The Visual Computer (2020) 36:2355–2368
https://doi.org/10.1007/s00371-020-01967-6

ORIGINAL ARTICLE

Valence optimization and angle improvement for molecular surface remeshing

Dawar Khan[1] · Alexander Plopski[2] · Yuichiro Fujimoto[1] · Masayuki Kanbara[1] · Zhanglin Cheng[3] · Hirokazu Kato[1]

Published online: 2 September 2020

Abstract
Molecular surface mesh generation plays a vital role in molecular modeling and visualization. However, meshes extracted directly from Protein Data Bank files have several issues such as small and large triangles, redundant elements, self-intersections, and irregular vertices. The state-of-the-art mesh improvement methods often fail to deal with these issues. In this paper, we present a novel method for valence optimization and angle improvement. For valence optimization, we remove the bad valence vertices with its neighbor triangle making regular holes in the mesh. The holes are filled in a careful manner to improve their valences as well as angle quality. We also use a segmentation-based surface remeshing which segments the mesh into random segments and then each segment is independently remeshed. In addition, a point insertion scheme is applied to minimize the ratio of obtuse triangles. Experimental results show that our method not only improves the maximal and minimal angles to an angle bound of [30° 120°] but also improves the vertices' regularity, reduces the ratio of obtuse triangles, preserves the area and volume, and always succeeds with downstream applications.

Keywords Molecular surface remeshing · Mesh quality · Valence optimization · Molecular modeling

1 Introduction

Molecular surface meshing and molecular modeling are an interesting research direction used in many fields including computer graphics, mathematics, molecular biology, biophysics, and chemistry. It plays a vital role in various phenomena such as protein folding, docking, implicit-solvent modeling, structure prediction, interaction of molecules, and measuring their areas/volumes [1,2]. However, meshes generated directly from molecular data have several defects including self-intersections, small and large angles, and redundant and irregular vertices. There are several methods to improve the mesh quality prior to its use in the downstream applications. Some of these defects such as self-intersections have been addressed in SMOPT [3]. However, SMOPT fails in maximal and minimal angle improvements. Similarly, small and large angles have been improved up to an acceptable level by a recently proposed cut-and-fill (CAF) method [1]. However, CAF fails to achieve a considerable improvement in valence optimization.

Figure 1 shows the main steps in molecular surface remeshing. Protein Data Bank (PDB) files are downloadable from their online repository (https://www.rcsb.org/). PDB2PQR [4] is a tool used to convert these files from PDB format to PQR format. The next step is mesh generation, where the mesh generating tools such as TMSmesh [5] are used to generate surface mesh from a PQR file. Unfortunately, these meshes are generated in raw form, thereby containing low-quality elements including zero degree angles, large angles, and irregular vertices [1]. Such raw meshes with low-quality elements are difficult to be directly used in downstream applications such as TetGen [6] or visualization (Fig. 2).

Therefore, remeshing for quality enhancement is desired at this stage to refine the raw mesh. However, previous methods have no significant improvement in mesh quality. Figure 2

✉ Dawar Khan
dawar@is.naist.jp

[1] Interactive Media Design Lab, Nara Institute of Science and Technology, Nara 630-0192, Japan

[2] Department of Information Science, University of Otago, Dunedin, New Zealand

[3] Shenzhen VisuCA Key Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

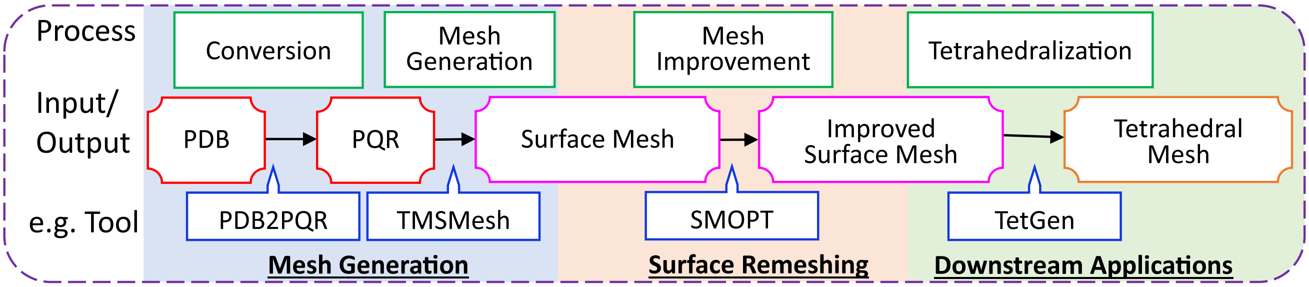


Fig. 1 The main pipeline of the molecular surface mesh generation and surface remeshing. The red color boxes represent the protein files (PDB and PQR), the magenta color boxes are surface meshes, the brown box represents tetrahedral mesh, the green boxes are processes, and the blue boxes are examples of available tools for each particular process

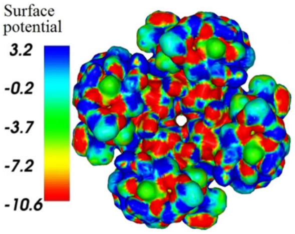


Fig. 2 Electrostatic potential on molecular surface, calculated with AFMPB [7]

shows an example of the molecular surface mesh in the downstream applications, i.e., the electrostatic potentials calculated with AFMPB [7] mapped on the molecular surface.

Typically, a triangle with an angle smaller than 30° or obtuse angle (i.e., $\geq 90°$) is called a bad triangle [8]. Similarly, it has been shown in the previous research that valence optimization can speed up the convergence. A vertex with valence (no. of adjacent edges) equal to four for a boundary vertex and valence 6 for an interior vertex is called optimal or regular vertex, while the others are non-optimal. Generally, valence 5, valence 6, and valence 7 vertices are treated as optimal valences and are found easy for surface remeshing [1,8,9].

In generic surface remeshing, the state-of-the-art methods have a significant improvement in the mesh quality. However, these methods from generic remeshing fail in molecular surface meshes due to additional issues of these meshes. Like generic meshes, it is challenging to preserve the area and volume as well as features and topology during molecular surface remeshing. Unlike the generic meshes, the meshes generated from molecular data have additional challenges including:

- The existence of very small and large angles (the small angles are equal to or nearly to 0^0).
- High ratio of the defective elements such as isolated vertices, self-intersections, and redundant elements.
- Complexity of the molecular surface meshes in their shapes and number of vertices.

In this paper, we present a simple method for valence optimization and angle improvement of molecular surface remeshing. We start with centroidal Voronoi tessellation (CVT) [10] initialization. The bad valence vertices are removed with their adjacent triangles in a careful way to create holes in regular structures. The holes are refilled with insertion of new vertices in the calculated feasible position which improves the vertices regularity and avoids the creation of small and large triangles. Then, we apply mesh segmentation to divide the mesh into random independent segments. Each segment is independently remeshed using real-time adaptive remeshing (RAR) [11]. After, the segment-wise remeshing, we remove small-angle triangles ($\theta < 30°$). Then, we apply a point insertion scheme that inserts a new point near each obtuse triangle to eliminate the obtuse triangle. In this manner, the ratio of obtuse triangles is significantly reduced. Finally, we apply Laplacian smooth and RAR method to the bad triangles and their neighbor triangles locally, with constraints to avoid the creation of new small triangles. We compared our results with four existing methods. Experimental results show that our method performs better than existing molecular surface remeshing methods in terms of angle improvements and valence optimization. In addition, there is no failure of the downstream applications for our output mesh. In summary, we have the following contributions in this paper.

- We propose a novel mechanism for valence optimization of molecular surface remeshing that improves vertex regularity without losing the mesh quality.
- A divide-and-conquer strategy is used for segment-based surface remeshing which improves the mesh quality without introducing self-intersections or other defects. It preserves the input area and volume.
- Our method not only improves vertex regularity, but also achieves an angle bound of [30° 120°], and improves other mesh quality parameters. It reduces the ratio of obtuse triangles significantly.

2 Related work

In generic surface remeshing, numerous methods have been proposed. These methods can be classified as mesh simplification-based methods [12], Delaunay insertion methods [13], advancing front-based method [14], field-based approaches [15], and local operators-based mesh optimization [11]. Recent methods have a significant improvement in minimal and maximal angles and meshing quality for graphical models. For example, Yan and Wonka [16] used additional operators with CVT to avoid short Voronoi edges and remove obtuse and small triangles ($< 30°$). However, their method does not work for noisy meshes.

The use of edge-based operators (edge split, edge collapse, vertex translation) [17] also has a significant achievement in meshing quality. These methods can improve the mesh quality for common graphical models, such as CAD models and human-made meshes. However, as molecular remeshing as additional challenges, these methods are often failed. The meshes generated from protein files have complex structures with many defects. Therefore, the generic meshing methods do not work to improve these meshes.

Researches reveal that small and large triangles badly affect the simulation results, and even a single bad triangle can affect the results [1,8,18]. To the best of our knowledge, a recent method of non-obtuse remeshing [8] in computer graphics better performs for angle improvement achieving an angle bound [30° 90°]. However, for molecular surface remeshing, there is no significant achievement in angle improvement with consideration of their area and volume preservations, valence optimization, and features preservations. Irregular vertices are also problematic in remeshing algorithms [9,19]. In this regard, Aghdaii et al. [9] introduced regular meshes called 567 meshes, which was recently improved with a low budget method [19], achieving comparatively higher geometric fidelity.

Molecular surfaces have different structures such as the van der Waals (VDWs) surface [20], the solvent accessible surface (SAS) [21], the solvent excluded surface (SES) [22], the minimal molecular surface [23], the molecular skin surface [24], and the Gaussian surface [25]. TMSmesh [2,5] is an efficient tool used for generating the triangular mesh of the Gaussian surface of biomolecules. Similarly, SESs are efficiently generated using alpha-shapes [26], marching tetrahedra [27], and analytical expressions [28].

In the context of molecular surface remeshing, several methods have been proposed. For example, SMOPT [3] is a tool specially designed for molecular surface mesh refinement which eliminates all redundant elements and self-intersections from the raw mesh. However, mesh refined with SMOPT still contains small angles. Recently, Khan et al. [1] proposed a method that starts with initialization using real-time adaptive remeshing (RAR) [11] followed by aspect ratio enhancement and a cut-and-fill module to eliminate invalid regions and small triangles. These selected regions are further improved with edge splitting, edge collapsing, and vertex translation. The results show that this method [1] better performs than SMOPT and other previous methods and that this method is able to generate a good quality mesh with an angle bound of [30°, 120°]. However, it fails to achieve a significant improvement in vertices regularities and it has a high ratio of obtuse triangles. Valence optimization can speed up the convergence. Vertices regularity is also a quality parameter used in surface remeshing [29].

Decherchi and Rocchia [30] proposed a ray-casting method for the triangulation of complex manifold surfaces in the nano-bioscience field. They summarized various applications of molecular surfaces in implicit-solvent modeling and simulations using the boundary element method (BEM) and the finite element method (FEM). Ray-casting methods [31] can also be used for fixing facet orientations. ISO2mesh [32] is a free available MATLAB/octave-based toolbox used for mesh generation and processing. It is used to create tetrahedral meshes from surface meshes, and 3D binary and grayscale volumetric images such as segmented MRI/CT scans. However, it fails to handle self-intersecting triangle pairs and small-angle triangles.

Wang and Yu [33] proposed a mesh smoothing scheme based on surface fitting used for discrete, general-purpose mesh models. Initially, this smoothing scheme starts with the projection of each vertex onto the fitted surface. For a detailed study of molecular mesh generation and molecular visualization, please refer to the recent survey paper [34]. The curvature is used to label all the vertices into four categories. Finally, post-processing modules are used to improve mesh quality. The experimental results reveal that this method [33] can handle different types of meshes, including molecular meshes, imaging data, and industrial models with mesh quality improvement and removal of small angles and short edges. However, there is no specific limit for small or large angles that this method can achieve.

Similarly, Cheng and Shi [36] used the restricted union of balls to improve the quality of molecular surface meshes. The method is able to improve the quality of large molecular meshes; however, it has a comparatively lower efficiency than other alternative methods. Similarly, Quan and Stamm used the advancing front approach and proposed a patch-wise molecular surface remeshing method [37] which is based on another previous method [38]. In addition to quality improvement and handling self-intersections, their method [37] fills the holes in molecular surface meshes. However, it fails in complex molecular surfaces.

In summary, there is a significant work on angle improvement and obtuse triangle removal in generic graphical models. However, there is no such algorithm for molecular surface remeshing. To the best of our knowledge, CAF method is the

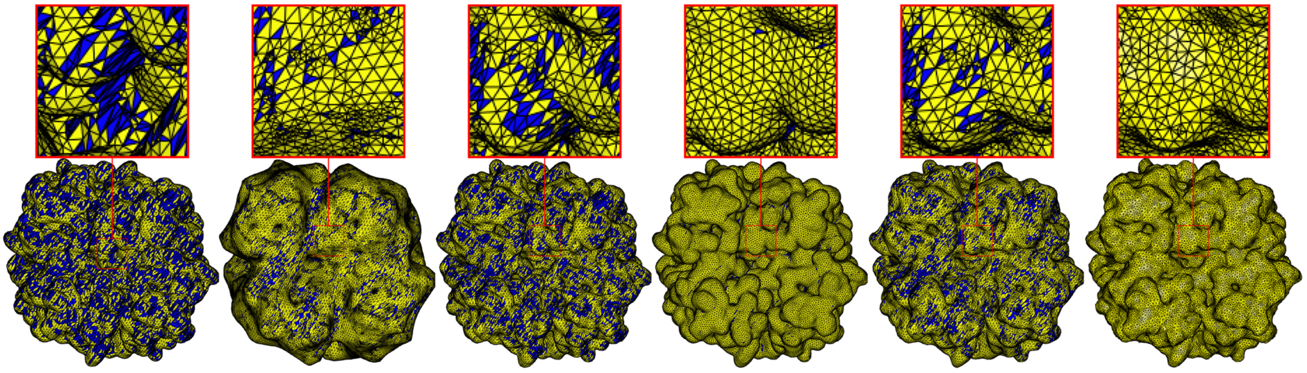


Fig. 3 Results from the state-of-the-art methods in molecular surface remeshing. Blue color indicates triangles with angle(s) smaller than 30°. From left to right, the input mesh (i.e., the mesh generated by TMSmesh 2.0 [5]), ISO2mesh [32], the Taubin method [35], CVT [10], the SMOPT method [3], and CAF method [1]. (PDB ID/ Molecular name $NaR1R4$)

latest method that is specifically focused on angle improvement (see Fig. 3). However, CAF also has a high ratio of obtuse triangles. Furthermore, it has no significant improvement in vertices regularity.

3 Our method

The main steps are given in Algorithm 1. Our method starts with CVT [10] initialization followed by valence optimization. Then, we segment the mesh into random patches and apply RAR [11] method to each patch independently. After this, we eliminate small angles $(\theta < 30^\circ)$. Then, we use a point insertion strategy to minimize the number of obtuse angles. Finally, the segmentation is merged back to a single mesh and local refinements are applied using RAR and Laplacian method. The submodules are described in more detail as follows.

Algorithm 1 Molecular Surface Remeshing

```
1: CVT [10] Initialization.
2: Valence Optimization. // Algorithm 2.
3: Segment and apply RAR to each segment
4: Remove small angles
5: Queue all obtuse angles in pentagonal structures
6: Make holes in pentagonal structures
7: Fill holes
8: Merge segments
9: Repeat steps 4 to 7
10: Apply RAR Locally
11: Local Smooth
12: END
```

3.1 Initialization

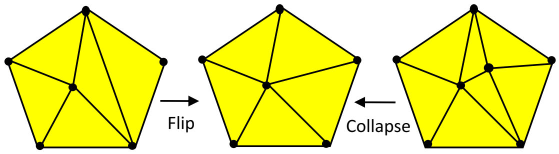


Fig. 4 Valence optimization with edge flip and collapse

The mesh generated from the PQR file is typically found with different issues including zero degree angles, redundant vertices, and self-intersection, which makes the failure of different remeshing operators. Therefore, we apply CVT [10] as initialization (42 iterations as a default value). We select CVT for initialization since it provides a smooth distribution of the vertices in an efficient manner avoiding the major defects. Various methods including [1,39–41] have used CVT either for initializations or to improve it.

3.2 Valence optimization

The valence optimization method is presented in Algorithm 2. For valence optimization, first, we apply two operators including edge flip and edge collapse operators (Fig. 4) with condition that it does not create an angle smaller than 30°.

Algorithm 2 Valence Optimization

```
1: Valence Flip
2: Edge Collapse
3: Delete vertices with bad valences (make holes)
4: for each hole do
5:   if (boundary vertices are 5 or 6) then
6:     Calculate feasible position and insert a single vertex
7:   else
8:     Insert two vertices
9:     Repeat step (1) and (2) (if required)
10:    Apply RAR [11] on newly filled hole
11:  end if
12: end for
13: END
```

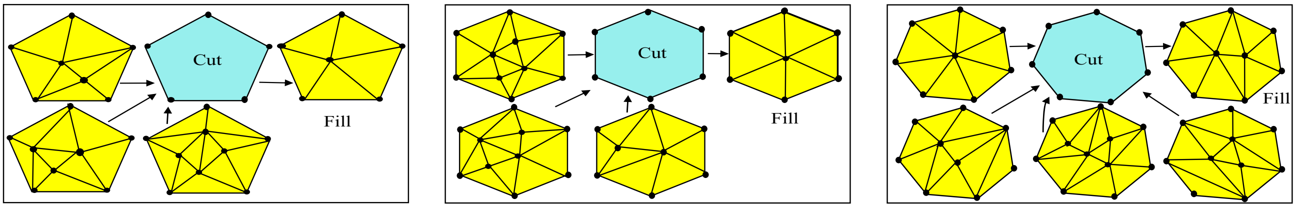


Fig. 5 Vertices removal and filling strategy. From left to right: holes with 5, 6, and 7 boundary vertices

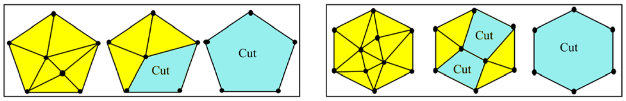


Fig. 6 Two phase vertices removal. First, bad valence vertices (not V567) are removed, and then, bad vertices directly connected to holes are removed. Making a hole with 5 (left) and 6 (right) boundary vertices

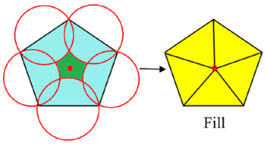


Fig. 7 Hole filling with angle improvement

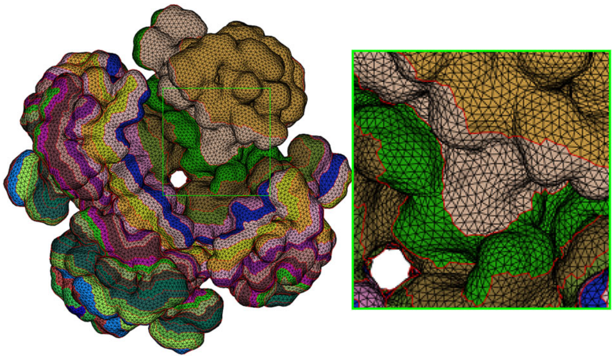


Fig. 8 Mesh segmentation for patch-wise remeshing. The red lines are locked, and each segment is independently remeshed

In addition, we apply a vertex removal and refilling strategy for further valence optimization. We remove all vertices having non-optimal vertices (i.e., not v567). The holes created with vertices removal are filled in a careful manner to improve the regularity of the vertices (see Fig. 5). The vertices are removed in two phases (see Fig. 6). In the first phase, we remove the bad valence vertices with their adjacent edges. In the second phase, we remove vertices directly connected to the holes created in phase 1 and satisfying two conditions. (1) Vertex is a bad valence vertex (not V567). (2) The total of area triangles removed (in both phases) is lesser than $9 \times \overline{A}$, where $\overline{A}$ is the average area of a triangle. These holes are filled carefully as shown in Fig. 5. For the holes with five or six boundary vertices, we insert one vertex, whereas for holes with seven or more boundary vertices, we insert two vertices. In case of the two vertices insertion, if the valence of the vertices at the hole boundary is not optimal, it is optimized with edge collapse and edge flip operators.

The new points are inserted carefully to maintain the angle quality, i.e., to avoid the creation of small or large angles. In this regard, for a single point insertion (i.e., v5 and v6 vertices) we draw circles/sphere around each boundary edge of the hole. The region outside all these circles/spheres is called feasible region for vertex insertion. Figure 7 shows a v5 vertex insertion in a calculated feasible region (the green color). Vertex insertion at this feasible position improves the regularity as well as the angles. For two vertices insertions, we also try to follow the same rules with an additional constraint that the edge between the two vertices must long enough that the opposite angles on both sides are 30° or above. In case, if these conditions are not satisfied, we filled the holes and apply the RAR method to the newly filled regions for angle improvements.

3.3 Mesh segmentation

The molecular meshes are typically generated with high complexity, where the common operations sometime either fail or create new issues such as self-intersections. Similarly, deformation and shrinkage are another issue which causes decreasing the area/volume of the input model (see results of the ISO2mesh [32] in Fig. 3). To solve these issues, we follow a divide-and-conquer approach [42] of segment-based surface remeshing. We divide the mesh into random patches and apply the remeshing operations (RAR method [11]) to each patch independently. The lines drawn for segmentations are locked during this patch-wise remeshing. We used RAR method [11] for the segment-based remeshing because this method is optimal for local modification. It is a simple method with easy implementation. More importantly, it is computationally efficient. RAR uses an adaptive module $L(x)$ which calculates edge length L. Edges having a length longer $\frac{4}{3}L$ are split, whereas edges shorter than $\frac{4}{5}L$ are collapsed.

Figure 8 shows the mesh segmentation used in our method. Segment-based surface remeshing on one side addresses the issue of mesh complexity by dividing the input mesh into segments. It preserves the area and volume of the input model, by locking the segmentation boundaries and thereby avoiding the possible deformation and shrinkage during surface

Fig. 9 Vertex translation

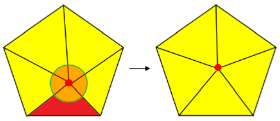
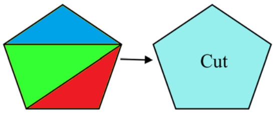


Fig. 10 Making a hole (to be refilled) for obtuse triangle (red color) removal



remeshing. Furthermore, it helps in features and topology preservations. After remeshing each segment with RAR method, and angle improvements, the segmentation boundaries are unlocked and the mesh is further refined with local smoothing (Sect. 3.6).

3.4 Small angle improvement

Our method eliminates all angles smaller than 30°. For small angle removal, we used edge-based operators including edge collapse, edge flip, and vertex translation. Initially, short edges having an opposite angle smaller than 20° are collapsed. Then, we locally apply the Laplace smoothing to the vertices having small angles ($< 30°$). Finally, we delete all vertices with small angles and refill the holes by directly connecting the boundary vertices of the holes. Then, we apply RAR and Laplace smooth to newly filled holes. Similarly, we apply vertex translation to each vertex of each small triangle. For vertex translation, we calculate the new position via Laplace operator. In case, if it improves the small angle, the translation is applied, otherwise the new position is calculated near the old position of the vertex. The translation is only applied if it improves the small angle and does not introduce a new small angle. This process is repeated until all small angles ($< 30°$) are removed.

3.5 Maximal angle improvement

One of our research aims is to improve the maximal angle and minimize the ratio of obtuse triangles. The angles have been generally improved via RAR (Sect. 3.3) and further refined via small angle improvement (Sect. 3.4). With small angle improvement, the maximal angle is also improved, and all the triangles with angle $\geq 120°$ are removed. To further improve the maximal angle, we label the angles greater than the maximal angle threshold as bad angles. Similarly, the triangles with a bad angle are labeled as bad triangles. Then, we apply special operators (vertex translation and point insertion) to each bad triangle. The maximal angle threshold is changed dynamically. First, we set the maximal angle threshold as 110° and translate one vertex (i.e., the central vertex of the bad angle) of each bad triangle randomly around its old position to reach a feasible position where the triangle has no angle $\geq 110°$. The same process is repeated for maximal angle threshold of 100° and 90°, respectively. Figure 9 shows the vertex translation, where the circle indicates the feasible region for vertex translation and red triangle is the bad triangle. For each bad triangle, the vertex is translated if it improves the maximal angles and does not introduce a small angle ($< 30°$), otherwise the translation is skipped. For each of these thresholds, the process is iterated 20 times.

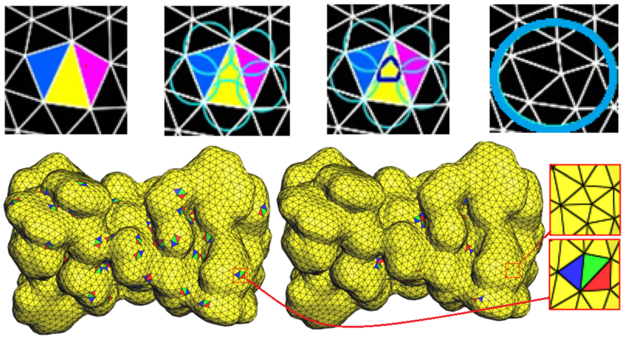


Fig. 11 Pentagonal structures and obtuse triangle removal. Top: an example from a 2D non-obtuse meshing method [40]. Bottom: an example with a molecular surface mesh. Bottom left: before point insertion. Bottom right: after point insertion

In addition to vertex translation, we also use hole-filling strategy in pentagonal structures to minimize the ratio of obtuse triangles. Inspired by a recent 2D non-obtuse remeshing method [40], we design pentagonal structures around each obtuse triangle [see Fig. 10 (left)]. The pentagon contains three triangles including an obtuse triangle, its adjacent triangle on the longest edge, and the next adjacent on the longest edge. The two inside edges of the pentagonal structure are removed, which makes a hole [see Fig. 10 (right)]. Like hole filling for valence optimization (see Fig. 7), the hole is filled with a new vertex insertion in a calculated feasible region. We draw circles/spheres around each boundary edge of the pentagon and calculate the feasible region where a new vertex is inserted. Figure 11 shows an example of pentagonal structure design and point insertions for obtuse removal. Unlike 2D meshing [40], we use surface normal to ensure the three triangles are on the same plane. Our method easily eliminates all angles greater than or equal to 120° and also significantly reduces the ratio of obtuse triangles. However, due to the complexity of molecular meshes and several other challenges, we are unable to eliminate all obtuse triangles. For example, if the obtuse triangle has no adjacent triangle on the same plane to make a pentagonal structure for point insertion. Similarly, if the pentagonal structure is not regular enough, the feasible region (see the green region in Fig.7) for new vertex insertion might be empty.

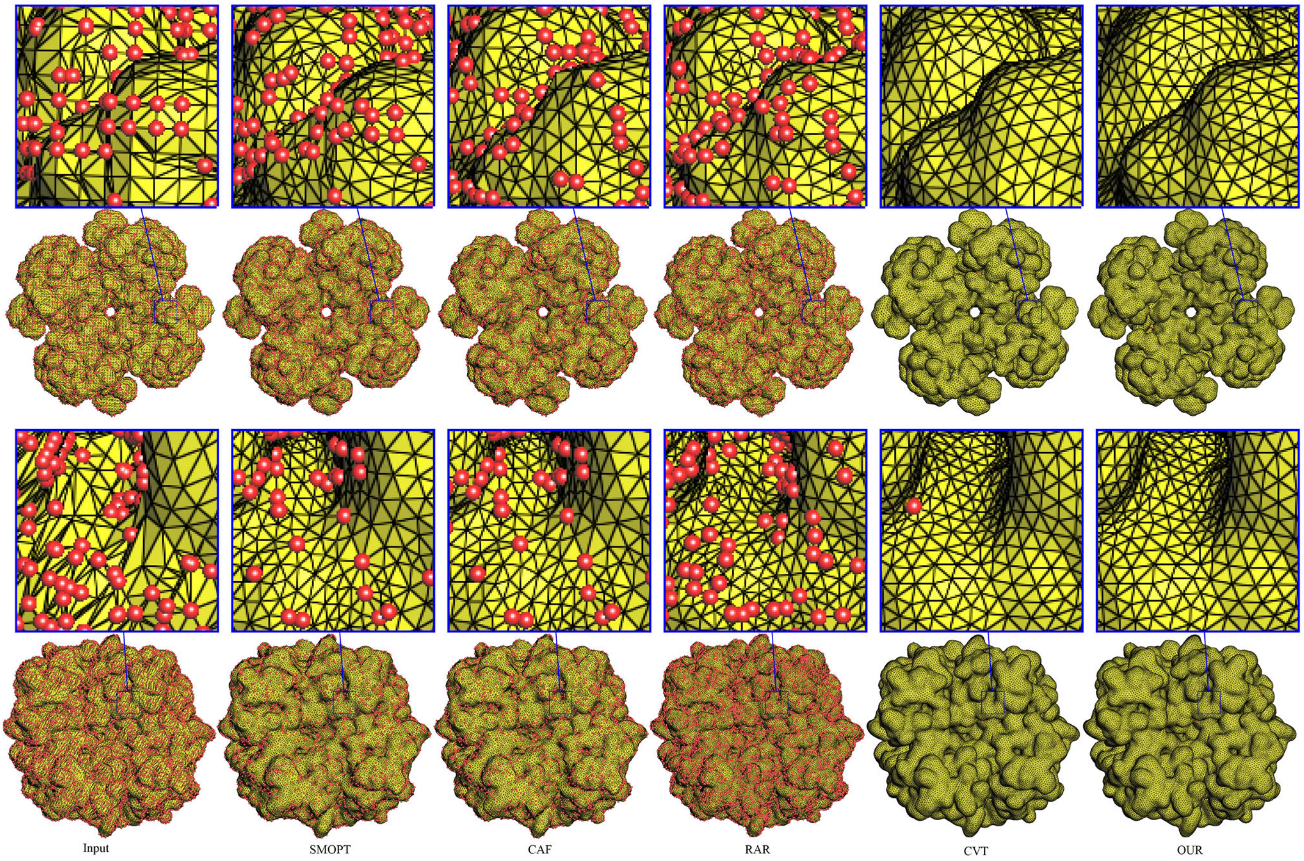


Fig. 12 Valence optimization results. The red color represents irregular vertices (i.e., not V567). From left to right: Top (1bl8): Input (V567: 75.84%), SMOPT (V567: 76.19%) [3], CAF (V567: 79.27%) [1], RAR (V567: 72.88%) [11], CVT (V567: 99.57%) [10], and Our (V567: 99.81%). Bottom (NaR1R4): Input (63.51%), SMOPT (64.96%), CAF (83.95%), RAR (76.52%), CVT (99.92%), and Our (99.96%)

3.6 Local smoothing

We flag the obtuse angles and small angles (if any) and apply local smooth. The local smooth is applied to the obtuse triangle and the triangle directly adjacent to it. The vertices are translated with a Laplace operator if it improves the maximal/minimal angles and does not create small angles. We also apply the RAR with the same constraints of small angles.

4 Experiments and results

We implemented our algorithm using Graphite [43]. We performed the experiments using Intel Core i7 3.60 GHz with 32 GB RAM on a 64-bit Windows 10 operating system. We compared our results with relevant methods including SMOPT [3], CAF [1], RAR [11], and CVT [10].

4.1 Meshing quality parameters

For results analysis, we used different statistical parameters to measure the mesh quality. These parameters include the minimal and average quality of triangle(s) denoted by $Q_{\min}$ and $Q_{\text{avg.}}$, respectively. The quality of a triangle t is calculated as: $Q(t) = \frac{6}{\sqrt{3}}\frac{A_t}{p_t h_t}$, where A_t is the triangle area, p_t is half-perimeter, and h_t is length of the longest edge of triangle [44]. The percentage ratio of the regular vertices (vertices with valence 5, 6 or 7) is also calculated which is denoted by $V567$.

Similarly, minimal angles ($\theta_{\min}$), the average value of minimal angles $\overline{\theta}_{\min}$, maximal angle ($\theta_{\max}$), and the aspect ratio (AR) of triangles (maximum and average) are noted. AR represents the ration of the circum-radius to the twice of the in-radius of a triangle. It is calculated as:

$$\text{AR} = \frac{abc}{8(S-a)(S-b)(S-c)};$$

where a, b, and c are the lengths of three sides of a triangle and $S = \frac{a+b+c}{2}$. An equilateral triangle has AR equal to 1. A high AR suggests low triangle quality. The smaller value indicates good quality unless AR = 1 [42].

In addition, the percentage ratios of the obtuse triangles ($\geq$ 90°) and the triangles with small angle ($<$ 30°)

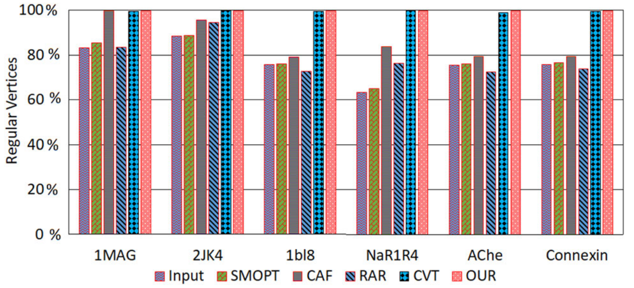


Fig. 13 The % value of regular vertices (V567), and comparison with the state-of-the-art methods using different molecular surface meshes

are also calculated. Furthermore, the number of atoms for each molecule, the area, volume, genus, self-intersections, and applicability in the downstream application (TetGen and AFMPB) are also recorded. In summary, our results are in three main categories including vertices regularity (valence optimization), mesh quality ($Q(t)$, $\theta_{\min}$, $\theta_{\max}$, etc.), and applicability in the downstream applications.

4.2 Valence optimization results

The state-of-the-art methods often fail for molecular surface remeshing. To the best of our knowledge, a recent method called CAF [1] gives comparative good results achieving an output mesh with an angle bound of [30° 120°]. However, this method has no significant improvement in valence optimization. Furthermore, the ratio of obtuse angles is also high. Our method attempts to handle these two issues with consideration of area and volume preservation and avoiding self-intersections. Figure 12 shows the valence optimization results where the bad valence vertices are highlighted in red. The results show that our method can achieve better results than CAF [1] with a considerable improvement in valence optimization. Similarly, Fig. 13 shows a chart of comparison of our method with existing methods in terms of the regular vertices (V567). The ratio of regular vertices is found higher than CVT. The quantitative results are shown in Table 1.

4.3 Mesh quality results

From the state-of-the-art, we found that some methods have improvements in angle quality, while others have significant improvements in vertex regularity. Some methods fail to remove self-intersections. Our method attempts to give an optimal output mesh having a significant improvement in all these parameters. Table 1 shows the quantitative results of our method and few existing methods including SMOPT, CAF, RAR, and CVT. Figure 14 shows the visual results in the form of different molecular meshes. The obtuse triangles are highlighted in red color. Our method not only maintains the angle bound of [30° 120°] like CAF [1] but also has a significant improvement in obtuse angle removal and provides higher regularity than CVT.

4.4 Applicability in the downstream applications

Table 2 shows results concerned with area and volume preservation during remeshing, and the success/failure of the downstream applications on the output meshes. AFMPB (solvation energy calculation) and TetGen (volumetric meshing) are used as downstream applications. Furthermore, we noted the Genus and counted the number of self-intersecting triangles. We found our method has no self-intersection and always succeeds in the downstream applications. Furthermore, it has minor changes in the area and volume of the input mesh. However, RAR method preserves the input genus better than our method.

4.5 Discussion

The literature review reveals that molecular surface remeshing is an active research area with special challenges. For general graphical models, there are very good methods to improve mesh quality. However, these methods often fail for molecular surface remeshing. There are a number of methods specifically used for molecular surface remeshing. However, the molecular surface meshing has no significant improvements. For example, SMOPT is a recent molecular surface remeshing method that can remove all intersecting elements; however, it fails to remove small and large angles. Similarly, CAF is another recent method for molecular surface remeshing that improves the minimal and maximal angles; however, the valence optimization is not satisfactory.

The results reveal that our method improves the angles to an angle bound of [30° 120°]. The ratio of the obtuse triangles is also significantly reduced. Furthermore, our method outperforms in terms of vertex regularities than existing methods including CVT. There was no defect found that causes the failure in the downstream applications.

Our method improves mesh quality in an optimal way and provides better mesh quality than CAF and other molecular surface remeshing methods. Furthermore, it gives a higher valence regularity than CVT and removes all defects that may cause the failure of the downstream applications. Similarly, the area and volume preservations are also significant. However, unlike RAR, the input genus is not preserved with our method. Furthermore, there are two main limitations of our current work. First, our method significantly reduces the number of obtuse triangles; however, it is unable to remove all obtuse triangles. Second, our method is strongly dependent on the previous methods including CVT (for initialization) and RAR.

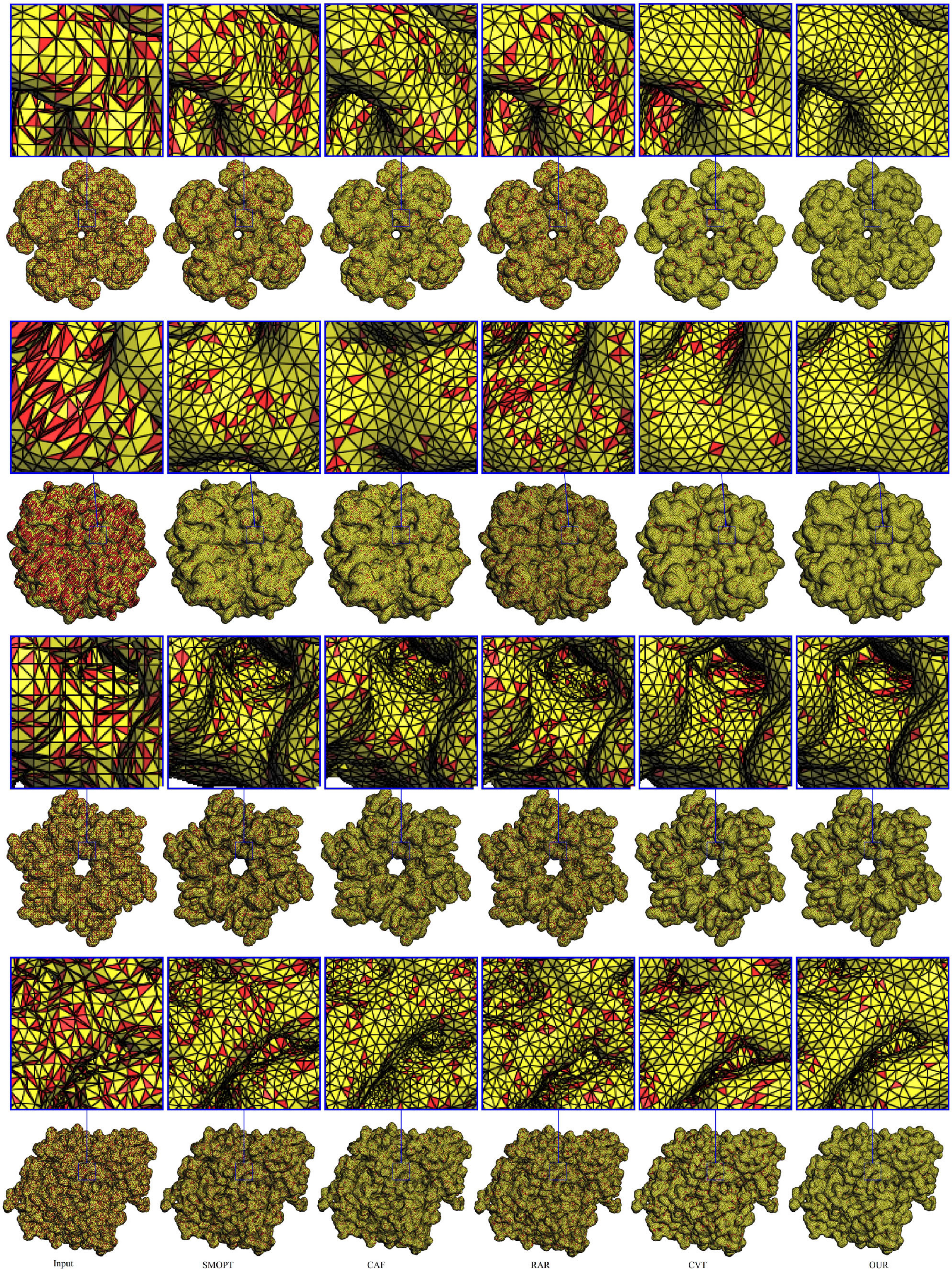


Fig. 14 Surface remeshing results. The red color represents obtuse triangles. From top to bottom, the protein ID or molecular names are: 1bl8, NaR1R4, Connexin, and AChE

Table 1 Comparative surface remeshing results. Our method shows a significant improvement in the mesh quality. The angles are measured in degree. The model names are the PDB IDs/molecular names

Model	Method	#v	$Q_{\min}$	Q_{avg}	$\theta_{\min}$	$\overline{\theta}_{\min}$	$\theta_{\max}$	$\theta < 30°$ (%)	$\theta \geq 90°$ (%)	Reg. v's (%)	$AR_{\max}$	AR_{avg}
1MAG	Input	4824	0.0002	0.6701	0.0069	36.41	176.09	27.67	26.25	83.31	Undef.	Undef.
	CVT	4811	**0.4316**	**0.9024**	25.32	52.15	123.42	0.10	1.32	**99.58**	2.41	**1.04**
	SMOPT	4718	0.0808	0.7958	3.37	44.19	167.94	8.42	17.28	85.59	51.30	1.21
	CAF	4719	**0.5523**	**0.9054**	**30.52**	52.30	109.043	**0.0**	1.13	**99.84**	1.67	**1.03**
	RAR	4888	0.2758	0.8244	15.3	52.48	142.58	2.29	11.19	83.67	5.17	1.12
	OUR	4892	**0.5598**	**0.9043**	**30.4**	**54.01**	**89.58**	**0.0**	**0.0**	**99.93**	**1.55**	**1.03**
2JK4	Input	23,312	0.1137	0.75867	5.20	41.51	161.27	12.97	20.49	88.41	23.71	1.23
	CVT	23,106	0.2880	**0.8899**	11.53	50.84	**117.83**	0.16	**2.02**	**99.97**	3.40	**1.05**
	SMOPT	23,139	0.2700	0.8402	12.09	47.22	142.27	1.53	8.11	88.86	5.24	1.10
	CAF	23,134	**0.4660**	**0.8762**	**30.09**	49.97	**119.77**	**0.0**	3.02	95.73	**2.14**	**1.05**
	RAR	23,505	0.3782	**0.8735**	19.89	**53.13**	129.37	0.12	3.45	94.73	**2.96**	**1.06**
	OUR	23,435	**0.4157**	**0.8843**	**30.05**	**51.10**	**119.86**	**0.0**	**0.93**	**99.73**	**3.10**	**1.07**
1bl8	Input	85,904	0.0	0.5191	0.0	26.30	179.92	56.46	41.0	75.84	Undef.	Undef.
	CVT	85,466	0.0020	0.7262	0.07	39.66	179.32	23.71	20.32	**99.57**	Undef.	Undef.
	SMOPT	85,601	0.0046	0.726	0.22	39.20	179.87	19.04	26.59	76.19	Undef.	Undef.
	CAF	74,623	**0.4679**	0.8141	**30.0**	45.31	**119.53**	**0.0**	14.61	79.27	**2.13**	1.12
	RAR	83,274	0.0070	0.7785	0.46	49.44	179.08	6.95	19.77	72.88	7699.65	1.76
	OUR	78,735	**0.3956**	**0.8961**	**30.3**	**51.10**	**109.14**	**0.0**	**0.5**	**99.81**	**2.15**	**1.04**
NaR1R4	Input	63,310	0.0	0.5135	0.0	26.319	179.43	58.21	36.11	63.51	Undef.	Undef.
	CVT	61,247	0.03452	0.8151	1.62	45.56	174.9	9.24	11.96	**99.92**	Undef.	Undef.
	SMOPT	61,129	0.0437	0.7063	1.88	38.35	173.66	23.41	29.82	64.96	179.55	1.47
	CAF	57,024	**0.4682**	0.8233	**30.0**	45.90	**119.49**	**0.0**	11.83	83.95	**2.12**	**1.11**
	RAR	88,921	0.0241	0.7714	1.49	**49.29**	176.79	8.5	20.17	76.52	641.51	1.22
	OUR	59,781	**0.4615**	**0.8362**	**30.05**	**49.36**	**119.92**	**0.0**	**5.72**	**99.96**	**2.10**	**1.11**
AChE	Input	158,088	0.0	0.5162	0.0	25.95	179.99	57.59	41.53	75.58	Undef.	Undef.
	CVT	150,606	0.0008	0.6708	0.04	35.97	179.86	32.27	26.74	**99.07**	Undef.	Undef.
	SMOPT	155,565	0.0140	0.7250	0.59	39.10	177.19	19.62	26.57	76.16	Undef.	Undef.
	CAF	124,326	**0.4656**	0.8142	**30.0**	45.32	**119.81**	**0.0**	13.26	79.45	**2.14**	1.12
	RAR	153,193	0.0	0.7788	0.0	49.58	179.49	5.81	19.80	72.54	25428	1.80
	OUR	159,543	**0.4643**	**0.8311**	**30.02**	**49.90**	**119.92**	**0.0**	**9.90**	**99.75**	**2.23**	**1.10**
Connexin	Input	107,500	0.0	0.5248	0.0	26.53	179.99	56.21	40.97	75.87	Undef.	Undef.
	CVT	104,255	0.0040	0.7407	0.16	40.41	179.35	20.0	20.0	**99.64**	Undef.	Undef.
	SMOPT	105,645	0.0236	0.7364	1.08	39.85	176.58	16.98	25.37	76.78	614.1	1.46
	CAF	101,574	**0.4658**	0.8158	**30.0**	45.44	**119.79**	**0.0**	20.96	79.46	2.14	1.12
	RAR	110,255	0.0109	0.7848	0.70	49.88	178.55	4.8	18.73	73.81	3134	1.29
	OUR	110,255	**0.4172**	**0.8329**	**30.0**	**50.41**	**119.87**	**0.0**	**13.73**	**99.86**	**2.08**	**1.07**

Table 2 Results concerned with area (unit: A^2) and volume (unit: A^3) preservation and applicability in the downstream applications (AFMPB (unit: kcal/mol) and TetGen). The model names are the PDB IDs/molecular names

Model	Natoms	Method	Area	Volume	Genus	Self-intersect.	TetGen	AFMPB (solv. energy)
1MAG	552	Input	2807.82	4531.35	−1	0	√	4.07610e+01
		CVT	2625.30	4499.45	3	0	√	3.22092e+01
		SMOPT	2195.70	4174.41	3	0	√	2.63009e+01
		CAF	2624.00	4499.73	2	0	√	3.22525e+01
		RAR	**2771.85**	**4530.85**	**−1**	0	√	3.44965e+01
		OUR	**2705.92**	**4529.30**	**−1**	0	√	3.42362e+01
2JK4	4393	Input	14,787.54	37,962.87	4	0	√	−1.59041e+03
		CVT	14,307.57	37,908.52	5	0	√	−1.68641e+03
		SMOPT	12,770.33	37,284.66	6	0	√	−1.63068e+03
		CAF	14,286.88	37,911.41	5	0	√	−1.65811e+03
		RAR	**14,721.21**	**37,964.70**	4	0	√	−1.61630e+03
		OUR	**14,623.46**	**37,920.88**	5	0	√	−1.60238e+03
1bl8	5892	Input	22,094.98	50,919.54	103	0	√	Failed
		CVT	21,166.78	50,688.84	97	0	√	−1.34629e+03
		SMOPT	19,673.38	49,935.09	**103**	0	√	−1.50455e+03
		CAF	19,629.99	50,901.04	58	0	√	−1.39705e+03
		RAR	**21,612.95**	**50,918.6**	**103**	50	×	Failed
		OUR	**21,312.46**	**50,823.17**	32	0	√	−1.34286e+03
NaR1R4	7443	Input	21,301.15	76,894.66	−1	0	√	−1.54776e+03
		CVT	20,480.95	76,807.18	14	0	√	−1.59706e+03
		SMOPT	18,574.04	75,558.13	14	0	√	−1.85589e+03
		CAF	18,723.79	76,881.15	8	0	√	−1.74595e+03
		RAR	**20,924.04**	**76,970.50**	**−1**	2	×	−1.56296e+03
		OUR	**20,511.28**	**76,811.54**	**0**	0	√	−1.59712e+03
AChE	8280	Input	37,653.08	69,058.97	225	0	√	Failed
		CVT	35,760.42	68,699.76	341	0	√	−2.07574e+03
		SMOPT	33,222.25	67,940.30	345	0	√	−2.48270e+03
		CAF	30,099.40	70,371.76	93	0	√	−2.37904e+03
		RAR	**36,734.38**	**69,081.50**	**225**	385	×	Failed
		OUR	**35,978.74**	**69,011.32**	93	0	√	−2.14293e+03
Connexin	19,883	Input	55,604.89	209,415.59	31	0	√	Failed
		CVT	53,272.10	208,731.02	94	0	√	−1.04865e+04
		SMOPT	49,695.69	206,505.02	96	0	√	−1.08436e+04
		CAF	50,806.90	209,318.82	68	0	√	−1.06104e+04
		RAR	**54,600.48**	**209,396.56**	**31**	36	×	Failed
		OUR	**53,938.17**	**209,232.45**	74	0	√	−1.04332e+04

5 Conclusion

We proposed a novel method for valence optimization and angle improvement of the molecular surface remeshing. Our method searches for bad valence vertices and removes it with its neighbor vertices making holes in the mesh. The holes are filled based on the boundary vertices of each hole. In addition, we segment the mesh into random patches and improve the minimal angle and maximal angles in a divide-and-conquer fashion. To minimize the ratio of the obtuse triangles, we extended a 2D non-obtuse meshing algorithm [40]. We found our method with better performance in terms of mesh quality, vertices regularity, area/volume preservation as well as applicable for the downstream applications.

In the future, we are planning for non-obtuse molecular surface remeshing to remove all obtuse triangles. We are also planning to develop a mesh extraction algorithm that extracts surface mesh from PQR files without any major issues.

Acknowledgements This work is partially funded by the Japan Society for the Promotion of Science (JSPS) KAKENHI (19K24346), the National Natural Science Foundation of China (61972388), and Shenzhen Basic Research Program ($JCYJ$20180507182222355). We are thankful to the anonymous reviewers for their valuable comments.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

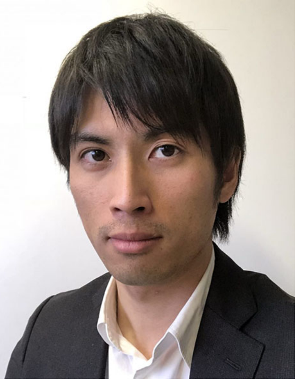
References

1. Khan, D., Yan, D.-M., Gui, S., Lu, B., Zhang, X.: Molecular surface remeshing with local region refinement. Int. J. Mol. Sci. **19**(5), 1383:1–1383:20 (2018)
2. Chen, M., Bin, T., Benzhuo, L.: Triangulated manifold meshing method preserving molecular surface topology. J. Mol. Graph. Model. **38**, 411–418 (2012)
3. Liu, T., Chen, M., Song, Y., Li, H., Lu, B.: Quality improvement of surface triangular mesh using a modified laplacian smoothing approach avoiding intersection. PLoS ONE **12**(9), 1–16 (2017)
4. Dolinsky, T.J., Nielsen, J.E., McCammon, J.A., Baker, N.A.: PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. Nucleic Acids Res. **32**(suppl–2), W665–W667 (2004)
5. Liu, T., Chen, M., Benzhuo, L.: Efficient and qualified mesh generation for Gaussian molecular surface using adaptive partition and piecewise polynomial approximation. SIAM J. Sci. Comput. **40**(2), B507–B527 (2018)
6. Tetgen, H.S.: A Delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw. **41**(2), 11:1–11:36 (2015)
7. Huang, J., Pitsianis, N.P., Sun, X., Lu, B., Zhang, B., Peng, B.: Parallel AFMPB solver with automatic surface meshing for calculations of molecular solvation free energy. Comput. Phys. Commun. **190**, 173–181 (2015)
8. Wang, Y., Yan, D., Liu, X., Tang, C., Guo, J., Zhang, X., Wonka, P.: Isotropic surface remeshing without large and small angles. IEEE Trans. Vis. Comput. Graph. **25**(7), 2430–2442 (2019)
9. Aghdaii, N., Younesy, H., Zhang, H.: 5-6-7 meshes: remeshing and analysis. Comput. Graph. **36**(8), 1072–1083 (2012). (Extended version from GI'12)
10. Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.-M., Lu, L., Yang, C.: On centroidal Voronoi tessellation—energy smoothness and fast computation. ACM Trans. Graph. **28**(4), 101:1–101:11 (2009)
11. Dunyach, M., Vanderhaeghe, D., Barthe, L., Botsch, M.: Adaptive remeshing for real-time mesh deformation. In *Eurographics short papers proceedings*, pp. 29–32 (2013)
12. Liu, Y.-J., Xu, C., Fan, D., He, Y.: Efficient construction and simplification of Delaunay meshes. ACM Trans. Graph. **34**(6), 174:1–174:13 (2015)
13. Cheng, S.-W., Dey, T.K., Shewchuk, J.R.: Delaunay Mesh Generation. CRC Press, Boca Raton (2012)
14. Schreiner, J., Scheidegger, C.E., Fleishman, S., Silva, C.T.: Direct (re)meshing for efficient surface processing. In: Computer Graphics Forum (Proc. EUROGRAPHICS), vol. 25(3), pp. 527–536 (2006)
15. Jakob, W., Tarini, M., Panozzo, D., Sorkine-Hornung, O.: Instant field-aligned meshes. ACM Trans. Graph. (Proc. SIGGRAPH Asia) **34**(6), 189:1–189:15 (2015)
16. Yan, D.-M., Wonka, P.: Non-obtuse remeshing with centroidal Voronoi tessellation. IEEE Trans. Vis. Comput. Graph. **22**(9), 2136–2144 (2016)
17. Hu, K., Yan, D.M., Bommes, D., Alliez, P., Benes, B.: Error-bounded and feature preserving surface remeshing with minimal angle improvement. IEEE Trans. Vis. Comput. Graph. **23**(12), 2560–2573 (2017)
18. Ma, G., Ye, J., Li, J., Zhang, X.: Anisotropic strain limiting for quadrilateral and triangular cloth meshes. Comput. Graph. Forum **35**, 89–99 (2016)
19. Vidal, V., Lavou, G., Dupont, F.: Low budget and high fidelity relaxed 567-remeshing. Comput. Graph. **47**, 16–23 (2015)
20. Gerstein, M., Richards, F.M., Chapman, M.S., Connolly, M.L.: Protein surfaces and volumes: measurement and use. In: International Tables for Crystallography, pp. f:531–545 (2000)
21. Lee, B., Richards, F.M.: The interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. **55**(3), 379–400 (1971)
22. Richards, F.M.: Areas, volumes, packing, and protein structure. Annu. Rev. Biophys. Bioeng. **6**(6), 151 (1977)
23. Bates, P.W., Wei, G.W., Zhao, S.: Minimal molecular surfaces and their applications. J. Comput. Chem. **29**(3), 380 (2008)
24. Edelsbrunner, H.: Deformable smooth surface design. Discrete Comput. Geom. **21**(1), 87–115 (1999)
25. Liu, T., Chen, M., Benzhuo, L.: Parameterization for molecular gaussian surface and a comparison study of surface mesh generation. J. Mol. Model. **21**(5), 1–14 (2015)
26. Chen, W., Zheng, J., Cai, Y.: Kernel modeling for molecular surfaces using a uniform solution. Comput. Aided Des. **42**(4), 267–278 (2010)
27. Chan, S.L.: Molecular surface generation using marching tetrahedra. J. Comput. Chem. **19**(11), 1268–1277 (1998)
28. Sanner, M.F., Olson, A.J., Spehner, J.-C.: Reduced surface: an efficient way to compute molecular surfaces. Biopolymers **38**(3), 305–320 (1996)
29. Men, Y., Shen, Z., Khan, D., Yan, D.-M.: Improving regularity of the centoridal Voronoi tessellation. In: SIGGRAPH Posters, pp. 66:1–66:2. ACM (2018)
30. Decherchi, S., Rocchia, W.: A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale. PLoS ONE **8**(4), 1–15 (2013)
31. Takayama, K., Jacobson, A., Kavan, L., Sorkine-Hornung, O.: A simple method for correcting facet orientations in polygon meshes based on ray casting. J. Comput. Graph. Tech. (JCGT) **3**(4), 53–63 (2014)
32. Fang, Q., Boas, D.A.: Tetrahedral mesh generation from volumetric binary and grayscale images. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1142–1145 (2009)
33. Wang, J., Yu, Z.: A novel method for surface mesh smoothing: applications in biomedical modeling. In: Clark, B.W. (ed) Proceedings of the 18th International Meshing Roundtable, pp. 195–210. Springer, Berlin (2009)
34. Gui, S., Khan, D., Wang, Q., Yan, D.-M., Lu, B.-Z.: Frontiers in biomolecular mesh generation and molecular visualization systems. Vis. Comput. Ind. Biomed. Art **1**(1), 7:1–7:13 (2018)
35. Taubin, G.: A signal processing approach to fair surface design. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95, pp. 351–358. ACM, New York (1995)
36. Cheng, H.-L., Shi, X.: Quality mesh generation for molecular skin surfaces using restricted union of balls. Comput. Geom. **42**(3), 196–206 (2009)
37. Quan, C., Stamm, B.: Meshing molecular surfaces based on analytical implicit representation. J. Mol. Graph. Model. **71**, 200–210 (2017)
38. Quan, C., Stamm, B.: Mathematical analysis and calculation of molecular surfaces. J. Comput. Phys. **322**, 760–782 (2016)

39. Chen, L., Holst, M.: Efficient mesh optimization schemes based on optimal Delaunay triangulations. Comput. Methods Appl. Mech. Eng. **200**(9), 967–984 (2011)
40. Khan, D., Yan, D.-M., Wang, Y., Kaimo, H., Ye, J., Zhang, X.: High-quality 2D mesh generation without obtuse and small angles. Comput. Math. Appl. **75**(2), 582–595 (2018)
41. Yan, D.-M., Wonka, P.: Gap processing for adaptive maximal Poisson-disk sampling. ACM Trans. Graph. **32**(5), 148:1–148:15 (2013)
42. Khan, D., Yan, D.-M., Ding, F., Zhuang, Y., Zhang, X.: Surface remeshing with robust user-guided segmentation. Comput. Visual Media **4**(2), 113–122 (2018)
43. Graphite. http://alice.loria.fr/index.php/software/3-platform/22-graphite.html. Accessed 21 Feb 2020
44. Frey, P., Borouchaki, H.: Surface mesh evaluation. In: 6th Intl. Meshing Roundtable, pp. 363–374 (1997)

Dawar Khan was an assistant professor with the Interactive Media Design Lab, Nara Institute of Science and Technology (NAIST), Japan, from 2018 to 2020. Since June 2020, he has been working as a postdoctoral researcher with the Social Computing Lab, Nara Institute of Science and Technology, Japan. He received the BS and MS degrees in computer science from the University of Malakand, Pakistan, in 2011 and 2014, respectively, and the PhD degree in computer science from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2018. During his PhD study, he was supported by the Chinese Government Scholarship and the National Natural Science Foundation of China. He was awarded with the 2018 Excellent International Graduates Award of UCAS. His research interests include computer graphics, computational geometry, mesh processing, and augmented reality.

Alexander Plopski is a Postdoctoral Fellow at the HCI group where he is working with Associate Professor Tobias Langlotz at University of Otago, New Zealand. Before coming to Dunedin, he worked as an Assistant Professor at the Interactive Media Design Lab at the Nara Institute of Science and Technology (NAIST), Japan. He received his BSc and MSc Informatics at the Technical University of Munich, Germany, in 2010 and 2012, respectively. In 2016, he completed his PhD in Information Science and Technology at the Osaka University, Japan, and joined NAIST in April 2016. His research interests are computer vision, eye tracking, augmented, mixed, and virtual reality.



Yuichiro Fujimoto is an Assistant Professor at the Interactive Media Design Lab at the Nara Institute of Science and Technology (NAIST), Japan. He received the BS degree in engineering from Osaka University, Osaka, Japan, in 2010 and the MS degree and PhD in engineering from Nara Institute of Science and Technology, Nara, Japan, in 2012 and 2015, respectively. From 2015 to 2018, he was an Assistant Professor with the Engineering Department, Tokyo University of Agriculture and Technology in Japan. His research interests include spatial augmented reality and analysis of internal states of human from 2D/3D information.

Masayuki Kanbara received a PhD degree in engineering from Nara Institute of Science and Technology (NAIST) in 2002. He was an assistant professor at the Information Science Department at NAIST since 2002. He was a visiting researcher of University of California, Santa Barbara, in 2008–2009. He has been an associate professor at NAIST since 2010. Fields of research are focused on augmented reality, computer vision, and human–robot interaction.

Zhanglin Cheng is an associate professor and PhD supervisor at Shenzhen Key Laboratory for Visual Computing and Analytics (VisuCA), Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences, Shenzhen, China. He received the PhD degree in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences in 2008. His research interests and experience span a wide range of topics including computer graphics, computer vision, and visualization.

Hirokazu Kato is a professor in the Nara Institute of Science and Technology (NAIST), Japan. He received his BE, ME, and Dr. Eng. degrees from Osaka University, Japan, in 1986, 1988, and 1996, respectively. He joined the Faculty of Engineering Science at Osaka University, from 1989 to 1999. In 1999, he joined the Faculty of Information Sciences at Hiroshima City University, Japan. In 2003, he joined the Graduate School of Engineering Science at Osaka University. Since 2007, he has been with the Graduate School of Information Science of NAIST. He studied pattern recognition and computer vision. When he joined the Human Interface Technology Laboratory (HIT Lab) at the University of Washington as a visiting scholar in 1998, he started a research on Augmented Reality which is currently his main research topic. He received the Virtual Reality Technical Achievement Award from IEEE VGTC in 2009, and Lasting Impact Award at the 11th IEEE International Symposium on Mixed and Augmented Reality in 2012.