Midterm Assignment 4

May 14, 2021

Indicaciones generales

1. Fecha de publicación: 18 de mayo de 2021.
2. Fecha de entrega: 24 de mayo de 2020 hasta las 23:55.
3. Único medio de entrega: `https://e-aulas.urosario.edu.co`.
4. Formato de entrega: código en Python 3.
5. Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.
6. La actividad **debe** realizarse **individualmente**.
7. Los grupos pueden consultar sus ideas con los profesores para recibir orientación; sin embargo, la solución y detalles del ejercicio debe realizarlos **individualmente**. Cualquier tipo de fraude o plagio es causa de anulación directa de la evaluación y correspondiente proceso disciplinario.
8. El grupo de trabajo debe indicar en su entrega de la solución a la actividad cualquier asistencia que haya recibido.
9. El grupo no debe consultar ninguna solución a la actividad que no sea la suya.
10. El grupo no debe intentar ocultar ningún código que no sea propio en la solución a la actividad (a excepción del que se encuentra en las plantillas).
11. Las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.
12. `e-aulas` se cerrará a la hora acordada para el final de la evaluación. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de `e-aulas` será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.

Problem set

1. In computer-aided design it is imperative to efficiently organize, access, and modify geometric objects such as curves, surfaces, etc. It therefore is important to represent geometric objects such that computer calculations are feasible. In this problem, we will deal with a particular way of representing a regular surface on a computer using compression.

   Although the strategy proposed here, modulo some minor modifications, should work on a wide class of regular surfaces, we will focus on the so-called level surfaces in $\mathbb{R}^3$. For that, consider the function $f_n : \Omega \subset \mathbb{R}^3 \to \mathbb{R}$ defined by

$$f_a(x, y, z) = \frac{x^2 + y^2}{a^2} + z^2, \tag{1}$$

   where $a \in [1/2^4, 2^4]$ defines the scale of the surface. A level surface in $\mathbb{R}^3$ is defined as the set of points

$$\mathcal{M}(c) = \{\mathbf{p} \in \Omega \,|\, f_a(\mathbf{p}) = c\}, \tag{2}$$

   where $\mathbf{p} = (x, y, z)$. Notice that the set $\mathcal{M}(c)$ is a regular surface for any positive $c \in \mathbb{R}^+$. From now on we set $c = 1$

   We can encode a level surface in $\mathbb{R}^3$ by keeping track of the constant c, calculating the Gauss map of the surface, and filter the vectors on the unit sphere that are almost parallel, according to the threshold $\tilde{\varphi} > 0$. Notice that as a increases $f_a(\mathbf{p})$ looks more and more like a flat surface; hence, it's expected that less and less different vectors on the range of the Gauss map be necessary to accurately reconstruct the surface.

Construct a class that makes use of the Gauss map to efficiently represent a level surface in $\mathbb{R}^3$ such as $f_a(\mathbf{p})$. In order to accomplish that, implement a method for each of the following items according to the specifications stated.

(a) `vector_field(c, a, M)`: That takes c, a and M as parameters and uniformly samples M points, $\mathbf{p}_i$, $i = 1, \ldots, M$ in the domain of f_a. For such a set of points, calculate the (discrete) vector field associated to the Gauss map. The result should be a size-M container of unit vectors.

(b) `filter_by_angle(vecs, phi)`: This method takes the vector field calculated before and the angle threshold $\tilde{\varphi}$. It must keep only those vectors in `vecs` whose orientation differs by an angle that is strictly larger than $\tilde{\varphi}$ and discarding vectors otherwise. The resulting filtered vector field must be of size smaller than or equal to M.

(c) `plot_info(sifted, nlist)`: That takes a container of filtered vector fields and a container of values of n associated to the filtered fields in `sifted` and plots the size of each filtered vector field as a function of `nlist`. Before calling this method, the above mentioned methods must be invoked for each value of n specified in `nlist`.

Notice that the size of the filtered vector field accounts for the 'minimum' amount of information necessary to reconstruct and operate on/with the regular surface f_a. This size will of course depend on the values of a and $\tilde{\varphi}$. Although we could generate a triangulation to sample the surfaces discussed above, start by generating a squared grid of N points. Choose a moderate value for $N \sim 10 - 100$ and gradually increase it up to 10^5. Also, choose several values for the thresholds starting from an initial value of 0.1; then, try gradually decreasing it down to 0.001. Similarly, select Ω and R such that distinct features about the surfaces being interrogated are clearly visible.

2. So you have chosen to spend your summer as an intern at Professor Plateau's lab, whose work focuses on experiments with elastic surfaces and materials physics. Lately, the lab has been experimenting with a new elastic material which is believed to be useful for the construction of architectural pieces such as columns, stair railings, fountains, etc.

   For this elastic material to actually be useful, it must be synthesized such that there is no pressure within the material, or at least it should be minimal. The effect of a small pressure implies that the material is malleable, automatically making it structurally more stable and durable. In other words, we need to synthesize pieces of *minimal surface*.

   Using this elastic material, the lab has produced a series of prototype shapes using a synthesis method that is controlled by the temperature of an oven. Prof. Plateau is an outstanding physicist but not really good at math to be honest. So he has asked you to calculate numerically the pressure along the surface of each of the prototypes produced at the lab and understand the effect of temperature. That is to say, verify whether minimal surface prototypes are being produced at some temperature or not.

You, as an expert on computational and mathematical modeling, have decided to 3D scan the prototypes and come up with the following regular surface model

$$
\begin{aligned}
\mathbf{x}_t^\alpha(u,v) = &\cos t(\sinh v \sin u, -\sinh v \cos u, u + \sin^\alpha u)\\
&+ \sin t(\cosh v \cos u, \cosh v \sin u, v + \cos^\alpha u),
\end{aligned}
$$

where t labels the prototype name (number) and α (even integer) is directly proportional to the temperature in the oven. More importantly, you know that pressure is proportional to surface tension which, in turn, is proportional to the mean curvature on the surface of the prototype. Hence, you know how to perform the calculations that Prof. Plateau has requested! In order to get to the final calculation you have to:

(a) Plot the prototypes first. For that, parametrize the prototype number as $t = n\pi/16$ where $n = 0, 1, \ldots, 8$; such that there are 9 prototypes total.

(b) Calculate the mean curvature H for each prototype. Replot each prototypes color coding the surface according to its mean curvature.

(c) Redo the calculations and plots of the previous item, but using the Gauss curvature K for each prototype.

(d) Go the extra mile and calculate the area of each prototype and plot this value as a function of n for a fixed given temperature α in a single graph.

Implement this operations as member methods of a single class in Python. You must follow this procedure for two values of 'temperature': $\alpha = 0$ and 2. An appropriate range for the values of the parametrization of the prototypes are $u \in [0, 2\pi]$ and $v \in [-1, 1]$. The plan above will help the team and Prof. Plateau at the lab to fine tune the temperature that will produce the pieces of minimal surface and thus of minimal pressure. Consequently, synthesizing architectural pieces that are malleable and structurally sound.

References

M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Dover Publications, 2016.

J. Oprea. *Differential Geometry and Its Applications*. The Mathematical Association of America, 2007.