Exercise 4: 3D surface mesh manipulation

PART 1: MESH DATA REPRESENTATION

Before we start with mesh processing you will need a basic understanding of meshes. In this part we will cover the most important aspects and properties of mesh representations. The exercise requires the available data:

- **`bunny_zipper.ply`** : a 3D model of the "Stanford Bunny"
- **`read_ply.m`**: a 3rd party MATLAB file to open **`.ply`** files in MATLAB
- **`QuickPatch.m`** MATLAB function to visualize a mesh with **`patch`**

1. Load the **`'bunny_zipper.ply'`** file with the **`read_ply`** function to get the faces and vertices of the bunny.
2. Look at the number of faces and the number of vertices. Is the mesh compressed or uncompressed? How can you tell?

The mesh is compressed as we can see number of faces is 69451, number of vertices is 35947. Their ratio: Nv/Nf=0.52<3, which is the value for uncompressed mesh

3. Show a 3D visualization of the bunny. Use the function **`patch`** and use your knowledge from "Segmentation & Visualisation" to beatify the visualization. In the same figure, use **`plot3`** to show the position of the first vertex in the vertex list. Also, show the position of the first face in the face list. For that, you can also use **`plot3`**.
4. Keep the original faces and vertices lists. Now decompress the mesh of the bunny in a new set of faces and vertices. Do note that an uncompressed matrix has a number of $3n_f$ vertices. Hint: the first face refers to vertices 1,2,3 and the second face to vertices 4,5,6 etc.
5. Create a 3D visualization of the uncompressed bunny. Use **`plot3`** in the same figure to show the position of the first vertex in the vertex list. Show the position of the first face in the face list.
6. Compare the position of the first face and first vertex with the compressed mesh. Can you explain this position?

We can see that, for the uncompressed mesh, the face and the vertex are in the "same" position (vertex 1 is exactly one of vertices in face 1). However, for the compressed mesh, the face and the vertex with the same index 1 are not close to each other. This is understandable as the vertices' indices are identical for all the faces in the compressed mesh. Thus, the vertex 1 is not assigned for this faces if one of the other faces process this index. For the uncompressed mesh, vertex with index 1 will start with face 1.

7. Compress the decompressed mesh again. Hint: the function **`unique`** can be made to work on rows. **`[C,ia,ic] = unique(A,'rows')`**. Make sure you understand what **`A`**, **`C`**, **`ia`** and **`ic`** are!
8. Show a 3D figure of the compressed bunny. Again, insert the position of the first vertex in the vertex list, the first face in the face list.
9. Compare the position of the first face and first vertex with those of the uncompressed mesh. Can you explain this position?

The unique function provided us with the list of non-repeated vertices. With the compressed data, the vertex indexed 1 is corresponded to the 1st index used the uncompressed vertices array, which can be found as the first value of idexm(1) - or index 7616 in the uncompressed vertices. With the compressed data, the face also rearranged due to the reorder of the vertices, using the indexn of new indices for vertices. The first face of the compressed mesh is thus corresponded to the 301th face in uncompressed mesh. With this result, we can see that the first vertex is the 7616th vertex in uncompressed mesh, while the first face is the 301th face in the uncompressed mesh. These two value do not start together as we have in the uncompressed data.

10. Compare the vertex list of the original compressed bunny and the newly compressed bunny are they similar? Can you explain why?

They are not similar as the lengths of the data are different. The number of vertices of newly compressed bunny is shorter than the original one. The explanation for this might be the compression technique is different between two sets of data, or compression is carried twice on the data so the number of vertices is further decreased

11. Compare the face list of the original compressed bunny and the newly compressed bunny are they similar? Can you explain why?

Although the faces of the original compressed mesh and the new one have the same length of 69451, the order of these faces are not the same. The number of faces is a fixed number do not change as it correspond to the number of triangles that the mesh, which is constant through our process. Decompression and compression are based on compressing and arranging of the vertices, thus influencing on the order of faces and make their order become different.

12. If you have a given vertex id, do you know which face id it belongs to in a compressed mesh if you do not have a face list? And do you know in an uncompressed mesh?

For the compressed mesh, you cannot know which vertex index belong to which face without the face list, due to the distribution of the faces and vertex indices in the compression phase. However, for an uncompressed mesh this is possible due to the distribution of vertices is correllating on the arrangement of faces.

13. If you have a given face (with vertex ids belonging to it) can you think of a method to determine the neighboring faces in a compressed mesh? And for an uncompressed mesh?

For the compressed mesh, we can know the neighboring faces by checking their vertex ids. If a face has a vertex index identical to the face that we are interested in, we can determine that these two faces have the common vertex and are neighbor to each other. However, for the uncompressed mesh, it is impossible as there are no relationship between neighbor faces

14. Which mesh type would you recommend if you have limited storage space.

Compressed mesh type is recommeded for limited storage space as we have a shorter length of vertice.

15. Which mesh type would you recommend if you want to smooth out an area?

Compress mesh type is recommend for smoothing out an area. As neighboring area can be defined in this mesh, the curvature of surface can be known from this type of mesh. Therefore, to smooth out an area, we can generate algorithm (such as low-pass filter) to reduce the curvature that is too sharp (or noise).

PART 2: MESH NORMALS

In this part we will cover the most important aspects and properties of mesh normals.

16. Load the '`t_hand.ply`' file with the `read_ply` function to get the faces and vertices of the bunny.
17. Calculate the normal of the first and last faces of the mesh using the formula for vertex normal calculation described in the syllabus.
18. In MATLAB it is possible to plot the normals of the mesh using the standard function "`quiver3`". Plot the normals of the complete dataset originating from the face centers using this function.
19. Instead of computing the face normals, it is also possible to compute the vertex normals. Compute the vertex normal of the first and last faces. Can you think of specific applications in which vertex normals are used instead of face normals?

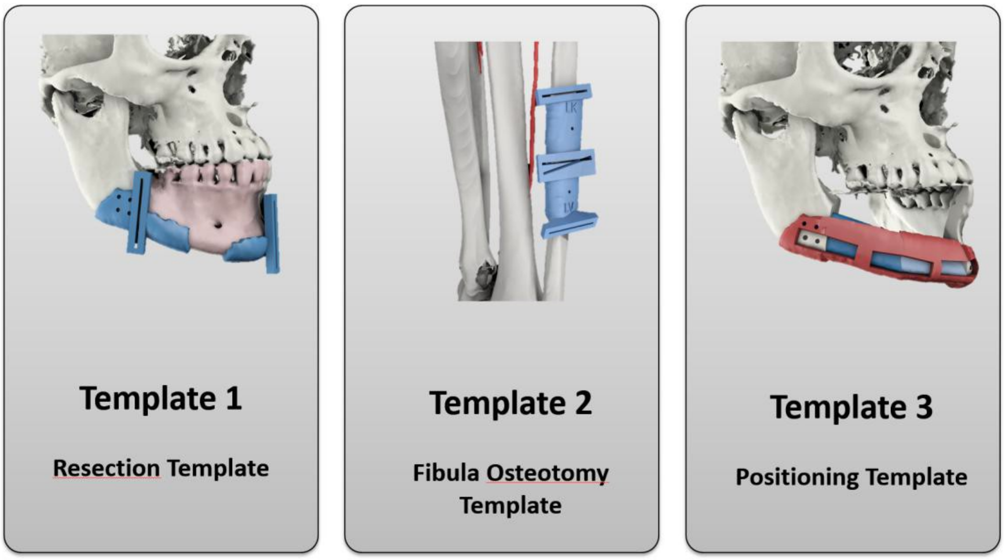
Vertex normals can be used in the situation when face normals does not significant change according to the space, such as rounded surfaces like a sphere. They can help to obtain a better approximation of smooth areas comparing with face normals, which might be useful to apply to "sharpy" objects with flat surfaces, such as cubes.

20. Compute the vertex normals of the complete set.
21. Once you have the complete vertex normal, it is possible to compute weighted vertex normals. Weighted normals can be computed in different ways. One common method is to take the 3D surface area of a vertex into account. Compute the weighted vertex normal of vertices 10, 100, 500, 1000. How do you calculate these weighted vertex normals and what are the parameters you need to take into account?

To calculate weighted vertex normals, we first find all the faces that share the same vertex. Secondly, we calculate he normals for each face with the cross product formula as we calculated before. These vectors can be normalized and time to the surface area of the face (weighted). However, as we work with triangle faces, this normalizing calculation can be shorten by dividing these normals to 2. Finally, we take the mean of all neighboring faces to find normal vertex. These vertex normals can also be normalized. The parameters we need to take into account is vertice's index, faces' index of neighboring and faces' normals. As faces' area can be achieve from faces normals (half of it), we can shorten the calculation without finding the face's area, and also shorten normalizing face normals

One application that is used often in 3D medical image processing is extrusion of a mesh structure. Extrusion can be used to generate surgical templates which will have a perfect fit on an anatomical structure. This technique is generally used in the reconstruction of large mandibular defects. See the figure.

22. Use the vertex normals to extrude the mesh by 1 mm. Next, use the vertex normals to extrude the mesh 5 mm.



Oromandibular reconstruction using 3D planned triple template method.
C Coppen, WM Weijs, SJ Bergé, TJJ Maal
J Oral Maxillofac Surg. 2013

23. What problem areas do you find during an extrusion of the bunny mesh?

Mesh of one provides the result that is natural and in proportions as before. However, we can observe there are some disnormality located within the fingers. Some of the information is lost, especially the locations where the surface curvature is turning sharp, such as within fingers. It happens because of two sides of the 2 opposite fingures, where the normals opposite to each other, got expand and on the way to overlap each other
In 5 mm extruding situation, the volume of the mesh increased significantly as we expect but overlaping with the other fingers also increased. The hand is swallow with lot of details are lost. Especially between fingures, imperfections are clearly seen.

24. Can you think of a way to minimize these problems from occurring during extrusion? What kind of solution would you propose?

As we mentioned, the imperfections happen mostly in the sharp curvatures of the surface. The extrusion can be done by putting costs on the side of these sharp curvatures, to less extrude in the direction where faces are counter with other faces and prevent collapsing. The suggestion is using simplification technique of the mesh. By using this techniques, we increase the size of the mesh while still keeping the features of shape. With bigger mesh size, or smaller number of vertices, the effect of extrusion will be less. Other suggestion is that maybe we can cluster the vertices to each other if we want to shrinking them together.

25. Instead of extruding the mesh, try to shrink the mesh with 1 mm and 5 mm.

26. What are the problems that occur during shrinking?

We can see that the mesh does not shrink really uniformly, as the fingers become smaller comparing with the inner part of the hand. The sharp curvatures, such as between fingure, becomes more flat and wider than before.

27. Can you think of a way to minimize these problems from occurring during shrinking a mesh? What kind of solution would you propose?

Instead of collapsing the two sides of two opposite fingers, shrinking mesh make the two sides of one fingers collape to each other. At these top of the fingers, simplification can also be used to reduce less effect of the shrinking. However, at the base of the figures, more vertices are needed to be added to the sharp curvatures to make them less squared and flat.

PART 3: APPROXIMATING MESHES

The following files are needed:

- **`ICO-Blob.stl`**: a 3D model of icospheres
- **`Skin_SMTHF.stl`**: a 3D model of a face
- **`stlread.m`**: a 3rd party MATLAB file to open .stl files in MATLAB
- **`patchcurvature.m`**: a 3rd party MATLAB file to calculate various curvatures

28. Load the '**`ICO-Blob.stl`**' file with the **`stlread`** function to get the faces and vertices of the ICOsphere blob.

29. Stl-files are usually uncompressed. To determine curvatures, we need a compressed mesh to obtain neighborhood information. Compress the ico-blob mesh.

30. Visualize and examine the mesh. The big spheric body has a radius of 1. The attached and imprinted medium sized spheric bodies have a radius of 0.5 and the smallest attached and imprinted spheric bodies have a radius of 0.25. The cube is 1x1x1 and the cylinder has a length of 2 and a radius of 0,5. All with arbitrary units.

31. Use **`Patchcurvature`** on the mesh with third rings enabled and use a smoothing of 1.

32. Plot the mesh with the following color overlays. Make sure you choose an appropriate color scale (**`caxis`**) per overlay based on the information given in step 31.

    a. Mean curvature

    b. Gaussian curvature

    c. Minimum Principal curvature value

    d. Maximum Principal curvature value

    See [MATLAB doc patch properties]on FaceVertexCData, and [MATLAB doc caxis] for help.

    What were the caxis values per overlay. Why did you choose these?

We know that to coloring the 3D surface, we need to map from a color scale/ grid to each point of the mesh. This color scale/ grid is defined by the limitation given by 4 values we specified below: a. Mean curvature =[-5.303, 4.058] b. Gaussian curvature =[-10.095, 16.475] c. Minimum Principal curvature value =[-4.053, 5.040] d. Maximum Principal curvature value =[-11.338, 4.054] We choose these caxis values from the standard variation values we calculated from the mesh.

33. Can you explain the differences between the minimum and maximum principal curvature for:

    a. The cube

    b. The cylinder

    c. The sphere (with each body and imprint separate).

a) Cube: We can see that the curvatures meshes is 0 at the flat surface, as both principal curvature for flat planes are null. However, at the edges of the cube, we have the sharp curve to the next surface, thus the maximal principal curvature become larger. In contrast, this values is zero for minimal principal as there is no curve along the cube edge, except for the corners.
b) Cylinder: Again, both meshes we can see that the flat part is nearly zero. Similar to the edge of the cube, for the curving surface, the maximum principal curvature is stable as the curvature is constant as they run along the circle. The minimum principal curvature is zero as it runs along the line from top to bottom of the cylinder, until this line change at the edge
c) Spheres: In the 3 spheres bodies and 2 imprints, we can see that both principal curves are homogeneous as the curves in two direction should be equal. The smaller sphere body has the higher curvature value as their radius is smaller, while the imprint will be negative. The biggest difference can be seen is the conections between spheres, which is similar to edge of the cube or cylinder. The maximum principal curvature is very high due to the sharp change, while for the second principal curvature is lower as it run along the diameter circle of the sphere

34. What is the mean curvature for:
    a. The center of a plane of the cube
    b. The center top of the cylinder
    c. The cylinder wall
    d. The big sphere
    e. The imprinted spheres (medium and small)
    f. The attached spheres on the big sphere (medium and small)

a. it is zero as there is no curvature
b. It is also zero as there is no curvature
c. The cylinder wall is almost 1.1
d. The big sphere is also around 1.01
e. The medium sphere is -2.03 and the small sphere is -4.06
f. The medium sphere is also 2.03 and the small sphere is 4.06, showing the same size as the imprinted spheres

35. What is the Gaussian curvature for:
    a. The center of a plane of the cube
    b. The center top of the cylinder
    c. The cylinder wall
    d. The big sphere
    e. The imprinted spheres (medium and small)
    f. The attached spheres on the big sphere (medium and small)

a. Similarly, the value is close to zero
b. Similarly, the value is also close to the wall
c. The cylinder wall has the values of 0.017
d. The big sphere is around 1.028
e. The medium sphere is 4.11 and the small sphere is 16.45
f. The medium sphere is 4.12 and the small sphere 16.44

36. The majority of the mean curvatures and Gaussian curvatures are positive. However, there are some areas that are predominantly negative. Which areas are those for the mean curvature? And which areas for the Gaussian curvature? Can you explain why?

The mean curvature is negative when the curve is dominantly concave, as the formula m=(1/2)(k1+k2) is negative only if both the curves are negative, or the bigger k is much negative.
For the Gaussian, from the formula G=k1*k2, we can see that the curvature is negative if the two curvatures are different sign. This can be seen in the saddle-shape surface (border of attached sphere) or the edge of the cube (maybe the cube edge is not totally straight) .

37. It is also possible to compute the spheric radius equivalent curvature based on the Gaussian curvature. Compute the spheric radius curvature. Determine the spheric radius equivalent curvature for:
    a. The big sphere
    b. The imprinted spheres (medium and small)
    c. The attached spheres on the big sphere (medium and small)

    Give the values and explain why these values are plausible:

We use the formula: K=1/(r^2) to calculate the data
Big sphere: 0.985
Imprinted sphere medium: -0.493, Imprinted sphere small: -0.246
Attached sphere medium: 0.493, Attached sphere small: 0.246
Comparing the information that is given, we can see that the values are very close and plausible. The value of imprinted pheres are negative also make sense as it has concave shape.

38. Load the '**`Skin_SMTHF.stl`**' file with the **`stlread`** function to get the faces and vertices of a face obtained from a CT-scan. Compress the mesh.

39. Use Patchcurvature on the mesh with third rings enabled and use a smoothing of 1. Plot the mesh with the following color overlays. Make sure you choose an appropriate color scale

(**`caxis`**) per overlay. A useful tool to work with “natural data” is to determine the 5-95% percentile range (**`prctile(C,[5 95]);`**).

a. Mean curvature

b. Gaussian curvature

c. Minimum Principal curvature value

d. Maximum Principal curvature value

40. Can you think of a way to extract local “Features” from (a combination of) the curvatures used in the previous question? A feature is a distinguishable attribute or aspect of a mesh. You can think of spheric/conic regions, saddle-shape regions, tube-like regions, etc. Also think of using the highest/lowest values or ratio of two curvatures. Implement this and specify what kind of features you extract.

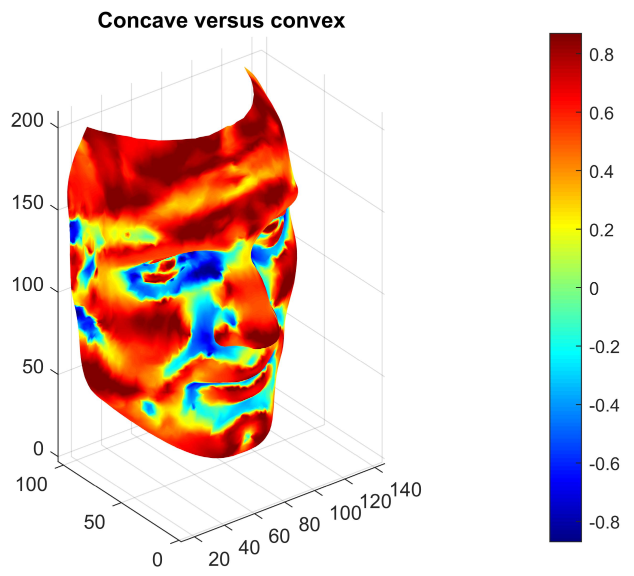
From the figure, we can use thresholds on the curvature data to be able to extract features. We focus on the nose and eyes of of the face, as well as features that peak-like. For example, the top of the nose, as well as forehead or chin are convex so we can utilize the the value from mean curvature, while the nasal bone area is saddle, which can be achieve from Gaussian's negative points. The eyes is concave too but the curvature is not as sharp as for the nose, thus the maximum principal curvature can be used.

41. Segmentation is also a common option of using curvatures. In the article by Zhang et al. (2008) they use a technique to enhance the visualization of the differences between the concave and convex areas of a mesh using the following equation:

$$K_c = \frac{2}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2}$$

Implement this equation and use this in an additional plot.

Add plot:



42. Change the values of the "**`usethird`**" and smoothing of the **`Patchcurvature`** on the mesh. What parameters, and which curvature rules (minimum/maximum threshold, region growing maximum change, etc.) would you use to create to isolate the following structures:
    a. Nose
    b. Eyes
    c. Zygomatic bone area
    d. Upper lip
    e. Lower lip

> a. As the nose has convex shape, the nose can be isolated using a threshold of of 0.5 in the Zhang et.al model. To isolate the nose from other convex part of the face, we use limits in the location of x from 55 till 100, while z from 75 till 135. All the values above 0.5 will be kept
> b. For the eyes, we use Gaussian curvature data with threshold of 2.8*10^-3 as the forehead is easier to separated than other graph. Limits in z is 120 till 140, limit in x for right eye is 30 till 60, left eye is 90 till 140. All the values above 2.8*10^-3 will be kept
> c. For the zygomatic bone, we also use Zhang et al. model, where the threshold of curvature is between 0 to 0.4. x limit for the left from 20-40, z limit is 110-140, for the right: x limit is right 120-140, limits in z is 110-140
> d. For the upper lip, we use mean curvature data, where the threshold is above 0.04, and the limits of x [50 110], and z [57 70]
> e. For the lower lip, we use similar as above with limits of x [50 110], and z [35 57]

MESH RE(SAMPLING), SIMPLIFICATION, RAYCASTING

Download the files provided in this exercise from blackboard:

- **`Aneurysm_Pre.stl`**: A mesh of a model of an Aneurysm before treatment.
- **`Aneurysm_Post.stl`**: A mesh of a model of an Aneurysm after treatment.
- **`stlread.m`**: a 3rd party MATLAB file to open .stl files in MATLAB
- **`Qslim.exe`**: a 3rd party executable for mesh simplification from: Garland, M., & Heckbert, 1997
- **`toolbox_graph.zip`**: a 3rd party tool for mesh analysis and manipulation[1]
- **`TriangleRayIntersection.zip`**: a 3rd party tool implementing the raycast function[2] from: Möller and Trumbore, 1997

43. Load both the **`Aneurysms_Pre.stl`** and **`Aneurysm_Post.stl`**. Compress the meshes. Plot the meshes. The meshes have already been aligned in a similar reference frame.

44. We are interested to see how much the aortic aneurysm has shrunken or grown. We can do this by raycasting the meshes as a function of each other. However, we need some pre-processing. First, we need to remove any parts of the mesh we are not interested in. This can be done by determining bounds of the aneurysm. Choose bounds were both aneurysms would fall in and leave an open bottom and top. Remove the faces that have vertices outside the bounds. This leaves redundant vertices but this is no issue for the further calculations. You can check this by plotting both aneurysms again. What bounds did you choose for x, y and z?

[1] https://nl.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph
[2] https://nl.mathworks.com/matlabcentral/fileexchange/25058-ray-triangle-intersection

xMin = 50
xMax = 300
yMin = 30
yMax = 225
zMin = 0
zMax = 300

45. Both meshes have a great number of vertices and faces that will not significantly contribute to see the growth of the aneurysm. Use the function "**`perform_mesh_simplification`**" from the toolbox_graph toolbox to simplify either the pre-treatment or post-treatment aneurysm. Make sure you have the file **`Qslim.exe`** in the active MATLAB directory. Determine a good value for simplification by checking this with plotting. What happens if you use an extreme low number of faces for output? Can you think of an issue for raycasting using this new mesh as the source of the rays? Can you think of an issue using this new mesh as the target of the rays?

With low number of vertices, the shape is understandably squared and is not an exact shape as we expected.
If this new inaccurate mesh is the source of the ray, as the mesh to pass the ray through is not high resolution, our result of measurement will be also less correct.
If the new mesh is the target of the ray, the projected faces might become too big, which supposed to consist several face. This leads to the disproportional of images, and also the mesh might be collided.

46. Choose which mesh you will use for raycasting and which mesh you use as a target. Determine the normals of the vertices you will use to perform the raycasting (normalize these normals!). Perform raycasting using the vertices of the mesh you choose as a source and use the normals of these vertices as directions. Use the function **`TriangleRayIntersection`**. Carefully read the options you have using this function. Make sure you use '**`lineType`**' as '**`line`**' and set '**`fullReturn`**' to 1 (read what they do!). Determine the minimum distance of each raycast and store this in an array. Hint: Your vert1 vert2 vert3 can be determined once and used in a loop. Loop over the origins and directions.

47. Plot the heatmap of the aneurysm based on the choices made in the previous question. Make sure you choose an appropriate color scale (caxis). A useful tool to work with "natural data" is to determine the 5-95% percentile range (**`prctile(C,[5 95]);`**).

48. What is the highest reduction in mm at the aneurysm (1 voxel = 0.34x0.34x0.34 mm). What is the estimated reduction at the residual aneurysm itself?

The maximum value of voxel is found as 28.49218 (we do not use value of 29.4472 due to noise reduction), so the highest reduction is 28.49218*0.34mm=9.6873mm in diameter. At the residual aneurysm, from the graph, we can see that at one side, the post mesh is different from pre mesh approximately 26 voxels, which is closely similar to our value of 29.447.

49. Take home messages = preparation for the oral exam

Prepare yourself for the oral exam by making a list of new concepts that you have learned in this exercise. Make sure that you know these concepts, and that you understand them. Examples of questions that may be asked during the oral are:

		level
1.	What are the 3 mathematical representations of a 3D surface?	understanding
2.	How can you tell whether you are looking at the outside or in the inside of a face?	knowledge
3.	What is the difference between a compressed and uncompressed mesh?	knowledge
4.	How can you calculate the normals of a surface?	understanding
5.	How does extruding works?	understanding
6.	How is a tangent plane defined? And how curvatures?	understanding
7.	How are local neighborhoods of a surface points described by curvatures?	understanding
8.	What is a possible principle-of-operation for resampling?	understanding
9	How can you register a surface with another surface?	understanding
10	How can you calculate the distance between two surfaces?	understanding