

25th International Meshing Roundtable

# A Local Frame based Hexahedral Mesh Optimization

Xifeng Gao<sup>a</sup>, Guoning Chen<sup>a,\*</sup>

<sup>a</sup>*Department of Computer Science, University of Houston, 3551 Cullen Blvd, Houston, USA*

---

## Abstract

We propose a new hexahedral mesh optimization technique that is based on the smoothing of the local frame system embedded in each hexahedral element. This smoothing process can be formulated as a quadratic optimization problem that can be solved efficiently. We have applied this technique for the improvement of the quality of a number of hexahedral meshes.

© 2016 The Authors. Published by Elsevier Ltd.

Peer-review under responsibility of the organizing committee of IMR 25.

*Keywords:* Hexahedral mesh; optimization; local frame

---

## 1. Introduction

Structured volumetric meshes, especially hexahedral (or hex-) meshes, are preferred by many critical applications, including mechanical analysis using finite element methods, kinematic and dynamic analysis of mechanisms, bio-mechanical engineering, and computational fluid dynamics. This is mostly due to its naturally embedded tensor product structure, larger tolerance for anisotropy and less numerical stiffness, compared to unstructured meshes (e.g., tetrahedral (or tet-) meshes). These preferred properties enable the convenient imposition of a simulation basis with a higher derivative smoothness between elements of the mesh, and the handling of large deformation during simulations.

*Previous work* . While various techniques have been proposed to generate hex-meshes for some classes of objects via sweeping or paving [1–3], volumetric polycubes construction [4–7], and frame field based method [8–10], respectively, many initial hex-meshes obtained from these methods, while topologically valid, contain poorly shaped and even inverted elements (i.e., with negative Jacobian), making the subsequent computation (e.g., finite element simulations) carried out on them unstable [11]. Therefore, an optimization process that alters the vertex positions, while maintaining the mesh connectivity, to regulate the element shapes is typically required [12]. However, such an optimization is still a challenging problem, because the shape quality metrics expressed as a function of vertex positions are usually non-linear.

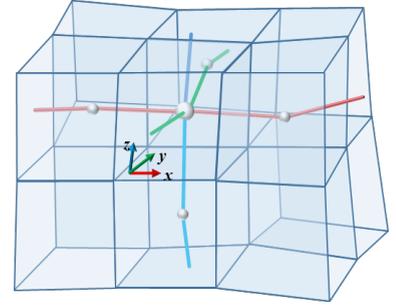
*Overview of our method* . To mitigate this challenge, we introduce a local frame based optimization framework. As a tensor product (i.e., a local frame, as shown in the inset to the right) can be imposed into each hexahedral (hex-)

---

\* Corresponding author. Tel.: +1-000-000-0000 ; fax: +1-000-000-0000.

*E-mail address:* [gchen16@uh.edu](mailto:gchen16@uh.edu)

element, along its three principle directions three families of curves can be *traced out* by connecting the center of each hex-element with the centers of its six direct neighbors, respectively. We refer to these curves the *frame lines*. If all the elements of the mesh have regular shapes, each element will have orthogonal local frame and the three families of curves are all straight lines. However, this ideal scenario is usually not achievable, as the boundary surface of the resultant mesh should match the input surface which is typically curved. Nonetheless, one can optimize the configuration of these local frames to tend toward the ideal setting in order to optimize the corresponding hex-mesh indirectly. This observation motivates our framework. In particular, we formulate the optimization of these local frames into a quadratic problem via the definition of an energy function based on the angles along the three families of frame lines and the angles at their intersections. Ideally, the angles along the frame lines should be as close to  $180^\circ$  as possible and the angles at their intersections should be close to  $90^\circ$ . The minimization of this energy function can then be reduced to the solving of a linear system. Our method shares some similarity to the recently introduced hex-element optimization based on a cone descriptor [13]. However, compared to that method, our method is intuitive and need not perform element orientation and normal rectification, which makes the implementation simpler and faster to compute.

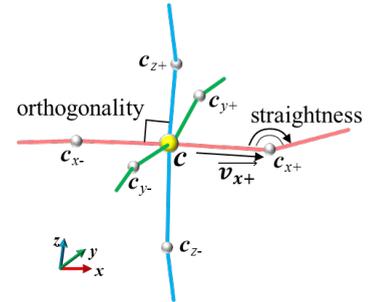


Although our method provides no theoretical guarantees on the solution, our experiments show that it typically produces inversion-free meshes when applied to a number of hex-meshes with various degrees of poorly shaped elements.

## 2. Our Methodology

Consider a valid, manifold all-hex mesh  $\mathcal{H}$  that consists of  $|H|$  hex-elements. For a hex-element with none of its six quadrilaterals on the boundary, we denote its center (also the origin of its local frame) as  $\mathbf{c}$ , and the centers of its six direct neighboring elements are  $\mathbf{c}_{x-}$ ,  $\mathbf{c}_{x+}$ ,  $\mathbf{c}_{y-}$ ,  $\mathbf{c}_{y+}$ ,  $\mathbf{c}_{z-}$ , and  $\mathbf{c}_{z+}$ , respectively. The vector pointing from  $\mathbf{c}$  to  $\mathbf{c}_{i\bullet}$  is denoted by  $\mathbf{v}_{i\bullet}^{\rightarrow}$  (e.g.,  $\mathbf{v}_{x+}^{\rightarrow}$  denotes the vector  $\mathbf{c}_{x+} - \mathbf{c}$ ).  $\bullet$  represents + or –.

**Orthogonality.** Given the above notation, the orthogonality at  $\mathbf{c}$  can be measured by  $\| \langle \mathbf{v}_{i+}^{\rightarrow}, \mathbf{v}_{j+}^{\rightarrow} \rangle \|$ ,  $\| \langle \mathbf{v}_{i+}^{\rightarrow}, \mathbf{v}_{j-}^{\rightarrow} \rangle \|$ ,  $\| \langle \mathbf{v}_{i-}^{\rightarrow}, \mathbf{v}_{j+}^{\rightarrow} \rangle \|$  and  $\| \langle \mathbf{v}_{i-}^{\rightarrow}, \mathbf{v}_{j-}^{\rightarrow} \rangle \|$  where  $i, j \in \{x, y, z\}$  and  $i \neq j$ . The closer to the zero are these inner products, the more orthogonal the corresponding two vectors are. We then can define the energy that measures how far away the configuration of the local frame at  $\mathbf{c}$  is from an orthogonal frame as follows.



$$E_o = \sum_{k=0}^{|H|-1} \sum_{i,j \in \{x,y,z\}} \| \langle \mathbf{v}_{i\bullet}^{\rightarrow}, \mathbf{v}_{j\bullet}^{\rightarrow} \rangle \|^2 \quad \bullet \text{ represents } + \text{ or } - \quad (1)$$

**Straightness.** Considering one family of frame lines, say along  $x$  direction, the straightness at  $\mathbf{c}$  can be measured by  $\| \langle \frac{\mathbf{v}_{i-}^{\rightarrow}}{\|\mathbf{v}_{i-}^{\rightarrow}\|}, \frac{\mathbf{v}_{i+}^{\rightarrow}}{\|\mathbf{v}_{i+}^{\rightarrow}\|} \rangle + 1 \|$ . The closer to the zero this measurement is, the straighter the frame line is at  $\mathbf{c}$ . We then can define the energy that measures how far away the configurations of all frame lines are from perfect straight lines below.

$$E_s = \sum_{k=0}^{|H|-1} \sum_{i \in \{x,y,z\}} \| \langle \frac{\mathbf{v}_{i-}^{\rightarrow}}{\|\mathbf{v}_{i-}^{\rightarrow}\|}, \frac{\mathbf{v}_{i+}^{\rightarrow}}{\|\mathbf{v}_{i+}^{\rightarrow}\|} \rangle + 1 \|^2 \quad (2)$$

Combining the above two energy terms, we define the following energy to measure how far away the local frame configurations of the individual elements are from the ideal settings.

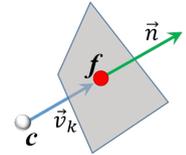
$$E_{frame} = E_o + E_s$$

Note that for a hex-element that has at least one of its quadrilaterals on the boundary, we consider the centers, denoted as  $\mathbf{f}$ , of those boundary quadrilaterals for the measurement of orthogonality and straightness described above.

**Boundary Handling.** For hex-meshes, flipped elements usually appear near the boundary. To alleviate this issue, we enforce the family of frame lines intersecting the surface boundary to have the same direction with the normal of the surface at the intersecting point (i.e.,  $\mathbf{f}$ ) by the formula below, as illustrated by the inset.

$$E_{align} = \sum_{k \in B} \| \langle \frac{\vec{v}_k}{\|\vec{v}_k\|}, \mathbf{n} \rangle - 1 \|^2$$

where  $\vec{v}_k$  is the vector from  $\mathbf{c}$  to  $\mathbf{f}$ , and  $B$  is the set of hexahedra with at least one quadrilateral on the boundary.  $\mathbf{f}$  can be represented by  $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)/4$  where  $\mathbf{v}_i$  represent the four vertices of the quadrilateral on the boundary. By employing the boundary conformity term in [13], we can optimize  $\mathbf{f}$  by directly optimizing the positions of the boundary vertices of the mesh. For completeness, we restate the boundary conformity term [13] as below,



$$E_{conformity} = \sum_{\mathbf{v} \in S} \beta (\langle \vec{\mathbf{n}}, \mathbf{v} \rangle + d)^2 + \sum_{\mathbf{v} \in F} (\alpha \|\mathbf{v} - (\hat{\mathbf{v}} + a\mathbf{t})\|^2 + a^2) + \sum_{\mathbf{v} \in C} \alpha \|\mathbf{v} - \hat{\mathbf{v}}\|^2 \quad (3)$$

where all the boundary vertices are classified into three groups, i.e.,  $S$ ,  $F$ , and  $C$  representing flat surface vertices, feature vertices, and corner vertices, respectively.  $\beta$  and  $\alpha$  are user specified parameters.  $\hat{\mathbf{v}}$  is the reference, or closest, surface position for each vertex.  $\langle \vec{\mathbf{n}}, \mathbf{v} \rangle + d$  is the implicit equation of the plane passing through  $\hat{\mathbf{v}}$  and orthogonal to the input surface normal  $\vec{\mathbf{n}}$  at that point, and  $\mathbf{t}$  is the feature tangent at  $\hat{\mathbf{v}}$ .  $a$  is an auxiliary variable added to the system to enable feature preservation.

Therefore, we have the following formulation for direction alignment and boundary conformity,

$$E_{boundary} = E_{align} + E_{conformity} \quad (4)$$

Therefore, given the centers of the individual elements (i.e., the origins of their corresponding frames) and the boundary vertices as variables, we minimize the following energy that takes into account both the boundary conformity and the local frame quality.

$$E = E_{frame} + E_{boundary} \quad (5)$$

Given the dot product of variables and the normalization of the vector involved in the above definition of the energy function, directly optimizing this non-linear formula is impractical. Therefore, we use the following strategy to turn the above non-linear formulation into a tractable quadratic problem.

- for dot product computation  $\langle \vec{\mathbf{v}}_a, \vec{\mathbf{v}}_b \rangle$ : we use the current mesh to estimate either  $\vec{\mathbf{v}}_a$  or  $\vec{\mathbf{v}}_b$  to eliminate the coordinate variables.
- for normalization: we approximate the length of a local frame edge by computing their target lengths [13].

With the above approximation, minimizing the energy  $E$  can be achieved by solving a linear system.

After optimizing the locations of the centers of the elements, we obtain the geometric locations of the mesh vertices of the output hex-mesh by averaging the centers of its neighboring hexahedra.

To summarize, the pipeline of our approach can be described as the following iterative process:

1. Compute target lengths for the vectors (i.e.,  $\vec{\mathbf{v}}_{i\bullet}$ ) of the local frames;
2. Minimize  $E$  (Equation 5);
3. Reconstruct the hex-mesh. If there is non flipped elements, output; otherwise, go to step 1.

### 3. Results

We have applied our technique to a number of hex-meshes with various levels of inverted elements (i.e., many elements have negative scaled Jacobian). Figure 1 provides some preliminary results of the proposed optimization technique. The Fandisk (614 hexes), Hanger (6, 633 hexes) and Joint (17, 784 hexes) models are used. Two versions of input for each model were generated from their corresponding hex-meshes produced by the state-of-the-art techniques [10] by perturbing the interior vertices with various degrees of noise (the first row for each experiment in

Figure 1). Compared to the cone descriptor based approach [13], our method does not require the estimation of the valid (or positive) cone direction, which could be computation expensive and may fail. Therefore, our method can produce an inversion free mesh much faster than the cone descriptor based approach. For the reported experiments, after applying our technique by a couple iterations (at most two), all these hexahedral meshes have positive minimum scaled Jacobian. Specifically, the results for the Fandisk1, Fandisk2, Hanger1 and Joint2 were all obtained after one iteration, while Hanger2 and Joint1 require two iterations to produce the inversion free meshes. For all the reported results, all boundary vertices are marked as corners. Also, we set  $\alpha = 10000$  in Equation 3.

#### 4. Discussions and Future Work

Although our initial results are promising, to produce hex-meshes with comparable quality to the cone descriptor based approach we still need to fine-tune the above pipeline by adjusting the relevant parameters and weights. In addition, for simplicity the boundary vertices of the input meshes in our experiments are not perturbed, which we will relax for further assessment of our method. Furthermore, similar to the cone descriptor based optimization [13], our method does not have theoretical guarantee on the solution. To completely address this challenge, the algorithm first needs to know whether an inversion free mesh can indeed be obtained given the initial connectivity of the mesh and its boundary. If the inversion free mesh is not achievable, it will be meaningless to perform any optimization without changing the mesh connectivity. On the other hand, even if an inversion mesh can be obtained, how good the quality of the mesh can be? That is, is there an upper bound for the quality of the mesh (say, measured by scaled Jacobian) given its initial connectivity? Knowing this upper bound will help assess the performance of an optimization algorithm. We plan to address these open questions in the future work.

#### References

- [1] S. J. Owen, A survey of unstructured mesh generation technology, in: INTERNATIONAL MESHING ROUNDTABLE, 1998, pp. 239–267.
- [2] J. F. Shepherd, C. R. Johnson, Hexahedral mesh generation constraints, *Eng. with Comput.* 24 (2008) 195–213.
- [3] A. Sheffer, M. Etzion, A. Rappoport, M. Bercovier, Hexahedral mesh generation using the embedded voronoi graph, *Engineering with Computers* 15 (1999) 248–262.
- [4] J. Gregson, A. Sheffer, E. Zhang, All-hex mesh generation via volumetric polycube deformation, *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing 2011)* 30 (2011) 1407–1416.
- [5] M. Livesu, N. Vining, A. Sheffer, J. Gregson, R. Scateni, Polycut: Monotone graph-cuts for polycube base-complex construction, *ACM Trans. Graph.* 32 (2013) 171:1–171:12.
- [6] J. Huang, T. Jiang, Z. Shi, Y. Tong, H. Bao, M. Desbrun, L1-based construction of polycube maps from complex shapes, *ACM Trans. Graph.* 33 (2014) 25:1–25:11.
- [7] X. Fang, W. Xu, H. Bao, J. Huang, All-hex meshing using closed-form induced polycube, *Transactions on Graphics (Proc. SIGGRAPH 2016)* 35 (2016).
- [8] J. Huang, Y. Tong, H. Wei, H. Bao, Boundary aligned smooth 3d cross-frame field, *ACM Trans. Graph.* 30 (2011) 143:1–143:8.
- [9] M. Nieser, U. Reitebuch, K. Polthier, Cubecover- parameterization of 3d volumes, *Comput. Graph. Forum* 30 (2011) 1397–1406.
- [10] Y. Li, Y. Liu, W. Xu, W. Wang, B. Guo, All-hex meshing using singularity-restricted field, *ACM Trans. Graph.* 31 (2012) 177:1–177:11.
- [11] P. P. Pébay, D. Thompson, J. Shepherd, P. Knupp, C. Lisle, V. A. Magnotta, N. M. Grosland, New applications of the verdict library for standardized mesh verification pre, post, and end-to-end processing, in: *Proceedings of the 16th International Meshing Roundtable*, Springer, 2008, pp. 535–552.
- [12] P. M. Knupp, Hexahedral mesh untangling & algebraic mesh quality metrics, in: *Proceedings, 9th International Meshing Roundtable*, 2000, pp. 173–183.
- [13] M. Livesu, A. Sheffer, N. Vining, M. Tarini, Practical hex-mesh optimization via edge-cone rectification, *ACM Transactions on Graphics (TOG)* 34 (2015) 141.

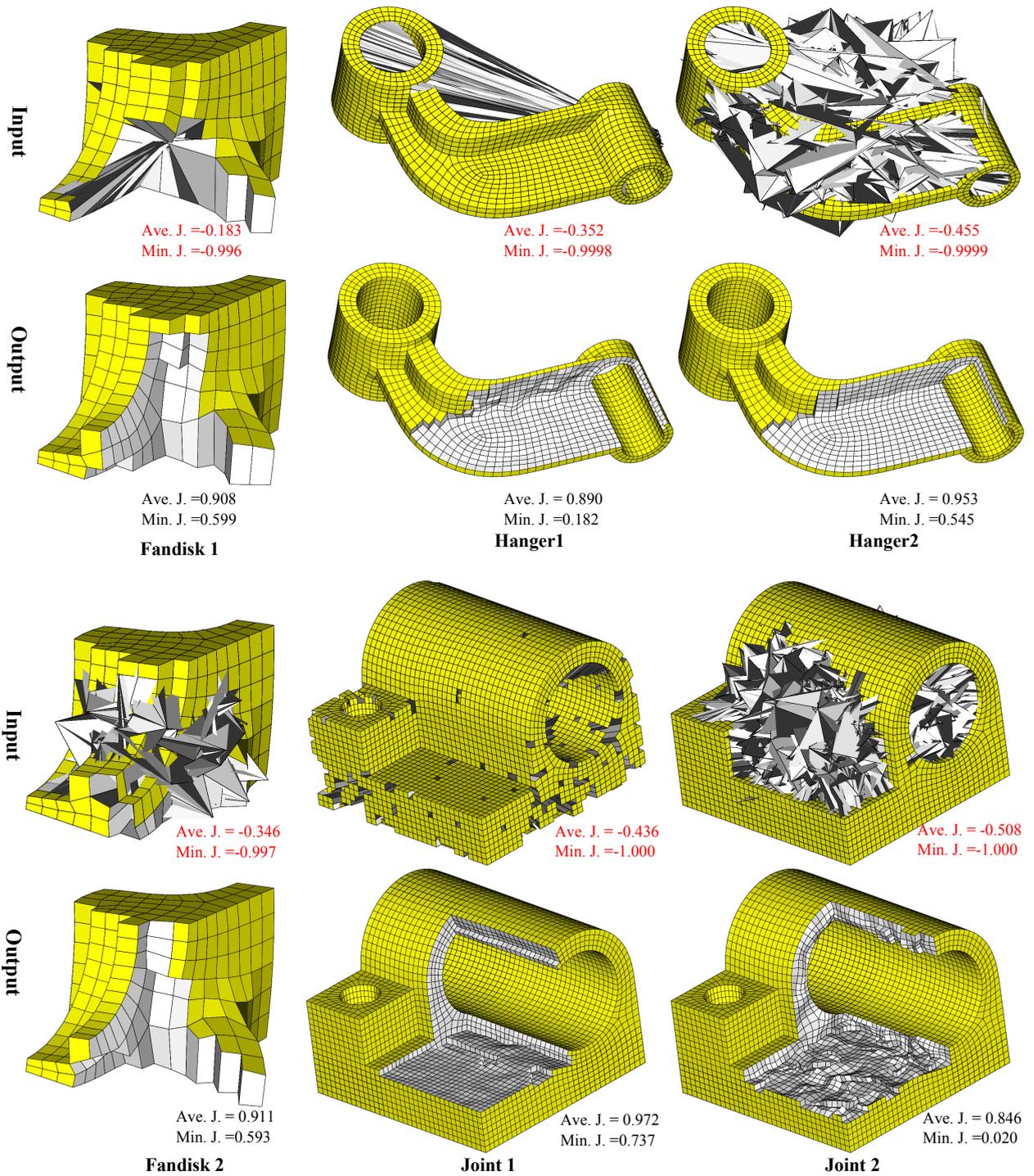


Fig. 1. Results of our optimization framework. The top row shows the input meshes and their corresponding scaled Jacobian information. The bottom row shows the optimized meshes with our method.