2DCentreOfMass: A Python library for the computation of the centre of gravity of a polygon

Maxime Bell, Marcel Głuszak and Tanishq Kumar

26-04-2024

Abstract

The 2DCentreOfMass library is a Python-based software tool designed to solve geometrical problems focusing on calculations related to the centre of mass in two-dimensional spaces. This tool supports interactive user manipulation as well as the computation of geometric properties of user-defined shapes.

1 Summary

The centre of gravity, also known as the centre of mass or centroid, refers to the point within an object where the entire mass of the object is concentrated. That is to say, the centre of gravity is the point at which an object would balance perfectly if supported at that point.
Several examples of when the centre of mass is useful include:

- Predicting rotational movement;
- Simplifying complex systems and applying Newton's laws of motion;
- Geometric proofs such as the centroid theorem.

The centroid of a polygon is calculated using the following formulae:

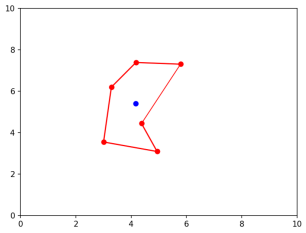
$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1}y_i),$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1}y_i).$$

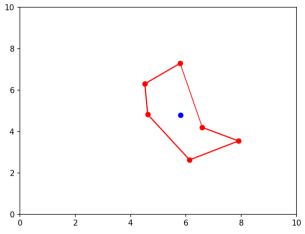
Where A is the area of the polygon calculated by the following:

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1}y_i),$$

and $(x_0, y_0), (x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_{n-1}, y_{n-1})$ are the coordinates of the polygon's vertices. We also define $(x_0, y_0) = (x_n, y_n)$ so that the polygon is closed. C_x and C_y represent the x and y coordinates of the centroid respectively. A useful resource to learn more about centroids, including the centroid of a three-dimensional shell is [1].
The 2DCentreOfMass library makes using these formulae much faster. Once launched from the command line, the user must choose an input mode and axis size. They can then input in any amount of vertices by plugging in their coordinates, so long as the shape does not intersect with itself, as seen in Figure 1a.



(a) The polygon before being pinned.



(b) The polygon after being pinned.

2 Statement of Need

Although finding the centre of mass of simple polygons (such as squares and triangles) may be straightforward, this process becomes complex and time-consuming when more vertices are added. For this reason, the use of software is required when computing the centre of gravity of more complex polygons.
One piece of software that can do this is Shapely [2]. Shapely is a BSD-Licensed Python package designed for the manipulation and analysis of two-dimensional geometric objects on a flat plane. In the field of geospatial analysis, Shapely is widely recognised as a powerful tool for computational geometry involving centroids. Please see [3] for more information on geospatial development using Python. However, a limitation of this software is that the user cannot see the polygon they input in real time. Moreover, the function that finds the centroid only returns the coordinates of the centroid and does not visualise where it would lie in the shape. Shapely also requires the user to have good knowledge of other coding languages, since it is built off of GEOS which is a C/C++ library.
2DCentreOfMass is a Python library specifically created to solve geometrical problems related to the centre of mass of polygons. It provides a dynamic graph that updates in real time as vertices are added, thus allowing the user to have a better understanding of the resulting polygon and its centroid.
Finally, the ability to 'pin' a vertex provides a practical way to demonstrate the relationship between the chosen vertex and centroid under the effect of gravity, as seen in Figure 1a and 1b.

3 Conclusion

This paper has given a brief description and explanation of the 2DCentreOfMass Python library, the mathematics behind it, and background information to justify the need for such a library.
The current limitations, however, provide the opportunity to further develop the program's ability in the future. This could involve expanding it into the third dimension, in addition to incorporating the ability to input and find the centre of mass of shapes which have varying density throughout.
The source code can be accessed here.

References

[1] Paul Bourke. Calculating the area and centroid of a polygon. *Swinburne Univ. of Technology*, 7, 1988.

[2] Shapely. Shapely 2.0.4 documentation. https://shapely.readthedocs.io/en/stable/. Accessed 24/04/2024.

[3] Erik Westra. *Python geospatial development*. Packt Publishing Birmingham, UK, 2010.