

J3D: using moments to quantify the shape of molecular clouds in 3D

Sarah E. Jaffa¹, Anthony P. Whitworth², and Seamus D. Clarke³

1 Centre for Astrophysics Research, University of Hertfordshire, Hatfield, AL10 9AB, UK **2** School of Physics and Astronomy, Cardiff University, The Parade, Cardiff, CF24 3AA, UK **3** I. Physikalisches Institut, Universität zu Köln, Zùlpicher Str. 77, D-50937 Köln, Germany

DOI: [10.21105/joss.0XXXX](https://doi.org/10.21105/joss.0XXXX)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Editor Name](#) ↗

Submitted: 01 January XXXX

Published: 01 January XXXX

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Molecular clouds are the birthplace of stars. In these chaotic environments, the interplay of gravity, pressure, turbulence, chemistry, radiation, heating and cooling, and feedback from young stars creates a wide variety of structures. We can observe these projected on the sky in two dimensions (position-position, PP), and, if we use line radiation to infer radial velocity from the Doppler Effect, in three dimensions (position-position-velocity, PPV). Line emission at different radial velocities is often presumed to come from physically separate structures along the line of sight, so used as a proxy for the third spatial dimension. We can also simulate many of these processes and create theoretical molecular clouds in two (PP) or three dimensions (PPP), and then use our understanding of the physics and chemistry of the interstellar medium to create synthetic observations of these clouds (PPV), which can then be compared with real observations. In all of these studies, the key is to be able to compare pixelated greyscale images in two or three spatial and velocity dimensions.

J plots is a tool to quantify the shape of 2D (PP) greyscale images using their moment of inertia. Written in Python, this allows users to import their data set from any format Python can understand (including the commonly used FITS format using the [pyfits package](#)) to make it into a numpy array and segment it using a method of their choice into meaningful structures (dendrograms are used in the example scripts, relying on the [astrodendro](#) package, but other techniques such as [clumpfind](#) (Williams, de Geus, & Blitz, 1994) or simple thresholding could equally be used). These structures can be individually or collectively analysed to reveal trends in their shape with environment, observational or simulated constraints, or segmentation method. This allows astronomers to quantify these chaotic structures and compare them in a statistical sense.

The J plots method

The basics of the J plots method, as described in Jaffa, Whitworth, Clarke, & Howard (2018), takes as input set of real-valued pixels representing a part of a greyscale image (called a 'structure'). For each structure, we calculate the area, A (number of pixels), mass M (sum of pixel values), centre of mass and principal moments of inertia, $I_{\{1...D\}}$, where D is the number of dimensions. We also calculate what the moments would be for a reference shape of the same mass and area, I_0 . The J moments are then defined as

$$J_i = \frac{I_0 + I_i}{I_0 - I_i}, \quad i = 1, 2, 3.$$

The two-dimensional case

In two dimensions the reference shape is a filled circle of constant surface density, so $I_0 = AM/4\pi$. If the shape is centrally concentrated, such as a collapsing core, both principle moments will be smaller than I_0 so both J values will be positive. For a hollow ring shape such as bubble blown in the cloud by stellar feedback, both principal moments will be greater than I_0 so both J values will be negative. For elongated shapes such as filaments, which are prevalent in molecular clouds, one moment will be larger and one smaller, so J_1 will be positive and J_2 will be negative. This gives us a simple diagnostic of these common shapes and allows us to place quantitative restrictions on shapes that fall between these categories.

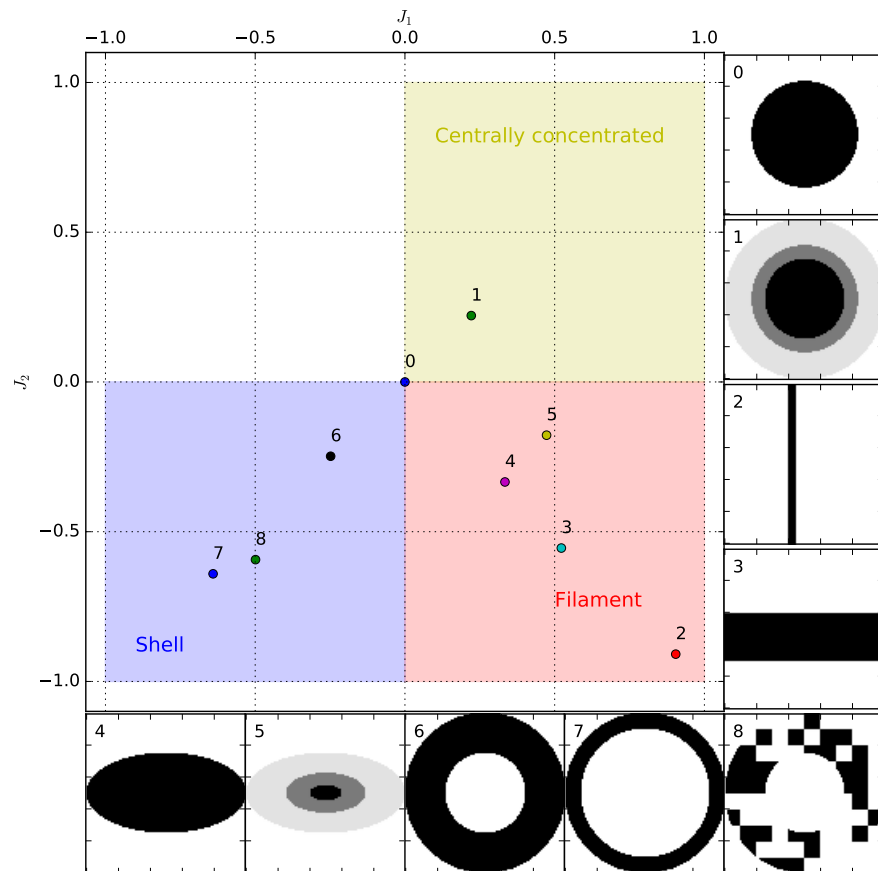


Figure 1: Proof of concept of 2D J plots. The J values of several simple shapes are plotted, representing morphologies observed in molecular clouds. This demonstrates that distinct categories of shape are distributed in different regions on the J plot.

J plots has been used in 2D to analyse the shape of structures within simulated filaments. Clarke et al. (2018) examined the J values of sub-filaments (small elongated structures formed inside the main filament identified in 3D PPP and projected into 2D PP) and fibres (elongated structures inside the main filament identified observationally in PPV, then projected into the PP) and found that the PPV detected fibres did not represent the same gas as the PPP identified sub-filaments. Their distributions of J values showed that the shape of the

structures recovered was changed by the nature of the observations, so observed velocity coherent structures should not be taken to represent physically separate objects.

J3D

J3D now also allows analysis of 3D objects without projection. This is mainly intended for the analysis of simulations in their native space, but can also be used for observations including velocity, as will be discussed later. In PPP space, the reference shape analogous to a uniform surface density circle becomes a uniform volume density sphere. For the moment of inertia, we care about the projection on a uniform density sphere into two dimensions. Therefore in PPP,

$$I_0 = \frac{2}{5} * M * \left(\frac{3V}{4\pi}\right)^{2/3},$$

where M is the mass of the structure (sum of pixel values) and V is the volume (number of pixels).

Again, hollow shapes where the mass is further from the centre than this reference shape will have all negative J values, centrally concentrated shapes will have all positive J values, and shapes that are elongated on one or two dimensions will have one or two positive J values and the others negative. In this case, the delineation between astrophysically relevant shapes is not as clear because most of the shapes we often discuss are defined in observations and therefore 2D, and the prevalence of filaments, oblate or prolate spheroids, and thin sheet-like structures is only discussed theoretically or in simulations. Algorithms to identify these kinds of shapes and studies of their relevance to astrophysical phenomena (such as turbulence, feedback, gravitational collapse, etc) are not well studied. We believe J3D meets an important need for robust quantification of 3D structures in simulations to enable statistical comparison of different studies.

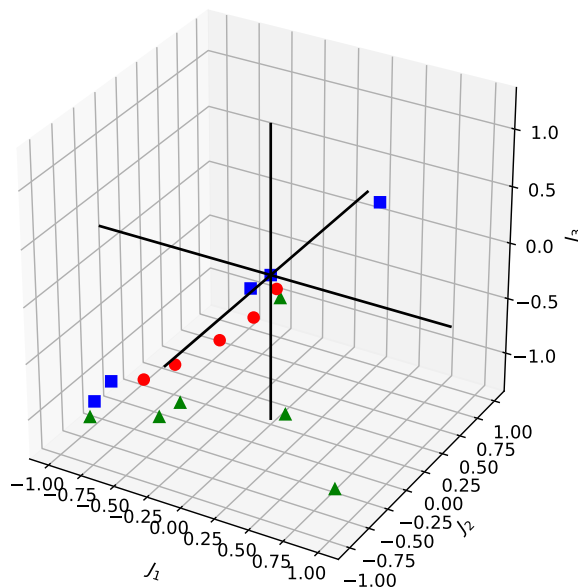


Figure 2: Proof of concept of J3D. The J values of a sample of simple shapes are plotted, showing how they are separated and classified. This plot can be generated to be explored in more detail using the example scripts included in the GitHub repository.

Velocity dimension

Real observations of molecular lines can provide information on the velocity along the line of sight of the gas, and this can help us understand the motion of the interstellar medium. Simulations also have velocity information, so can theoretically be analysed in 6-dimensional PPPVVV, or in projected PPV space to be comparable to observations. In such a case we caution against the blind interpretation of velocity as another dimension to be treated the same as spatial dimensions. One axis of the moment analysis can be confined to be along the velocity axis to keep the types of dimensions separate, but this option is left up to the user. Combining dimensions may provide some interesting insights, but will need much more thoughtful analysis to draw physically meaningful conclusions.

Concluding remarks

We present J3D, a code to quantify the three-dimensional structure of astrophysical objects in simulations or observations, based on the two-dimensional J plots. Comparing the moment

of inertia to a suitable reference shape separated hollow, centrally concentrated, elongated or elliptical shapes which may relate to important physical processes. Quantifying these shapes allows us to analyse and compare them in a statistical sense. The code is available on [GitHub](#). This release is also updated to be Python 3 compatible.

Acknowledgements

This research made use of *astrodendro*, a Python package to compute dendrograms of Astronomical data (<http://www.dendrograms.org/>) and *Astropy* (Astropy Collaboration et al., 2013; Price-Whelan et al., 2018). SEJ gratefully acknowledges support from the STFC grant number ST/R000905/1. APW gratefully acknowledges the support of a Consolidated Grant (ST/K00926/1) from the UK Science and Technology Facilities Council (STFC). SDC acknowledges support from the ERC starting grant no. 679852 ‘RADFEEDBACK’.

References

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., et al. (2013). *Astropy: A community Python package for astronomy*, 558, A33. doi:[10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068)
- Clarke, S. D., Whitworth, A. P., Spowage, R. L., Duarte-Cabral, A., Suri, S. T., Jaffa, S. E., Walch, S., et al. (2018). Synthetic C¹⁸O observations of fibrous filaments: the problems of mapping from PPV to PPP, 479(2), 1722–1746. doi:[10.1093/mnras/sty1675](https://doi.org/10.1093/mnras/sty1675)
- Jaffa, S. E., Whitworth, A. P., Clarke, S. D., & Howard, A. D. P. (2018). J plots: a new method for characterizing structures in the interstellar medium, 477(2), 1940–1948. doi:[10.1093/mnras/sty696](https://doi.org/10.1093/mnras/sty696)
- Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., et al. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package, 156, 123. doi:[10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f)
- Williams, J. P., de Geus, E. J., & Blitz, L. (1994). Determining Structure in Molecular Clouds, 428, 693. doi:[10.1086/174279](https://doi.org/10.1086/174279)