RAA 2021 Vol. **21** No. **2**, 48(8pp) doi: 10.1088/1674-74527/21/2/48
© 2021 National Astronomical Observatories, CAS and IOP Publishing Ltd.
http://www.raa-journal.org *http://iopscience.iop.org/raa*

Research in
Astronomy and
Astrophysics

Algorithm for extracting intersections in PPV space of filaments

Chao Zhang[1,2], Zhi-Yuan Ren[1], Chen Wang[3], Jing-Wen Wu[1] and Xiao-Yun Ma[1]

[1] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; *zhangchao920610@126.com*
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] CSIRO Data61, Sydney, NSW 2015, Australia

Received 2020 April 22; accepted 2020 July 24

Abstract A filament is an important structure for studying star formation, especially intersections of filaments which are believed to be more dense than other regions. Identifying filament intersections is the first step in studying them. Current methods can only extract two-dimensional intersections without considering the velocity dimension. In this paper, we propose a method to identify three-dimensional (3D) intersections by combining Harris Corner Detection and Hough Line Transform, which achieve a precision of 98%. We apply this method for extracting intersection structures of the OMC-2/3 molecular cloud and to study its physical properties and obtain the associated PDF distribution. Results show denser gas is concentrated in those 3D intersections.

Key words: ISM: molecules — ISM: structure — methods: data analysis — techniques: image processing

1 INTRODUCTION

Filamentary structures have been recognized in the interstellar medium for well over a century (Clarke et al. 2016; Takahashi et al. 2013; Gómez et al. 2018). Filaments are universal structures for dense molecular gas and they are important in star formation (Schneider & Elmegreen 1979; Arzoumanian et al. 2011). In many cases, the intersecting filaments are closely associated with multiple or clustered cores or young stellar objects (Myers 2011; Schneider et al. 2012; Wang et al. 2016; Lu et al. 2018; Treviño-Morales et al. 2019), suggesting increased star-forming activities therein. Schneider et al. (2012) asserted that intersecting filaments are more closely related to dense cores and young stars. Schneider et al. (2012)'s target of interest was the Rosette Molecular Cloud. Around the filament intersections, more dense core candidates or stars are found, possibly since merged filaments have yielded locally higher densities. Schneider et al. (2012) confirmed a high degree of coincidence between high column densities, filament intersections and the locations of embedded infrared clusters. Although the intersections are important for early-phase gas assembly and dense-core formation, the physical connections between intersections and dense cores, in particular those at early stages, are still to be further explored.

As described above, the intersections would be considerably important for inducing gas concentration and star formation. Therefore, the next critical job is to accurately locate all the intersections. There are two steps to locate intersections, to identify filaments and to mark out intersections of filaments.

There are already plenty of methods developed to identify filaments, including Curvelet Transform (Starck et al. 2003), HiFIVE (Hacar et al. 2013), Filfinder (Koch & Rosolowsky 2016) and DisPerSE (Sousbie 2011). Curvelet Transform has the advantage of enhancing contrast (including elongated features) in the data. Filfinder is capable of uniformly extracting the hierarchical filament structures. It performs well even if the image has large intensity variation. Although these methods have been confirmed to be effective and are widely implemented, lack of three-dimensional (3D) capacity has prevented them from detecting filament intersections. The specialty of HiFIVE is to find connected and aggregated gas structures in PPV space. Its capability is demonstrated well by the impressive results of intertwined filaments (Hacar et al. 2013). However, HiFIVE is not focused on resolving branched and intersected filaments. DisPerSE

(Sousbie 2011) can sensitively extract the elongated filamentary structures in any 3D data cube and resolve the branched and intersected filaments. So, we choose to utilize DisPerSE to extract filaments in 3D space.

For detecting intersections, manually marking intersections is time consuming and error-prone, especially for large data sets, which makes it almost impossible to be done. Apparently, this job should rely on an automatic way which can improve efficiency and eliminate the uncertainties. There are automatic methods like Harris Corner Detection[1] which is widely applied in 2D space scenarios, but it cannot be used in 3D space. We invent a new automatic method to identify 3D intersections and develop a procedure which integrates DisPerSE and our algorithm into a pipeline to detect intersections of filaments with favorable efficiency and completeness.

In this paper, we describe the details of the algorithm and show preliminary results on its efficiency and completeness by applying the algorithm to Orion A filaments observed by ALMA. The workflow is presented in Section 2. In Section 3 we explain the design of the algorithm and architecture of the pipeline. We highlight the performance of the algorithm in Section 3.3. Then, we discuss ‘pseudo-intersections’ and study the probability distribution function of density (PDF) of the filamentary structures and intersections by the pipeline in Section 4. A conclusion is provided in Section 5.

2 WORKFLOW

The detection of intersections in a filamentary structure is not straightforward as the structure has a 3D representation and does not have a regular shape. We design an algorithm to address this problem. Its workflow is expressed below:

1. Extract the filamentary structures in the PPV space (RA, Dec and radial velocity) by DisPerSE. Following the previous studies (Arzoumanian et al. 2011; Schneider et al. 2012), we adopt the 5σ level as the threshold of intensity.
2. Project the 3D filaments to three orthogonal 2D planes (RA-Dec, RA-radial velocity, DEC-radial velocity). Apply Harris Corner detection on each plane to obtain 2D intersections. Then, acquire the corresponding third dimension coordinates to get 3D candidate intersections.
3. Extract data cubes around each candidate intersection and inspect these cubes by applying the Hough Line Transform to eliminate false positive cases.

The workflow is illustrated in Figure 1, and the architecture and design of the algorithm are described in Section 3.

3 INTERSECTION DETECTION

Among the three steps to identify intersections, extracting filaments using DisPerSE is already commonly utilized. The other two steps are newly introduced and critical in the pipeline.

3.1 Find Candidate Intersections in 3D Space

To identify intersections, we projected 3D filaments into three two-dimensional (2D) planes to avoid overlap which will lead to intersections being missed. Overlap happens when two lines intersect in a plane perpendicular to the projected plane. Figure 2(a) displays this situation. Line 1 and line 2 intersect in 3D space, but this situation cannot be detected by only projecting onto the RA-Dec plane.

We introduce Harris Corner Detection which is a general method applied to extract intersections in 2D space. Mathematically, intersections are regions in an image with large variation in intensity in all the directions. Harris Corner Detection finds the difference in intensity for a displacement in all directions to detect intersections. We use it in step two of the workflow to detect candidate intersections in a projected plane. For each 2D plane, we provide the ‘2D filaments’ as input to the Harris Corner Detection, calculate the 2D coordinates of intersections (called projected intersections) and then convert the projected intersections into 3D coordinates. For this step, we need to consider false negative situations and remove some points.

Candidate intersections detected in each 2D plane are a subset of 3D candidates. After false positive elimination, the complete set of 3D intersections is the union of intersections detected in three 2D planes.

3.2 Remove False Positive Intersections

There are mainly two kinds of false positive intersections; one is filaments which intersect on the sky-plane, but are spatially separate along the velocity-axis. These filaments look like they intersect merely due to the projection effect; we call this situation “pseudo-intersection”. The other one is endpoints, which can also be detected by Harris Corner

[1] More details of this algorithm can be found at the website: `http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html`

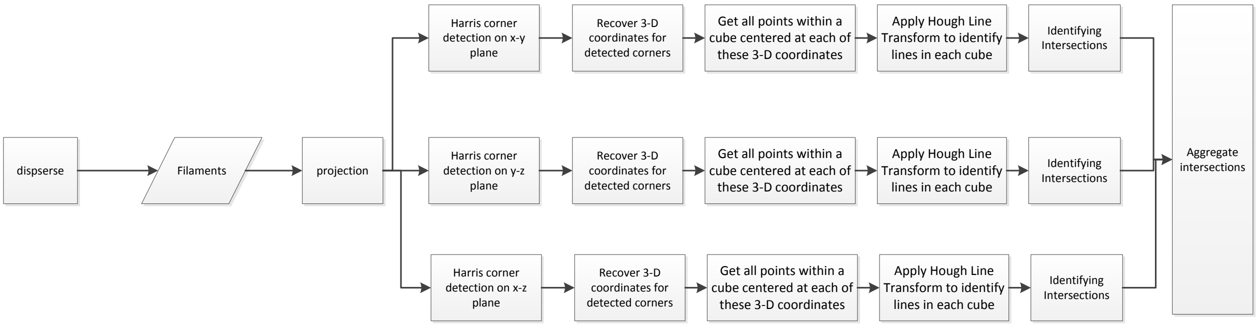

dispserse
Filaments
projection
Harris corner detection on x-y plane
Recover 3-D coordinates for detected corners
Get all points within a cube centered at each of these 3-D coordinates
Apply Hough Line Transform to identify lines in each cube
Identifying Intersections
Harris corner detection on y-z plane
Recover 3-D coordinates for detected corners
Get all points within a cube centered at each of these 3-D coordinates
Apply Hough Line Transform to identify lines in each cube
Identifying Intersections
Harris corner detection on x-z plane
Recover 3-D coordinates for detected corners
Get all points within a cube centered at each of these 3-D coordinates
Apply Hough Line Transform to identify lines in each cube
Identifying Intersections
Aggregate intersections

Fig. 1 The workflow of extracting intersections of filaments in 3D space.

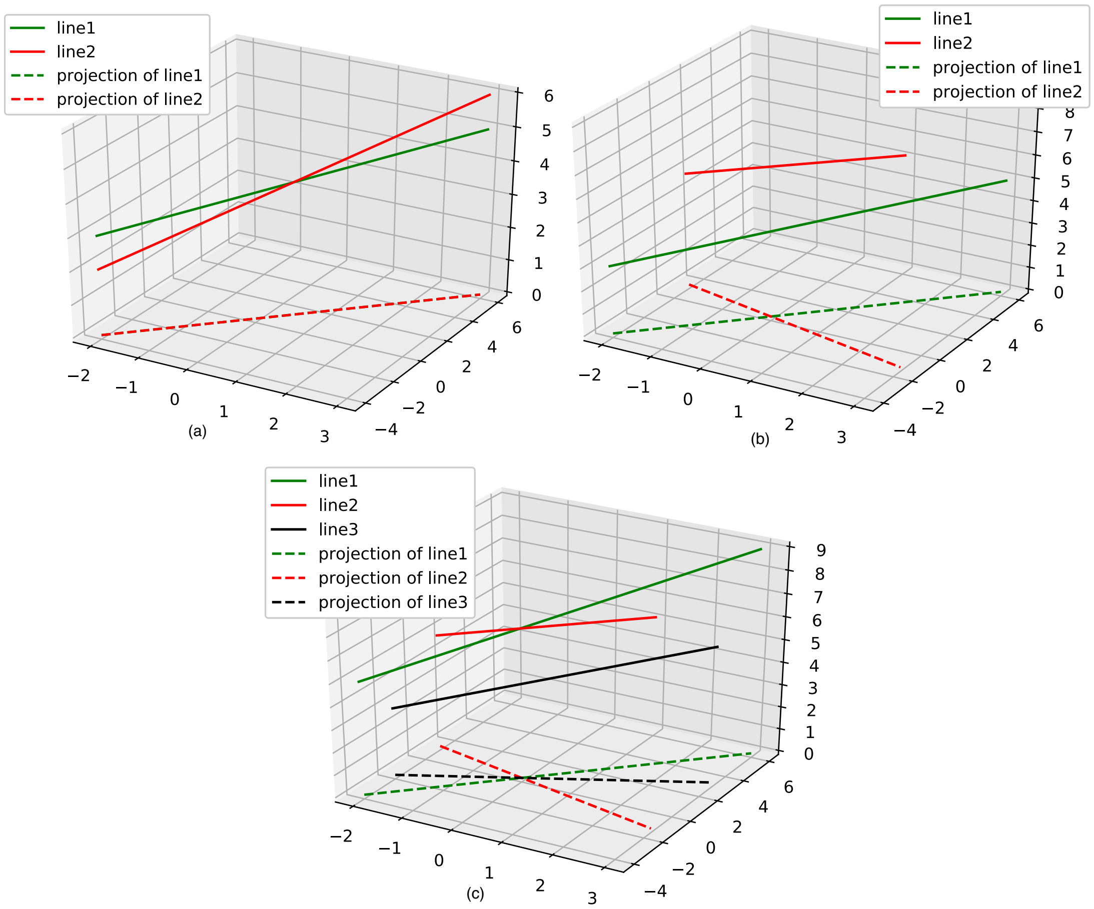


Fig. 2 Three special situations. *Solid lines* are 3D lines and *dotted lines* are the projection of *solid lines*. (a) One plane situation, (b) Pseudo-intersection situation, (c) pseudo and true intersection situation. These panels are described in the text.

Detection. These cases arise because of large variation in intensity in all directions.

Figure 2(b) displays an example of pseudo-intersections. Line 1 and line 2 intersect on the projected plane, but are actually spatially separated along the velocity axis.

To exclude pseudo-intersections, with the assumption that each filament has a distinct radial velocity, if overlapping filaments have two or more velocity components, their intersection would be regarded as a pseudo one. This means we can delete pseudo-intersections based on whether projected intersections have more than one velocity component. Here we adopt a threshold to

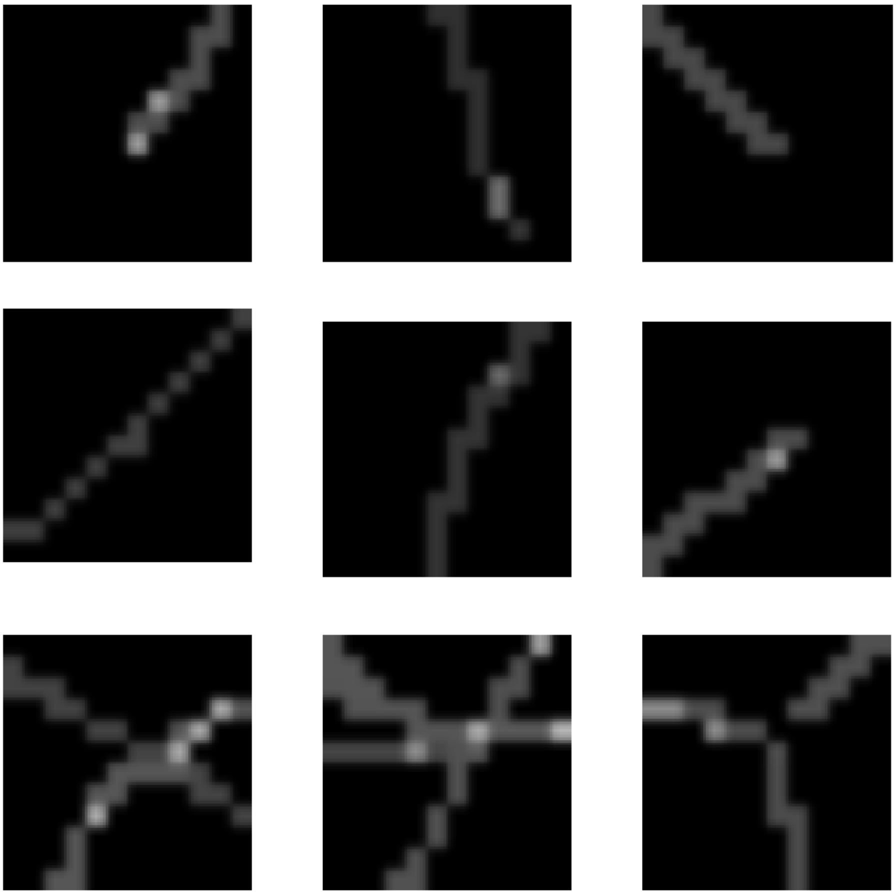


Fig. 3 Examples of projections of a cube. The first two rows feature endpoints and straight lines but the algorithm can discriminate and delete them. The last row shows real intersections and the algorithm can recognize them effectively by the Hough Line Transform.

distinguish the different velocity components. Velocity components within the threshold are treated as one component, while those beyond the threshold are regarded as pseudo-intersections.

However, there is an exception. Figure 2(c) illustrates an example. Line 1 and line 2 intersect at point A, while another line passes point A when projected on the sky-plane (x, y). In this case, the intersection of line 1 and line 2 is real there, but there are two components on the velocity-axis, which will be wrongly excluded by only use of the number of velocity components. This situation can easily be confused with pseudo-intersections. We call this situation 'pseudo and true intersections'. In order to solve these problems, the velocity components are further examined.

For differentiating between the situation of 'pseudo-intersections' and 'pseudo and true intersections' and removing endpoints, we introduce the Hough Line Transform[2]. Hough Line Transform is a popular technique for detecting straight lines. It can detect a line even if it is broken or distorted a little bit. Firstly, we select all points which have the same position but different velocities compared to projected intersections to form a new set of intersections (called 'candidate intersections'). Then, we take a cube around each candidate intersection and project the cube onto a plane to identify lines using the Hough Line Transform. We recognize the candidate intersection as true once there are two (or more) lines identified. To meet the prerequisite of applying Hough Line Transform, the cube should be small enough that curves can be regarded as straight lines.

As described above, the threshold applied to differentiate velocity components is an important parameter in our algorithm. It is related to the velocity sensitivity and average line width which need to be determined by real data. For example, in the OMC-2/3 dense molecular cloud,

[2] More details on this algorithm can be found at the website: *http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html*

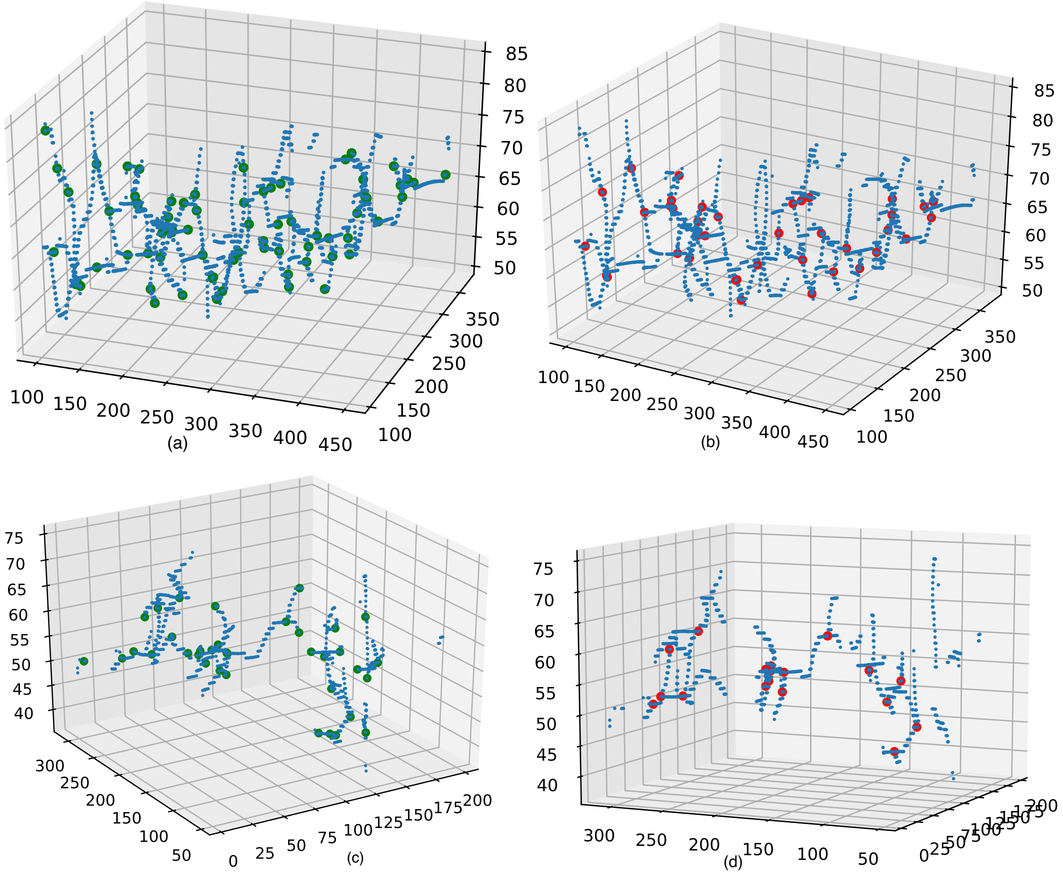


Fig. 4 The intersections extracted by the algorithm. *Blue points* represent filaments (a part of Orion A), *green points* in (a) and (c) are 'candidate points' and *red points* in (b) and (d) are real intersections (false positive intersections excluded).

the minimum value of threshold should be higher than the velocity sensitivity of a high-resolution $N_2H^+(1-0)$ spectral cube observed with ALMA which is $0.1\,\mathrm{km\,s^{-1}}$ in this case. In the meanwhile, if the velocities of two (or more) components are more than the average line width of filaments, we think a candidate intersection corresponds to multiple velocity components. In OMC-2/3, the average line width of filaments is $0.5\,\mathrm{km\,s^{-1}}$, so we choose $0.5\,\mathrm{km\,s^{-1}}$ as the velocity threshold in this work.

3.3 Experimental Results

To evaluate the performance of the algorithm, we applied it to a high-resolution $N_2H^+(1-0)$ spectral cube from the dense molecular cloud OMC-2/3 observed by ALMA. Figure 3 displays examples of projections of a cube. The first two rows feature endpoints and straight lines but the algorithm can discriminate and delete them. The last row shows real intersections and the algorithm can recognize them effectively by the Hough Line Transform. Figure 4 plots the results of extracted intersections using the algorithm. Blue points represent filament paths, green points in (a) and (c) are 'candidate points' and red points in (b) and (d) are the intersections which eliminated false positives. Here we adopted a threshold of $0.5\,\mathrm{km\,s^{-1}}$ to differentiate velocity components.

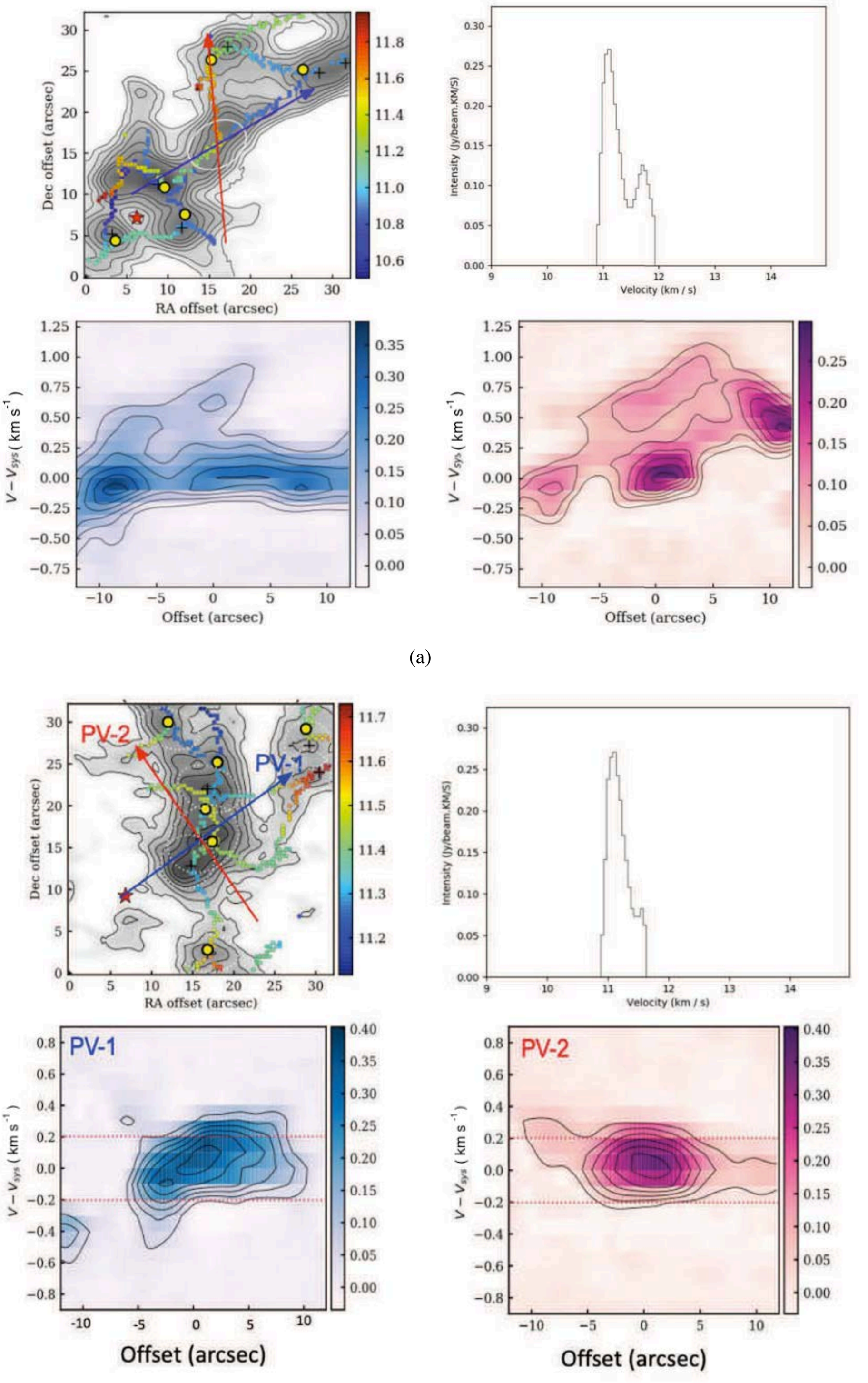
We consider intersections identified by human experts as the perfect ground truth to evaluate the result produced by the algorithm. We find accuracy is 93%, precision is 98% and recall is 95%.

4 DISCUSSION

4.1 Pseudo-intersections and Real Intersections

In this section, we present pseudo-intersections and real intersections identified in a real astronomical scenario which demonstrates a meaningful result obtained by our algorithm.

In Figure 5, for each set of four panels that form a square shape, the upper left panel is the integrated $N_2H^+(1_0-0_1)$ image (gray scale and contours). The contour levels are 15, 30, 45, 60, 75 and 90 percent of the local maximum. The colored dots represent the filament structures modeled by DisPerSE, with the color scale



(a)

(b)

Fig. 5 The gas distributions, spectral profile and position-velocity diagrams over two representative gas structures, (a) pseudo-intersection and (b) real intersection. These images are described in the text.

indicating the radial velocity. The directions to plot the Position-Velocity diagrams are signified with red and blue arrows. The two directions are along the major and minor axes of the filament. The upper right panel in each set is the spectrum profile of intersection. The lower left and right panels in each set are the Position-Velocity diagrams along with the major and minor directions, respectively.

Figure 5(a) displays an example of pseudo-intersections. Clearly, it is an intersection on the projection plane, but there are two velocity components in the spectral

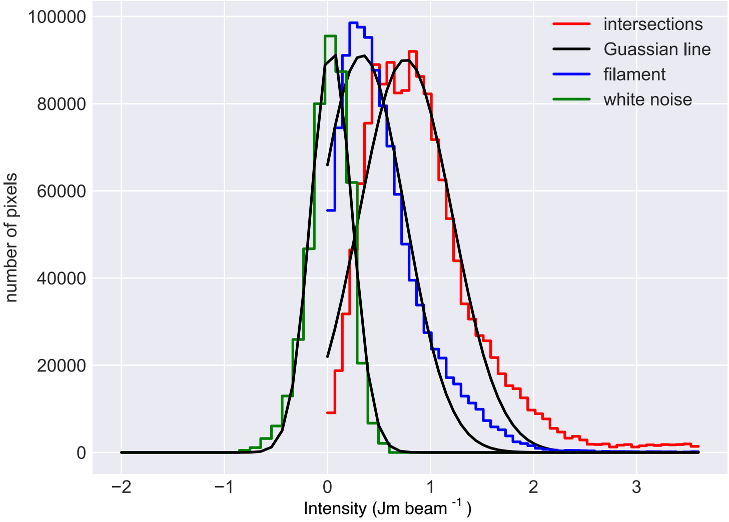


Fig. 6 The distribution of filaments and intersections. The *blue* and *red lines* represent the set of filaments (including intersections) and the set of intersections. The *black lines* are obtained by Gaussian fitting. The values of the peaks are 0.33 Jy beam^{-1} and 0.72 Jy beam^{-1}. The *green line* represents the distribution of uncertainty and the root mean square (RMS) of original data is 0.02 Jy beam^{-1}. (Note: The vertical axis is the number of pixels which are normalized.)

profile. The Position-Velocity diagrams along the major and minor directions also depict two velocity components. Our algorithm deletes this candidate correctly. Figure 5(b) shows the real intersections which have only one velocity.

4.2 Physical Properties of Extracted Structures

After identifying the filaments and intersections by our algorithm, we compare the different gas distributions of filaments, intersections and the rest of the clouds. We study the PDF of the gas pixel intensities in the extracted structures. The PDF is estimated for three components of our maps as plotted in Figure 6: (1) the white noise, (2) filaments and (3) intersections. The PDF of the white noise is sampled from a no-emission region, which we chose from a 100×100 pixel square area in the data cube without apparent N_2H^+ (1-0) emission. Since the extracted filaments and intersections from DisPerSE are merely lines and points, we need to identify pixels that belong to the filaments or intersections for statistical comparison. We classify pixels of filaments and intersections with the following strategy: for each point A along a filament, we look for the direction in which the intensity gradient descends fastest. We locate a position B along this direction where the intensity decreases to 80% of the peak value, and take the distance between A and B as a radius, to make a circle around A. All the pixels within the circle are counted as filament pixels. As point A goes through the whole filament, we mark all pixels for this filament. When A is an intersection, the pixels within the radius will be the intersection pixels. By this method, we collect filament pixels in an area of 15 579 arcsec^2 and interaction pixels in an area of 3728 arcsec^2 from our data cube. We utilize the Kolmogorov−Smirnov test (K−S test) to check whether the intensity datasets of filaments and intersections are different. The resulting p-value of this test is 0, indicating the two datasets differ significantly.

The green line in Figure 6 is the distribution of white noise, with a Gaussian distribution that peaks close to zero. The intensity PDF of the filaments peaks at $S = 0.33$ Jy beam^{-1}. The intersection pixels take up a fraction of 0.24 among the filament sample and 0.08 among the entire data volume. The PDF of intersections peaks at $S = 0.78$ Jy beam^{-1}, which is comparable to the full width at half-maximum (FWHM) of the filament PDF. The offset between the peaks of PDF are obvious in Figure 6 from noise to filaments, and to intersections. The intersections have significantly higher pixel intensity than the filaments at higher densities, with an obvious excess at the densest end of the distribution. This suggests a considerable fraction of the dense gas in the filaments tends to be concentrated around intersections, which present obvious overdensity.

4.3 The Limitation of Our Method

As mentioned before, there are two steps to locate intersections: to identify filaments and to mark out intersections. Identifying filaments accurately is a precondition to intersection identification. The ability to extract filaments will limit the quality of intersection detection.

Besides, threshold setting involves subjective factors. If the threshold is too high, it will mix the multi-velocity components into a single velocity component, and the pseudo-intersection cannot be removed. If the threshold is too low, the intersection may be mistaken as a pseudo-intersection. A good threshold can improve the accuracy of the result, which needs to be tried based on the properties of data.

5 CONCLUSIONS

In summary, we develop a new procedure to identify intersections of filaments in PPV space. The algorithm uses Harris Corner Detection and Hough Line Transform to extract intersections of filaments identified by DiSperSE, and achieve a precision of 98%.

Our algorithm has two main steps. The first step is getting candidate intersections applying Harris Corner detection. The next is to delete all the false positive intersections which include “pseudo-intersection” and endpoints using Hough line transform. The sensitivity for identifying false positive intersections can be adjusted by setting a threshold, which is related the velocity sensitivity and average line width of the observation.

In this paper, we also explore the use of our algorithm to extract intersection structures of the OMC-2/3 molecular cloud, study its physical properties and obtain the associated PDF distribution. We find a considerable fraction of the dense gas in the filaments tends to be concentrated around intersections, which present obvious overdensity.

With the rapid development of new telescopes and instrumentation technology, the data volume has grown to a level that makes detecting intersections by eye almost impossible. Our algorithm demonstrates its capacity for extracting filament intersections automatically, especially in PPV space, which is expected to have a better performance for eliminating false positive intersections than a 2D algorithm.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant Nos. 11988101, 11725313, 11403041, 11373038 and 11373045), CAS International Partnership Program (No. 114A11KYSB20160008) and the Young Researcher Grant of National Astronomical Observatories, Chinese Academy of Sciences. This paper makes use of the following ALMA data: ADS/JAO.ALMA#2013.1.00662.S.

References

Arzoumanian, D., André, P., Didelon, P., et al. 2011, A&A, 529, L6

Clarke, S. D., Whitworth, A. P., & Hubber, D. A. 2016, MNRAS, 458, 319

Gómez, G. C., Vázquez-Semadeni, E., & Zamora-Avilés, M. 2018, MNRAS, 480, 2939

Hacar, A., Tafalla, M., Kauffmann, J., & Kovács, A. 2013, A&A, 554, A55

Koch, E. W., & Rosolowsky, E. W. 2016, FilFinder: Filamentary Structure in Molecular Clouds, Astrophysics Source Code Library, ascl:1608.009

Lu, X., Zhang, Q., Liu, H. B., et al. 2018, ApJ, 855, 9

Myers, P. C. 2011, ApJ, 735, 82

Schneider, N., Csengeri, T., Hennemann, M., et al. 2012, A&A, 540, L11

Schneider, S., & Elmegreen, B. G. 1979, ApJS, 41, 87

Sousbie, T. 2011, MNRAS, 414, 350

Starck, J. L., Donoho, D. L., & Candès, E. J. 2003, A&A, 398, 785

Takahashi, S., Ho, P. T. P., Teixeira, P. S., Zapata, L. A., & Su, Y.-N. 2013, ApJ, 763, 57

Treviño-Morales, S. P., Fuente, A., Sánchez-Monge, Á., et al. 2019, A&A, 629, A81

Wang, K., Testi, L., Burkert, A., et al. 2016, ApJS, 226, 9