

Concept of Skeleton Texture Mapping

M. Madaras

madaras@fmph.uniba.sk
Comenius University Bratislava, Slovakia

Abstract

This article introduces an idea for novel way of mapping textures onto a surface of 3D model. Our technique is based on two interlocking mappings, the first one maps surface vertices onto computed skeleton and the second maps surrounding area of each skeleton segment into a rectangle with dimensions based on surface properties around this segment. Furthermore, these rectangles are packed into a squared texture by approximately solving an optimization problem which seems to be NP-complete. With our technique, we are able to map the texture onto the surface without any precomputed or stored texture coordinates. Our texture mapping approach is also suitable for surfaces with topology non-homotopic to a sphere.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Texture mapping is a commonly used and the most successful technique of improving visual quality of 3D surfaces in computer graphics. Problem of texture mapping can be formulated as retrieving a pair of texture coordinates for each surface vertex. Some techniques have been developed for automatic mapping of textures on complex surfaces as unfolding the polygon mesh, two-part texture mapping [BS86] or particle system based approaches. These methods might have serious mapping and texture distortion problems with complex surfaces when they are not homotopic to a sphere.

2. Skeleton Texture Mapping

Mapping between the surface and the skeleton works in a deterministic way and the benefit is that storing of uv coordinates is not needed. In the first stage, skeleton is extracted from an input mesh. As algorithm for skeleton extraction [ATC*08] is used. The algorithm extracts the skeleton from closed 2D-manifold mesh using iterative Laplacian contraction. If the input surface is with boundaries, polygon soup or point cloud, we can use adaptation of this algorithm [CTO*10], which can handle such surfaces. During the skeleton construction we store the mapping (i.e. which mesh vertices were collapsed into which skeleton node). Furthermore, we compute and store some surface proper-

ties of collapsed vertices into corresponding skeleton node. These properties serve for priority weight estimation. Priority weights encode, how important the segment is in a sense of level of detail and how much of space it requires in the final texture. To avoid aliasing and waxy look of textures the filtering has to be applied in texture mapping. For each texel within rectangles we have to guarantee that it has well defined local neighborhood. We do this in such a way that each rectangle is enclosed in mirrored parts of itself.

3. Segment Mapping and Packing Problem

Each skeleton segment is mapped into a rectangle as shown in Figure 1 and relative texture coordinates to the rectangle origin are then computed as in Equations 1 and 2. For each skeleton segment priority weights are computed as weighted linear combination of parameters as number of polygons, surface area ratio, curvature and the segment length, which are used to determine parameters as distance d and dimensions of the rectangle w, h

$$u = \frac{\theta}{2\pi}h \quad (1)$$

$$v = \begin{cases} d(1 - \frac{\alpha}{\pi/2}) & t < 0 \\ d + t(w - 2d) & t \in [0, 1) \\ w + d(\frac{\beta}{\pi/2} - 1) & t > 1. \end{cases} \quad (2)$$

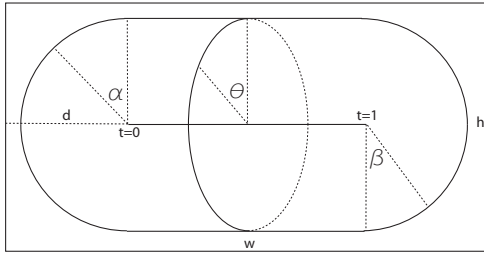


Figure 1: Skeleton segment parameterization between two nodes. Coordinates are computed from parameters d, w, h and pairs of angles θ, α and θ, β which correspond to spherical coordinates of the mapped vertex.

Rectangle with higher priority needs better storing of details, hence it will be stored in bigger area than rectangle with lower priority. Determining relative size ratio between the rectangles, we can formulate the storing of these rectangles as a packing problem of storing N rectangles with dimensions (R_i^W, R_i^H) into a unit square. We maximize the sum of area they cover (3) and we can scale them by an arbitrary constant $s \in \mathbb{R}^+$

$$\forall s \in \mathbb{R}^+ : \max \sum_{i=0}^{N-1} s R_i^W R_i^H \leq 1. \quad (3)$$

We find the best packing of rectangles within the square texture by exploring all acceptable configurations. We use a binary search to find the correct scaling ratio s to fit the rectangles. We explore configurations for assigned s and if there is an acceptable configuration, we increase s and iterate again. If there is not next acceptable configuration, we take the last one as the solution.

4. Applications and Future Work

We see some useful applications of our approach in computer graphics. We define here two texture types, which are further used in proposed applications:

Skeleton Texture Map (STM) is an arbitrary texture, where uv coordinates are computed by our mapping technique and encode texels around skeleton segments. To map STM onto the surface we need either an input mesh to compute the skeleton, or the skeleton itself. The mapping can be used for extracting STM from textured mesh with classical uv coordinates or applying STM onto a surface without parameterization.

Skeleton Displacement Map (SDM) is a special type of STM. This texture encodes vertex displacements from skeleton. SDM can be used as a data structure for model representation. The skeleton and SDM are extracted from an input

and they are later used for reconstruction of the model surface.

Recently, polygonal meshes are either of high resolution or low resolution with supporting textures for detail enhancement as normal, bump or displacement maps. The idea is, if we have such a detailed displacement map, how much can a low resolution mesh be downsampled. Using SDM, it can be downsampled to the skeleton level. It encodes models topology and the original surface can be reconstructed from the displacement map.

An application of STM concerns mapping of procedurally generated textures onto the surface. If procedural textures enhancing models surface are generated, we need acceptable texture mapping to apply them. Using STM has a few advantages. Textures can be aligned by skeleton axis, skeleton can store important surface parameters and apply them during seamless mapping.

Furthermore, STM can be used for seamless texture space diffusion. For example, in [dLE07] a texture space diffusion is used for real-time approximation of subsurface scattering. If the topology is complex and there are non-occupied areas in texture space, there come problems with seams during applying the blur. And that is the place, where our method can be used. The STM should be generated in such a way, that rectangle encoding each segment is enclosed in mirrored parts of itself. These parts have to be only as large as the radius of the widest blur operator.

5. Acknowledgment

Our project was partly supported by the Slovak Scientific Grant Agency (VEGA), project No. 1/0763/09 and from the grant No. UK/90/2011.

6. Conclusion

We have presented a novel approach of mapping textures on model skeletons. Our technique is deterministic and no storing of uv coordinates is needed. The mapping can take its place in representing models geometry, mapping procedurally generated textures or for seamless texture space diffusion.

References

- [ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–10. 1
- [BS86] BIER E., SLOAN K.: Two-part texture mappings. *IEEE Computer Graphics and Applications* 6 (1986), 40–53. 1
- [CTO*10] CAO J., TAGLIASACCHI A., OLSON M., ZHANG H., SU Z.: Point cloud skeletons via laplacian based contraction. In *Proceedings of the 2010 SMI Conference* (2010), pp. 187–197. 1
- [dLE07] D'EON E., LUEBKE D., ENDERTON E.: A system for efficient rendering of human skin. In *ACM SIGGRAPH 2007 sketches* (2007), SIGGRAPH '07. 2