A Deviation-based Vertex Reordering Technique for Mesh Quality Improvement

Harrim Kim[1], Junhyeok Choi[1], Shankar P. Sastry[2], Jibum Kim[1]
[1]Incheon National University, Incheon, South Korea. [2]Intel Corporation, Hilsboro, OR, USA

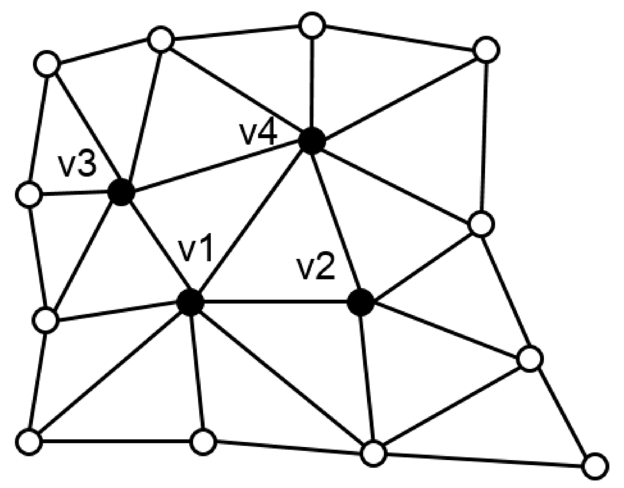
Abstract

We devised a vertex reordering technique to exploit the inequality of mesh elements so that poor quality elements are improved at the cost of high-quality elements. We reorder vertices based on how likely they are to improve the quality of adjacent elements. The estimation of how effective a vertex movement is based on the gradient of the element quality with respect to the vertex location. The heuristic technique is based on the theory of nonsmooth optimization.

Motivation

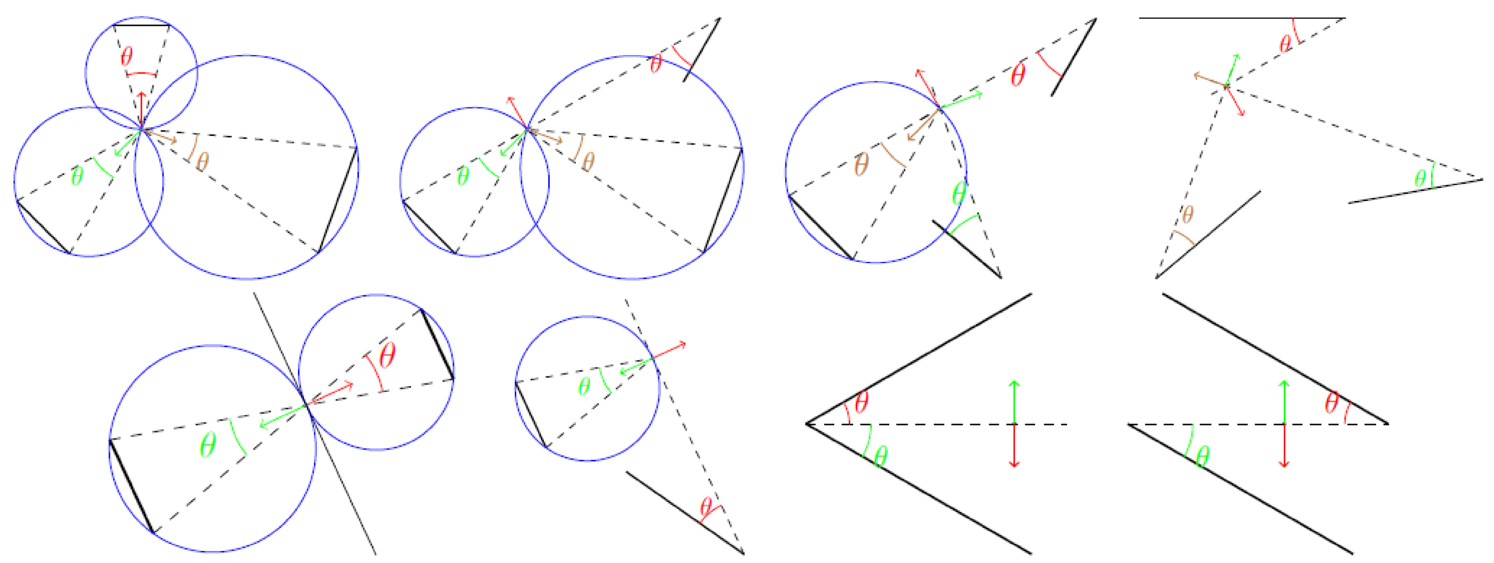
The ordering of vertices is important when the quality of a mesh is defined as the quality of its worst element.

The worst element improves only in the first iterations and optimization routine was stuck when mesh optimization is performed.



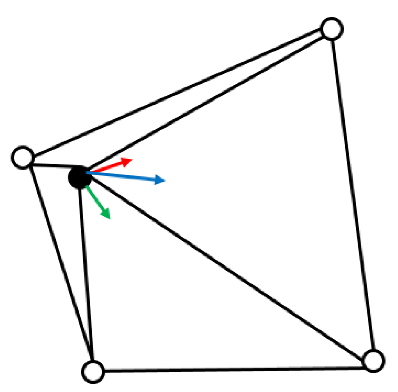
Theory

Two or three angles in 2D meshes are identical in an optimal patch.



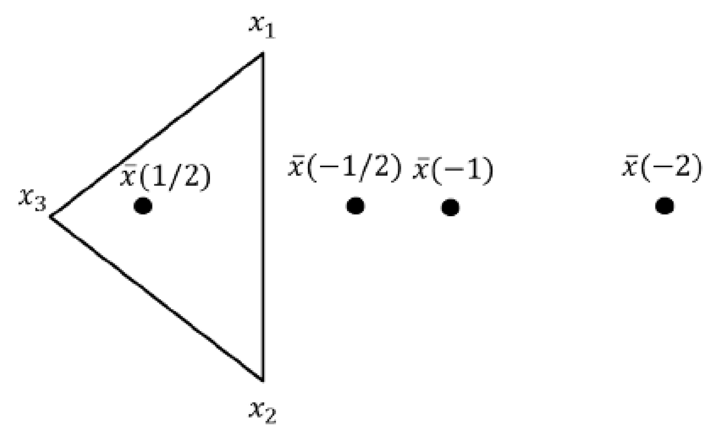
Algorithm

Input: A mesh and a list of free vertices
Output: An ordering of the free vertices
for all the free vertices do
    compute the quality of all elements around the vertex
    compute their gradient with respect to the vertex
    pseudo-ActiveSet ={}
end
while pseudo-ActiveSet's gradient are unsuitably directed do
    add the next element from the sorted set into pseudo-ActiveSet
end
Compute the difference in the quality of the best and worst element in the pseudo-ActiveSet
Sort the vertices in the decreasing order of the difference in the quality computed in the loop above



Local Mesh Optimization

Solves the optimization problem for one free vertex at a time
q_i: i^{th}element quality
F=min (max q_i)
Solves F using a downhill simplex method



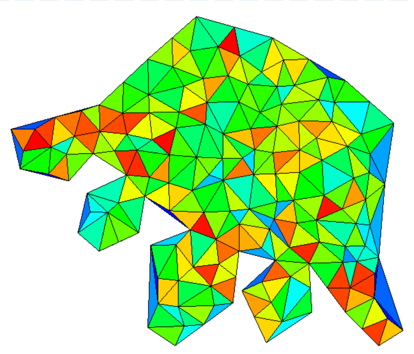
Vertex Reordering Schemes

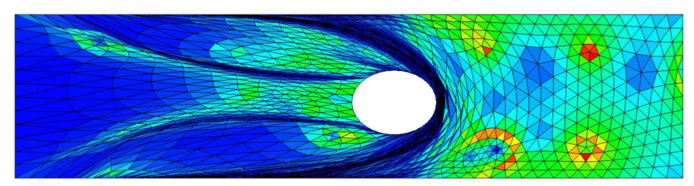
1. Null ordering: do not reorder the vertex list

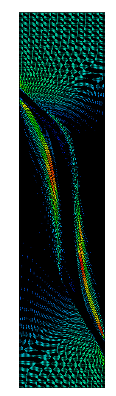
2. Static ordering: vertex reordering is performed only one in the beginning of an iteration

3. Dynamic ordering: decide which vertices to move after the movement of some constant number of vertices

Examples


Crown


Cylinder


Shashkov

Mesh	# Vertices	# Elements	Worst element quality
Crown	154	262	255.5082
Cylinder	1,196	2,240	2282.6032
Shashkov	4,225	8,192	277.0463

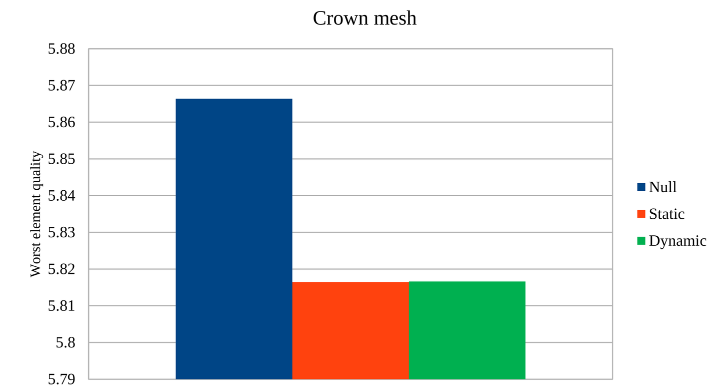
Results

The worst element quality was improved by up to 7.2% after the proposed vertex reordering method was used.

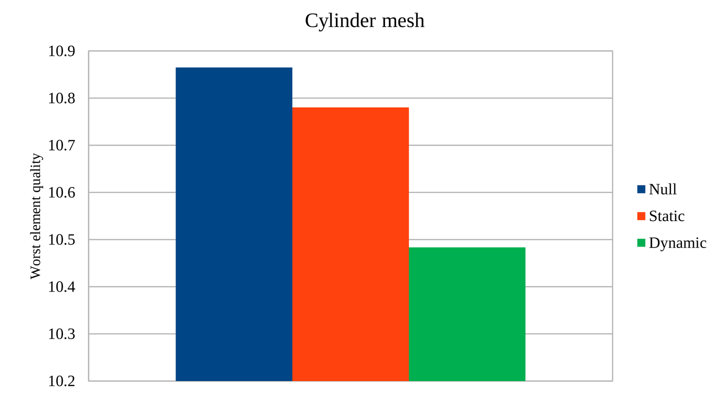
Dynamic vertex reordering showed the best output results in terms of the worst element quality.

Static vertex reordering both improved the worst element quality and reduced the mesh optimization time up to 29.1%.

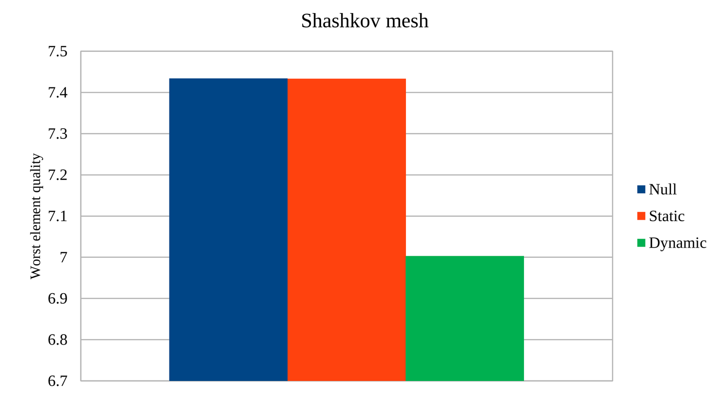
Crown mesh	Worst element quality	Time (sec)
Initial mesh	255.5080	-
Output mesh (null)	5.8663	2.4
Output mesh (static)	5.8164	1.7
Output mesh (dynamic)	5.8165	2.2



Cylinder mesh	Worst element quality	Time (sec)
Initial mesh	2282.6132	-
Output mesh (null)	10.8648	85.0
Output mesh (static)	10.7805	73.9
Output mesh (dynamic)	10.4835	89.0



Woody mesh	Worst element quality	Time (sec)
Initial mesh	277.0462	-
Output mesh (null)	7.4342	161.1
Output mesh (static)	7.4331	125.4
Output mesh (dynamic)	7.0031	141.7



References

[1] L. Freitag and P. Plassmann, "Local optimization-based simplicial mesh untangling and improvement,", *Int. J. Numer. Meth. Eng.*, vol. 49, pp. 109–125, 2000.
[2] S.P. Sastry, ""Maximizing the Minimum Angle with the Insertion of Steiner Vertices," in Proc. of the 27th *Canadian Conference on Computational Geometry*, 2015.

Acknowledgement

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1C1B1007080).