 | **381.1** | 360.4 | 305.8 |
| Both hands up | 510.8 | 499.8 | 463.2 | 488.1 | 413.1 | 461.8 | 709.2 | 499.2 | 461.8 |
| band | 602.5 | 597.7 | 552.2 | 597.7 | 463.8 | 552.8 | 465.8 | 597.7 | 552.7 |

## REFERENCES

[1] Nicu D. Cornea, Deborah Silver, Patrick Min, "Curve-Skeleton Properties, Application, and Algorithms," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, 2007, pp. 530-548.
[2] Gunilla Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics, and Image Processing*, vol. 34, 1986, pp. 344-371.
[3] Gunilla Borgefors, "Distance transformation in arbitrary dimensions," *Computer Vision, Graphics, and Image Processing*, vol. 27, 1984, pp. 321-345.
[4] Gunilla Borgefors, "On digital distance transforms in three dimensions," *Computer Vision and Image Understanding*, vol. 64, 1996, pp. 368-376.
[5] Frank Y.Shih and Christopher C.Pu, "A skeletonization algorithm by maxima tracking on Euclidean distance transform", *J. Pattern Recognition*, vol. 28, 1995, pp. 331-341.
[6] Franz Aurenhammer, "Voronoi diagrams - A Survey of a fundamental geometric data structure," *ACM Conputing Surveys*, vol. 23, 1991, pp. 345-405.
[7] Jonathan W. Brandt and V. Ralph Algazi, "Continuous skeleton computation by Voronoi diagram," *CVGIP : Image Understanding*, vol. 55, 1991, pp. 329-338.
[8] Kenneth E. Hoff III, Tim Culver, John Keyser, Ming Lin and Dinesh Manocha "Fast computation of generalized Voronoi diagrams using graphic hardware," in *Proc. 26th annual Conf. Computer graphics and interactive technique*, 1999, pp. 277-286.
[9] Kalman Palagyi, Erich Sorantin, Emese Balogh, Attila Kuba, Csongor Halmai1, Balazs Erdohelyi, and Klaus Hausegger, "A Sequential 3D Thinning Algorithm and Its Medical Applications," in *Proc. 17th international Conf. IPMI*, vol. 2082, 2001, pp. 409-415.
[10] Kalman Palagyi and Attila Kuba, "A 3D 6-subiteration thinning algorithm for extracting medial lines," *Pattern Recognition Letters*, vol. 19, 1998, pp. 613-627.
[11] Kalman Palagyi and Attila Kuba, "Directional 3D thinning using 8 subiterations," in *Proc. 8th international Conf. DGCI*, vol. 1568, 1999, pp. 325-336.
[12] Ta-Chih Lee, Rangasami L. Kashyap and Chong-Nam Chu, **"**Building skeleton models via 3-D medial surface/axis thinning algorithms," *CVGIP : Graphical Models and Image Processing*, vol. 56, 1994, pp. 462-478.
[13] H. Fujiyoshi and A. J. Lipton, "Real-time human motion analysis by image skeletonization," *4th IEEE Workshop on Application of Computer Vision*, 1998, pp. 15-21.
[14] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen and Suh-Yin Lee, "Human Action Recognition Using Star Skeleton," in *Proc. 4th ACM international workshop on Video surveillance and sensor networks*, 2006, pp. 171-178.