

Step2: Surface flattening

Revision 1: Written by Dongjun, Oct. 03, 2007. Heat kernel smoothing is implemented and applied to U-shape object and mandible image is presented. Heat kernel smoothing algorithm-SPHERE (modified algorithm) is proposed. The result for simple U-shape object and mandible image is presented.

Revision 2: Written by Dongjun, Oct. 12, 2007. Surface flattening using interpolation is implemented and applied to simple U-shape object. Experiments for comments are included: (1) average distance and optimal σ in heat kernel smoothing & (2) measurement of area distortion of heat kernel smoothing-SPHERE.

Revision 3: Written by Dongjun, Oct. 22, 2007: Area regularization is proposed and its experiment results are provided. The function 'isosurface()' was check and it is found that there are several different types of triangles. Area distortion experiment results (color of each triangle of figure 16-18 and figure 24-31) are corrected. Vertex with distance=0 is added in the plot of optimal σ in heat kernel smoothing.

Revision 4: Written by Dongjun, Oct. 29, 2007: Area-regularized HKS with shortest-distance interpolation is proposed. The experiment results for simple U-shaped object are provided. Answers for the comments are provided, especially comments for figure 15.

1. Data Processing Procedure

- (1) Load the volume data.
- (2) Remove the points with the value '1'.
- (3) Put another slice below the first slice (see '8. issues' for the detail).
- (4) Apply hole-patching algorithm-CLOSING.
- (5) Transform the volume data into isosurface using the function 'isosurface'.
- (6) Compare # of vertices, edges, faces and EC of the original isosurface with those of the hole-patched isosurface.
- (7) Apply heat kernel smoothing.

2. heat kernel smoothing (Original Method): Algorithm

Notation:

p_i : coordinate of each vertex

\bar{p} : mass center

r_i : the radius of p_i , i.e. $r_i = \|p_i - \bar{p}\|$

\bar{r} : average radius

Algorithm:

- (1) Input: Hole-filled isosurface
Parameter: σ , # of iterations
- (2) Calculate maximum degree and obtain neighborhood information.
- (3) Compute weight W_σ for each vertex using heat kernel with σ and neighborhood information.
- (4) Center the object w.r.t. mass center \bar{p} , i.e. $p_i = p_i - \bar{p}$.
- (5) Make average radius $\bar{r}=1$ by $p_i = p_i / \bar{r}$ for all i .
- (6) For $i=1$:[given # of iterations]
 - A. Center the object w.r.t. \bar{p} .
 - B. Make average radius $\bar{r}=1$ by $p_i = p_i / \bar{r}$.
 - C. Heat kernel smoothing $p_i = W_\sigma * p_i$.
- (7) Output: Object mapped onto S^2 .

3. heat kernel smoothing (Original Method) - Simulation Study of Simple U-shaped object

Figure 1 shows original isosurface. The object does not have any hole or handle. Note that the mass center is placed outside of the object. The heat kernel smoothing with $\sigma=1$ is applied to the object. After 100 iterations (figure 2), the surface became smoother. After 200 (figure 3) and 500 iterations (figure 4), the surface became even smoother but it is still U-shaped and the object became thinner. Even though the average radius remains as 1, the radius of longer side is about 1.5 and that of shorter side is just about 0.5.

4. heat kernel smoothing (Original Method) - Mandible image results

Figure 5, 6, 7 show the mandible image after 1000, 2000, 10000 iterations ($\sigma=1$). As the iteration increases, the surface became smoother but the object became thinner, as in

the simulation study.

5. heat kernel smoothing-SPHERE: Motivation

I faced three problems:

- (1) The object did not become close to the sphere.
- (2) The object became thinner (similar as the plane) after much iterations.
- (3) The mass center was not placed inside the object if the object is U-shaped.

[Moo] This is the problem I didn't expected when I originally suggested the method. I think we need to modify the part where we make average radius = 1. Note that the above algorithm $p_i = p_i / \bar{r}$ scales mesh vertices p_i identically by amount \bar{r} so the actual shape is never changed. We need to change the mesh vertices p_i in a spatially adaptive fashion as we discussed during the discussion using the interpolation technique. Try this interpolation technique first and see if it works. Check first if the interpolation scheme works. If not, we will pursue mean curvature flow approach. I will provide some literature.

I thought about projecting the smoothed surface onto the sphere. But this object is not star-shaped and hence the surface cannot be projected onto the sphere without overlapping. Hence, I consider another approach that projects the surface onto the sphere first and then smoothes the surface while overlapped surface is permitted at the initial mapping. The result should be the same as the original heat kernel smoothing because the heat kernel smoothing is distance preserving flattening. So this overlapped surface will be flattened and the place of vertices will be stabilized after sufficient iterations.

[Moo] This is a very good and simple solution and this is one way of removing the overlap and singularity in the map (If you follow the James Gee's talk today, he also talked about mapping without overlap or singularity while preserving the topology. The underlying mathematical idea is the same). This approach will in fact flatten out any mesh surface to a unit sphere. But the problem with this approach is that areas of triangles around the overlapped regions seem to either enlarged or shrink out substantially.

[Moo] It is not enough to come up with any kind of map from mandible surfaces to a sphere. The map has to be really smooth. The simplest way to check if the map is

smooth or not is the following. Suppose a vertex p on the spherical mesh is mapped onto vertex q on the mandible surface. Then you can plot the length $\|p-q\|$ on the spherical mesh. Check if this length is smooth visually. If you see sudden change in color, that means the map is not smooth enough. If you have overlapping region, heat kernel smoothing of those regions will converge to a point making the overlapped regions to shrink down to a point. Hence this procedure will produce unsmooth map.

Hence, the original heat kernel smoothing is modified and heat kernel smoothing-SPHERE is defined as in the next section. Moreover, the mass center is chosen by user so that the mass center is placed inside the object and the mapped surface has least overlapped surfaces.

6. heat kernel smoothing-SPHERE: Algorithm

Notation:

p_i : coordinate of each vertex

\bar{p} : mass center

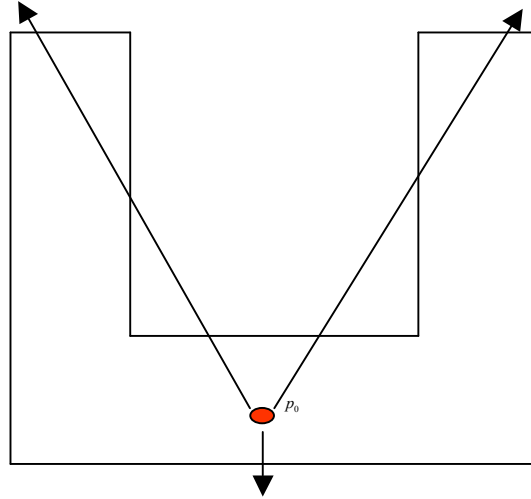
r_i : the radius of p_i , i.e. $r_i = \|p_i - \bar{p}\|$

Algorithm:

- (1) Input: Hole-filled isosurface
Parameter: initial mass center p_0 , σ , # of iterations
- (2) Calculate maximum degree and obtain neighborhood information.
- (3) Compute weight W_σ for each vertex using heat kernel with σ and neighborhood information.
- (4) Center the object w.r.t. p_0 , i.e. $p_i = p_i - p_0$.
- (5) Make $r_i=1$ by $p_i = p_i / r_i$ for all i .
- (6) For $i=1$:[given # of iterations]
 - A. Center the object w.r.t. \bar{p} .
 - B. Make radius of every point $r_i = 1$ by $p_i = p_i / r_i$.
 - C. Heat kernel smoothing $p_i = W_\sigma * p_i$.
- (7) Output: Object mapped onto S^2 .

7. heat kernel smoothing-SPHERE - Simulation Study of Simple U-shaped object

Figure 8 shows the simulation object projected onto the sphere. The initial mass center is given as $p_0 = (6, 15, 15)$, which is placed inside the object. The picture below describes this initial mapping procedure. Red point is the initial mass center p_0 and the vertices are projected onto the sphere w.r.t. p_0 (arrow).



Because some arrows meet the surface more than once, some of surfaces are overlapped. Heat kernel smoothing with $\sigma = 1$ was applied to flatten this overlapped surface. After 100 iterations (figure 9), most of overlapped surface were flattened. After 1000 iterations (figure 10), all of the surface were flattened and the coordinate of vertices were stabilized.

8. heat kernel smoothing-SPHERE - Mandible image results

Figure 11 shows the initial mapping of the mandible image. The initial mass center is given as $p_0 = (160, 170, 30)$ which is placed inside the lower part of the mandible. The surface was overlapped at the initial mapping and it became flattener after 1000 iterations (figure 12). It became even flattener after 2000 iterations (figure 13) of the heat kernel smoothing-SPHERE ($\sigma = 1$) but there are still overlapped surfaces. Hence, more iteration, e.g. # of iteration=10000, will be needed to flatten all the overlapped surfaces.

9. Issues in heat kernel smoothing (original method) and heat kernel smoothing-

SPHERE

(1) Average distance and optimal σ in heat kernel smoothing

I coded MATLAB routine to measure the distance between vertices. Mean and standard deviation of inter-nodal distance are obtained as follows:

Object	Mean of distance	Standard deviation of distance
Simple U-shape object	1.1087	0.2090
Mandible image	0.9860	0.2465

Average distance is around 1 in both simple U-shape object and the mandible image. Assuming that average inter-nodal distance is 1, I did simple experiment to investigate which σ should be chosen in order to make heat kernel smoothing as weighted average rather than simple average. Figure 14 shows the result assuming that there are four neighbors with distance²=0, 0.4, 0.7, 1.0, and 1.2. From this, we can expect that σ might need to be less than 0.5 in order to do weighted average given that average distance is around 1 as in our images.

(2) Area of triangles on original surface

To check how the function 'isosurface()' works, area of each triangle on the original surface was calculated. Figure 15 shows the result of simple U-shape object. There are several different triangles. In short, area of triangles around corners < area of triangles around edges < area of triangles on faces. Area of these triangles and number of triangles are given below for the case of simple U-shape object.

Place of triangles	Area of triangle	# of triangles
On the corners	<0.21	15
Around the edges	0.35	510
On the faces	0.50	2523

(3) Area distortion experiment for heat kernel smoothing-SPHERE

To check how well heat kernel smoothing-SPHERE maps the original isosurface onto

the sphere, I compare the area of triangles between before smoothing and after smoothing. As a criterion for such comparison, area distortion measure of i -th face is defined as follows:

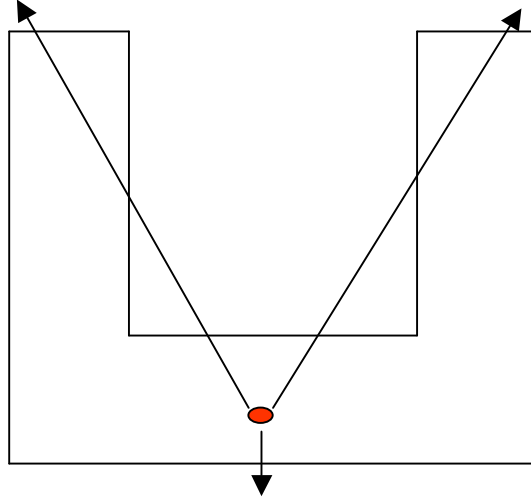
$$\text{area distortion measure}_i = \log \left[\left(\frac{\text{area smoothing}_i}{\text{area original}_i} \right) / \text{mean} \left(\frac{\text{area smoothing}}{\text{area original}} \right) \right]$$

Area distortion measure is positive if the area of i -th face expands and negative if the area shrinks. Larger positive value of area distortion measure means more severe expanding and smaller negative value means more severe shrinking.

Figure 16-18 show the results of the simple U-shape object after iteration 10, 100, and 1000. Heat kernel smoothing-SPHERE was implemented with $\sigma=0.5$. The color of the face is close to red if the area expands and close to blue if the area shrinks. As predicted, there is area distortion around the folded surfaces after 10 iterations (figure 16). After 100 iterations (figure 17) and 1000 iterations (figure 18), all surfaces are unfolded and overall area distortion decreases. However, there is still some area distortion around the originally folded surfaces.

10. Area regularization: Motivation

From chapter 9, we found that heat kernel smoothing-SPHERE cannot prohibit some triangles converge to points even though it removed overlapped surfaces and decreases area distortion. I conclude that some triangles converge to points because the mass center is placed in the middle of U. When points are mapped on the sphere, the farther the points are placed from the mass center, the smaller triangles become on the sphere. Moreover, most of points are placed far from the mass center in U-shape object. So area-preserving mapping may be not so easy in U-shape object (except the case that the mapping itself optimizes the surface with the area-preserving constraints).



I tried to solve the problem with indirect approach. Our objective is to map the original surface onto the sphere while preserving area. Hence, I think that the triangles converging to points can be removed by regularizing the area of triangles. Regularization here means making small triangles larger and large triangles smaller. After such regularization, area of all triangles becomes similar. If we apply the heat kernel smoothing again on this regularized sphere, then coordinate of points are smoothed locally and the result should be close to what we want.

[Moo] What you are doing is not directly related to area preserving mapping. In the area preserving mapping, the algorithm uses the information about the area of triangles in the original mesh. In the area regularization method, the algorithm doesn't use this critical information.

11. Area regularization: Algorithm

Notation:

\bar{a} : average area of all triangles

$\bar{a}_{i,j}$: average area of triangles around j-th neighbor of i-th vertex

p_i : coordinate of each vertex

\bar{p} : mass center

r_i : the radius of p_i , i.e. $r_i = \|p_i - \bar{p}\|$

Algorithm:

(1) Input: Smoothed and non-overlapped isosurface mapped onto S^2 by heat-kernel smoothing-SPHERE

Parameter: α , # of iterations

(2) Calculate maximum degree and obtain neighborhood information.

(3) For $i=1$:[given # of iterations]

A. Calculate \bar{a} .

B. For $j=1$:# of vertices

1. Find the neighbors of j -th vertex.

2. Calculate $\bar{a}_{j,k}$ of j -th vertex.

3. If $\min_k (\bar{a}_{j,k}) < \bar{a}$,

then move p_j closer to its neighbor with largest area

by $p_j = \alpha p_j + (1 - \alpha) p_{j,\max}$ where $p_{j,\max} = \arg \max_k (\bar{a}_{j,k})$

End

C. Center the object w.r.t. \bar{p} .

D. Make radius of every point $r_i = 1$ by $p_i = p_i / r_i$.

(4) Output: Area-regularized object mapped onto S^2 .

[Moo] Let me summarize your algorithm. Without plain English explanation, it is difficult to follow this algorithm. Basically we are moving the vertex only if the minimum of its neighboring average areas is smaller than the global average. We move along the edge that connects to the region of the maximum neighboring average area such that after the movement, the neighboring areas become more uniform.

This is a very nice algorithm and I can see potentials in various applications in graph data structure. The simplicity of the algorithm for doing extremely complicated process (area regularization is a nonlinear problem and hard if you try to solve it exactly). What you are doing is what Fisher once said (Fisher once said that an approximate solution to the correct formulation is way better than the perfect solution to incorrect formulation).

There are two VERY critical issues with the proposed approach.

1] This algorithm is developed in order to unwrap the overlapped singular regions caused by brute spherical projection of the original surface. If the projection technique is more or less properly done, we don't need area regularization. So the area regularization itself is a good idea for other applications, in our application, it simply adds more complexity.

Suggestion: Instead of keep trying to perform spherical projection, try different type of projection. We have discussed the idea of projecting toward the shortest distance. This is not hard to implement. In MATLAB, if you have the $3 \times m$ array K of vertex coordinates. $\text{Mean}(\text{Sum}((K-p).^2,2),2)$ will give you the length vector from given vertex p to every vertex in K . Then finding the index of the shortest distance can be done by $\text{find}(\text{Mean}(\text{Sum}((K-p).^2,2),2) == \min(\text{Mean}(\text{Sum}((K-p).^2,2),2)))$. This is just one example of how to find the vertex index corresponding to the shortest distance to show you it is really trivial. Can you implement this and see if it works. Intuitively it should introduce less area distortion. Then you can apply the area regularization algorithm.

2] The problem with the area regularization algorithm is that it never uses the area information problem the original mesh. In order to make your algorithm similar, in principle, to what area preserving mapping does, you have to use this information. Here is the fix.

Suggestion: Find the neighboring triangle areas from the original surface. Let them be A_1, A_2, \dots, A_m . Let B_1, \dots, B_m be the neighboring triangle areas after above projection. Then we need to have $A_1/B_1 = A_2/B_2 = \dots = A_m/B_m$. If the scale is similar between two surfaces, this ratio is close to 1. If the scale is different, it will be some constant. Try to move the vertex p in such a way that the ratio is preserved as much as possible. Obviously this is a nonlinear problem and may not have a unique solution if you try to solve it exactly. However, there will always a solution in "optimization sense".

Try to find $\min(A_i/B_i)$ and $\max(A_i/B_i)$. Then, applying the area regularization algorithm, move vertex p toward the region of $\max(A_i/B_i)$ only if $\min(A_i/B_i)$ is small. Again this is one suggestion but you really have to use the area information from the original surface.

[Answer] See area-regularized HKS with interpolation.

12. Area regularization - Simulation Study of Simple U-shaped object

Figure 19-21 show the results of area regularization after 10, 100 and 500 iterations. After 10 iterations (figure 19), small triangles started to be enlarged. After 100 iterations (figure 20), area regularization removed all of triangles which had converged to points. After 500 iterations (figure 21), area of all triangles became similar. Note that range of area distortion measure is $-1 \sim +1$. Then, this sphere was smoothed with heat kernel smoothing with few iterations. Figure 22 and 23 show the results after 1 and 10 iterations. After heat kernel smoothing, the sphere became much closer to what we want.

13. Interpolation surface flattening (general form): Algorithm

Notation:

p_i : coordinate of each vertex

\bar{p} : mass center

r_i : the radius of p_i , i.e. $r_i = \|p_i - \bar{p}\|$

Algorithm:

(1) Input: Hole-filled isosurface

Parameter: initial mass center p_0 , σ , α , # of iterations-HKS-PRE,
of iterations-INT, # of iterations-WITHIN

(2) Calculate maximum degree and obtain neighborhood information.

(3) Compute weight W_σ for each vertex using heat kernel with σ and neighborhood information.

(4) Center the object w.r.t. mass center \bar{p} , i.e. $p_i = p_i - \bar{p}$.

Make average radius $\bar{r}=1$ by $p_i = p_i / \bar{r}$ for all i.

(5) Pre-smoothing: heat kernel smoothing with [# of iterations-HKS-PRE].

(6) For $i=1:[\# \text{ of iterations-INT}]$

A. Center the object w.r.t. \bar{p} .

B. Heat kernel smoothing $p_i = W_\sigma * p_i$ with iteration [# of iterations-HKS-WITHIN].

C. Interpolation: $p_i = \alpha p_i^{original} + (1 - \alpha) p_i^{sphere}$,

where $p_i^{sphere} = p_i / r_i$ and $p_i^{original} = p_i / \bar{r}$.

End

(7) Center the object w.r.t. mass center \bar{p} , i.e. $p_i = p_i - \bar{p}$.

Mapping onto the sphere by $p_i = p_i / r_i$.

(8) Output: Object mapped onto S^2 .

14. Interpolation surface flattening: Simulation Study of Simple U-shaped object

Interpolation surface flattening is applied to simple U-shaped object. The experiment was implemented with several different settings. $\sigma = 0.5$ at all experiments.

(1) With pre-smoothing: Figure 24 shows pre-smoothed object using heat kernel smoothing with 150 iterations. After that, interpolation with $\alpha = 0.9$ is applied and heat kernel smoothing was applied once at each iteration of interpolation. Figure 25-28 shows the result after 10, 100, 500 and 1000 iterations. The results are not so different from the results of heat kernel smoothing-SPHERE.

(2) Without pre-smoothing: Figure 29 shows the interpolation result without pre-smoothing. Interpolation with $\alpha = 0.9$ is applied 1000 times and heat kernel smoothing was applied once at each iteration of interpolation. The results are not so different from the results of the interpolation with pre-smoothing.

(3) Different α values: Interpolation surface flattening was implemented with different α values. No pre-smoothing was applied because there is not much difference in the results. Interpolation was applied 1000 times and heat kernel smoothing was applied once at each iteration of interpolation. Figure 30 and 31 show the results with $\alpha = 0.5$ and $\alpha = 0.1$. The results are not so different from the results of interpolation with $\alpha = 0.9$.

(4) More heat kernel smoothing at each iteration of interpolation: Heat kernel smoothing was applied 3 times at each iteration of interpolation. Figure 32 show the result with $\alpha = 0.9$ and 200 iterations. The results were not so different from the results when

heat kernel smoothing was applied once at each iteration.

15. Area-regularized HKS with shortest-distance interpolation: Motivation and Explanation

Previous area regularization algorithm succeeded to make each area of triangles similar. However, area regularization doesn't actually preserve area because it does not use information of original area of triangles. So in order to make the algorithm area-preserving, the new algorithm, area-regularized HKS, uses ratio of current area to original area, instead of current area on the surface as in previous algorithm. In other words, if the area of triangle is distorted w.r.t. its original triangle, then area-regularized HKS regularizes the area.

Moreover, during some experiment with area-regularized HKS, I found good property of area-regularized HKS. Because area-regularized HKS regularizes area at each iteration, the object is smoothed in much smoother way. When the mass center was placed outside the object and original HKS was applied on the surface, the object became thinner and the object went farther from the mass center. See figure 3 and 4. In contrast, when area-regularized HKS is applied, the object is still thick enough and its surface is very smooth even after much iteration. Moreover, the mass center moved closer to the object at each iteration and it moves into inside of the object smoothly after enough iteration. See figure 33 and 34. So we now don't need to pick up the initial mass center manually any more.

In plain language, the algorithm can be described as follows:

[Presmoothing with area-regularized HKS]

First, center and adjust the size so that average radius=1. Then smooth this object with area-regularized HKS iteratively. During the iteration, if HKS shrinks some triangles w.r.t. its original triangles, then regularize the area so that area of triangles are reserved. After enough iteration, the surface of the object becomes smoother and the mass center is placed inside the object.

[Shortest-distance interpolation with area-regularized HKS]

Then interpolate every points of the pre-smoothed object with its closest point on the sphere iteratively. At each iteration of interpolation, several iterations of area-regularized HKS are implemented. After enough iteration, the object is mapped onto the sphere and

its surface is smoothed while the area is preserved.

16. Area-regularized HKS with shortest-distance interpolation: Algorithm

Notation:

p_i : coordinate of each vertex

\bar{p} : mass center

r_i : the radius of p_i , i.e. $r_i = \|p_i - \bar{p}\|$

\bar{r} : average radius

$\bar{a}_{i,j}$: average area of triangles around j-th neighbor of i-th vertex in original surface

when $\bar{r}=1$

$\bar{b}_{i,j}$: average area of triangles around j-th neighbor of i-th vertex in smoothed surface

Algorithm: Presmoothing with area-regularized HKS

(1) Input: Hole-filled isosurface

Parameter: σ , α , # of iterations

(2) Compute weight W_σ .

(3) Center the object and make $\bar{r}=1$.

(4) Calculate $\bar{a}_{i,j}$ for j-th neighbor of i-th vertex, for all i, j.

(5) For i=1:[given # of iterations]

A. Center the object and make $\bar{r}=1$.

B. Heat kernel smoothing $p_i = W_\sigma * p_i$.

C. Area regularization

For j=1:# of vertices

1. Find the neighbors of j-th vertex.

2. Calculate $\bar{b}_{j,k}$ of k-th neighbor of j-th vertex.

3. If $\min_k (\bar{b}_{j,k} / \bar{a}_{j,k}) < 1$,

then move p_j closer to its neighbor with largest area ratio

$$\text{by } p_j = \alpha p_j + (1 - \alpha) p_{j,\max} \text{ where } p_{j,\max} = \arg \max_k (\bar{b}_{j,k} / \bar{a}_{j,k}).$$

End

(6) Output: Smoothed isosurface (mass center is within the mass), $\bar{a}_{i,j}$, W_σ

Algorithm: Shortest-distance interpolation with area-regularized HKS

(1) Input: Pre-smoothed isosurface with area-regularized HKS

Parameter: α , β , σ , $\bar{a}_{i,j}$, W_σ ,

[# of iterations-interpolation], [# of iterations-smooth]

(2) Center the object and make $\bar{r}=1$.

(3) For $i=1$:[# of iterations-interpolation]

A. Center the object and make $\bar{r}=1$.

B. Calculate $p_j^{sphere} = 2 \max_j (r_j) p_j / r_j$ for all j .

C. Shortest-distance interpolation

$$p_i = \alpha p_i^{original} + (1 - \alpha) p_i^{sd},$$

$$\text{where } p_i^{original} = p_i / \bar{r} \text{ and } p_i^{sd} = \arg \min_{p_j} |p_i - p_j^{sphere}|.$$

D. Heat kernel smoothing $p_i = W_\sigma * p_i$.

E. Area regularization

For $j=1$:# of vertices

1. Find the neighbors of j -th vertex.

2. Calculate $\bar{b}_{j,k}$ of k -th neighbor of j -th vertex.

3. If $\min_k (\bar{b}_{j,k} / \bar{a}_{j,k}) < 1$,

then move p_j closer to its neighbor with largest area ratio

$$\text{by } p_j = \alpha p_j + (1 - \alpha) p_{j,\max} \text{ where } p_{j,\max} = \arg \max_k (\bar{b}_{j,k} / \bar{a}_{j,k}).$$

End

(4) Center the object and make $r_i = 1$.

(5) Output: Object (smooth and area-regularized) mapped onto S^2 .

17. Area-regularized HKS with shortest-distance interpolation: Simulation study of simple U-shaped object

Figure 33 shows the pre-smoothed surface with area-regularized HKS with $\sigma = 0.1, \alpha = 0.9$ after 300 iterations. We can see that the object did not become thin. Figure 34 is the same object as figure 33, which is seen from the above. We can see that the mass center is placed inside the object. Here, the mass center is the origin because the object is centered at each iteration. Figure 35 shows the object after 100 iterations of shortest-distance interpolation with $\beta = 0.9$. At each iteration of interpolation, 5 iterations of area-regularized HKS with $\sigma = 0.1, \alpha = 0.9$ were implemented. There are no more triangles converging to points. Area corresponding to edges on original surfaces is still smaller than area corresponding to faces on original surfaces even after mapping onto the sphere.

[Moo] Suggestion for writing MATLAB. Make following functions separately: 1) shortest distance projection, 2) area-regularization, 3) heat-kernel smoothing. After step 3, I will ask you to upload the MATLAB codes to webspace and I will test them myself.

Figure 1. Original Surface of simulation object

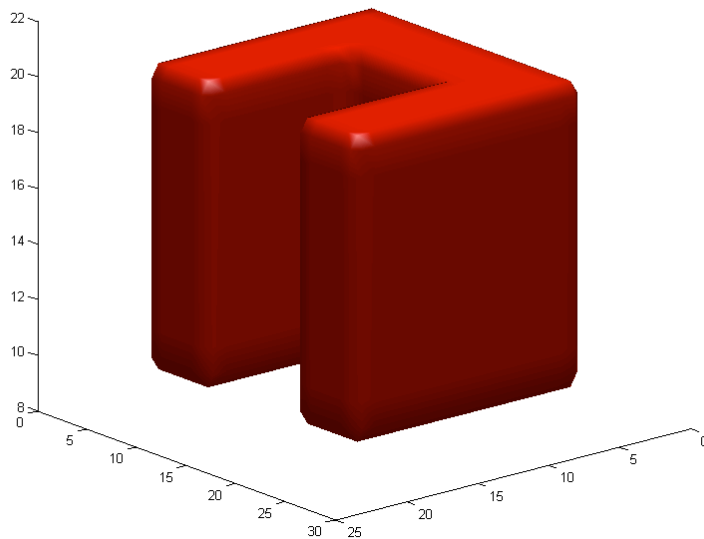


Figure 2. Simulation object after 100 iterations of heat kernel smoothing (original method)

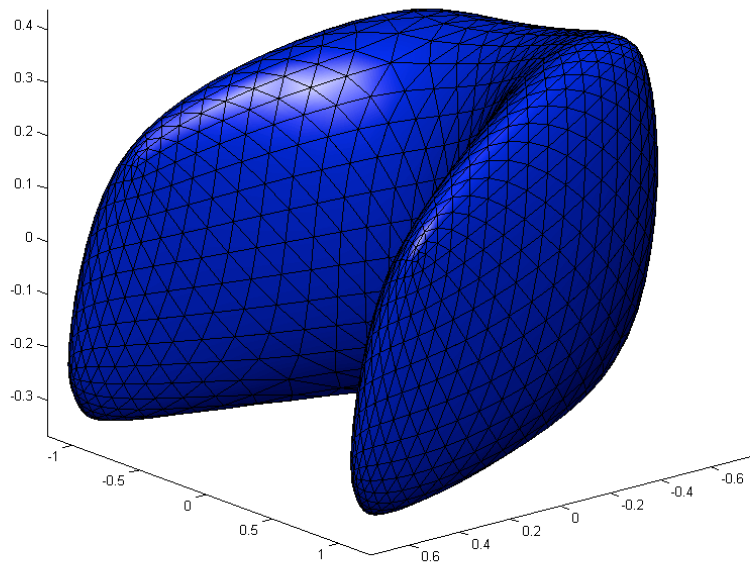


Figure 3. Simulation object after 200 iterations of heat kernel smoothing (original method)

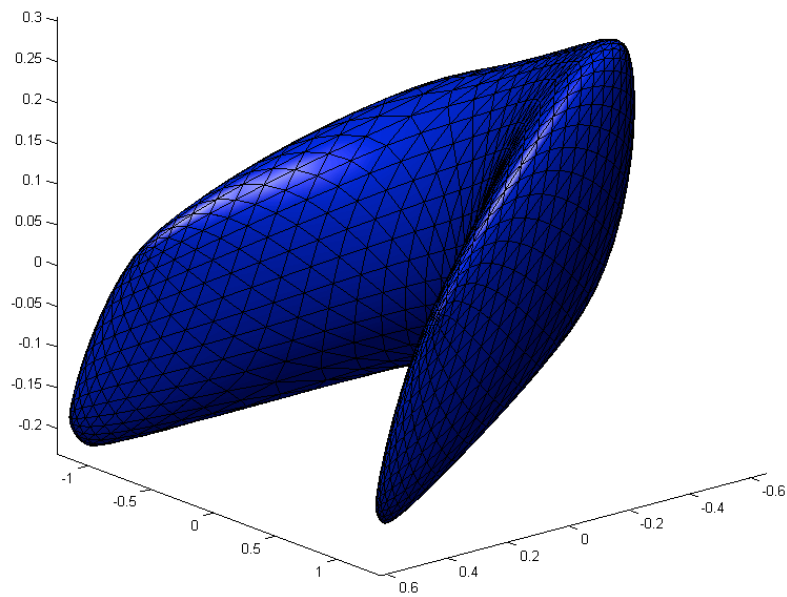


Figure 4. Simulation object after 500 iterations of heat kernel smoothing (original method)

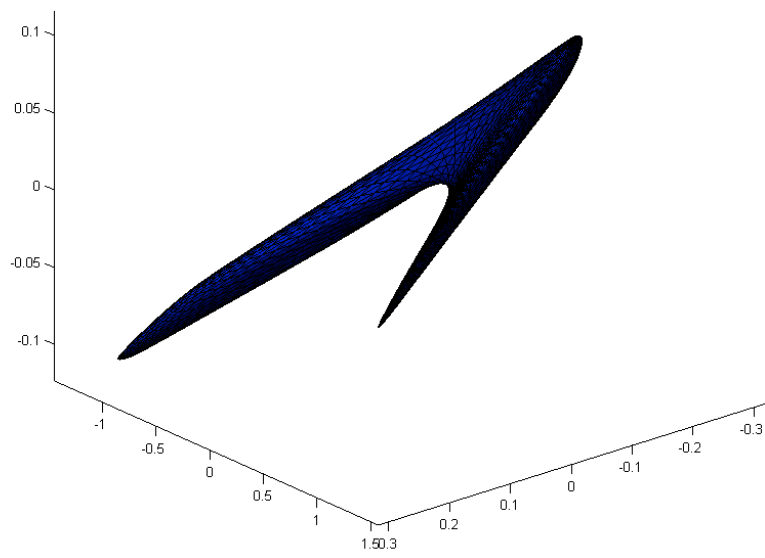


Figure 5. The mandible image after 1000 iterations of heat kernel smoothing (original method)

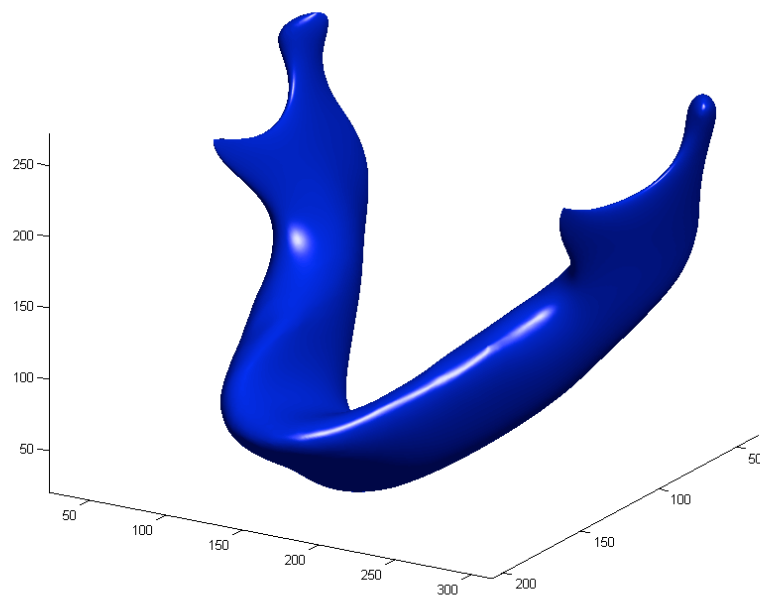


Figure 6. The mandible image after 2000 iterations of heat kernel smoothing (original method)

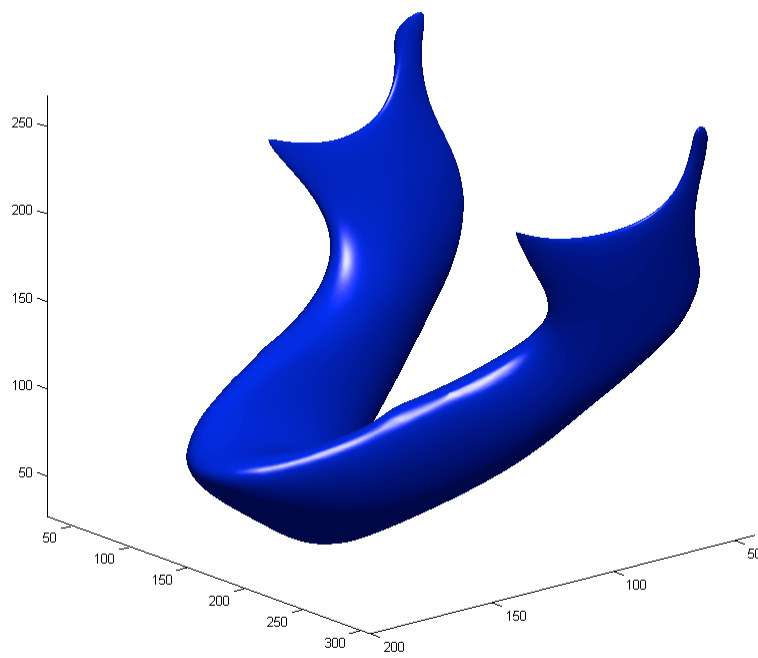
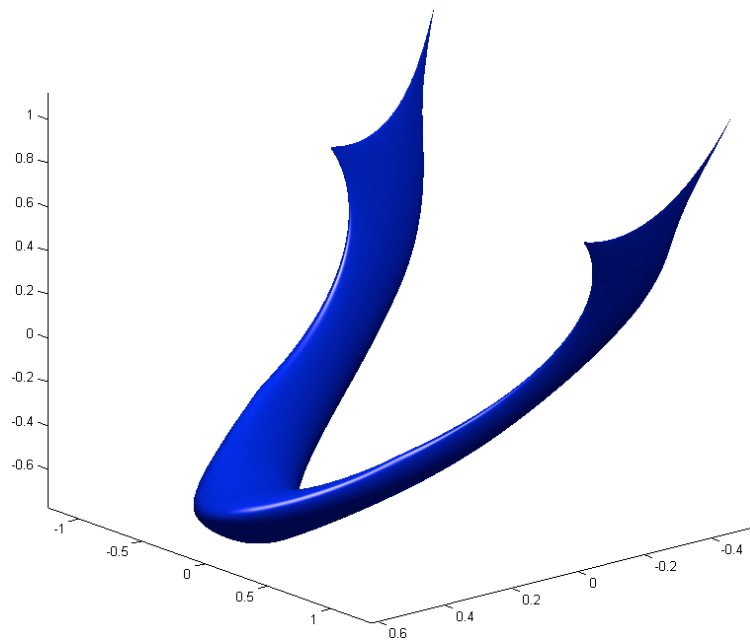


Figure 7. The mandible image after 10000 iterations of heat kernel smoothing (original method)



[Moo] A new research idea. If we perform this process further it seems we can get the so-called skeleton representation of a complex object. So this is another way of reducing very complex geometrical data. This can be a very good future research problem.

Figure 8. The simulation object mapped onto the sphere

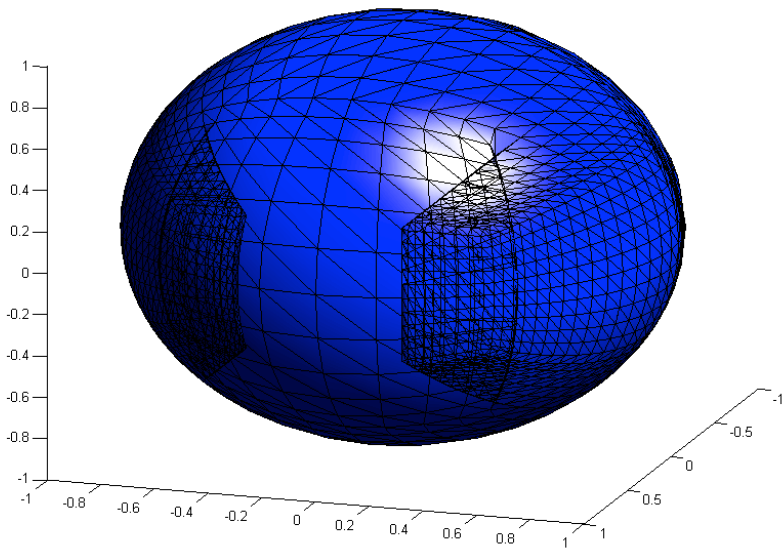


Figure 9. Simulation object after 100 iterations of heat kernel smoothing

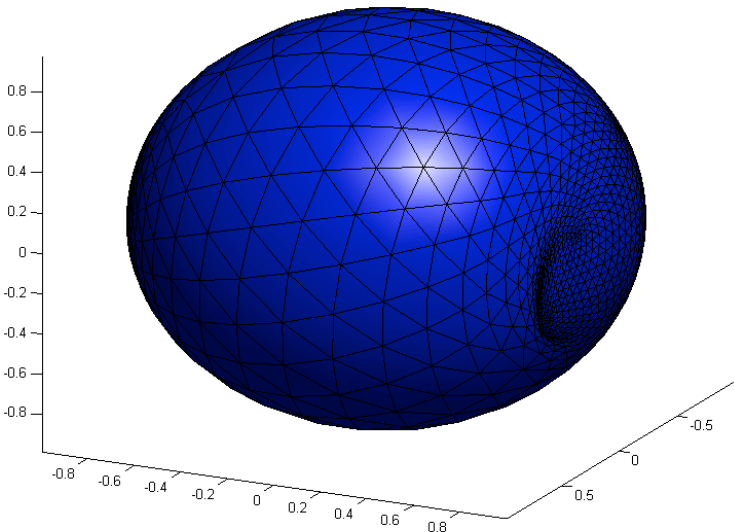


Figure 10. Simulation object after 1000 iterations of heat kernel smoothing

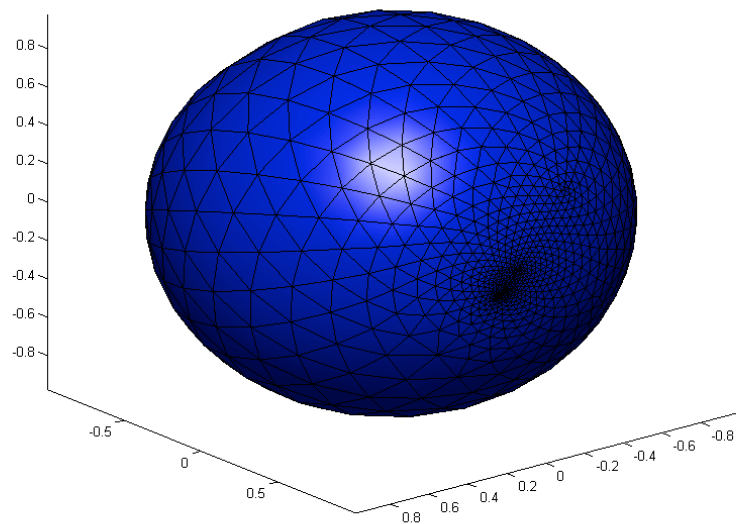


Figure 11. The mandible image mapped onto the sphere

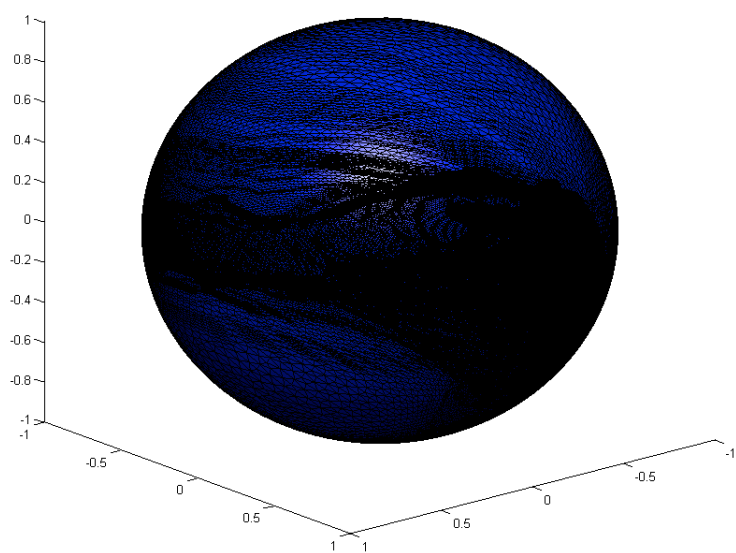


Figure 12. The mandible image after 1000 iterations of heat kernel smoothing

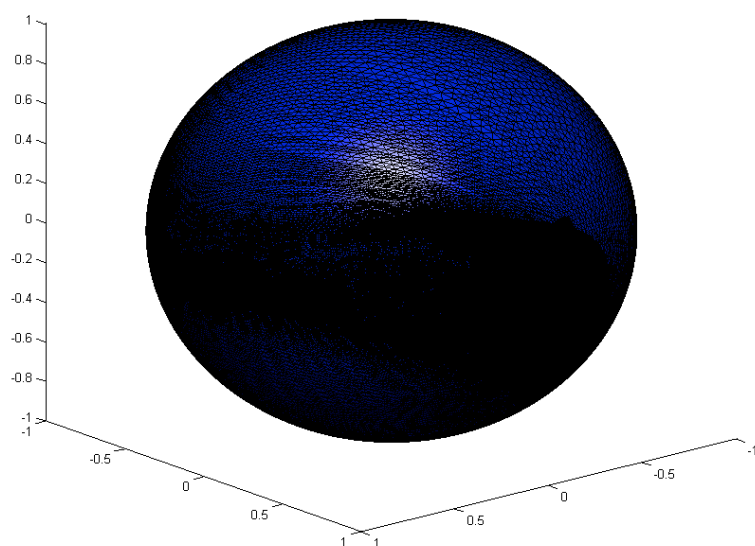


Figure 13. The mandible image after 2000 iterations of heat kernel smoothing

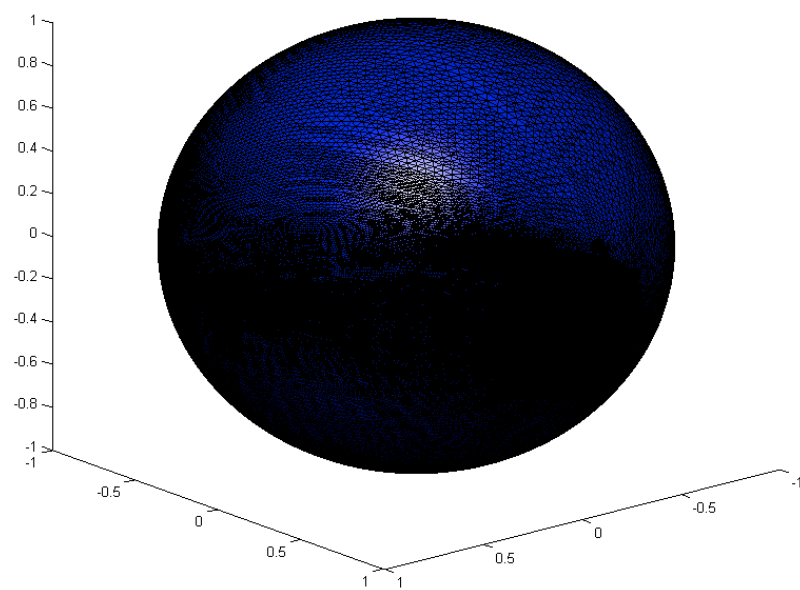


Figure 14. Plot of weight for each neighbor vs. σ , given 4 neighbors with distance²=0, 0.4, 0.7, 1.0, 1.2

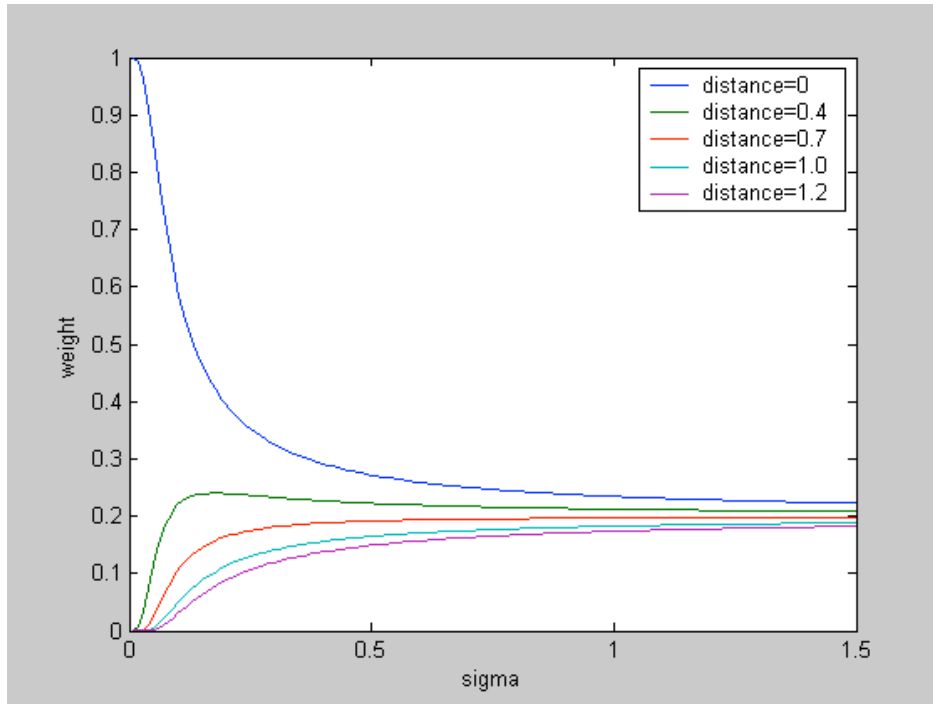
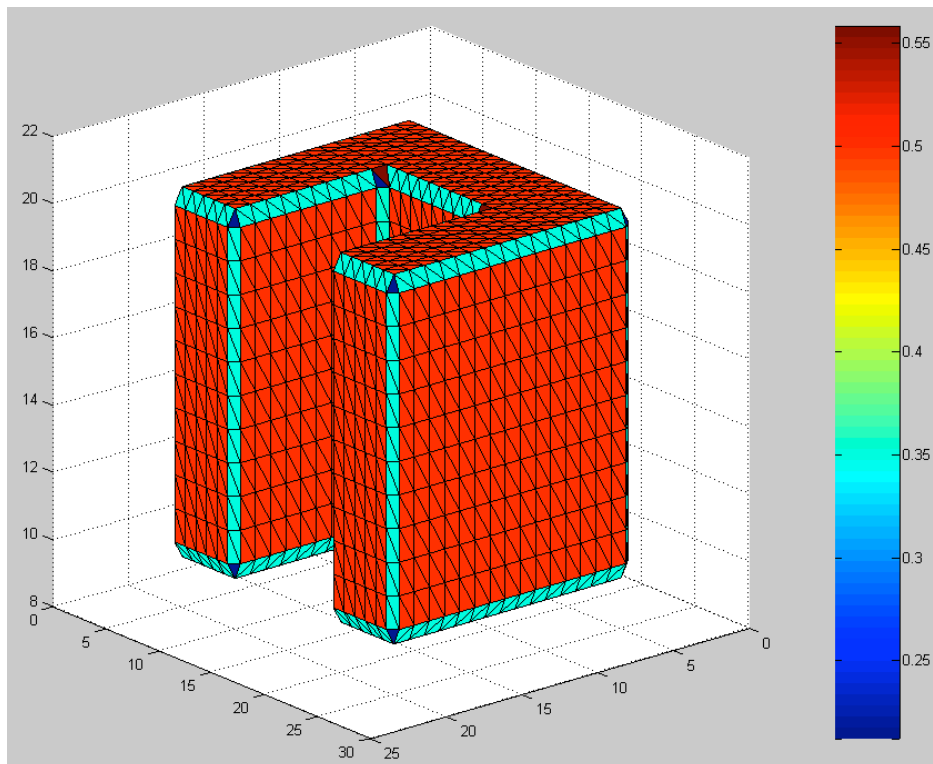
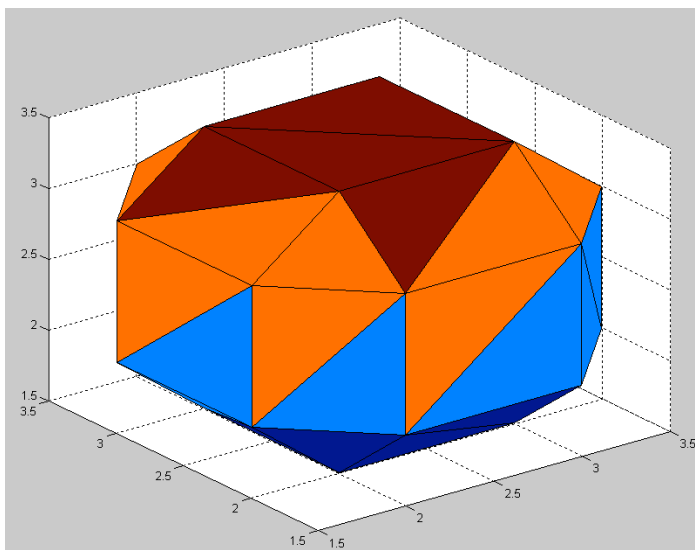


Figure 15. Areas of triangles on original surface of simple U-shape object (triangle areas: around corner<0.21, around the edges=0.35, on the faces=0.50).



[Moo] There is something wrong with this surface. The original binary image must be constructed using 1 and 0. So the edges of the binary object must have 90 degree angle. In this isosurface, the edges are chopped off too smooth out somehow and it has 45 angle. Why?

[Answer] The figure below is figure 3 from 'hole patching-8.Sep.21.2007.doc'. This is the isosurface constructed from the cube $[2,3]*[2,3]*[2,3]$. The cube $[2,3]*[2,3]*[2,3]$ is also binary image with 8 points in the volume data. We can think this cube as the simplified one of figure 15.



The below are 15 unique cube configurations of marching cube. From the above figure, we can see that the triangles on faces correspond to 6th element of configurations. Likewise, triangles of edges correspond to 3rd element and triangles of corner correspond to 2nd element of configurations.

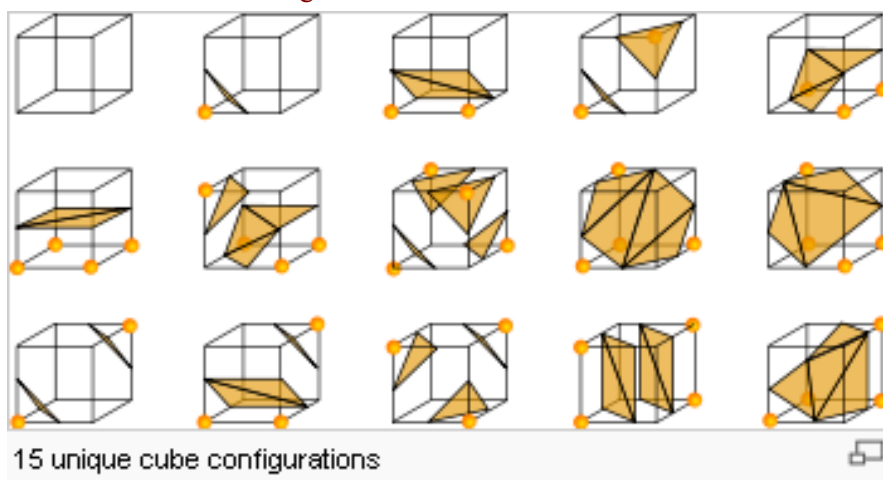


Figure 16. Simple U-shape object after 10 iterations of heat kernel smoothing-SPHERE

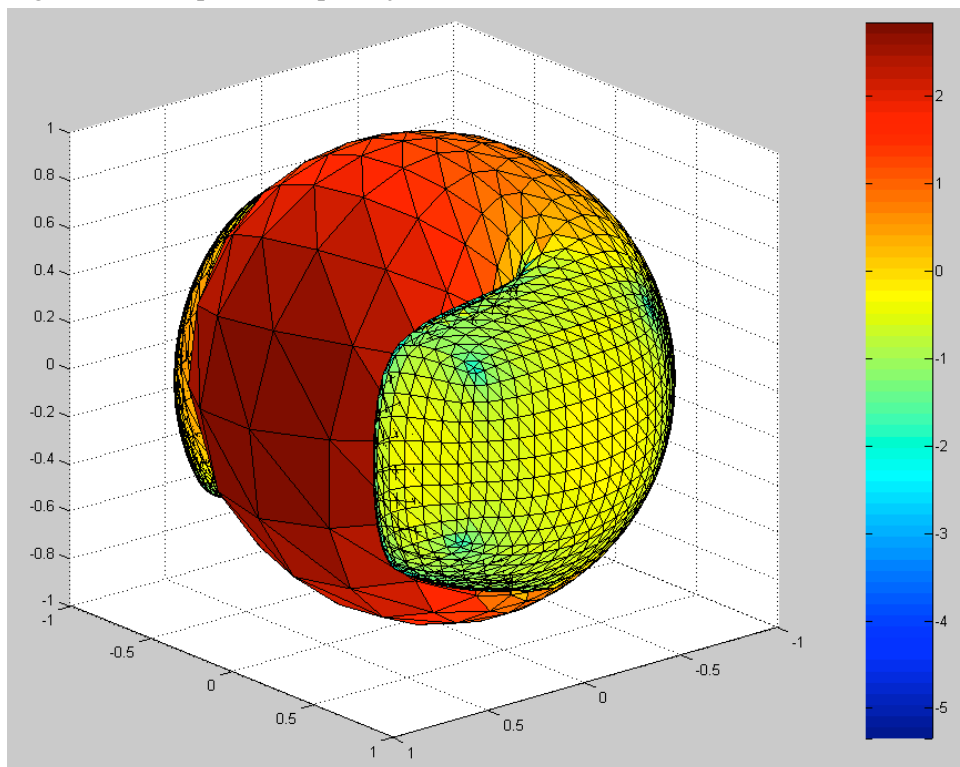


Figure 17. Simple U-shape object after 100 iterations of heat kernel smoothing-SPHERE

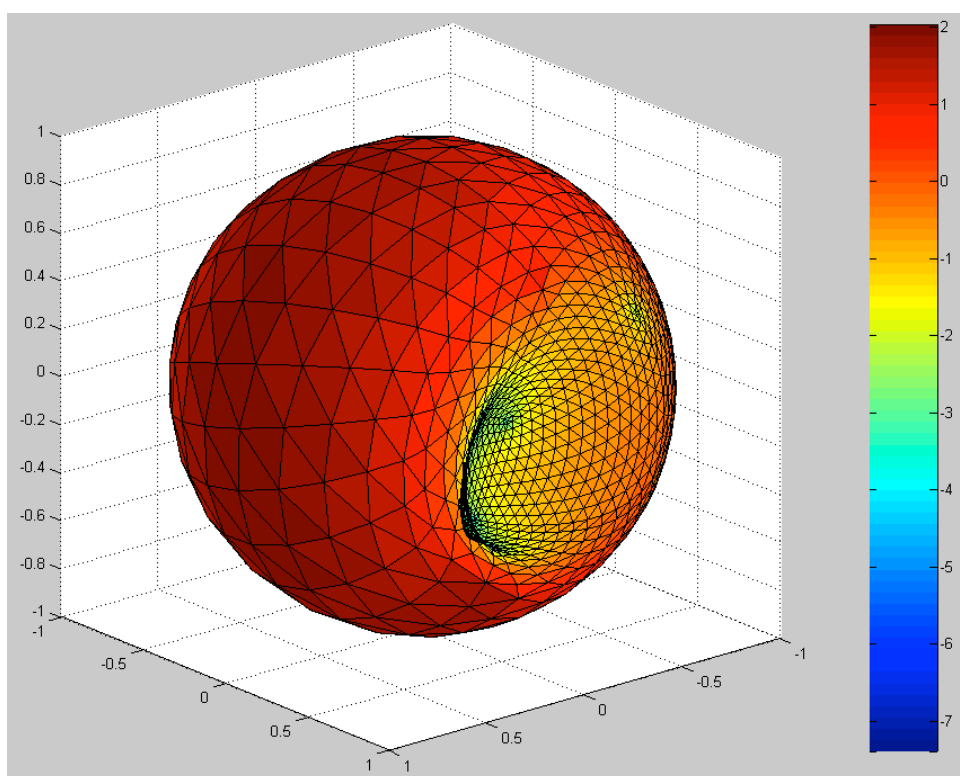


Figure 18. Simple U-shape object after 1000 iterations of heat kernel smoothing-SPHERE

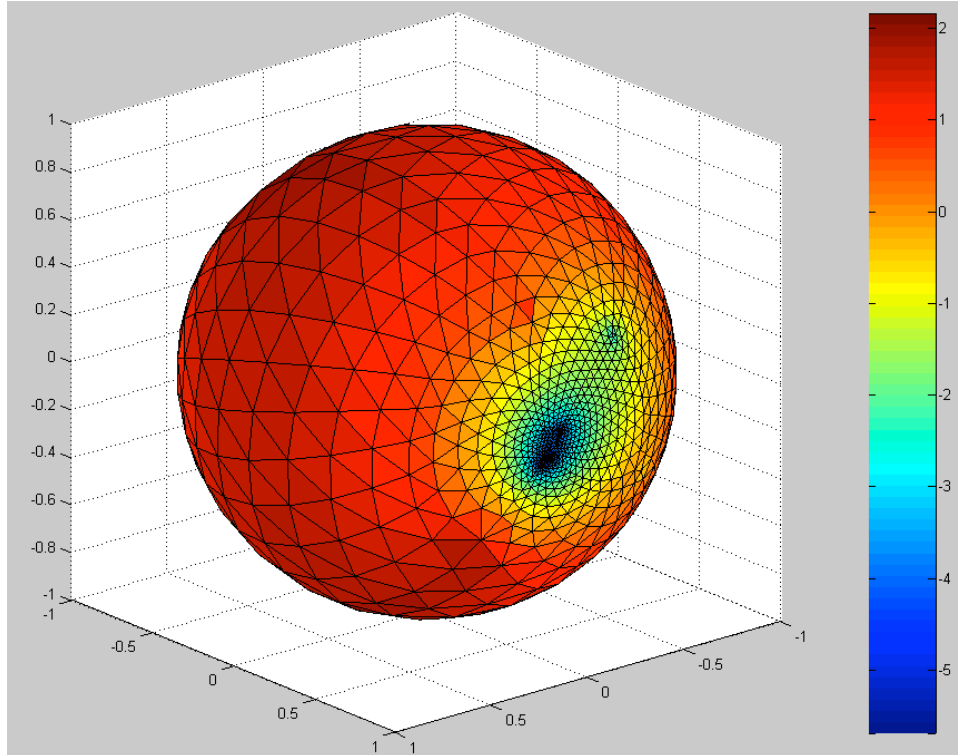


Figure 19. Simple U-shape object after 10 iterations of area regularization with $\alpha=0.9$

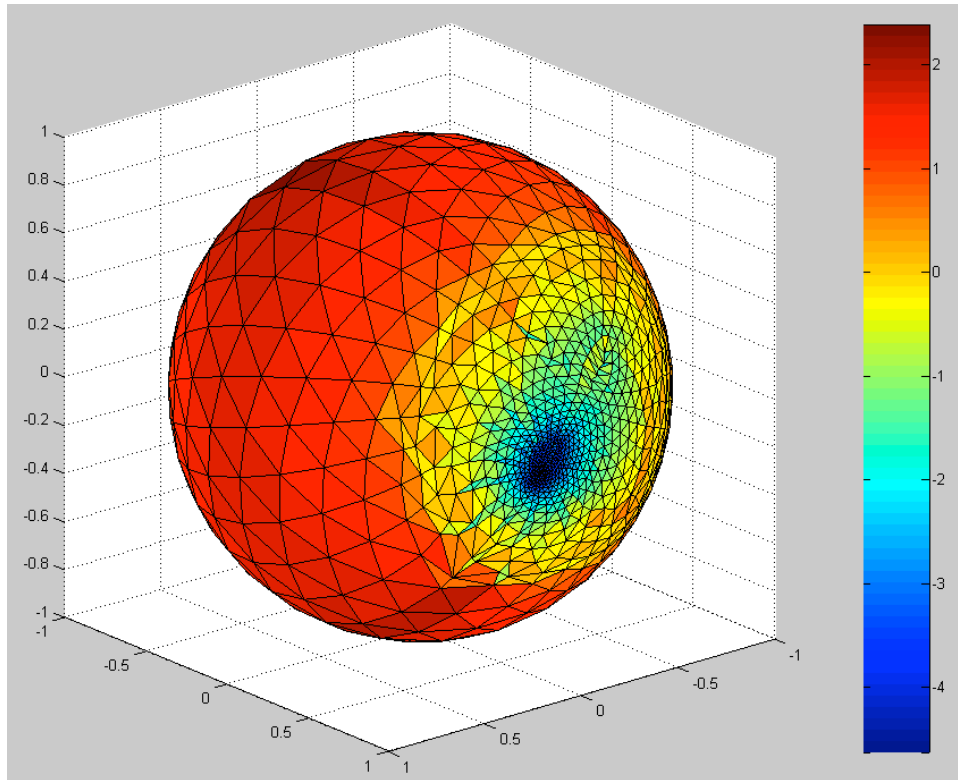


Figure 20. Simple U-shape object after 100 iterations of area regularization with $\alpha=0.9$

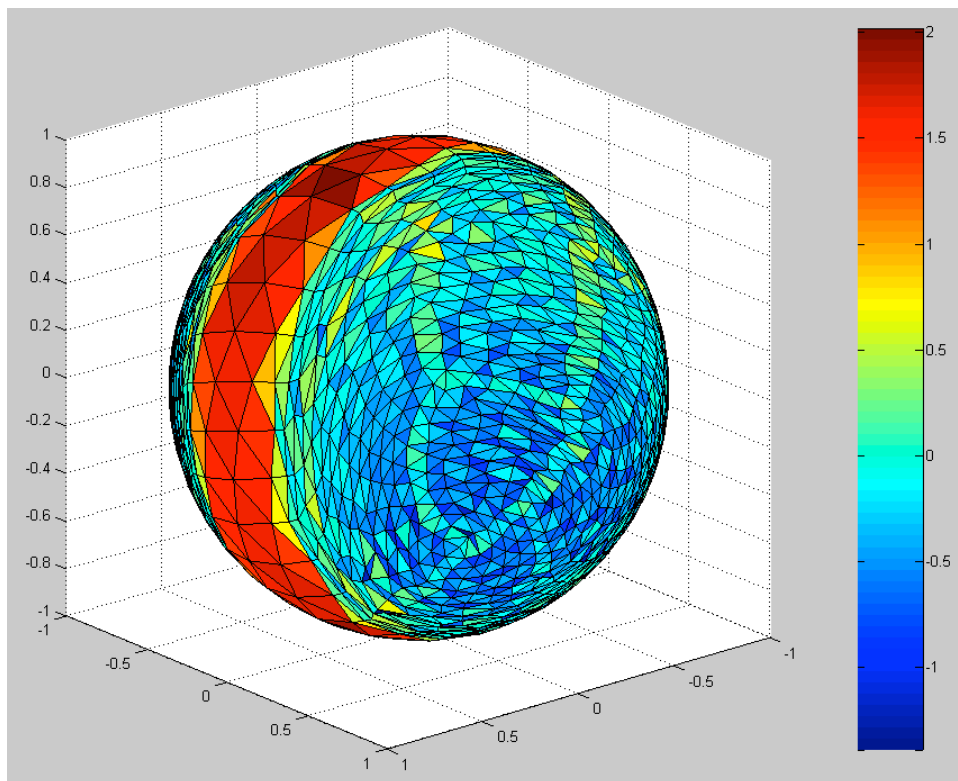


Figure 21. Simple U-shape object after 500 iterations of area regularization with $\alpha=0.9$

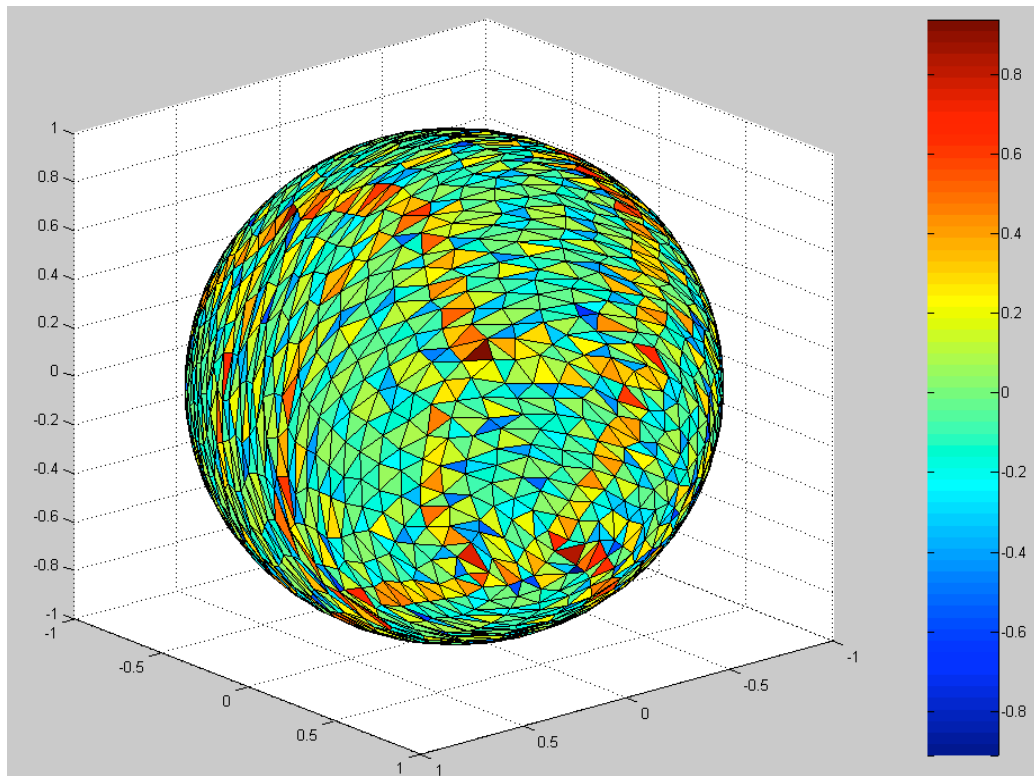


Figure 22. Simple U-shape object after 1 iteration of heat kernel smoothing on the object of figure 21

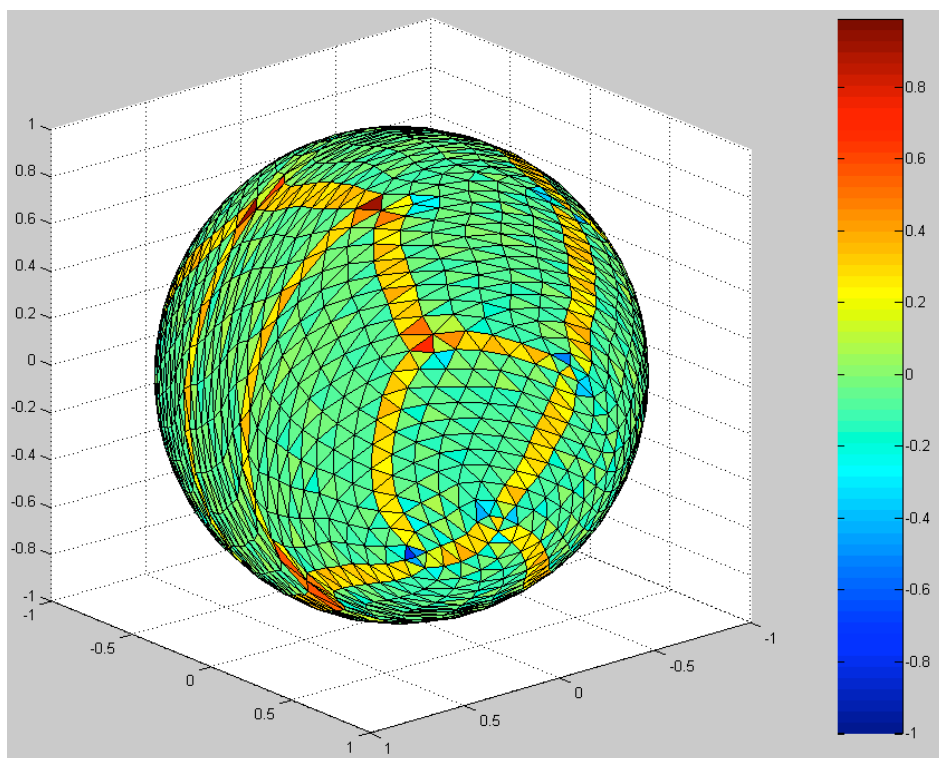
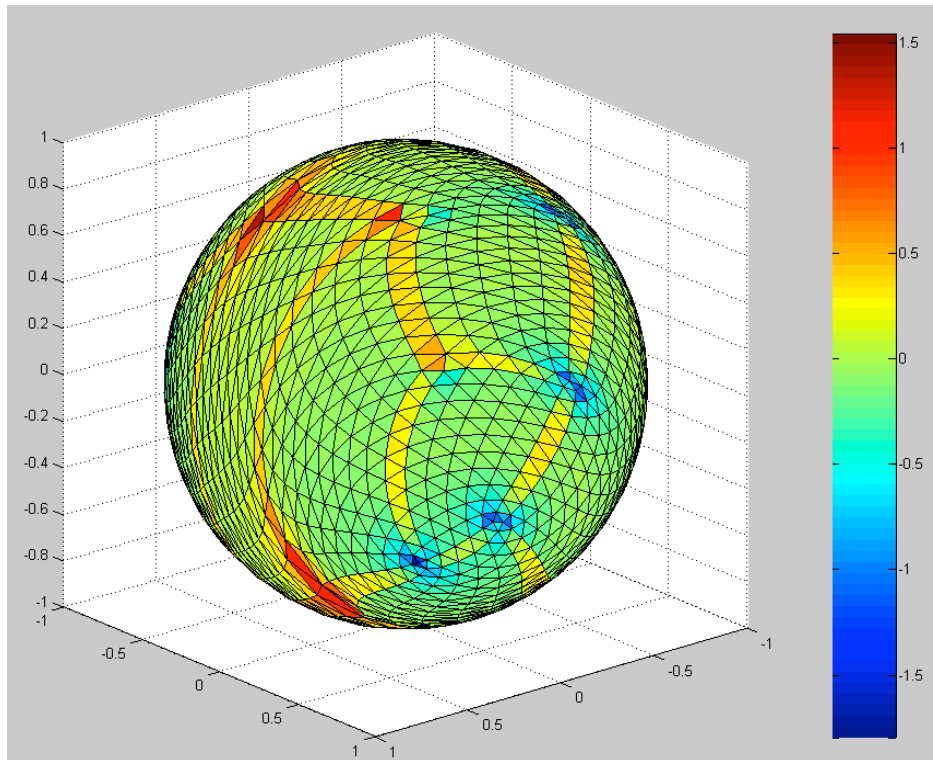


Figure 23. Simple U-shape object after 10 iteration of heat kernel smoothing on the

object of figure 21



[Moo] This figure is deceiving. The area regularization algorithm never directly uses the area of the original mesh while the area distortion metric uses the information about the areas of the original mesh. So this figure showing the edge after heat kernel smoothing is basically a fake. The edges are supposed have smaller triangles while the area regularization map is expected to have more or same area, the area distortion map should be able to detect this pattern.

Also figure 23 should give smooth map, it may not be as good as figure 21 in terms of regularizing areas. I think you should iteratively use area regularization and heat kernel smoothing as follows. Here is the flow of the image processing.

1. Perform heat kernel smoothing on original mesh (to smooth out small kinks in the original surface).
2. Do shortest distance projection to sphere.
3. Perform area regularization with modification $\min(A_i/B_i)$ and get sufficiently regularized surface. In computing this ratio, you have to use the area obtained from the original mesh.

4. Then perform heat kernel smoothing (to smooth out the mapping from original mesh to sphere).

5. Go to step 3 and iterate the process many times.

You may have to toy with different bandwidth, alpha etc. This is one suggestion. Also you have to use it to mandible surfaces since these parameters may depends on the mesh.

[Answer] See area-regularized HKS with interpolation.

Figure 24. Simple U-shape object after 150 iterations of heat kernel smoothing before interpolation

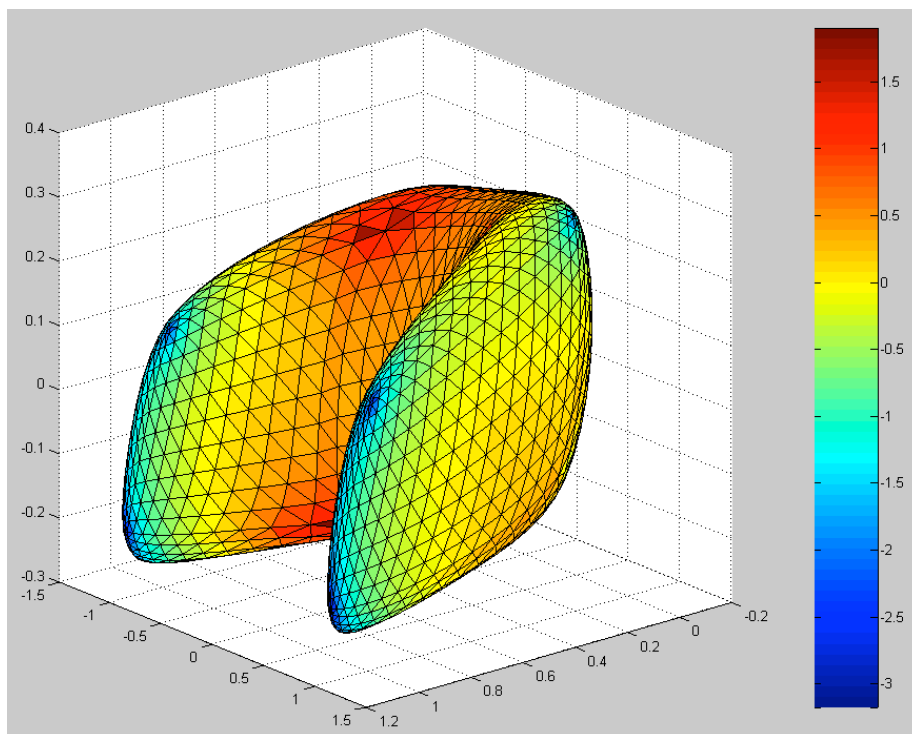


Figure 25. Simple U-shape object after 10 iterations of interpolation with $\alpha = 0.9$ on heat kernel smoothed object in figure 24

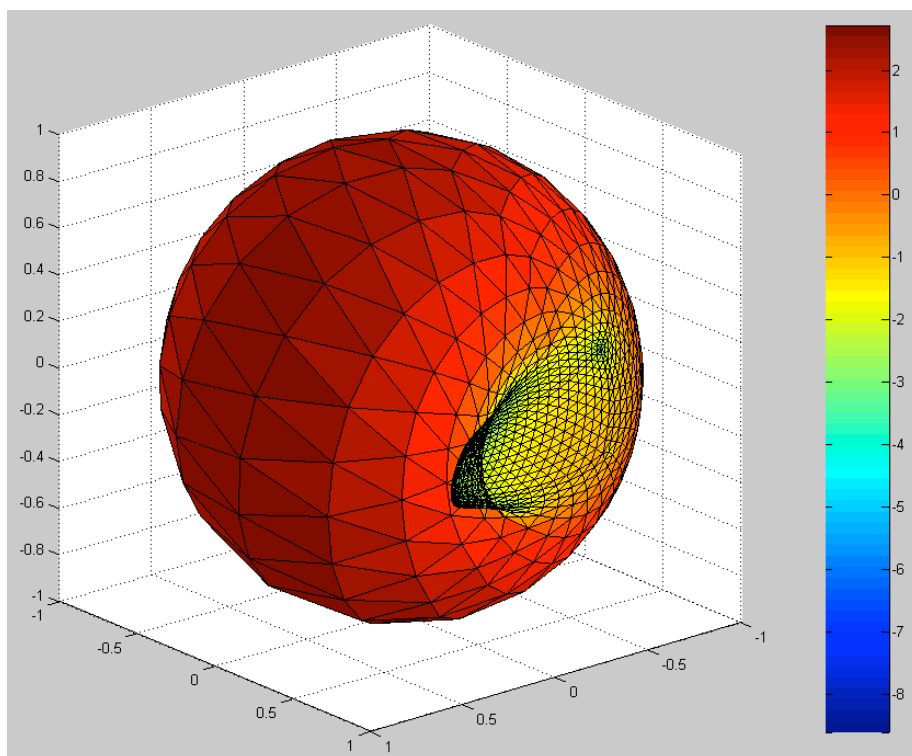


Figure 26. Simple U-shape object after 100 iterations of interpolation with $\alpha = 0.9$ on heat kernel smoothed object in figure 24

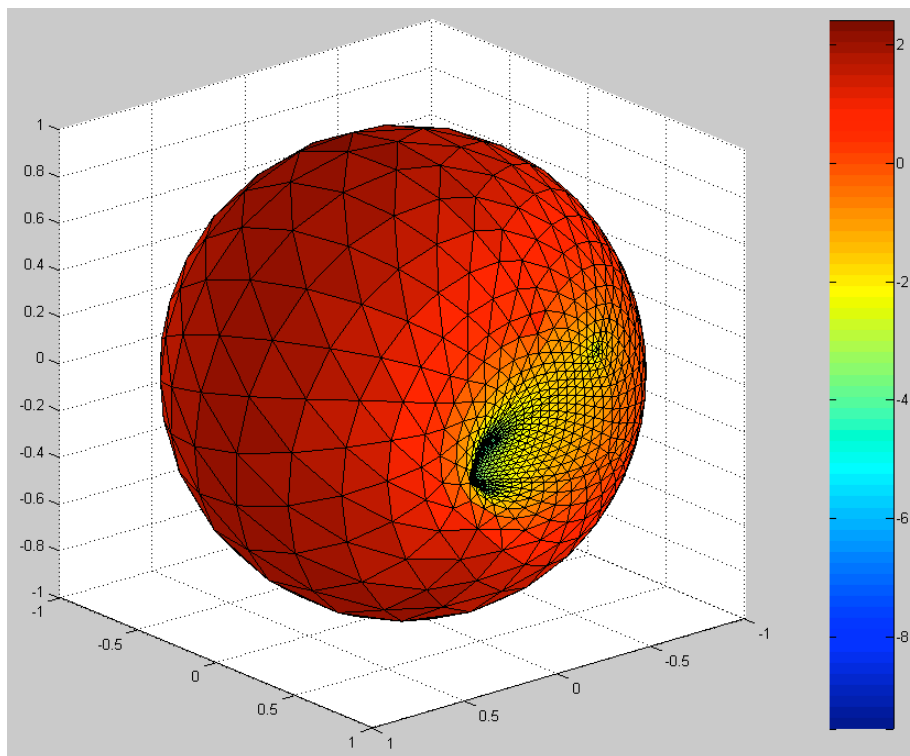


Figure 27. Simple U-shape object after 500 iterations of interpolation with $\alpha = 0.9$ on heat kernel smoothed object in figure 24

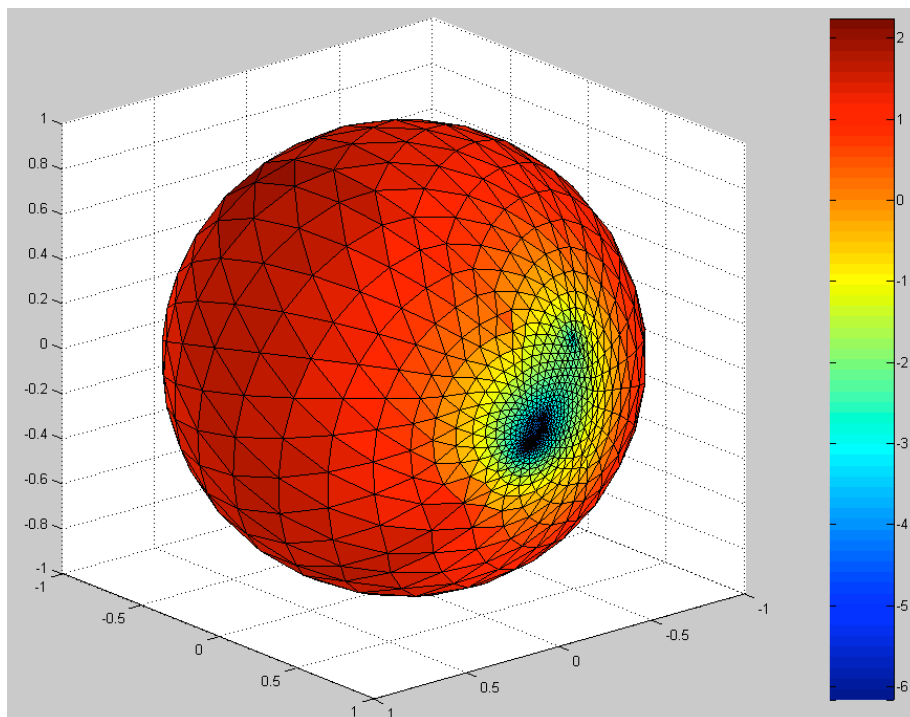


Figure 28. Simple U-shape object after 1000 iterations of interpolation with $\alpha = 0.9$ on heat kernel smoothed object in figure 24

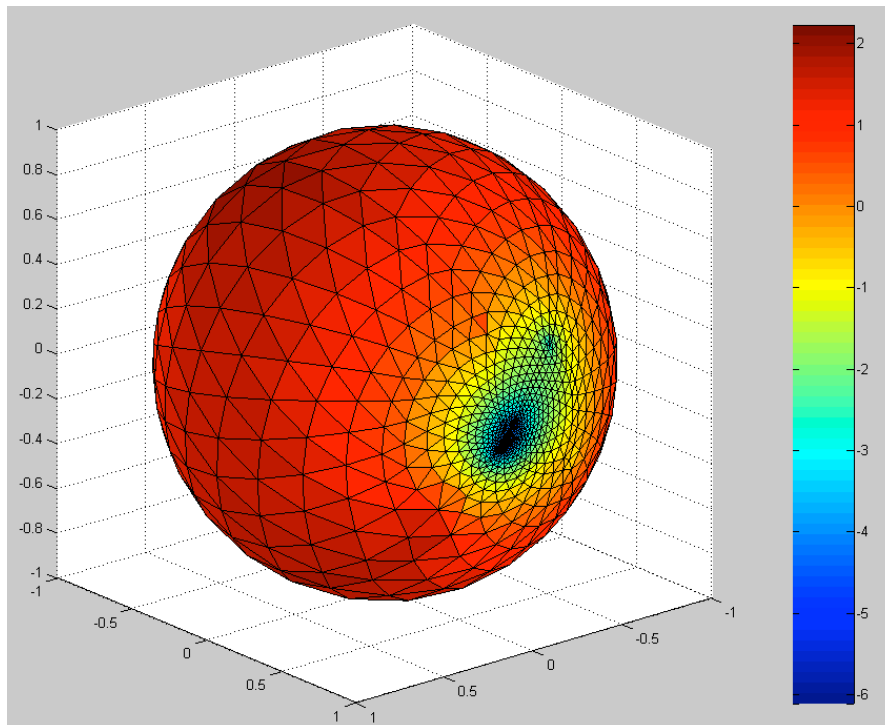


Figure 29. Simple U-shape object after 1000 iterations of interpolation with $\alpha = 0.9$ (without pre-smoothing)

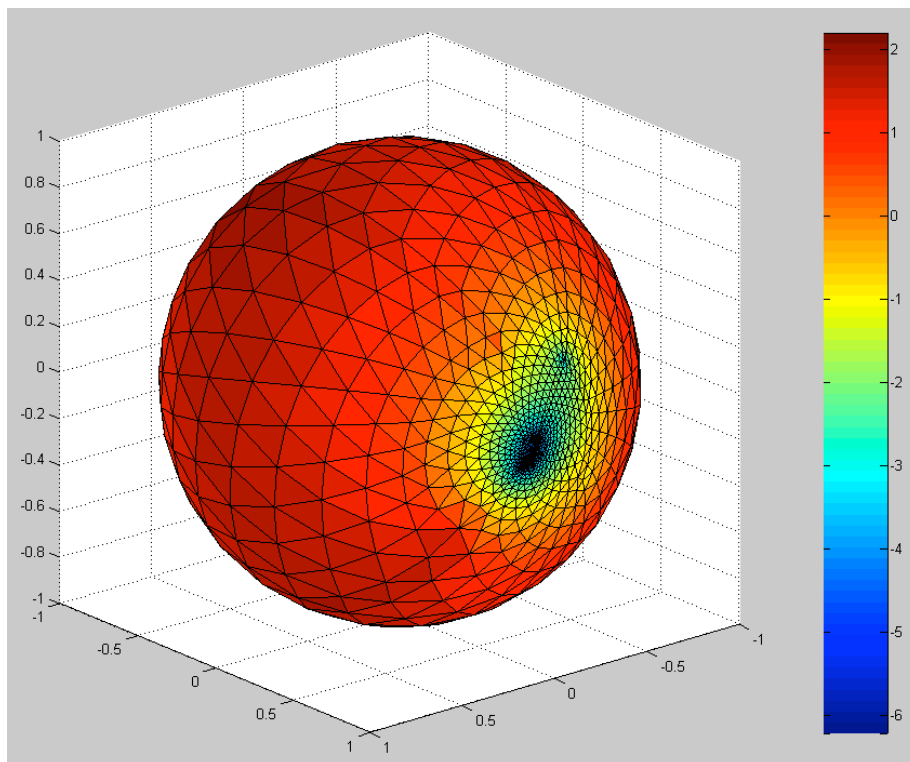


Figure 30. Simple U-shape object after 1000 iterations of interpolation with $\alpha = 0.5$ (without pre-smoothing)

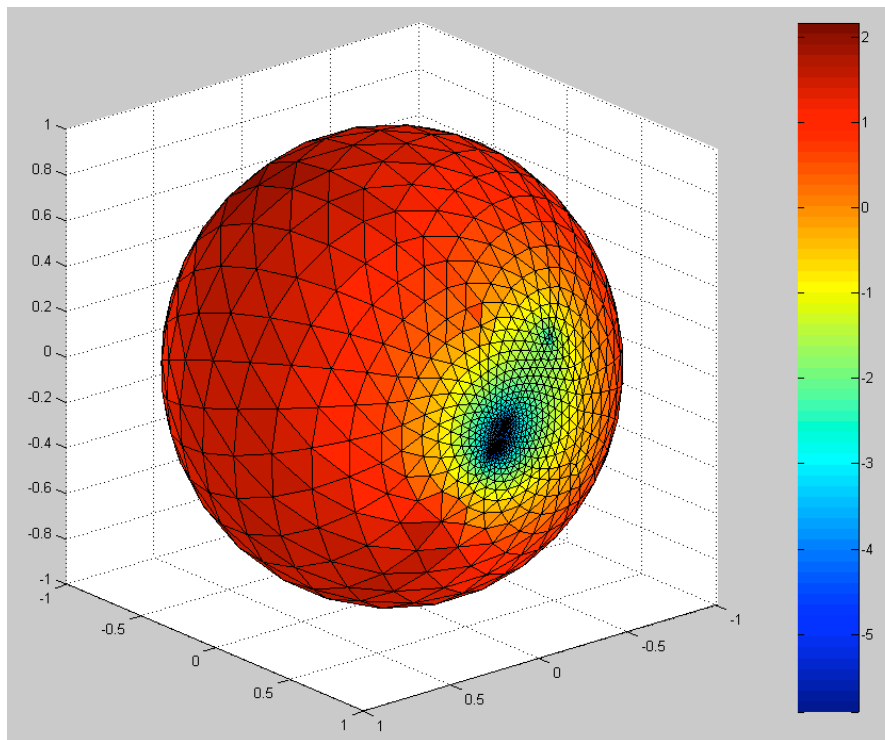


Figure 31. Simple U-shape object after 1000 iterations of interpolation with $\alpha = 0.1$ (without pre-smoothing)

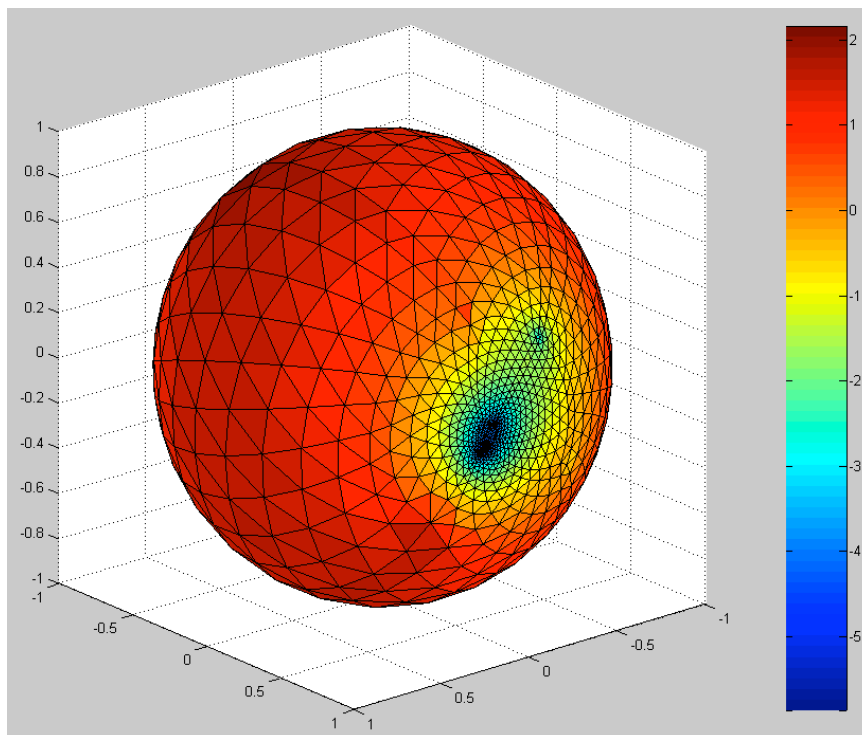


Figure 32. Simple U-shape object after 200 iterations of interpolation with $\alpha = 0.9$ (without pre-smoothing) – heat kernel smoothing was applied 3 times at each iteration

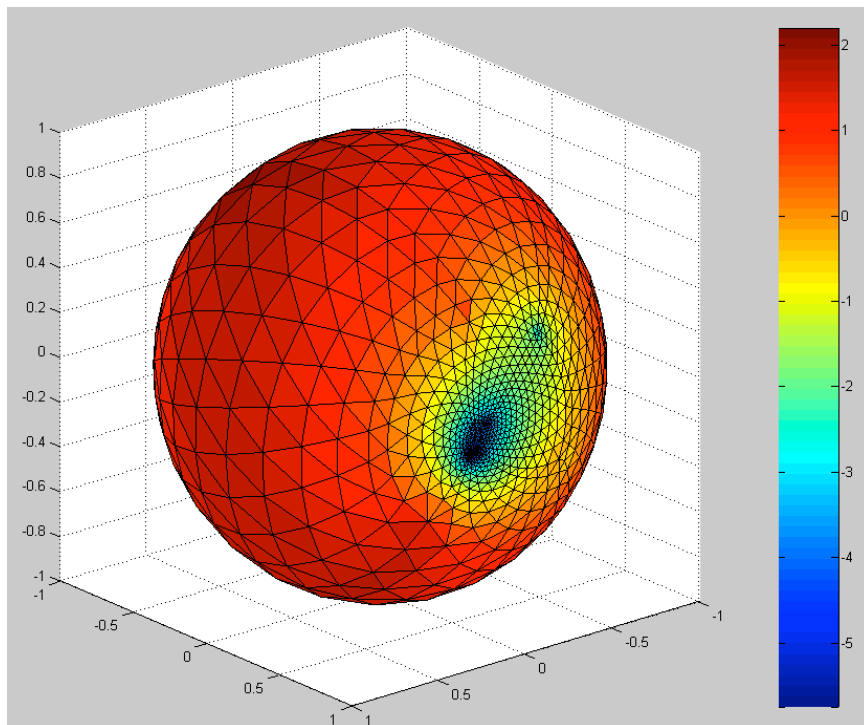


Figure 33. Simple U-shape object after 300 iterations of presmoothing with area-regularized HKS with $\sigma = 0.1, \alpha = 0.9$

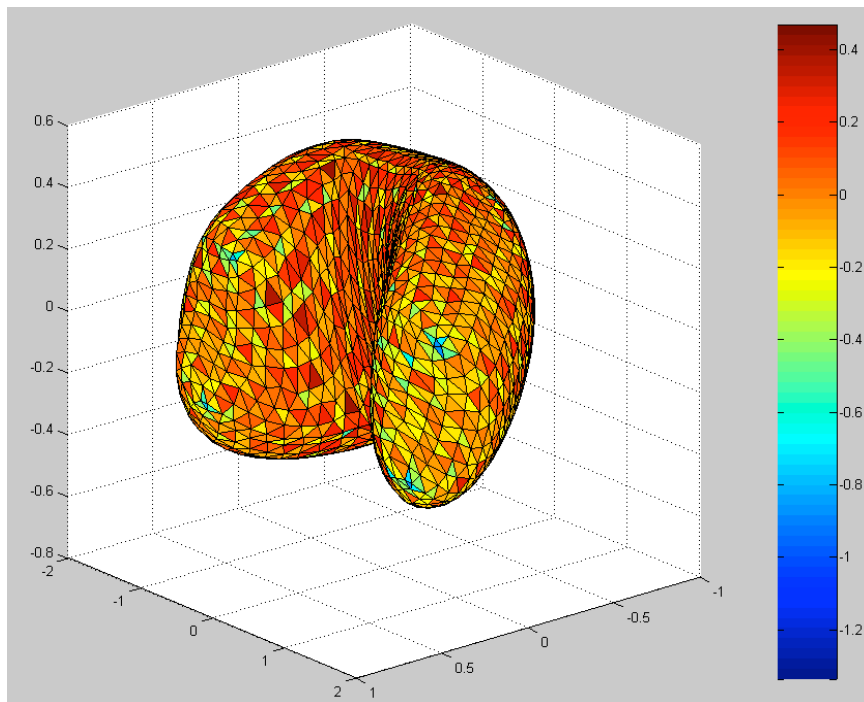


Figure 34. The same object as figure 33, which is seen from the above

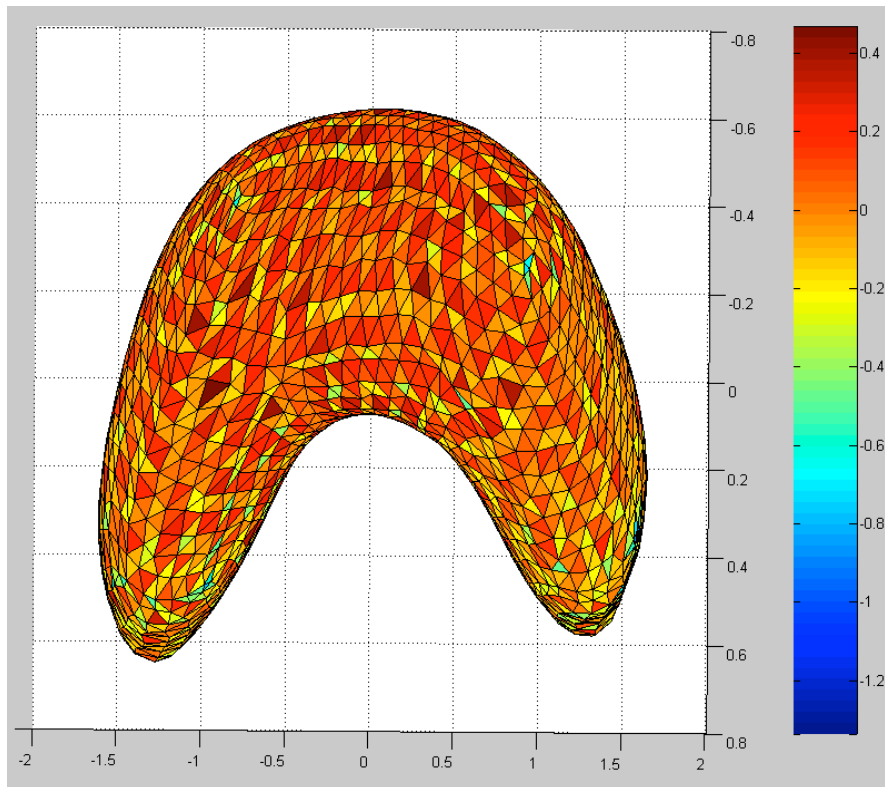


Figure 35. Simple U-shape object after 100 iterations of shortest-distance interpolation with area-regularized HKS, when $\sigma = 0.1, \alpha = 0.9, \beta = 0.9$

