

Skeleton Extraction of 3D Objects with Visible Repulsive Force

Fu-Che Wu, Wan-Chun Ma, Ping-Chou Liou, Rung-Huei Liang and Ming Ouhyoung

Communication and Multimedia Laboratory
Dept. of Computer Science and Information Engineering
National Taiwan University

Abstract

Skeleton should be the reduced object representation that conforms to human visual perception, and is very useful in applications such as object matching, mesh representation, computer animation, etc. A novel skeletonization algorithm to extract visually satisfactory skeleton from arbitrary 3D objects is proposed. The skeletonization can deal with arbitrary shape representations, such as polygonal models or parametric surfaces. In implementation, we choose triangulated models to represent 3D objects without losing generality.

The skeletonization algorithm contains two major steps. First, all the 3D model faces are regarded as charged planes. We initiate seed points with negative charges from the 3D model vertices. These seed points are pushed by electric-static force and they finally converge to local minimum positions. The force model we use here is called "visible repulsive force", which is the sum of all repulsive forces derived from the visible charged planes. In other words, faces that are invisible from the seed point do not contribute in the force model. Then, these local minimum positions are connected to complete the skeleton. Every connection in the skeleton is determined according to the neighborhood relationship defined by the 3D model edge connectivity information. Currently, we have implemented an object deformation system and the generated skeletons are used to be the deformation reference lines.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computation Geometry and Object Modeling

1. Introduction

Skeleton is widely adopted in areas such as object matching⁸, mesh representation⁵, computer animation²², collision detection and mesh editing²¹. Since skeleton conforms more to human perception than other shape descriptions do, it is suitable for human-interacted applications.

However, the requirements for a skeleton differ with applications. For example, object matching may need the skeletons to preserve principal features such as morphology (e.g. branching and termination) and geometrical information, so that we can regard these skeletons as major keys and use them to find object similarities. On the other hand, object reconstruction needs the skeleton that can store complete geometrical information of the original object to guarantee that we can use the skeleton to rebuild the object with minimal error.

A traditional skeleton definition is medial axis transform (MAT), which was proposed by Blum⁴. Medial axis transform in 3D is the locus of the inscribe spheres inside an object. MAT can describe the object surface precisely and it is appropriate for surface reconstruction. However, this characteristic is also a drawback: if the curvature of the object surface varies everywhere (such as the object contains many noise signals on the surface), MAT is going to produce many undesired skeleton branches¹² which is morphologically meaningless. Due to this reason, many previous studies focus on how to simplify the noisy MAT skeleton by methods such as pruning^{16, 18}.

In this paper, we present a skeleton extraction algorithm based on visible repulsive force (VRF), a special force field derived from visibility. For convenience, we name the extracted skeleton as visible domain skeleton, or VDS in short.

VDS is a one-dimensional skeleton that maintains primary morphology structure of the object. Therefore, it is a good representation for morphologically meaningful objects, such plants, animals and so on.

2. Previous Work

MAT skeleton has been widely studied for decades ^{6, 11, 15, 19, 20, 23}. Under current implementations, two approaches are used to approximate MAT: Voronoi diagram and distance field methods. Amenta et al. ² proposed a Voronoi diagram method called the power crust algorithm for MAT approximation and 3D surface reconstruction. It is easy to understand that MAT is the subset of the Voronoi diagram if the input object is a polygonal model or point set. The algorithm first computes the Voronoi diagram of a point set, which is sampled from a 3D object. Then it determines a set of polar ball, which takes the Voronoi vertex as center and the surface of the ball can touch the nearest sample points. By labeling the polar balls that are inside the object, the object now can be represented with these interior polar balls. Connecting the interior polar ball centers according to the connectivity of the Voronoi diagram cells forms a good approximation of the MAT.

Wade and Parent ²² presented a distance field method for automated control skeleton generation. The algorithm has several steps. First, a Euclidean distance map is computed to construct a discrete medial surface (DMS), which can be regarded as discrete MAT approximation. A path creation algorithm then generates a tree-structured set of voxel paths spanning along the DMS. Simplification of these paths leads to a control skeleton for the object. Their system produces reasonably good control skeletons for 3D objects. Zhou and Toga ²⁴ used a voxel-coding method, which is based on recursive voxel propagation, to generate 3D skeletons. The algorithm starts from a set of seed voxels, and then uses a coding schema to construct connectivity relationship and a distance field. Bitter et al. ³ proposed a penalized-distance algorithm to extract skeleton from volumetric data. The algorithm first uses a distance field and Dijkstra algorithm to locate an initial skeleton. Then, the skeleton is refined iteratively by discarding its redundant voxels. However, how to determine the voxel size in the voxelization process to preserve the fine feature of an object becomes a critical issue to all the voxel based approaches.

A comparatively diverse one-dimensional skeleton extraction method is to use the Reeb graph. The definition of the Reeb graph was introduced by Reeb ¹⁷. The idea of Reeb graph is to use a continuous function, usually a height function, to describe the topological structure and reveal the topological changes (such as merging or splitting). However, it is easy to realize that using a height function to construct Reeb graph does not guarantee the result to be affine-invariant, which is very important in shape analysis. Since we know that different functions may generate dif-

ferent Reeb graphs, Lazarus and Verroust ¹⁰ use this idea to construct Reeb graph skeleton by replacing the height function with a geodesic distance function which is derived from Dijkstra algorithm. They named the generated skeleton as level set diagram (LSD). Unfortunately, LSD is dependent on its source points, so different LSDs may be produced from the same model. Mortara and Patané ¹⁴ choose the source points in high curvature regions to guarantee the result is affine-invariant. In the aspect of real applications, Hilaga et al. ⁸ propose a technique called topology matching by comparing the multiresolutional Reeb graphs (MRGs) of 3D objects to match their similarities.

3. Problem Analysis

In our previous work ¹³, we tried to extract skeleton from 3D objects with radial basis functions (RBFs). RBF is used to build a distance field of a 3D object. We apply gradient descent algorithm to locate local extremes in RBF, which are branching or termination nodes in the generated skeleton. RBF is a continuous function and it is a good property to be able to use in gradient calculation. Nevertheless, how to construct an RBF to preserve the shape property of an arbitrary 3D model is still worth more consideration.

In this paper, we try to use a different method called VRF to locate local extreme positions. The VRF model is robust, easy to implement and most of all, suitable for all kinds of 3D object representation (polygonal model, parametric surface, etc.). If point a is visible to point b , we denote it as $a \rightarrow b$. For a point x which is in the interior of a surface S , we define the visible set $V(x)$ as

$$V(x) = \{v_i | v_i \rightarrow x, v_i \in S\}$$

Then, the VRF can be calculated as

$$\overline{VRF}(x) = \sum f(\|v_i - x\|_2) \cdot \overrightarrow{(v_i - x)} \quad (1)$$

where $v_i \in V(x)$. In here we take $f(r) = r^{-2}$ as the Newtonian potential function. One may imagine that all the 3D model faces are charged planes. Point x is a vertex charged with negative energy and enclosed by the 3D model. The point is pushed by electric-static force, which is the sum of all repulsive forces derived from the visible charged planes.

A visual domain skeleton (VDS) of S is defined as $D(S) = (Q, M)$, where Q is the set of local minima in VRF and M is a operator used to describe the topological relationship in Q . As previous discussed, it can be imagined that $V(x)$ is a visible charged surface and x is a negative charged point. All negative charged points would finally converge to stable positions in the force field. Therefore, we use a shrinking method similar to the gradient descent algorithm to locate these stable positions, which are also local minima. In Bitter's study ³, local extremes can be geometrically viewed as centers of the maximal inscribed spheres and this statement is consistent with the definition of MAT. We name these local minima as VDS nodes. Each point p_i on surface S converge

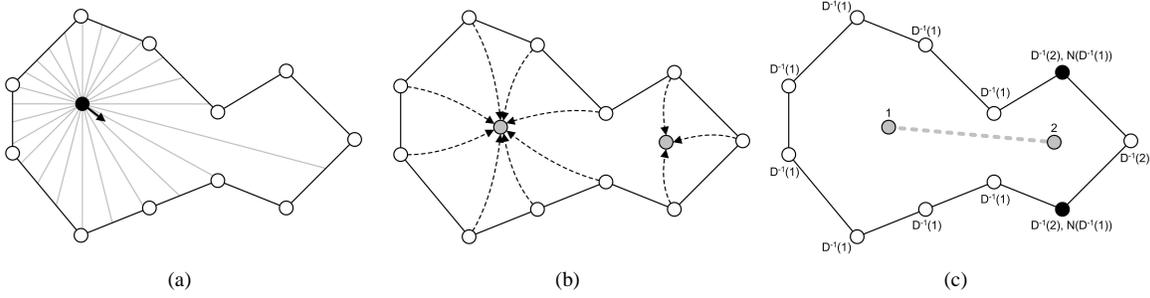


Figure 1: Illustrations of the skeletonization algorithm. (a) At point x (the black point), radial parameterization is used to sample the visible set. The black arrow represents the calculated VRF at that point. (b) For each vertex (in white color) on the model, apply the VRF shrinking procedure to it until a local minimum (in gray color) is reached. These local minima are regarded as VDS nodes. (c) Connect all adjacent VDS node pairs with the neighborhood relationship. The set of $M(1,2)$ is colored in black.

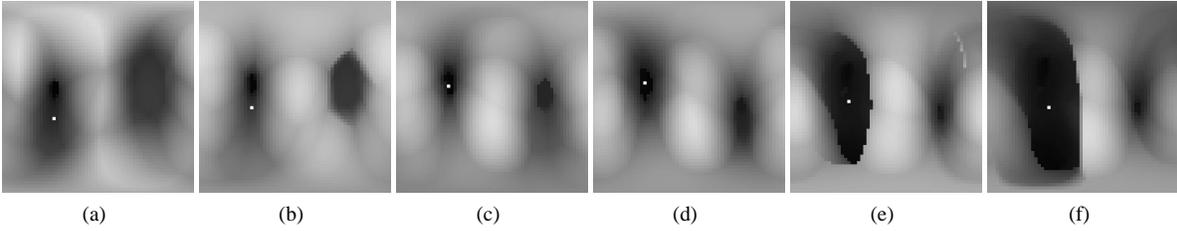


Figure 2: Radial parameterizations in the VRF local minima locating process. The images are generated by transforming radial parameterizations into 2D maps. One may imagine that these images are range images taken from the inside of an object (the lower the intensity, the longer the distance is). The coordinate of the white pixel in the map indicates the direction of the calculated VRF.

to a VDS node q_j , and we denote it as $D(p_i) = q_j$. Comparatively, for each VDS node q_j , we denote $D^{-1}(q_j)$ as a point set on the surface which belongs to q_j .

Connecting these VDS nodes according to their neighborhood relationship on the object surface completes the skeleton. If p_i and p_j are two adjacent points on the surface, we denote the neighborhood relationship N as $p_i \in N(p_j)$, and vice versa. Based on the neighborhood relationship, two VDS nodes q_i and q_j are adjacent and need to be connected if the set

$$M(q_i, q_j) = N(D^{-1}(q_i)) \cup D^{-1}(q_j) \quad (2)$$

is not empty.

4. Skeletonization Algorithm

Our skeletonization algorithm can be stated as follows:

1. For each seed point on the surface, apply the VRF shrinking procedure to it until a local minimum is reached.
2. Cluster the final positions of the seed points which are gathered in a controlled error distance to become one VDS node.

3. Connect all adjacent VDS node pairs with the neighborhood relationship.
4. Output the final graph as skeleton.

4.1. Locating VRF Local Minima

We use radial parameterization, which is shown in Figure 1(a), to sample the visible set. The visible set is acquired by calculating the intersection points of sampling rays and polygons in the model⁷. Once the visible set is obtained, the VRF can be derived by Formula 1. For each seed point p_i on the surface, we first assign the initial position by pushing p_i into the interior of the object toward the reverse surface normal direction, then apply the shrinking procedure:

$$p_{i+1} = p_i + \text{normalize}(\overrightarrow{VRF}(p_i)) \times \text{step}$$

where step is a small descent factor. Figure 2 shows the radial parameterizations of a seed point during the shrinking procedure. The iteration stops when

$$|VRF(p_{i+1})| > |VRF(p_i)|$$

and the final position of p_i is recorded as a VRF local minimum. The whole procedure stops when all of the points remain in corresponding local minima. Figure 1(b) illustrates

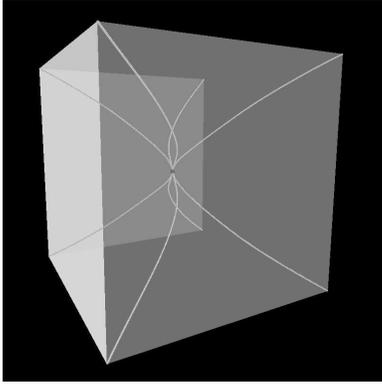


Figure 3: VRF local minimum locating process. Eight convergence paths and a VRF local minimum in the center of a unit cube model are shown in this figure.

the VRF local minimum locating process and Figure 3 shows the result of this procedure.

4.2. Clustering

In practice, the final positions of the seed points may be close to, but not exactly the same as where the local minima are. The error distance between the final position of a seed and its corresponding local minimum depends on the precision of the descent step. Due to this reason, we merge the final positions of the seed points in a given range to become one single VDS node.

4.3. Connectivity Information

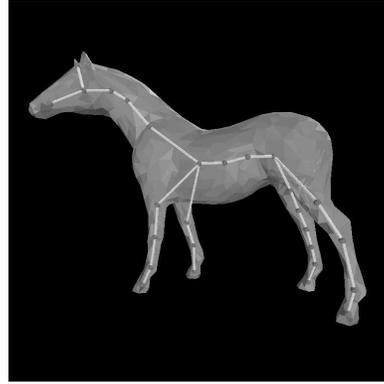
As we use the vertices of the 3D model to be initial seed points, it is intuitive that the edges of a model become the neighborhood relationship. As discussed in Formula 2, for two VDS nodes q_i and q_j , they should be connected if $M(q_i, q_j)$ is not empty. We iteratively add edges until all adjacent VDS node pairs are connected. Sometimes a small amount of undesired connections may be produced and currently we just remove them manually. Figure 1(c) illustrates the VDS node connection process.

4.4. Time Complexity Analysis

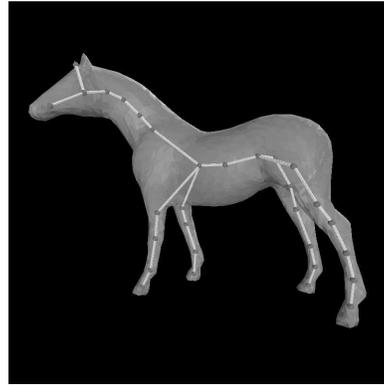
In the skeletonization process, the most time-consuming procedure is to locate VRF local minima. In this procedure, the time complexity should be:

$$O\left(\frac{n \times s \times t}{k}\right)$$

where n is the number of shrinking points, s is the visible set sampling number, t is the number of faces in input model, k is the stepping descent factor. By using an Octree structure to speed up the calculation of visible set sampling we may



(a)



(b)

Figure 4: (a) VDS extracted from a model with 1,996 faces. (b) VDS extracted from the same model with 7,984 faces. The two skeletons are similar in appearance.

reduced the variable t to $\log(t)$. Therefore, the complexity can be reduced to:

$$O\left(\frac{n \times s \times \log(t)}{k}\right)$$

The other two procedures, clustering and node connecting, can be implemented in constant time with auxiliary storage structure.

5. Results

Six VDSs extracted from different models are shown in Figure 6 in colors. The execution statistics for each model is shown in Table 1. Significant shape features are preserved well in the skeleton representation. Since we invoke radial parameterization to sample the surface, two models may have similar VDSs regardless of the number of faces if they are similar in shape. Figure 4 reveals this feature.

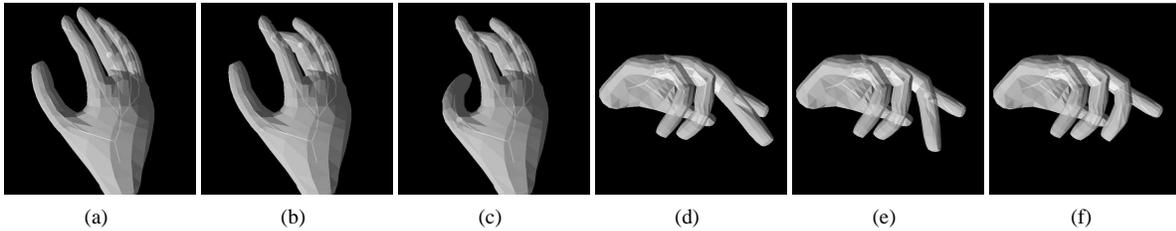


Figure 5: Deformation sequence of a hand model using the proposed visible domain skeleton (VDS).

Model	Face number	Vertex number	Skeletonization (sec.)	Total VDS nodes	Total VDS edges
Mouse	1,890	954	125	35	34
Bull	1,978	991	137	30	29
Octopus	2,000	1,002	53	68	60
Cobra	1,715	876	49	36	35
Frog	1,884	945	76	29	28
Horse	1,996	1,000	170	27	26
Horse (complex)	7,984	3,994	1,028	30	29

Table 1: Execution statistics on a desktop computer with Intel Pentium-4 processor running at 1.5 GHz and with 512 Mbytes RAMBUS memory. The visible set sampling number s is 197 and the stepping descent factor k is 0.0025. All models are scaled in a unit cube.

6. Conclusions and Future Work

In this paper, we propose an algorithm to extract skeleton from arbitrary 3D objects. The generated VDS maintains both topological and morphological information. We may further apply an active contour model⁹ to the skeleton in order to generate a deformed VDS for preserving the geometry of the object¹³. Currently a simple object deformation system has been implemented and it uses the VDS as the deformation reference. It shows that the VDS is potential to be applied to the computer animation applications. Figure 5 shows a deformation sequence of a hand model.

There are still many work need to be done. First one, and is also the most important thing, is to fully automate the skeletonization process. At the moment we need to remove undesired connections by hand. We hope that this problem can be solved in the future. Second, how to speed up the process is also a critical concern. Currently, if the skeletonization process takes large models (e.g. 10,000 faces) as inputs, it takes more than 20 minutes to complete the execution. As over 95% of the execution time is consumed by locating the VRF local minima, how to speed up the visible set sampling becomes a critical issue. Now we try to use Z-buffer (hardware acceleration) to calculate the VRF.

It is anticipated that VDS can be used in many areas. Take surface reconstruction as example. Since we may sample the visible set at each VDS node and store the visibility map, reconstructing the surface is simply done by recalling the surface points from these visibility maps. One may also regard VDS as a weighted connected graph; each node in the graph has different weight value, which represents the volume in the object that belongs to the node. By matching two weighted graphs, we may find the morphology similarities between these two objects.

7. Acknowledgements

We thank 3DCAFE¹ to provide all 3D models used in this paper. This work was partially supported by grants from National Science Council, Taiwan, R.O.C. under NSC91-2213-E-002-066 and Silicon Integrated Systems (SiS) Education Foundation.

References

1. <http://www.3dcafe.com>
2. N. Amenta, S. Choi, and R. Kolluri. The Power Crust.

- Proceedings of the ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001.
3. I. Bitter, A. E. Kaufman, and M. Sato. Penalized-Distance Volumetric Skeleton Algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):195–206, 2001.
 4. H. Blum. *A Transformation for Extracting New Descriptors of Shape*, pages 362–380. MIT Press, 1967.
 5. D.-T. Chen, S. M. Pizer, and A. L. Thall. M-reps: A New Object Representation for Graphics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–643, 1994.
 6. T. Culver, J. Keyser, and D. Manocha. Accurate Computation of the Medial Axis of a Polyhedron. *Proceedings of the ACM symposium on Solid Modeling and Applications*, pages 179–190, 1999.
 7. D. Badouel. An Efficient Ray-Polygon Intersection. *Graphics Gems I*, pages 390–393. Morgan Kaufmann, 1990.
 8. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. *SIGGRAPH 2001 Conference Proceedings*, pages 203–212.
 9. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1:321–331, 1987.
 10. F. Lazarus and A. Verroust. Level Set Diagrams of Polyhedral Objects. *Proceedings of the ACM Symposium on Solid Modeling and Applications*, pages 130–140, 1999.
 11. F. Leymarie and M. Levine. Simulating the Grass-fire Transform Using an Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):56–75, 1992.
 12. M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38:327–341, 1987.
 13. W.-C. Ma, F.-C. Wu, and M. Ouhyoung. Skeleton Extraction of 3D Objects with Radial Basis Functions. *Proceedings of Shape Modeling International 2003* (to appear).
 14. M. Mortara and G. Patané. Affine-Invariant Skeleton of 3D Shapes. *Proceedings of Shape Modeling International 2002*, pages 245–278, 2002.
 15. L. R. Nackman and S. M. Pizer. Three-Dimensional Shape Description Using the Symmetric Axis Transform I: Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):187–202, 1985.
 16. R. Ogniewicz. Automatic Medial Axis Pruning by Mapping Characteristics of Boundaries Evolving Under the Euclidean Geometric Heat Flow onto Voronoi Skeletons. *Harvard Robotics Laboratory Technical Report*, 95–4, 1995.
 17. G. Reeb. Sur les points singuliers d’une forme de Pfaff complètement intégrable ou d’une fonction numérique. *Comptes Rendus Acad. Science Paris*, 222:847–849, 1946.
 18. D. Shaked and A. M. Bruckstein. Pruning Medial Axes. *Computer Vision and Image Understanding*, 69(2):156–169, 1998.
 19. D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Shape Description By Medial Surface Construction. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):62–72, 1996.
 20. E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. An Algorithm for the Medial Axis Transform of 3D Polyhedral Solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):44–61, 1996.
 21. T.-S. Tan, X.-T. Li, T.-W. Woon, and Z.-Y. Huang. Decomposing Polygon Meshes for Interactive Applications. *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 35–42, 2001.
 22. L. Wade and R. E. Parent. Automated Generation of Control Skeletons for Use in Animation. *The Visual Computer*, 18(2):97–110, 2002.
 23. F.-E. Wolter. Cut Locus and Medial Axis in Global Shape Interrogation and Representation. *MIT Sea Grant Report*, 1992.
 24. Y. Zhou and A. Toga. Efficient Skeletonization of Volumetric Objects. *IEEE Transaction on Visualization and Computer Graphics*, 5(3):195–206, 1999.

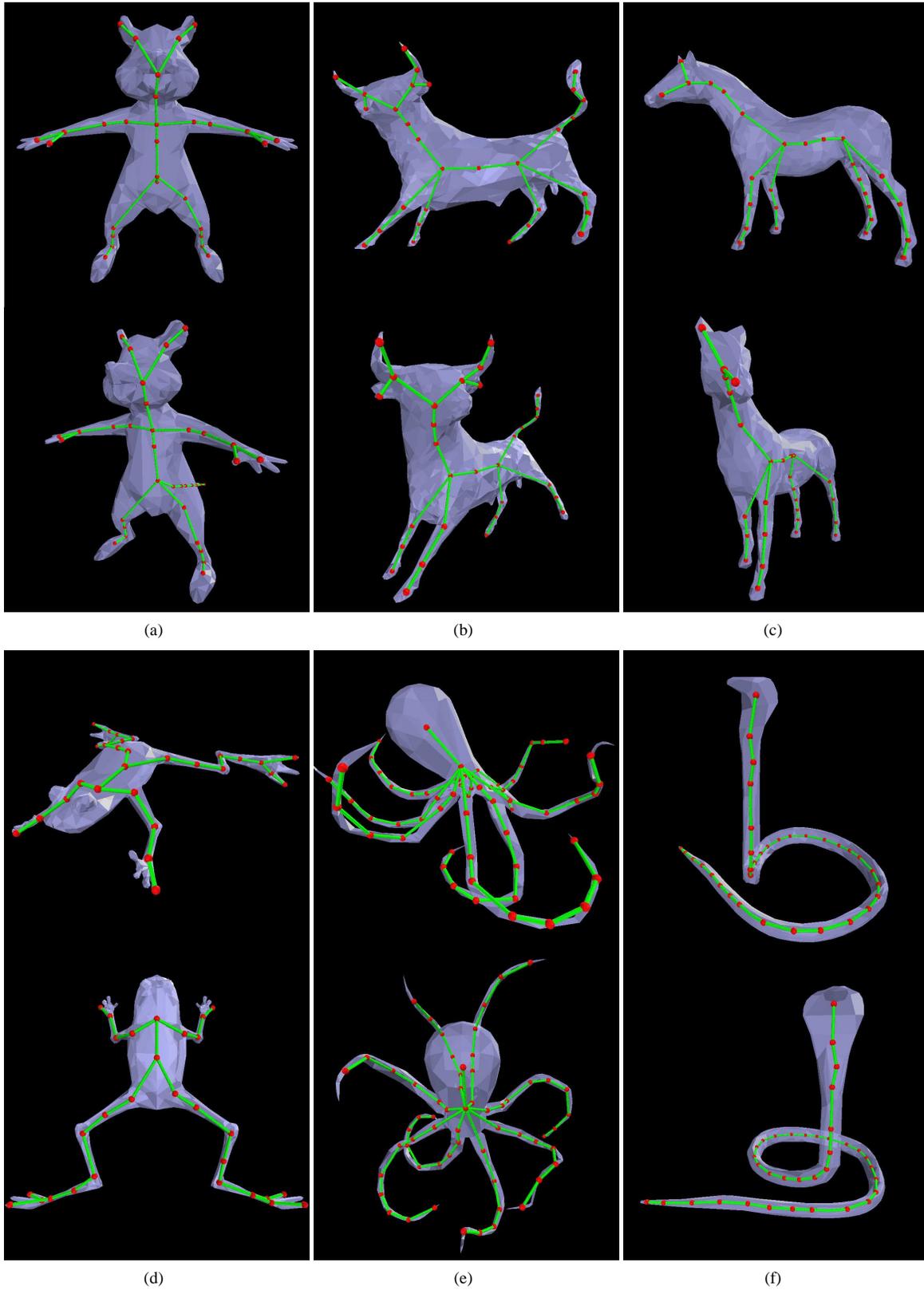


Figure 6: Skeletons extracted from 3D models. (a) Mouse, (b) Bull, (c) Horse, (d) Frog, (e) Octopus and (f) Cobra.
© The Eurographics Association 2003.