Curve-skeleton extraction using silhouettes' medial axes

Andrey Zimovnov, Leonid Mestetskiy
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
zimovnov@gmail.com, mestlm@mail.ru

Abstract

A new method of curve-skeleton extraction is proposed in this paper. Unlike most methods this approach retrieves information from a polygonal mesh via rasterization and extraction of silhouettes' continuous medial axes. Described method shows a great improvement in computational time comparing to existing methods. Method shows good extraction results for models with complex geometry and topology. Resulting curve-skeletons conform with most requirements to universal curve-skeletons.

Keywords: curve-skeleton, medial axis, visual hull, distance transform, mean-shift

1 Introduction

Curve-skeleton is a graph that depicts simplified versions of object's geometry and topology. An inscribed sphere is associated with every node of curve-skeleton. The envelope of all curve-skeleton spheres approximates the shape of the object. Curve-skeletons find many applications in problems where object's shape analysis is needed, such as object recognition, shape classification, object skeletal animation, object segmentation, finding visually similar objects in databases and others [Cornea et al. 2005].

Most existing methods analyze polygons or voxels in space [Cornea et al. 2005], which leads to slow extraction process. In paper [Mestetskiy and Tsiskaridze 2009] authors utilize the observation that on a projection of object without occlusions silhouette's medial axis is the projection of curve-skeleton's bones. Although object's occlusions lead to spurious curves in resulting skeleton, experiments [Livesu et al. 2012] show that even with occlusions projections store enough information for curve-skeleton extraction.

In this paper a new approach to curve-skeleton extraction is proposed. Our method uses continuous medial axes extraction from object's silhouettes [Mestetskiy and Semenov 2008] to organize an efficient iterative contraction process similar to [Au et al. 2008].

2 Proposed method

In this paper we assume that object is represented by a polygonal mesh. Proposed method is an iterative process. One iteration can be divided into the following steps:

- Curve-skeleton approximation with point cloud reconstructed from silhouettes' medial axes.
- Object contraction based on inscribed spheres radii reduction associated with every point of the approximating cloud.

2.1 Curve-skeleton approximation with point cloud

Proposed method is based on the observation that on a projection of object without occlusions silhouette's medial axis is the projection of curve-skeleton's bones. Assuming there're no occlusions on a projection, curve-skeleton can be reconstructed by medial axis back-projecting. To eliminate the influence of object's occlusions many projections are used. The main idea is to back-project object's parts that are visible without occlusions and filter out spurious occluded parts. Orthogonal projections are captured from uniformly distributed cameras on a sphere around the object (Fig. 1). Uniformly distributed points on a sphere are generated using hexahedron subdivision. Filtered parts acquired from different projections are combined into one graph and aligned.

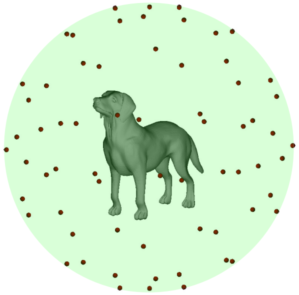

Figure 1: *Uniformly distributed cameras on a sphere.*

To sum up, the approximation process can be described with the following steps:

- Back-projection of silhouettes' medial axes
- Nodes and bones filtering in space
- Alignment of skeleton nodes and bones from different projections.

2.1.1 Back-projection of silhouettes' medial axes

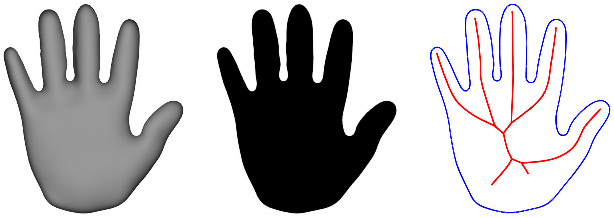

Figure 2: *Object's projection, silhouette and medial axis.*

For each silhouette medial axis is a continuous graph [Mestetskiy and Semenov 2008] with nodes and bones (Fig. 2). In order to back-project the node v_i of medial axis a ray is casted from the node towards camera view and two intersections with object are saved: $x_1^{(i)}$ — the first intersection with front-facing polygon and $x_2^{(i)}$ — the first intersection with back-facing polygon (Fig. 3). Calculation is done on GPU via depth buffer and culling, so there's no need to find intersections analytically.

Now the depth of curve-skeleton node V_i can be estimated as

$\frac{1}{2}[x_1^{(i)} + x_2^{(i)}]$. Such depth calculation helps to back-project the nearest to the camera parts of the object even with heavy occlusions.

The maximum inscribed ball radius R_i with the center in V_i can be estimated as follows:

$$R_i = \min \left[r_i, \frac{|x_1^{(i)} - x_2^{(i)}|}{2} \right],$$

where r_i is the radius of maximum inscribed circle with the center in v_i. Such estimation is robust due to implicit measuring of radius R_i in two orthogonal directions.

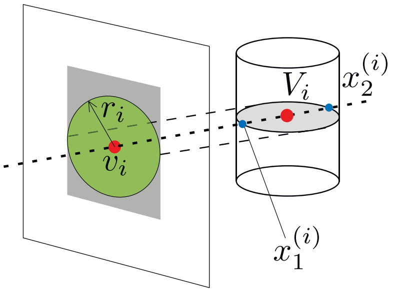

Figure 3: *Back-projection of the node v_i of medial axis in order to get curve-skeleton node V_i.*

2.1.2 Nodes and bones filtering in space

Occlusions in silhouettes produce a huge number of spurious nodes and bones (Fig. 4).

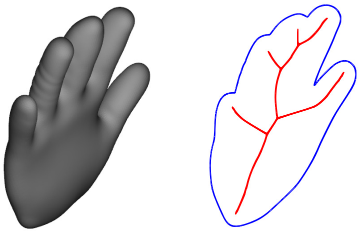

Figure 4: *Projection with occlusions and its medial axis. Medial axis doesn't represent object geometry well due to occlusions.*

Nodes filtering The basic idea of nodes filtering comes from curve-skeleton definition: this graph should reproduce object's shape with the set of maximum inscribed balls in its nodes. Thereby a node is considered spurious if a corresponding inscribed ball is not inside the object. For efficient nodes filtering Visual Hull is used: the object is approximated with an intersection of prisms with object's silhouettes in their bases. A strict definition of Visual Hull (VH) is as follows:

$$VH = \{x \in \mathbb{R}^3 : \forall i \ \mathrm{Pr}_i(x) \in S_i\},$$

where $\{S_i\}$ — set of object's silhouettes, $\mathrm{Pr}_i(x)$ — orthogonal projector of point $x \in \mathbb{R}^3$ on the plane of silhouette S_i.

To check whether a sphere is inside the Visual Hull, Distance Transforms (DT) of silhouettes are computed [Felzenszwalb and Huttenlocher 2004]. Thus to check a sphere on a single silhouette we need to compare its radius with the minimum distance from its projected center to the silhouette border acquired with DT. If the sphere radius is less or equal to DT values throughout all silhouettes from $\{S_i\}$, sphere is inside VH and the corresponding node is considered relevant. Otherwise, the node is deleted from resulting graph.

The resulting approximation of object with inscribed balls is much better after filtering (Fig. 5).

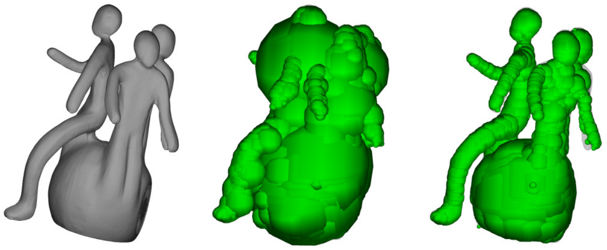

Figure 5: *An object, inscribed balls from step 2.1.1 and filtered inscribed balls.*

Bones filtering Occlusions lead to spurious bones after back-projecting. These bones usually stand out of the model due to occlusions of object's parts. Suggested method deletes bones that have small angle ($\leq \phi_0$) with corresponding camera view direction. This approach might delete relevant curves, but due to usage of the set of projections this curve will likely be captured from another projection.

The example of filtering result is shown in Fig. 6.

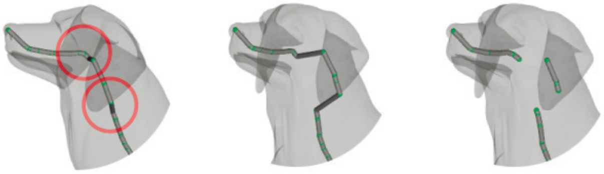

Figure 6: *Skeleton back-projection, back-projection from different view, filtered bones.*

2.1.3 Alignment of skeleton nodes and bones from different projections

Even without occlusions same part of the object on different projections can be back-projected into centered, but quite different curves. This happens when the object's cut is not a circle (Fig. 7). We suggest to organize an alignment process to join nodes and bones from different projections and to remove such noise.

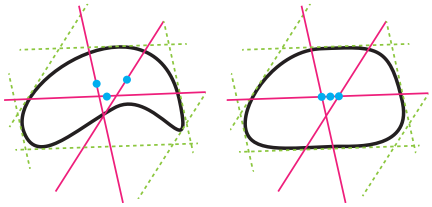

Figure 7: *Inconsistent back-projecting throughout different projections due to the concavity of the object's cut or its elliptic shape.*

At this stage nodes and bones are centered in the model and look like a very noisy curve-skeleton. The basic idea behind alignment is noise canceling by finding resulting nodes with the highest density of neighbors. For this purpose Mean-Shift algorithm is used, that iteratively moves every node from starting position to the average position of its neighbors. When the node stops to significantly change its position, the node is replaced with the final average. The algorithm doesn't take into account the skeleton bones, that are automatically shifted with adjacent nodes.

The following algorithm describes this approach:

Require: X — input nodes, ε — minimum shift threshold, α — minimum density of neighbors threshold, $K(x)$ — kernel function;

Ensure: Y — shifted nodes;

```
$Y := \varnothing$;
for $x_i \in X$ do
    $m := x_i$; {start Mean-Shift from $x_i$}
    repeat
        $m_{old} := m$;
        $m := \frac{\sum_{x_j \in X} K(x_j - m_{old}) x_j}{\sum_{x_j \in X} K(x_j - m_{old})}$;
    until $\|m - m_{old}\| \geq \varepsilon$
    if $\sum_{x_j \in X} K(x_j - m) > \alpha$ then
        $Y := Y \cup \{m\}$;
    end if
end for
```

The alignment algorithm is illustrated in case of points on plane in Fig. 8. There's an obvious drawback of such approach: the ending points of the line travel too far. We suggest to use a heuristic that forbids ending points of medial axis (which is a graph) to travel too far.

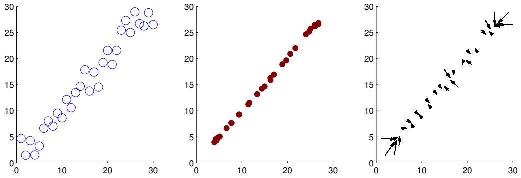

Figure 8: *The alignment algorithm for points on plane. From left to right: set of points, aligned points, travel paths.*

The result of such alignment is shown in Fig. 9. The result is much cleaner and looks like a good curve-skeleton approximation.

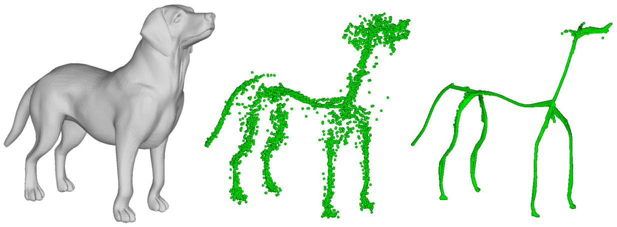

Figure 9: *An object, back-projecting and filtering result, aligned nodes and bones.*

2.2 Iterative contraction

We suggest to implement object contraction through spheres of curve-skeleton approximation radii reduction. Point cloud from section 2.1 approximates curve-skeleton and thus is centered and describes shape quite well, hence spheres radii reduction leads to visually correct object contraction (Fig. 10).

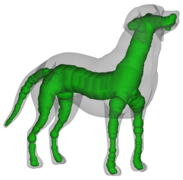

Figure 10: *Object contraction through spheres radii reduction.*

Contraction makes the object thinner, which helps to get rid of occlusions, thus making point cloud approximation better through iterations. Fig. 11 illustrates 4 iterations of object contraction. We can see that point cloud approximation of curve-skeleton is sufficient for object contraction.

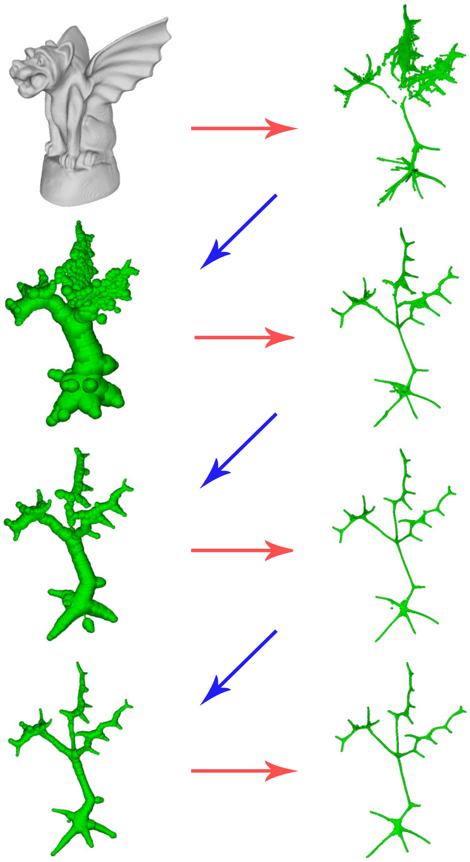

Figure 11: *4 iterations of object contraction. Red arrows depict curve-skeleton approximation with point cloud, blue arrows depict object contraction based on approximation.*

3 Experiments

Experiments were conducted on the computer with Core i5-3570K 3.4 GHz CPU, 8 GB RAM, Intel HD Graphics 4000 GPU. For continuous skeleton extraction library [Mestetskiy and Semenov 2008] was used.

Method used 66 projections with resolution 300×300. Bones filter was used with $\phi_0 = 45^\circ$. Alignment of nodes and bones was used with ε equal to the height of one pixel in model coordinates, α equal to 5%–10% percentile, kernel function $K(x) = \exp(-C\|x\|^2)$ with $C = 100$.

Table 1 shows timings in milliseconds for different stages of algorithm for different models. Due to the fact that method doesn't deal with polygons in space, model complexity (number of polygons) has little effect on extraction time.

Table 1: *Timings in milliseconds for different stages of algorithm.*

Model	Faces in model	Medial axes	Back-project	Filters	Mean-Shift	Total
Fertility	50,000	129	405	153	484	1,171
Memento	52,550	152	388	145	957	1,642
Horse	7,951	144	328	185	1,193	1,850
Elk	48,026	160	381	195	602	1,338

Fig. 12 shows curve-skeletons for different models. The resulting curve-skeletons depict models' geometry and topology well.

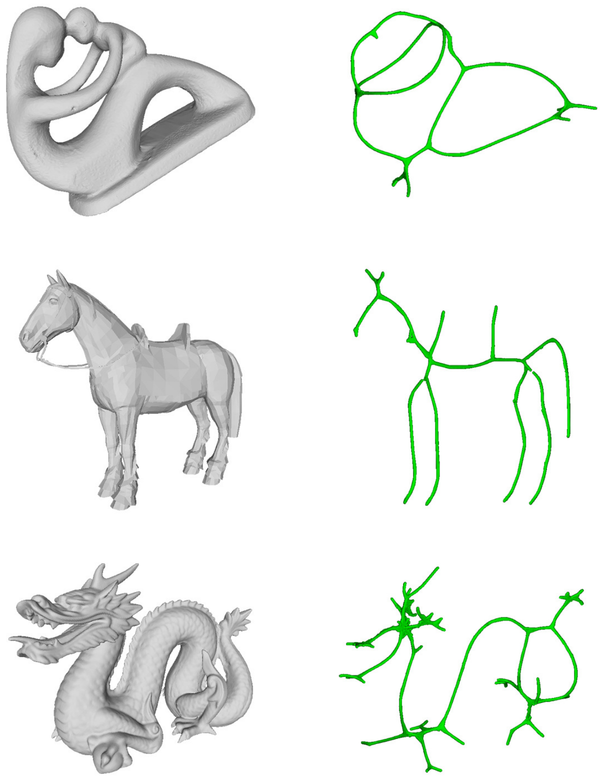

Figure 12: *Examples of curve-skeletons extracted with described method.*

4 Method analysis

Resulting curve-skeletons conform with most requirements to universal curve-skeletons based on research [Cornea et al. 2005]. This is achieved using filters, mean-shift and inherited properties of medial axes like centeredness and isometric invariance.

Compared to [Au et al. 2008] method shows 6 times increase in extraction speed (in average 3 sec. vs 19 sec. for different models), though it's hard to compare the quality of resulting skeletons due to poor formalization of the problem [Dey and Sun 2006].

Method looks ideal for parallel implementation as most stages can be done on GPU, which promises a huge decrease in computational time and possibly new real-time applications for curve-skeletons.

Method fails to extract curve-skeletons for heavily occluded objects with parts not visible without occlusions. Fortunately, most applications of curve-skeletons assume that we have a character-like object with significant visual branches.

5 Conclusion

A new method of curve-skeleton extraction is proposed in this paper. Unlike most methods this approach retrieves information from a polygonal mesh via rasterization and extraction of silhouettes' continuous medial axes. The usage of continuous medial axes reduces computational time and gives an ability to analyze medial axis as a graph. Described method shows good extraction results for models with complex geometry and topology.

Acknowledgements

The work was partially supported by RFBR, research project No. 12-07-31107-mol_a.

References

AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 44:1–44:10.

CORNEA, N. D., SILVER, D., AND MIN, P. 2005. Curve-skeleton applications. In *IEEE Visualization*, IEEE Computer Society, 13.

DEY, T. K., AND SUN, J. 2006. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '06, 143–152.

FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. 2004. Distance transforms of sampled functions. Tech. rep., Cornell Computing and Information Science.

LIVESU, M., GUGGERI, F., AND SCATENI, R. 2012. Reconstructing the curve-skeletons of 3d shapes using the visual hull. *IEEE Trans. Vis. Comput. Graph. 18*, 11, 1891–1901.

MESTETSKIY, L., AND SEMENOV, A. 2008. Binary image skeleton - continuous approach. In *VISAPP (1)*, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, A. Ranchordas and H. Araújo, Eds., 251–258.

MESTETSKIY, L., AND TSISKARIDZE, A. 2009. Spatial reconstruction of locally symmetric objects based on stereo mate images. In *VISAPP (1)*, INSTICC Press, A. Ranchordas and H. Araújo, Eds., 443–448.