Geometric-aware Partitioning on Large-scale Data for Parallel Quad Meshing

[Extended Abstract]

Wuyi Yu
wuyiyu@lsu.edu
Jian Tao
jtao@cct.lsu.edu

Qin Jim Chen
qchen@lsu.edu
Xin Li
xinli@lsu.edu

Center for Computation and Technology
Louisiana State University
Baton Rouge, LA 70803

1. INTRODUCTION

Recently the acquisition or generation of large high-resolution geometric datasets pose new challenges to the design of effective processing algorithms. These big and complex data are expensive to model and analyze using existing sequential algorithms, therefore parallel algorithms utilizing high-performance computers is desirable.

Divide-and-conquer is a natural and effective strategy in parallel mesh generation for large geometric data. The given region or object is first decomposed into a set of solvable and simplified subparts; then each subpart is distributed to a working processor for mesh construction; finally, individually generated meshes are merged to get the final result. Following this general paradigm, we aim to develop a partitioning algorithm on comlex and large-scale 2D regions for parallel quadrilateral mesh generation.

2. MOTIVATION

In geometric processing through divide-and-conquer, the partitioning of data often directly dictates the algorithm's efficiency and quality. We formulate **three main criteria** as follows: (1) The areas of the subregions should be similar; (2) The boundary length of each subregion should be small compared with it's area; and (3) Each subregion should have desirable geometric property, and more specifically for quad meshing, it should have corner angles close to $k\pi/2, k \in \{0, 1, 2, 3\}$. The **first two criteria** determine the parallel performance, while the **third criterion** on the subregion's geometry affects the quality of the final quad tessellation.

After data partitioning, we distribute subregions to different working processors for mesh generation. In our implementation, we use consistent boundary sampling and *advancing front* for parallel meshing on each subregion. The sub-meshes are finally merged together.

3. OUR APPROACH

3.1 Partitioning

We observe that: *if the cells of the initial tessellation has near-square geometry, then the partition performed on the dual graph of this tessellation tends to have smaller separator angle deviation.* Hence, we propose a two-stage partitioning scheme, which is shown in Figure 1.

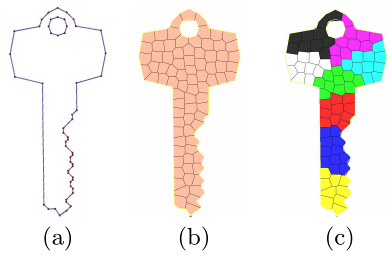

Figure 1: Partitioning a 2D "Key" Region for Parallel Quad Mesh Generation. (a) The input 2D boundary; (b) L_∞-norm CVT on the input boundary; (c) our partitioning result, with different subregions indicated using different colors.

First, we use L_∞-Centroidal Voronoi Tessellation to generate a tessellation with cells similar to squares. A 2D L_p *Centroidal Voronoi Tessellation* of a given set of distinct points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ in $\mathbf{R}^2$ is defined by a set of Voronoi cells $\{\Omega_i\}_{i=1}^n$ which minimizes the energy [3]:

$$F(\mathbf{X}) = F([\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]) = \sum_i \int_{\Omega_i \cap \Omega} \|\mathbf{y} - \mathbf{x}_i\|_p^p d\mathbf{y} \quad (1)$$

where $\|\mathbf{z}\|_p = (\sum_j |z_j|^p)^{\frac{1}{p}}$, z_j is the j-th coordinate of a 2D point $\mathbf{z}$.

After solving the L_∞-CVT, we get a tessellation of the 2D region. Then on it's dual graph, we solve the graph partitioning to get the sub-regions by the the hierarchical graph partitioning algorithm [2].

3.2 Parallel Quad Meshing on Subregions

After the entire 2D region is partitioned, subregions can be sent to different processors for simultaneous mesh generation. We use the **advancing front technique** [1] to tile the interior regions with quads. To ensure the individually constructed sub-meshes can be composed directly, we need to sample boundary vertices consistently on the shared edge of adjacent subregions. We use a simple sampling scheme: first, we compute an average edge length $\bar{l}$ from input boundary line segments, then on each interior partitioning boundary curve S we evenly sample $\frac{l_S}{2\bar{l}}$ points, where l_S is the length of S.

4. EXPERIMENTAL RESULTS

We perform experiments on our high performance computing clusters, SuperMIC, which consist of 128 computing nodes. Each node has two 2.6GHz 8-Core Xeon 64-bit Processors and 32GB memory. We test our algorithm on three coastal ocean/terrain regions : the Gulf of Mexico, Matagorda Bay, and West bay, with 230M, 250M, and 550M boundary vertices, respectively. We compare algorithms in three aspects: (1) **Decomposition Quality, Parallel Computation Efficiency** and **Meshing Quality**.

We perform thorough experiments to evaluate our partitioning and mesh generation algorithm. Compared with other data partitioning strategies, our new algorithm leads to better partition result and better final meshing quality.

5. CONCLUSION

We present a parallel quadrilateral mesh generation pipeline for large-scale 2D geometric regions. A main contribution of this work is the solving of data partitioning with effective incorporation of the geometric constraint on angles between separators. After partitioning, subregions are distributed to different processors for parallel mesh generation through advancing front. Finally, after composition, post-processing is performed near partitioning boundaries for refinement.

6. REFERENCES

[1] T. D. Blacker and M. B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):811–847, 1991.

[2] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM JSC*, 20(1):359–392, 1998.

[3] B. Lévy and Y. Liu. Lp centroidal voronoi tessellation and its applications. *ACM Trans. Graph.*, 29(4):119:1–119:11, 2010.