Computer-Aided Design and Applications

http://www.cadanda.com

EARS: A System for Geometric and Anthropometric Evaluation of Human Body Scans

Xuetao Yin[1], Brian D. Corner[2] and Anshuman Razdan[3]

[1]Arizona State University, Xuetao.Yin@asu.edu
[2]WarSTAR, U.S. Army Natick Soldier RDEC, brian.corner@us.army.mil
[3]Arizona State University, arazdan@asu.edu

ABSTRACT

We present the Enhanced Anthropometric Rating System (EARS), an automated system for evaluating the quality of 3D human body scans. EARS is able to detect and classify both the geometric and anthropometric features of a given mesh and rates its quality. These features and corresponding operations include the roughness of the scanned surface, the fairness of vertex location, area and position of missing body parts, anthropometrically guided segmentation, detection of landmarks, and wrinkles in clothing. The system ranks these features and operations based on their importance as determined by Anthropologists who have specific requirements with respect to understanding the anthropometry of the soldier of the 21st century. The data scans contain more than 100,000 vertices and over 300,000 facets. The system is able to provide real-time feedback on whether the mesh is suitable for downstream applications. The system will be used by the U.S. Army to do statistical studies on their large human body dataset.

Keywords: geometry processing, anthropometric landmarks, mesh evaluation.
DOI: 10.3722/cadaps.2009.431-445

1. INTRODUCTION

During the past few years, large numbers of 3D human body scans have been collected and examined for various purposes and applications [3]. To ensure only scans of high quality are saved for later processing and analysis, it is critical to have a tool that can evaluate the scan quality quickly, automatically, and reliably. This is the motivation for developing EARS. In this paper, we present an overview of the system as well as the algorithms used to identify and measure specific scan attributes.

1.1 Requirements

For quality evaluation, EARS takes the following aspects of an input into account:

- *Topological Soundness*: The scanned surface should be a manifold, meaning any point on the surface has a small neighborhood that can be mapped onto a plane.
- *Closeness*: All edges of the scanned surface should be adjacent to two faces.
- *Smoothness*: The scanned surface should be overall smooth. Rough regions indicate scan errors such as hair and scanner calibration issues.

In order to assess these three major measurements, some pre-processing tasks are required:

- *Uniform Alignment*: Models are embedded in individual coordinate system. A uniform embedding provides consistency among scans and comparisons.

- *Hole Filling*: Voids or holes appear in the scanned surface due to occlusions and a combination of material/camera calibration issues. These holes should be filled before segmentation (see below).
- *Differential Geometry Estimation*: The computation of normals and curvatures are required.
- *Segmentation*: Defects are labeled according to their location on the body; hence a meaningful segmentation procedure is needed.

1.2 Operation Hierarchy

EARS is organized as a set of independent procedures in the order of their relative importance to determine overall mesh quality. All procedures are operated sequentially. Fig. 1 shows the EARS pipeline.

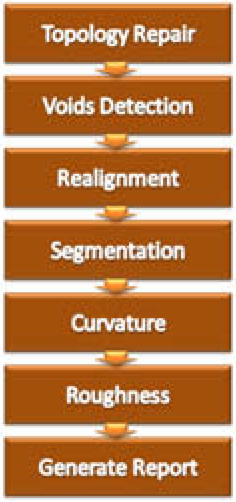


Fig. 1: The pipeline of the EARS.

The rest of the paper is organized as follows: From section 2 to 6, each component of the system is expanded upon; in section 7, we provide a system overview of EARS and discuss efficiency; the last section is a summary and discusses future improvements.

2. TOPOLOGY REPAIR

The operation of EARS begins with the user loading in a scan mesh data file (preferably .ply format). EARS utilizes the half-edge data structure provided in the Computational Geometry Algorithms Library (CGAL) [5]. Half edge data structure represents each edge in the mesh by two half-edges with opposite directions; information about the facet on the left hand side (adjacent facet), previous and next halfedge in the same face, and the vertex pointed to is associated with each of the half-edges as illustrated in Fig.2.

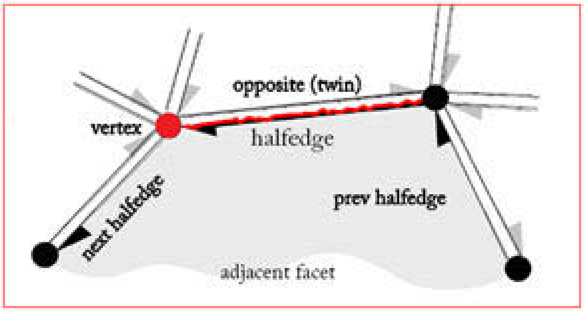


Fig. 2: The half edge data structure of CGAL [5].

This B-rep is efficient for adjacency queries; it is also intrinsically restricted to represent only manifolds. This limitation, however, works in our favor because one of the key requirements of scan evaluation is to detect and repair all topological errors, namely the non-manifold portion of the triangle surface, which is introduced by the triangulation algorithm that converts point cloud data into a triangle mesh surface. EARS marks the following defects and discards the facets that cause them.

- *Intruding Facet:* Identified as a facet that is adjacent to a vertex whose one ring neighborhood has already been completed. Concomitant errors: T-junction. Shown in Fig. 3 (left).
- *Conflicting Normals*: Identified as multiple appearances of a single halfedge. Shown in Fig. 3 (middle).
- *Outliers*: Facets that are not part of the largest connected component of the mesh. Detected after region growing. Example is shown in Fig. 3 (right).

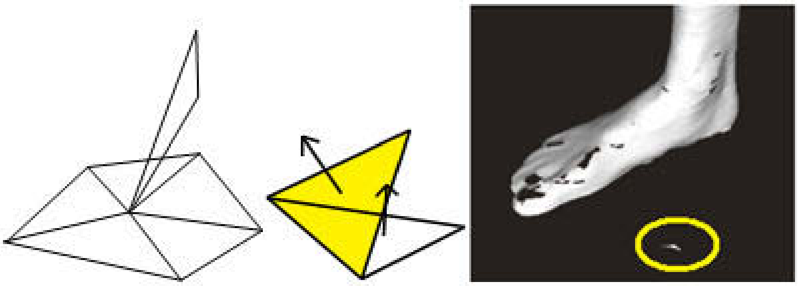


Fig. 3: Defects, from left to right, intruding facets, conflicting normals, outliers.

EARS reports the information about the removed facets as part of its quality feedback.

3. VOID DETECTION

Due to occlusions, material properties etc, some parts of the object are not captured by the scanner. This leaves holes in the reconstructed surface at places such as the crotch and armpits. The location and affected area of these holes (or voids) is critical in the determination of mesh usability. EARS identifies all the voids, along with their respective sizes and locations. Several filling options are also provided.

3.1 Definition and Detection

Voids are defined as sets or loops formed by connected boundary halfedges. The detection algorithm starts with collecting all the half-edges that have no adjacent facets. Following the *next* and *previous* half-edge adjacency queries, closed loops are collected as surface voids.

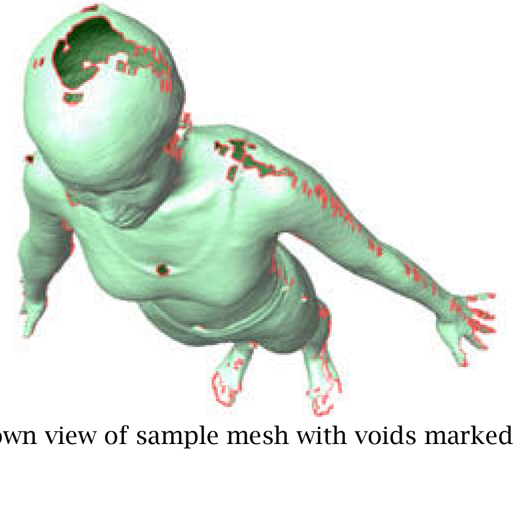


Fig. 4: Top-down view of sample mesh with voids marked in red.

3.2 Measurements and Filling

The most important measure of voids is the estimated surface area affected by the voids. EARS *virtually* fills each void with simple triangulation as shown in Fig. 5. This scheme treats voids with odd and even number of border edges differently. The resulting triangulation area is reported by the system as part of the feedback report. The sum of area of the voids is compared to the existing surface area in the report. This ratio, as we observed, is one of the indicators of scan coverage quality. It can be employed as a quick check for identifying poor scans that did not have adequate coverage. (Fig. 6)

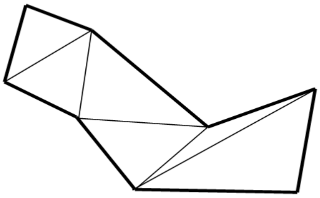


Fig. 5: (Left) A void with even number of vertices; (Right) a void with odd number of vertices.

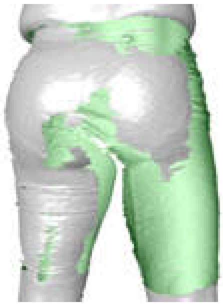


Fig. 6: Typical bad scans where large amount of surface is missing.

EARS employs two simple schemes for filling a hole: **naïve filling**, where a new vertex is added and positioned at the volumetric centroid of a target void and connected to all vertices on that void; **greedy filling**, an advancing front type of algorithm which adds triangles meeting certain requirements iteratively to the void until the gap is completed. It is a simplified version of the hole filling algorithm in [8]. At each iteration, assume that the current edge count of the void is *n*, then there are *n* candidate triangles for addition. The chosen triangle is the one with minimal maximal inner angle among all. The algorithm is illustrated in Fig. 7. Comparison between two schemes is shown in Fig. 8. An important point to note is that we did not employ a very sophisticated algorithm for hole filling because of two reasons. One, the execution time needed to be fast. Second, the filled area does not affect the evaluation of the rest of the surface. It is needed to close the surface for downstream processing such as segmentation; hence the quality of the fill was not a factor.

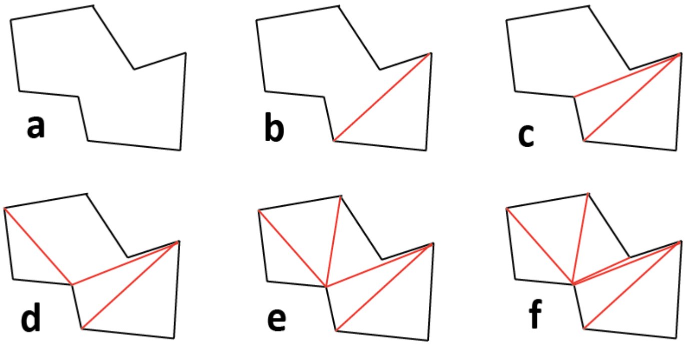


Fig. 7: Illustration of greedy triangulation.

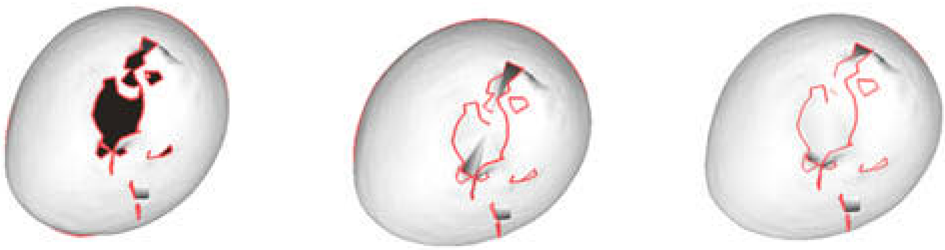


Fig. 8: (Left) A void in the top of the head; (Middle) naïve filling; (Right) greedy filling.

4. REALIGNMENT

EARS is designed to process human body scan data, hence positioning all the data in a unique stance and framework is desirable for downstream operations such as segmentation and classification. Advantages of a uniform model embedding include reliable geometry information and orientation of the scanned human body. An example of realignment is shown in Fig. 9. EARS defines the local coordinate system as follows: the z axis runs from the feet to the head, the y axis runs from the back to the front, and the x axis runs from right to left. In Fig. 9, red lines represents the x axis, green lines represents the y axis, and blue lines represents the z axis. The same configuration applies to Fig. 10 and Fig. 11.

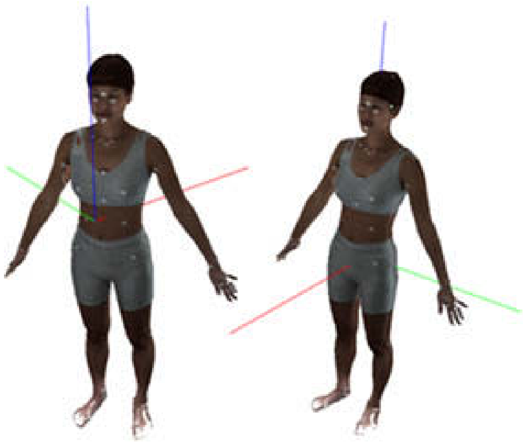


Fig. 9: (Left) Initial stance of a sample model; (Right) after realignment.

4.1 PCA

Principal Components Analysis (PCA) is one of the most commonly applied techniques in model realignment. By solving the eigen decomposition of the covariance matrix, we obtain the three principal directions and align the model along those directions using 3D transformations. In order to achieve the balance of efficiency and accuracy, we test run PCA using only a portion of the whole mesh. Optimal results are obtained when 90% of the vertices are used.

4.2 Orientation Adjustment

PCA is always not successful in determining if the realigned model is in its correct orientation. EARS expects every model to face in the positive x direction with the head pointing in the positive z direction. Adjustment is needed if the orientation of head and face is not correct. Example is shown in Fig. 10.

This adjustment is automatically carried out by EARS. To achieve this, EARS asks two questions: *Is the model positioned upside down?* And, *is the model facing forward or backward?* Different answer pairs cause different rotation sequences. In order to answer these, the vertex mass within certain regions are carefully examined as shown in Fig. 11 (region inside the boxes). In order to test head position, EARS attaches two boxes with adjustable sizes to both ends of the z axis. EARS can tell which box contains the feet by checking the number of connected components in them. The height of these boxes is $1/7^{th}$ of the height of the whole body by default. Areas near the top and the bottom of a scanned human body surface suffer the most content loss. If the boxes are restrained to these flawed areas, a poor choice of height might cut the mesh at places where the number of connected components is larger than 2, hence it can mislead the algorithm. On the other hand, the height of these boxes should not exceed the distance between the top of the head to the shoulder line for the algorithm to work correctly The choice of $1/7^{th}$ is based on average body size survey. EARS allows the user to adjust this setting if needed. To test the face direction, EARS attaches two adjustable boxes to the lower end of the z axis, one at the bottom contains the feet and the other contains the sin/calf areas. Both have height that is $1/14^{th}$ of the total body height by default (half the height of that of the previous box at the bottom). EARS determines if the toes are pointing in the right direction by checking the x coordinate of the vertices in both boxes.

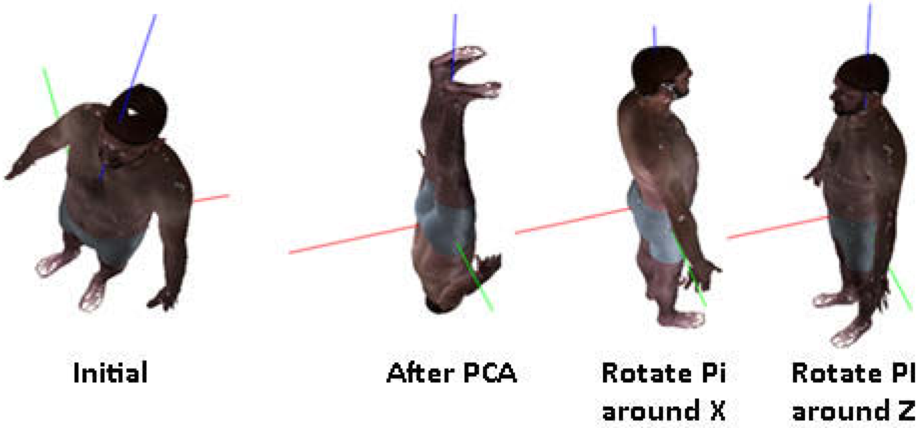


Fig. 10: Position adjustment example.

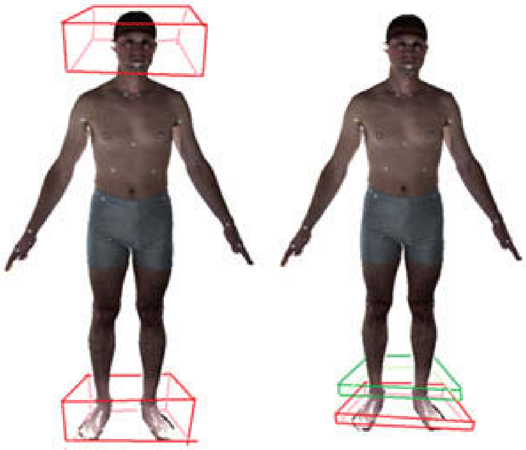


Fig. 11: (Left) Boxes used to answer the first query. The box enclosing feet has two connected components rather than one. (Right) Boxes used to answer the second query. The box enclosing feet should span longer in the x direction.

5. SEGMENTATION

Mesh Segmentation [1] is a technique that decomposes the mesh into *meaningful* parts. *Meaningful* is dependent on the end application. It has become an important tool in many applications in CAD and

Computer Graphics. In past years, many algorithms have emerged to tackle this problem. Generally, there are two groups of algorithms: one uses only geometric features of a given mesh hence should work on all meshes. Examples include fuzzy clustering [6]. The other group explores the characteristics of a specific type of models and cannot be applied to models outside of its target group. EARS falls into the latter group.

EARS employs mesh segmentation based on two reasons: first, EARS reports and rates mesh quality based on body parts rather geometrical location coordinates; second, downstream algorithms in EARS can have different parameter setting for each body part. The goal is to segment the body into head, torso, arms, and legs.

5.1 Algorithm Overview

EARS employs a similar approach as in [11]. It can be viewed as a simplified Reeb Graph where the indicating variable is simply the height. Essentially, the algorithm finds a topological abstract of the mesh and EARS places its cuts at the bifurcation joints. The algorithm implemented in EARS is constrained to process human scans in the upright standing position.

The algorithm starts from top of the model and traverses along the negative z direction. At each small interval (0.1% of body length as default setting in EARS), it slices the model surface using a plane that is perpendicular to the z axis. This slicing produces one or multiple connected strips of faces. The enclosed area of each strip is computed. We only examine the strips that have significant area so as to avoid small strips resulting from body parts such as ears. Multiple significant strips indicate a change in topology, i.e. regions to cut the mesh. A sample result is shown in Fig. 12.

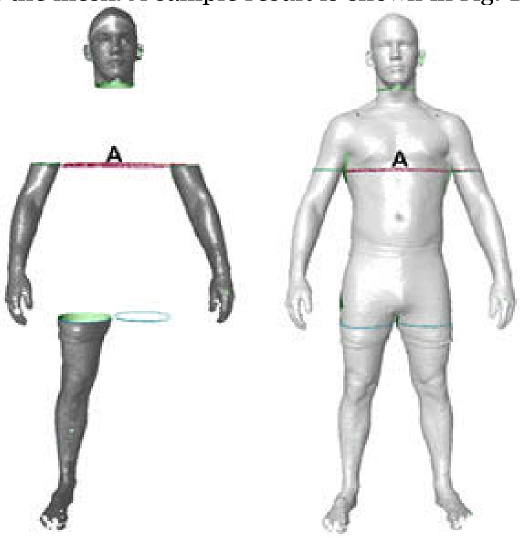


Fig. 12: Sample segmentation result. Rendered in EARS display window: The green-bluish strips are cutting strips while the one in maroon (marked with letter **A**) indicates the other strip generated along with the arm/torso cuts.

5.2 Head Cut

Not all cuts can be located solely using topological information. The cut that separates the head and torso needs geometric information. EARS implements three cut schemes, each associated with an anthropological specification.

- *Upper neck.* Defined as the fastest retreating point of the frontal profile. During the slicing procedure, the front most point (maximal in x direction) of a strip is recorded. The profile is a poly-line connecting all these points. Sample profile is shown in Fig. 15 (left) of section 5.3.
- *Shoulder.* Defined as the point where the span along the y axis increases the fast.
- *Valley.* Defined as the deepest dent in the frontal profile curve. It often coincides with the joint of the clavicles.

The choices of three cutting schemes offer flexibility to the user; it is also possible to combine these cuts into a weighted averaged cutting. Examples are shown in Fig. 13 and Fig. 14.

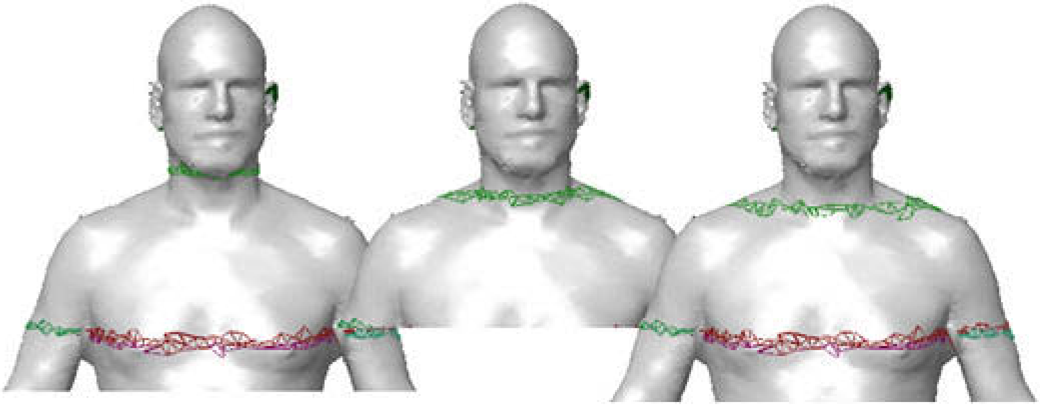


Fig. 13: Comparison of different head segmentation settings: (left) upper neck; (middle) shoulder; (right) valley point.

Fig. 14: Comparison of different head segmentation settings: (left) upper neck; (middle) shoulder; (right) valley point. Notice in Fig. 12 the valley point is above the shoulder instead of below it as in Fig. 11, which reflects the variances in the human form.

5.3 Variations in the Human Form

In Fig. 14, we show how segmentation algorithm produces different outcomes on an individual body. Since human form varies greatly with sex, age, and race, designing a scheme that would work for all possible input is difficult. Fig. 15 (left) shows how upper neck cut fails by cutting beneath the nose. Fig. 15 (right) shows that the subject has his thighs joined (as it appears in the mesh), hence topology cut fails, as it cuts above the knee caps.

To overcome these obstacles, detailed examination and statistical analysis of large dataset is critical. Only after anthropological landmarks are clearly defined, can they be translated into geometric measurements and then employed to guide segmentation and other shape related tasks. This is one of the many directions where EARS will be working on in the future.

5.4 Further Decomposition

There are cases where further decomposition is needed, such as segmenting a whole leg into foot, calf, and thigh. These cuts are based purely on geometric features. Generic segmentation algorithms, such as fuzzy clustering [6] can be applied as shown in Fig. 16. Further, since most segmentation happens at regions of concavity, some visualization techniques can be used to provide a hint.

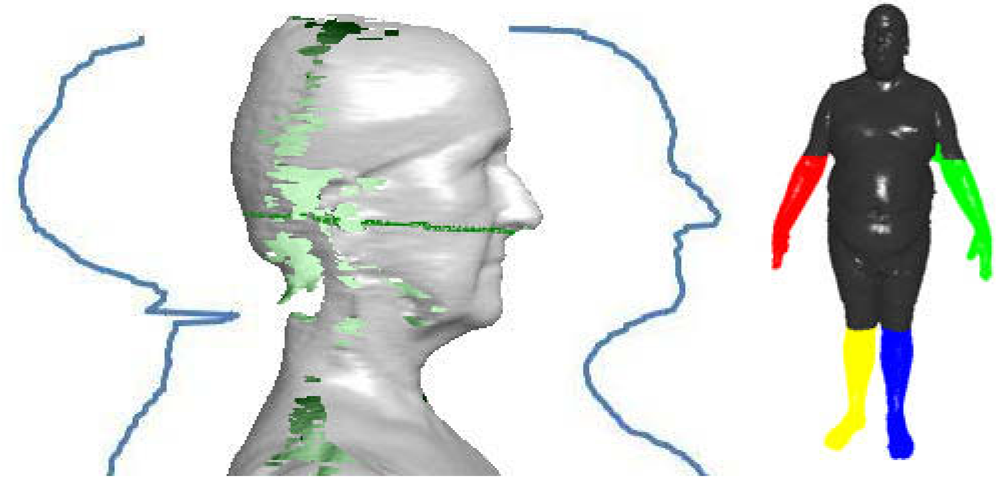


Fig. 15: (Left) The green strip is the cut line that should cut the head from torso. The two blue curves from left to right are the back profile and frontal profile respectively; (right) The torso is shaded in black, while the two legs are in yellow and blue. The thighs are identified as part of the torso. The scanner was unable to distinguish the two thighs due to excessive tissue in the subject's thigh area.

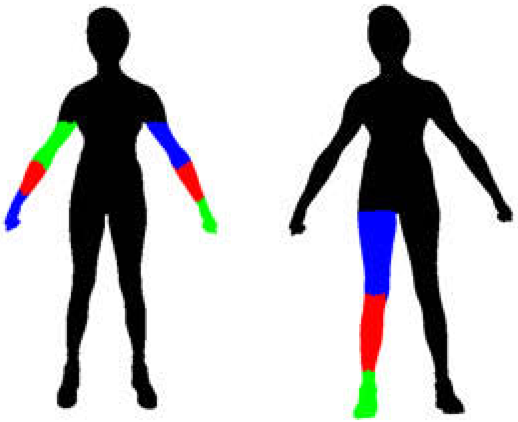


Fig. 16: Further decomposition on (left) arms and (right) legs.

Ambient Occlusion [12] is a technique that is commonly used in global illumination. The algorithm shoots *lights* from each point on the surface in random directions and counts the number of occlusions by nearby features. The more occluded a location is, the darker it is shaded. In Fig. 17, it is clear that a cut should be put at the neck region whose ambient color is the darkest.

These features are present in EARS as supplementary procedures. A common concern is about their efficiency. Both procedures take a long time to compute, which is not desirable when real time interactivity is required. A future addition to EARS might use simplification to resolve the problem. First the mesh is simplified, then the algorithms are run on the reduced mesh, and then the result is mapped back to the original input. For Ambient Occlusion, GPU programming can be used to reduce the running time.

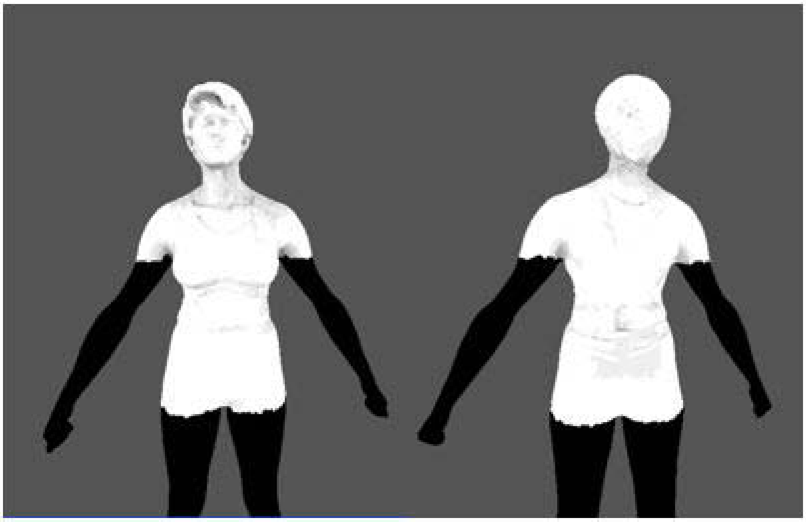


Fig. 17: (Left) Front view and (right) back view of a model whose *torso* is rendered using Ambient Occlusion. The neck area where the cut should be placed is darker than other parts of the body.

6. CURVATURE AND ROUGHNESS

The horizontal angle shift of the scanner sometimes causes certain areas of scanned surfaces to be noisy. One important task is to identify these rough areas. EARS computes a local roughness index for each vertex and ranks the vertices according to this measurement. A user defined threshold is needed for EARS to identify the *rough* regions.

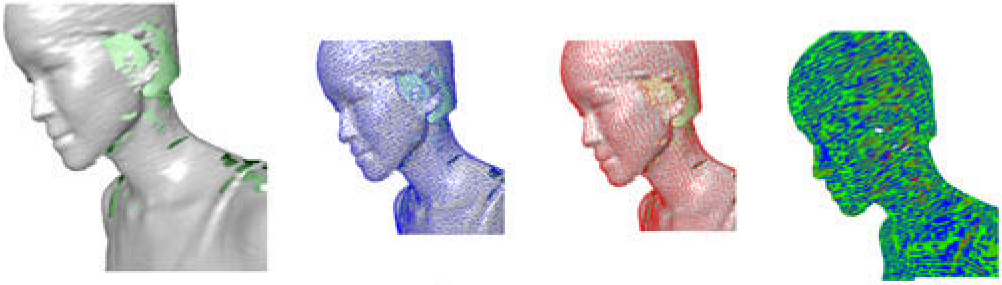
6.1 Curvature Evaluation

Roughness is closely related and largely based on an important differential geometry concept: the curvature. Curvature describes the second order derivatives (directional derivative of the normal) of a continuous 2-manifold surface and plays a central role in many geometry processing applications. At each vertex $\boldsymbol{P}$ in the surface $\boldsymbol{S}$, a normal curvature $\boldsymbol{k}_n$ is defined as the curvature of a planar curve that results from intersecting the surface with a plane spanned by vertex normal and a random vector $\boldsymbol{v}$ in the tangent plane at $\boldsymbol{P}$. The minimal and maximal of all normal curvatures are called minimal and maximal principal curvatures; the vector $\boldsymbol{v}$ that generate these curvatures are called minimal and maximal principal directions respectively. With triangle meshes, there are many algorithms (also referred to as discrete differential operators in differential geometry) proposed to compute curvature on a triangle mesh. There are two algorithms implemented in EARS: Normal Cycle [4] and Biquadratic Bezier Fitting [9]. Each of them represents one of two families that Curvature estimation algorithms fall into; one uses discretized formulas (Normal Cycle) and the other fits a parametric surface (Biquadratic Bezier) to the triangle mesh locally. Fig. 18 is generated using the Normal Cycle algorithm. Since curvature is sensitive to the noise, this is a hard task to achieve. On one hand, you want accurate curvature estimation to account for the geometry changes, yet the algorithm should be able to overcome inherent noise in the scanned surface.

It is shown in Fig. 19 that the latter algorithm generates better outcomes since the previous one is too sensitive to the surface quality. EARS uses Bezier method [9] as default because it deals with noisy data better.

6.2 Roughness Evaluation

In order to locate spikes and ripples such as on the sides of the thighs (as shown in Fig. 15), we compute roughness measure on each vertex. Vertex roughness [7] is computed in two steps. First, the k-ring (k >= 1) neighborhood vertices are collected. Second, EARS computes the difference between the vertex's curvature and the average curvature among the neighbors as described in Eqn. (6.1).



a b c d

Fig. 18: (a) Input sample mesh; (b) directions of kmin; (c) directions of kmax; (d) color coded mean curvature.

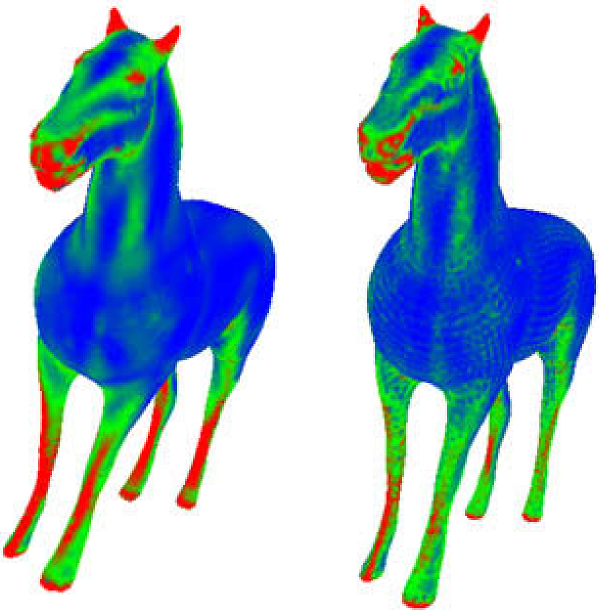


Fig. 19: (Left) Absolute curvature using Biquadratic Bezier Fitting [9]; (Right) Absolute curvature using normal cycle [4].

Here, the curvature can be any one of the following: mean, Gaussian, RMS, and Absolute curvature. Mean curvature is employed as the default because it is one of the most stable measurements. The spikes are defined as the vertices that have the highest roughness. They are selected using a threshold, which can be either one of three choices: percentage, for example the top 5% of the vertices on a mesh, specific cut-off value to force consistency in evaluation among meshes, or overall rank, for example the top 30 vertices in a mesh. In Eqn. (6.1), R_v is the roughness of target vertex *v*. N is its *k*-neighborhood. K_i is the curvature of vertex *i*.

$$R_v = K_v - \frac{\sum_{j \in N} K_j}{|N|} \tag{6,1}$$

Fig. 20 and Fig. 21 show some results of roughness computation. The elements that are identified as rough are highlighted in red. In Fig. 21, as we predicted, the rough regions on the side of the body are captured. These rough regions are caused not by the subject surface, rather by calibration problems in the scanner. To better catch these flaws, a study of a large dataset is needed in order to achieve the best threshold. The roughness computation also captures the wrinkles in clothing (scanwear).

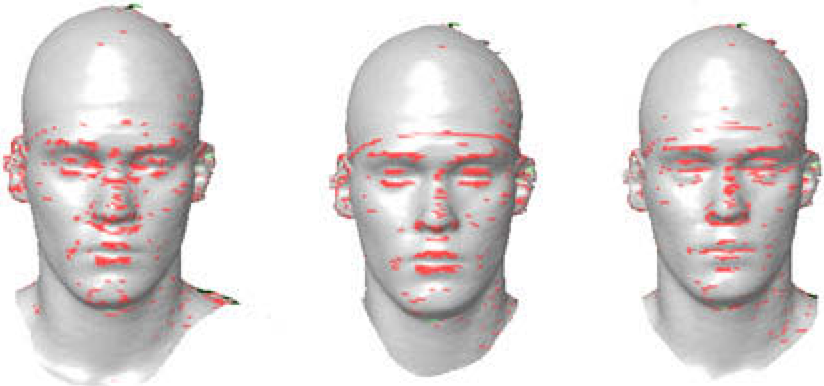


Fig. 20: Roughness using different types of curvature measurements: (left) Gaussian; (middle) mean; (right) RMS.

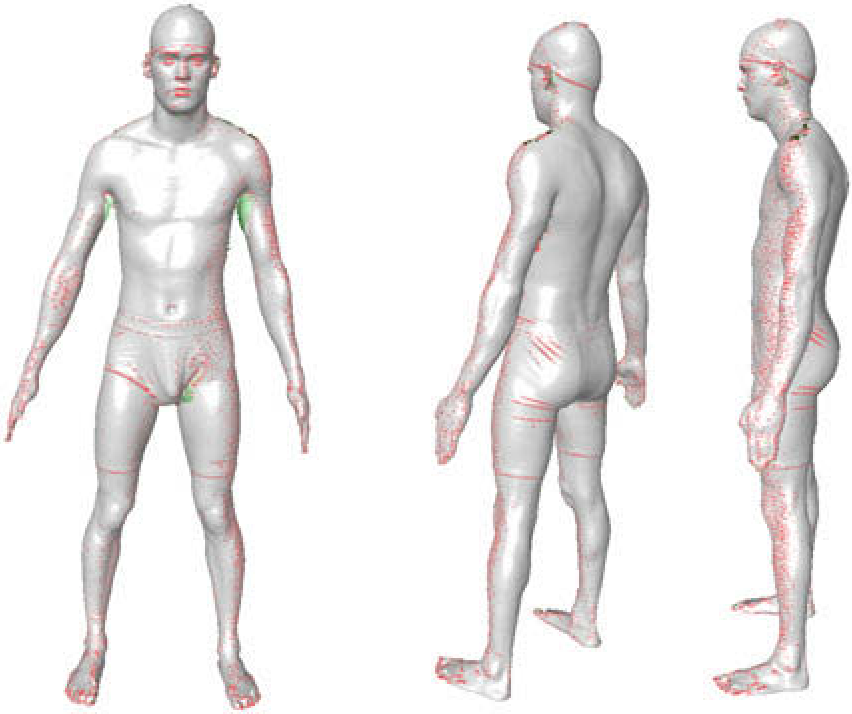


Fig. 21: Top 3% rough elements using mean curvature.

7. SYSTEM OVERVIEW

From the software engineering point of view, EARS is composed of three layers, as shown in Fig. 22. The lowest layer is consists of the infrastructure packages.

EARS infrastructure uses CGAL [4] for geometric representation, Boost libraries [2], whose C++ design helps improves the portability of the code, and TNT [9] for its linear algebra support. Built upon the infrastructure are many application algorithms (or procedures) from void filling to roughness computation. The top layer is the user interface. Two types of dialog bars are implemented: Read-only bars report system status and process result to the user; write-only bars take parameters from the user to customize the behavior of algorithms.

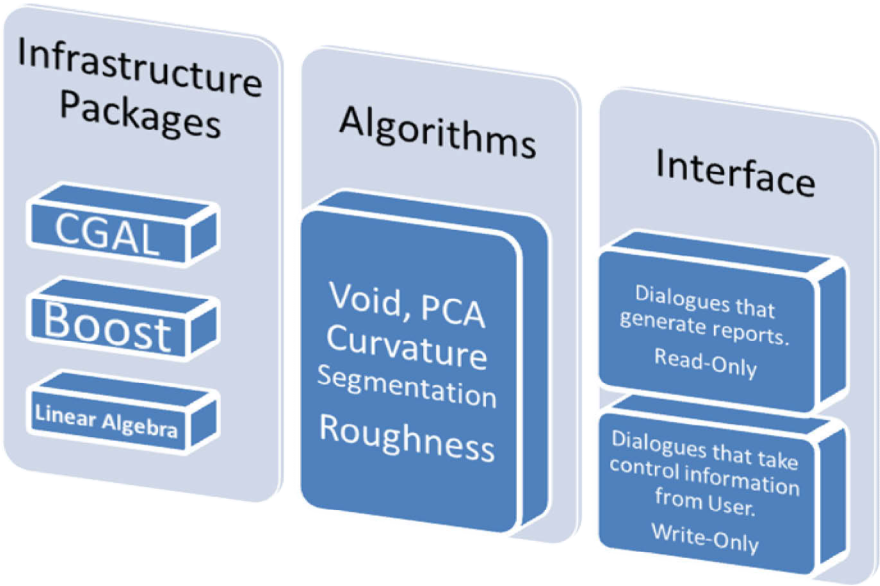


Fig. 22: The three layered EARS systems view.

7.1 Tunability

Most of the EARS algorithms and procedures are tunable via the interface. Tab. 1 summarizes the tunable arguments of each procedure.

Operation	*Arguments*
PCA	(1) Percentage of vertices used. (2) Box sizes used to identify orientation.
Segmentation	(1) Number of slicing planes per scan, range from 100 to 1000. (2) Acceptable ratio, range from 0.01 – 0.10. (3) Offsetting parameters used when head cut fails.
Curvature	Neighbor size
Roughness	(1) Neighbor size. (2) Curvature type. (3) Threshold.

Tab. 1: Tunable arguments.

7.2 Running Times

Most EARS algorithms and procedures are optimized. Because of the homogenous characteristic of the target input (most inputs are scans that have a similar number of elements), the running time is approximately the same. A sample runtime of EARS on a laptop (Processor: Intel(R) Core(TM) 2 Duo T7300 2.0GHz; RAM: 2.0GB) is summarized below (all times in seconds):

- Model loading, topology repair, normals computation: 4-5s
- PCA: 2-3s
- Segmentation: 3-4s
- Curvature: 2-3s
- Roughness: 5-6s

A sample running time display of EARS is shown in Fig. 23.

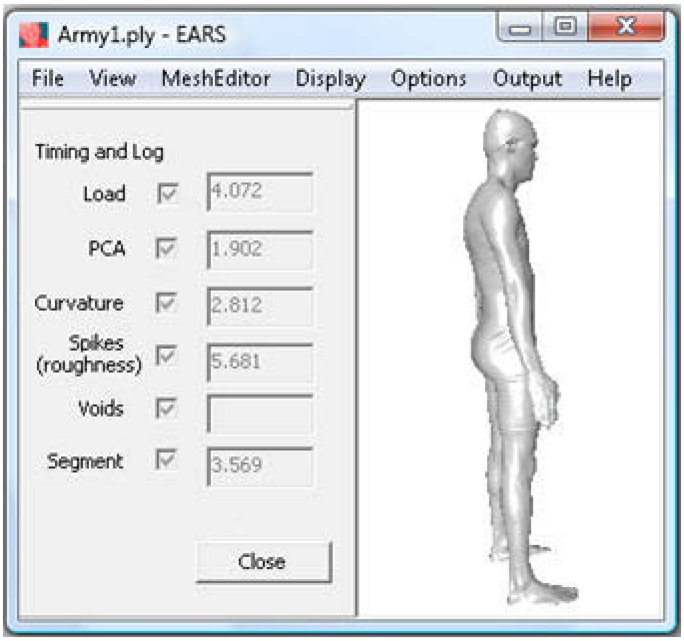


Fig. 23: Sample running time display

8. CONCLUSIONS

We present EARS, a prototype that integrates key algorithms for evaluating the quality of human body 3D scans. It provides the user a rating on how well the given mesh is formed and whether it is acceptable for downstream applications. The system operates in real time without human intervention; and its behavior and performance can be adjusted and customized. The three layer design of EARS software adds great extensibility to the system. For future improvement, automatic position identification and clothing identification can also be added. A study of a large dataset is needed for rough region identification; also, efficiency boost is needed for segmentation to work in real time.

ACKNOWLEDGEMENT

We would like to acknowledge the US Marine Corps Manager-Infantry Combat Equipment (USMC PM-ICE) for funding the work under grant SST-TCN-08007, and the U.S. Army Soldier Natick Research Development and Engineering Center for providing data.

REFERENCES

[1] Attene, M.; Katz, S.; Mortara, M.; Patane, G.; Spagnuolo, M.; Tal, A.: Mesh segmentation – a comparative study, Shape Modeling International, 2006, 14-25.
[2] Boost C++ Libraries, http://www.boost.org, Dawes, B; Abrahams, D.
[3] CAESAR project, http://www.sae.org/technicalcommittees/caesar.htm, SAE International.
[4] Cohen-Steiner, D.; Morvan, J.: Restricted Delaunay triangulations and normal cycle, Proceedings of the 19th Annual Symposium on Computational Geometry, 2003, 312-321.
[5] Computational Geometry Algorithms Library, http://www.cgal.org.
[6] Katz, S.; Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts, ACM Transactions on Graphics, 22(3), 2003, 954-961.
[7] Kushnapally, R.; Razdan, A.; Bridges, N.: Roughness as Shape Measure, Computer-Aided Design & Applications, 4(1-4), 2007, 295-310.
[8] Liepa, P.: Filling holes in meshes, Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2003, 200-205.

[9] Razdan, A.; Bae, M.: Curvature estimation scheme for triangle meshes using biquadratic Bezier patches, Computer-Aided Design, 37(14), 1481-1491.
[10] Template Numerical Toolkit, http://math.nist.gov/tnt/, U.S. National Institute of Standards and Technology.
[11] Xiao, Y.; Werghi, N.; Siebert, P.: A topological approach for segmenting human body shape, Proceedings of the 12th International Conference on Image Analysis and Processing, 2003, 82-87.
[12] Zhukov, S.; Iones, A.; Kronin, G.: An ambient light illumination model, In Rendering Techniques '98, 45-55.