Geometry Synthesis

Ares Lagae Olivier Dumont Philip Dutré

Department of Computer Science
Katholieke Universiteit Leuven *

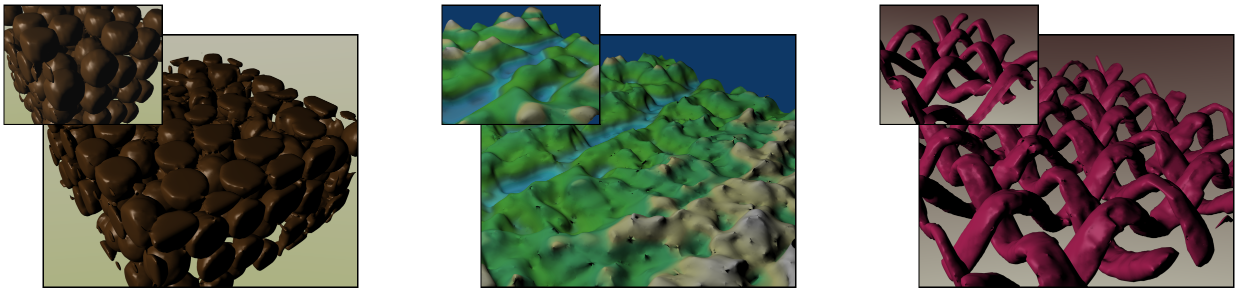

Figure 1: Geometry synthesis. The geometry in the large images is synthesized from the example geometry shown in the smaller images.

Introduction

Geometry synthesis relates to geometry modeling much like texture synthesis relates to texture modeling. Textures can be created by hand, generated procedurally, or synthesized from existing textures. Likewise, geometry can be modeled by hand, and specific types of geometry can be generated procedurally. However, there are currently no techniques to generate geometry *by example*.

Inspired by texture synthesis techniques, such as the one by [Liang et al. 2001], we present a method for geometry synthesis. Given an example of input geometry, we synthesize new output geometry that is perceived to be similar to the input geometry.

Geometry Synthesis

We use regularly sampled distance fields as an intermediate geometry representation. After computing the distance field of the input geometry, which we call the source, we create the target distance field that will hold the synthesized result. It is seeded with random samples from the source. Then, the best matching source block for a corner block in the target is located, and copied from the source to the target. A next block in the target that overlaps an already synthesized area is selected, and again its best match is copied to the target. This is repeated until the target is completely synthesized.

To enforce similarity on a larger scale, we apply this procedure iteratively; the source is repeatedly subsampled, and the first iteration works on the lowest resolution version of the source. The target is seeded with random samples from the source only for the first iteration. For next iterations, the target is seeded with the result of the previous iteration.

When copying a block from the source to the target, we output the corresponding geometry. Because it is unlikely that the geometry corresponding with two neighboring blocks in the target will fit exactly, we smooth the output geometry using mesh relaxation.

Geometry Matching

Matching geometry using distance fields, an essential operation in our algorithm, is handled by the geometry matching algorithm. Given a block of distance field samples, the geometry matching algorithm finds the most similar block in the distance field of the input geometry. To this end, we first organize a hierarchical representation of the distance field in a search tree, and then employ a tree search algorithm to locate the best match for the query block.

To construct the search tree, we first choose a block size. We then consider all possible (thus also overlapping) blocks in the distance field. For each of these blocks, we build a complete block pyramid by averaging (this bears some similarity to the construction of Gaussian image pyramids). Each level in the block pyramid of each block then becomes a node in the search tree. Finally, identical nodes in the tree are combined.

We define a similarity measure between corresponding levels of block pyramids, for which the following two properties hold: first, when comparing the levels of two block pyramids at subsequently increasing resolution, their similarity decreases. Second, a lower bound for the similarity between corresponding levels of two block pyramids can be obtained from their similarity at a lower resolution.

Our tree search algorithm, inspired by the $A^\star$ algorithm, exploits both properties. We have verified experimentally that it is three to four orders of magnitude faster than a linear search. Matching a block of size 32^3 in a 256^3 distance field takes roughly 10 seconds (on a 2 GHz desktop PC). This corresponds to a nearest neighbor query in a $32^3 = 32768$ dimensional space with about 11 million candidate points. A simple brute force linear search would take approximately 12 hours. Our geometry matching algorithm is very fast, considering that there are no general purpose (even approximate) nearest neighbor algorithms that can handle search spaces with a dimension larger than about 30 much more efficiently than a linear search. Because of our fast geometry matching algorithm, geometry synthesis becomes computationally feasible.

Conclusion

Figure 1 shows some geometry synthesis results. Our technique is able to mimic the appearance of the input geometry and synthesize similar output geometry.

Future work will concentrate on specialized geometry reconstruction operators, extending our technique to geometry synthesis over surfaces of arbitrary topology, and extending applications of texture synthesis to similar applications operating on geometry.

References

LAGAE, A., DUMONT, O., AND DUTRÉ, P. 2004. Geometry synthesis. Technical Report CW 381, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium.

LIANG, L., LIU, C., XU, Y.-Q., GUO, B., AND SHUM, H.-Y. 2001. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph. 20*, 3, 127–150.

*email:{ares,olivierd,phil}@cs.kuleuven.ac.be