

# WingSegment: A Computer Vision-Based Hybrid Approach for Insect Wing Image Segmentation and 3D Printing

Shahab Eshghi,\* Hamed Rajabi, Johannes Poser, and Stanislav N. Gorb

This article introduces WingSegment, a MATLAB app-designed tool employing a hybrid approach of computer vision and graph theory for precise insect wing image segmentation. WingSegment detects cells, junctions, Pterostigma, and venation patterns, measuring geometric features and generating Voronoi patterns. The tool utilizes region-growing, thinning, and Dijkstra's algorithms for boundary detection, junction identification, and vein path extraction. It provides histograms and box plots of geometric features, facilitating comprehensive wing analysis. WingSegment's efficiency is validated through comparisons with established tools and manual measurements, demonstrating accurate results. The tool further enables exporting detected boundaries as FreeCAD macro files for 3D modeling and printing, supporting finite element analysis. Beyond advancing insect wing morphology understanding, WingSegment holds broader implications for diverse planar structures, including leaves and geocells. This tool not only enhances automated geometric analysis and 3D model generation in insect wing studies but also contributes to the broader advancement of analysis, 3D printing, and modeling technologies across various planar structures.

## 1. Introduction

The study of insect wings presents a captivating challenge, as their intricate structures hold the key to unraveling the secrets of nature's most skilled aviators.<sup>[1]</sup> Understanding the complex morphology of insect wings is not merely an academic pursuit in multiple scientific domains, such as biomechanics, animal behavior, physiology, and ecology,<sup>[2–7]</sup> but an inspiration for innovative technologies in the field of engineering. These include the development of robust load-bearing structures,<sup>[8–10]</sup> bioinspired joints,<sup>[11–14]</sup> insect-inspired hinges,<sup>[15]</sup> bioinspired wing,<sup>[16,17]</sup> insect-inspired composites,<sup>[18,19]</sup> bioinspired attachment strategies,<sup>[20–22]</sup> and bioinspired grippers.<sup>[23]</sup>

As our understanding of the wing structures deepens, manual analysis methods face challenges in accurately


capturing the complex geometric features to characterize wing morphology<sup>[24,25]</sup> comprehensively. Furthermore, manual analysis of structure complexity requires precise consideration of geometrical parameters and proves time-consuming and prone to errors. To comprehensively explore the multifaceted nature of insect wings, automated methods powered by computer vision offer a promising avenue for precise and efficient analysis. In recent years, several techniques have emerged that facilitate convenient 3D scanning,<sup>[26]</sup> extraction of geometrical information,<sup>[27–31]</sup> and automated finite element modeling<sup>[29,32,33]</sup> of both insect bodies and wings. These advancements aim to enhance the precision, speed, and efficiency of analyzing insect wing morphology.

Network extraction tool (NET),<sup>[34]</sup> and network extraction from images (NEFI)<sup>[35]</sup> are tools designed to extract graphs from images of networks found in diverse domains. The kind of graph they extract is a representation of the underlying network structure within the images. They also encompass the aim of extracting networks (e.g., venation patterns) from insect wings. FijiWings<sup>[36]</sup> is a comprehensive tool that incorporates multiple integrated functions to accurately quantify cell number (trichome count) in the wings of *Drosophila*. DrawWing<sup>[37]</sup> is another tool for extracting vein junctions and drawing a fitted diagram of wing outlines and veins. DrawWing is specifically developed to be compatible with the wings of honeybees belonging to the genus *Apis*. However, the aforementioned tools primarily focus on extracting junction locations or venation network patterns, while there are other tools dedicated to automating the modeling of insect wings.

S. Eshghi, J. Poser, S. N. Gorb  
 Department of Functional Morphology and Biomechanics  
 Zoological Institute  
 Kiel University  
 Kiel 24118, Germany  
 E-mail: eshghi.shahab@gmail.com

H. Rajabi  
 Mechanical Intelligence (MI) Research Group  
 South Bank Applied BioEngineering Research (SABER)  
 School of Engineering  
 London South Bank University  
 London SE1 0AA, UK

H. Rajabi  
 Division of Mechanical Engineering and Design  
 School of Engineering  
 London South Bank University  
 London SE1 0AA, UK

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300712>.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300712

WingMesh<sup>[32]</sup> is one such tool previously developed by the authors, which aims explicitly at extracting cell boundaries from images of insect wings. It utilizes Distmesh2D,<sup>[38]</sup> a mesh generator in MATLAB, to generate a mesh on the insect wing domain. Embedded within WingMesh are methods that facilitate the replication and manipulation of a distinct file format known as Abaqus input file (i.e., INP files), utilized by Abaqus CAE. These methods enable the generation of orphan meshes.<sup>[39]</sup> Orphan meshes lack the necessary user-friendly functionality to change the wing's geometry easily. It should be noted that generating a wing model using WingMesh is considerably slow. WingGram<sup>[29]</sup> was later developed as a solution to tackle these challenges, incorporating computer vision techniques to accurately detect wing cell boundaries from two-dimensional images. It introduces a novel approach for assigning corrugations to the wing model using a secondary image. Moreover, WingGram has the capability to extract various geometric features from wings, including cell area, length, width, and vein junction locations. Python scripting in Abaqus was utilized within WingGram to transfer wing geometry from an image to a journal file (JNL). This enables users to import wing geometry without relying on generated meshes, allowing them to leverage the advantages of Abaqus faster and more optimal meshing tools compared to the embedded Distmesh2D in WingMesh.

Although the aforementioned tools have been acknowledged for their intended purpose, there is room for improvement further to enhance the analysis and study of insect wings. These enhancements may include extending the detection capabilities beyond venation networks to encompass venation patterns, enhancing robustness and accessibility, and providing additional tools for statistical analysis of insect wings. These improvements would contribute to the advancement of research in this field. Recognizing the limitations of NEFI and NET in detecting curved patterns, along with the species-specific nature of FijiWings<sup>[36]</sup> and DrawWings,<sup>[37]</sup> and the fact that WingMesh<sup>[32]</sup> and WingGram<sup>[29]</sup> exclusively generate Abaqus-compatible models, which may be inaccessible and costly to students, this study aims to address these shortcomings by introducing WingSegment as a fast, accessible, and reliable solution. WingSegment adopts a hybrid approach that integrates computer vision and graph theory to extract cell boundaries, junction coordinates, and vein path coordinates accurately. The methodology encompasses various techniques, such as region-growing,<sup>[40–42]</sup> Dijkstra path finding,<sup>[43–46]</sup> thinning,<sup>[47]</sup> and line simplification,<sup>[48,49]</sup> which have been employed in this research. Additionally, we have utilized the FreeCAD macro (FC macro) file format<sup>[50,51]</sup> to generate automated FC models from insect images, further enhancing the accessibility and flexibility of the software in both modeling and 3D printing of biomimetic wings.

WingSegment additionally produces Voronoi patterns within the wing area by utilizing the centroids of the extracted cells. Voronoi patterns, commonly found in insect wings, hold significant importance in their study.<sup>[52]</sup> These patterns offer insights into the physical constraints governing the organization of biological tissues, reflecting the crucial role wings play in insect radiation and contributing to their remarkable diversity in shape, size, and color patterns. With their efficient and complete plane tiling, Voronoi patterns serve as convenient models for the intricate structures of insect wings. By examining Voronoi patterns,

researchers can gain a valuable understanding of the physical and biological properties of wings, shedding light on insect evolution and adaptation.

## 2. Result

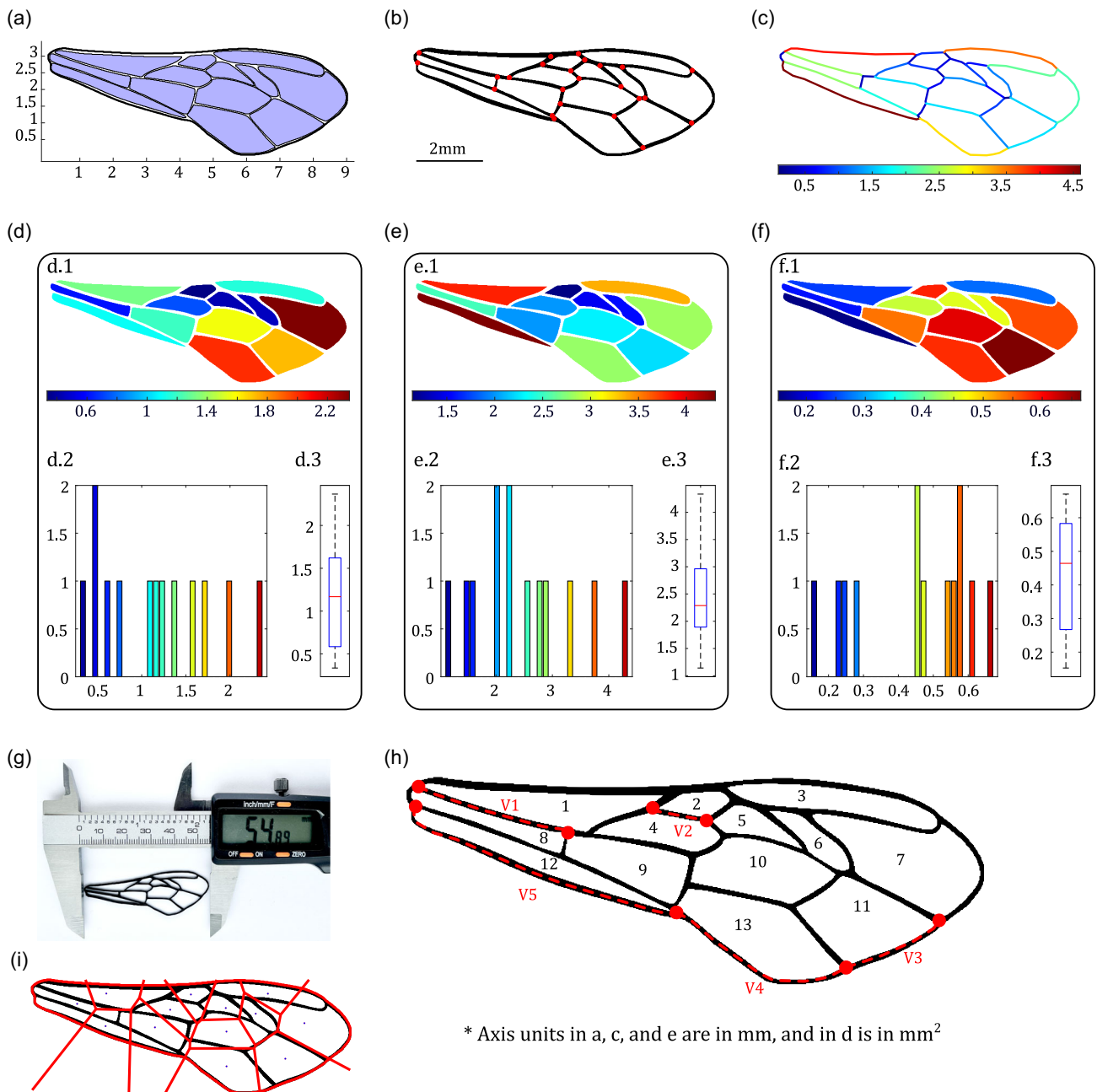
Our method successfully extracts the boundary of the wing outline, individual cells, and the intricate venation pattern. Additionally, it identifies the locations of junctions and generates a 3D-printable wing model. Moreover, our method could measure the wing's geometric properties, including cell area, length, and circularity distributions, which are visualized through heatmaps, boxplots, and histograms. Furthermore, we generate a Voronoi pattern based on the centroids of the segmented wing cells. The results obtained from *Apis mellifera* (Honeybee, Hymenoptera; forewings) are summarized and visually presented in **Figure 1**, which showcases several panels. Figure 1a displays the extracted cells, while Figure 1b illustrates the precise locations of the junctions. Figure 1c showcases the intricate venation patterns, with each vein color-coded according to its length, as indicated by the accompanying color bar.

Figure 1d provides insights into the distribution of cell areas, including a distribution plot (d.1), a corresponding histogram (d.2), and a boxplot (d.3). Similarly, panels (e) and (f) present the distributions of cell lengths and circularity, respectively, with their respective histograms and boxplots. Figure 1g illustrates a physical model of the segmented wing created through 3D printing. The model was generated in FC using the sketch obtained from WingSegment's FC macro export tool. Printing was carried out using a Prusa MK3S printer with a 0.25 mm nozzle and Prusa Jet black polylactic acid (PLA) filament. The file containing the 3D-printable model is provided as Model S1, Supporting Information.<sup>[53]</sup> Finally, Figure 1h presents the Voronoi pattern generated based on the centroids of the extracted wing cells.

### 2.1. Validation

The validation of WingSegment was conducted to assess the accuracy and reliability of the method in segmenting insect wing images and extracting vital features. **Table 1** and **2** present the results obtained during the validation process, demonstrating the performance of WingSegment. Figure 1h illustrates the numbered cells and five randomly selected veins represented by red dashed lines. The result regarding the length of all veins is stored as File S5, Supporting Information.<sup>[53]</sup> The subsequent analysis focused on comparing results between WingSegment and ImageJ. Table 1 provides the vein-related results, revealing an error rate of only 0.56%. This low error percentage underscores the precision of our method in detecting and measuring the intricate venation patterns present in insect wings. Moving on to cell analysis, Table 2 presents the obtained cell area measurements, circularity, and length from both WingSegment and ImageJ. The results indicate a high accuracy of cell area measurements, with negligible differences observed. The average error in cell circularity and length measurement is 3.70% and 1.59%, respectively.

Moreover, manual counting was performed on a complex *Crocothemis erythraea* (Scarlet dragonfly, Odonata: Anisoptera; forewings) to assess the cell and junction counting functionality



**Figure 1.** Visualization of the results obtained using WingSegment on an *Apis mellifera* (Honeybee, Hymenoptera; forewings). a) Identified cells, b) detected junctions, c) distribution of vein lengths, d–f) distribution of cell areas, lengths, and circularity. Each Figure panel consists of three subpanels: Panel 1 displays the heatmap, Panel 2 shows the corresponding histogram, and Panel 3 presents the boxplot. g) The 3D-printed physical wing model from the 3D model was generated using WingSegment. h) Labeled cells and veins corresponding with the results of Table 1 and 2 i) Generated Voronoi pattern.

of WingSegment. Video S1 and S2, Supporting Information<sup>[53]</sup> illustrate the manual marking process for cells and junctions using our custom code (Code S1, Supporting Information), respectively. The coordinates of cells' centroids and junctions are stored as File S1–S4, Supporting Information.<sup>[53]</sup> Remarkably, both methods yield the same count for cells (440) and junctions (816), highlighting the exceptional performance of WingSegment in accurately detecting cells and junctions.

### 3. Discussion

This study suggests a new hybrid method called WingSegment to segment insect wing images. The methodology employed in WingSegment combines region-growing and thinning algorithms for detecting cells and junctions, along with path-finding techniques for extracting venation patterns. This hybrid approach ensures efficient and accurate analysis of complex wing structures. It is important to emphasize that integrating the

**Table 1.** Comparison of measured vein lengths using WingSegment and ImageJ, along with the computed average error.

	WingSegment	ImageJ	Error [%]
	Vein Length [mm]	Vein Length [mm]	
Vein 1	2.35	2.33	0.77
Vein 2	0.84	0.83	0.72
Vein 3	1.57	1.56	0.45
Vein 4	2.97	2.95	0.68
Vein 5	4.61	4.60	0.17
		Average%	0.56

mentioned algorithms is crucial in this case, and considerable effort is required when employing these algorithms separately for the same task. In WingSegment, the embedded method extracts the locations of junctions by utilizing the thinning algorithm and the conditions introduced in the methodology. The pathfinding method then determines the path between the extracted junctions from the previous step. Additionally, the skeletonized image enables the algorithm to identify the available networks between junctions and uncover the true path, even if it is curved. This capability is not present in currently available tools like NET and NEFI. Performance evaluation was conducted on honeybee and *Crocothemis erythraea* (Scarlet dragonfly, Odonata: Anisoptera; hindwing) to assess the effectiveness of WingSegment (Figure 2). In addition, additional figures corresponding to the geometric features of a damselfly, a locust wing, and a Scarlet dragonfly forewing are presented in Figure S5–S8, Supporting Information.<sup>[53]</sup> This comprehensive evaluation allowed a thorough assessment of the software's performance across different insect species.

The results obtained from WingSegment demonstrate its effectiveness, as evidenced by the successful extraction of wing features with minimal errors in vein detection and cell measurement. The WingSegment validation involved comparing the results with those obtained from established tools such as ImageJ and manual measurements. The results obtained from WingSegment were in agreement with these reference methods, confirming the accuracy and reliability of our approach. It is important to note that both WingSegment and ImageJ employ image binarization as a preprocessing step. In WingSegment, we set the threshold value to 0.54, which provides optimal alignment with ImageJ results. Altering this threshold value would impact the outcome.

### 3.1. WingSegment versus WingGram

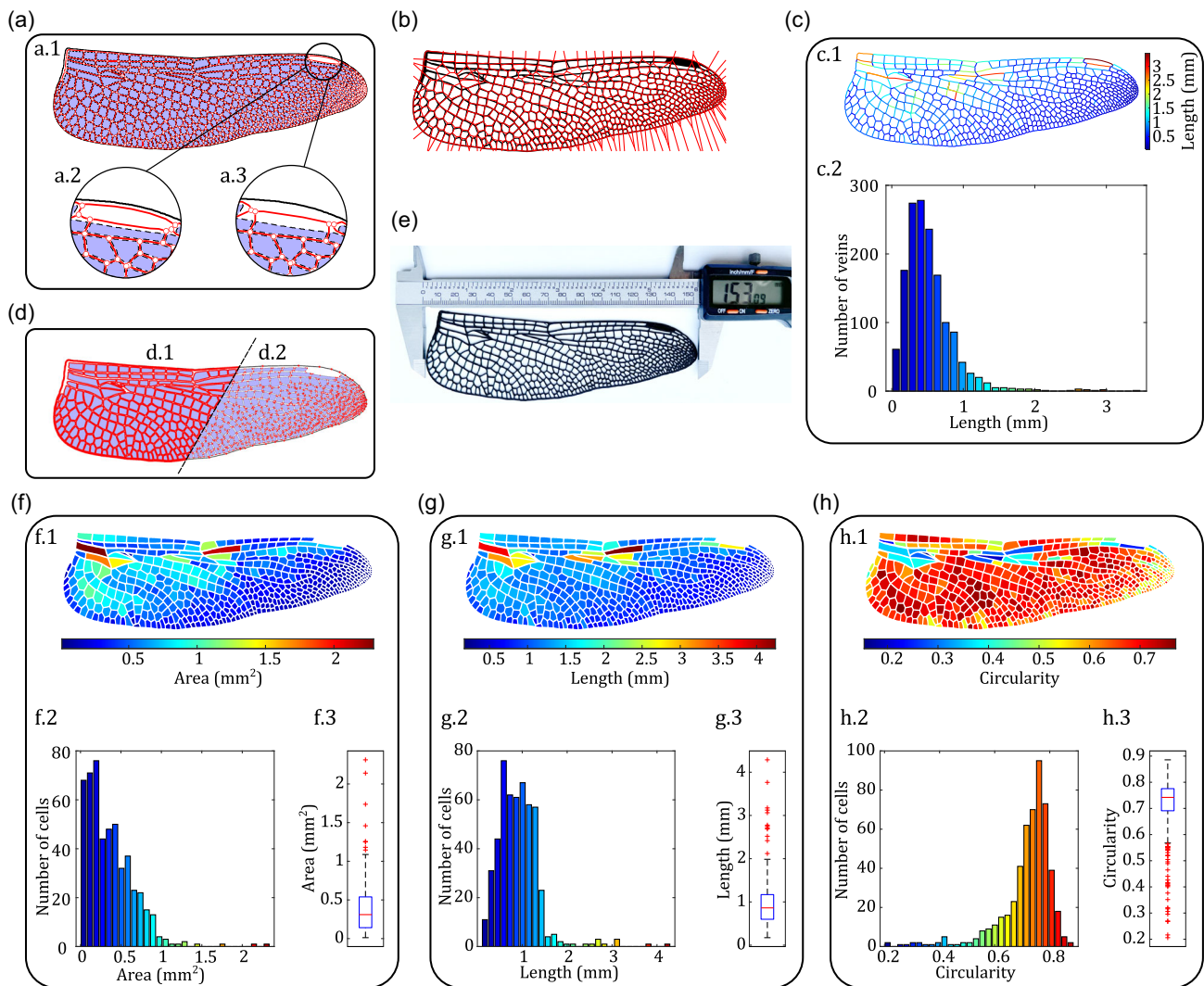
WingSegment, while visually resembling its counterpart WingGram,<sup>[29]</sup> incorporates notable distinctions and numerous enhancements. Both tools employ a similar approach to extract cell boundaries, but WingSegment addresses minor bugs from WingGram, resulting in enhanced reliability and speed. Figure 2a illustrates the extracted cells from a dragonfly wing, represented in blue color. While both WingGram and WingSegment provide measurements for cell area, length, and count, WingGram includes an additional feature that measures cell width by dividing the area by the length of cells. However, this approach was found to be imprecise and has been omitted from WingSegment. Instead, WingSegment incorporates cell circularity measurement, providing valuable information about cell shape.<sup>[27,28]</sup> Figure 2f–h depicts the distribution of cell area, length, and circularity using various visual formats, including heatmap (Panel 1), histogram (Panel 2), and boxplot (Panel 3). Panel 1 provides an overview of the distribution of cells and offers insights into their location based on area, length, or

**Table 2.** Comparison of measured cell area, circularity, and length using WingSegment and ImageJ, alongside the computed error.

	WingSegment			ImageJ			Error%		
	Area <sup>a)</sup>	Circularity	Length <sup>a)</sup>	Area <sup>a)</sup>	Circularity	Length <sup>a)</sup>	Area <sup>a)</sup>	Circularity	Length <sup>a)</sup>
Cell 01	6345	0.23	268.27	6348	0.24	266.71	0.05	2.10	0.59
Cell 02	1547	0.53	81.74	1547	0.57	79.46	0.00	6.37	2.87
Cell 03	5540	0.26	235.60	5543	0.27	234.03	0.05	2.58	0.67
Cell 04	3561	0.42	141.17	3563	0.44	140.01	0.06	4.08	0.83
Cell 05	2097	0.43	106.25	2100	0.45	103.92	0.14	5.76	2.24
Cell 06	2115	0.42	111.61	2116	0.45	108.63	0.05	5.61	2.74
Cell 07	11 343	0.55	199.46	11 347	0.57	199.74	0.04	3.00	0.14
Cell 08	2876	0.21	178.84	2878	0.22	174.44	0.07	2.78	2.52
Cell 09	5757	0.53	141.03	5760	0.55	137.06	0.05	3.83	2.90
Cell 10	7558	0.60	160.38	7564	0.62	159.70	0.08	3.87	0.42
Cell 11	8302	0.65	157.28	8303	0.67	154.11	0.01	3.15	2.05
Cell 12	5071	0.14	304.36	5076	0.15	301.94	0.10	2.07	0.80
Cell 13	9659	0.56	196.87	9662	0.58	193.30	0.03	2.93	1.84
				Average %			0.056	3.70	1.59

<sup>a)</sup>In this table, the units of area and length are  $\mu\text{m}^2$  and  $\mu\text{m}$ , respectively.





**Figure 2.** Evaluation of WingSegment performance on a *Crocothemis erythraea* (Scarlet dragonfly, Odonata: Anisoptera; hindwings). a) Extraction of cells, junctions, and venation patterns. Panels 2 and 3 demonstrate the impact of using and not using the distance transform function in Pterostigma detection, respectively. b) Generated Voronoi pattern based on extracted cell centroids. c) Visualization of vein length distribution. Panel 1 displays the heatmap, and panel 2 presents the histogram. d) Illustration of line simplification technique. Panels 1 and 2 depict point density before and after applying the line simplification method. e) Three-dimensional printed dragonfly wing. f-h) Analysis of cell area, length, and circularity, respectively. Panels 1, 2, and 3 show the heatmap, histogram, and boxplot.

circularity. This type of figure enables the examination of the relationship between cell size and its spatial positioning. The histogram in Panel 2 provides valuable information on how cells are distributed across different categories of area, length, and circularity. This allows for investigating the predominant size or shape of the cells within the population. Lastly, the boxplot in Panel 3 presents important statistical details such as the minimum, maximum, upper quartile, lower quartile, and median values of cell area, length, and circularity. These visualizations collectively provide insights into the characteristics and distribution of cells within the wing sample.

Furthermore, WingSegment offers additional features not present in WingGram. It provides distance measurements of cells centroids from the wing margins, allowing for a more

comprehensive analysis of the wing structure. Additionally, WingSegment includes automatic detection of the Pterostigma, a specialized region within the wing. In terms of junctions, both tools are capable of locating and quantifying junctions. However, WingSegment surpasses WingGram by identifying the ends of unblocked veins as junctions, which provides a more accurate representation of the wing's connectivity. Notably, both WingGram and WingSegment can detect cell centroids on insect wings. However, WingSegment stands out with its built-in feature for generating a Voronoi pattern based on the extracted centroids. Figure 2b illustrates the Voronoi pattern generated by WingSegment on a dragonfly wing. This Voronoi pattern is a significant aspect of WingSegment, as it offers insights into the arrangement and distribution of cells on the wing surface.<sup>[27,54]</sup>

### 3.2. WingSegment versus WingMesh

WingMesh is another software developed for mesh generation on insect wing geometries.<sup>[32]</sup> It was created by integrating image processing with Distmesh2D,<sup>[38]</sup> a well-known mesh generator in MATLAB. In WingMesh, the wing's geometry is extracted using image analysis, and Distmesh2D is then employed to generate a mesh on the extracted geometry. The output mesh in WingMesh is based on INP files compatible with Abaqus CAE. While WingMesh represents an advancement in automated mesh generation for insect wing geometries, it is noteworthy that it generates orphan meshes in Abaqus. Consequently, manipulating the geometry on the finite element (FE) models generated by WingMesh is impossible. Moreover, WingMesh was slow, particularly demonstrating reduced functionality on wings with complex geometries, such as dragonfly wings. **Table 3** provides a side-by-side comparison illustrating the key differences between WingGram, WingMesh, and WingSegment.

### 3.3. Venation Pattern Recognition

One distinguishing feature of WingSegment, which was not incorporated in WingGram, is its ability to extract the network pattern of insect wings and accurately determine the paths between junctions. While tools like NEFI<sup>[35]</sup> and NET<sup>[34]</sup> excel in extracting network patterns from various structures such as leaves and spider webs, WingSegment offers a unique advantage in capturing the intricate details of wing venation. NEFI and NET primarily focus on network extraction, representing connections between junctions with straight lines, which is suitable for their intended purposes as network analysis tools. However, WingSegment goes beyond network extraction by precisely capturing the curved paths between junctions, allowing for a more

**Table 3.** Comparison between WingSegment, WingGram, and WingMesh.

Tools and options	WingSegment	WingGram	WingMesh
Abaqus INP model	No	No	Yes
Abaqus JNL model	No	Yes	No
FreeCAD model	Yes	No	No
Cells' area	Yes	Yes	No
Cells' length	Yes	Yes	No
Cells' width	No	Yes	No
Cells' circularity	Yes	No	No
Cells' centorids	Yes	Yes	No
Cells' distance from margin	Yes	No	No
Voronoi pattern generation	Yes	No	No
Junction detection	Yes <sup>a)</sup>	Yes	No
Venation pattern detection	Yes	No	No
Pterostigma detection	Yes	No	No
Applicable for complex geometries	Yes	Yes	No
D printable model generation	Yes	Yes	No
Corrugation assignment	No	Yes	Yes

<sup>a)</sup>Improved and Modified.

comprehensive representation of the intricate venation pattern and providing valuable insights into the wing's structural characteristics. It is important to highlight the specific focus and strengths of NEFI and NET in network analysis while emphasizing that WingSegment complements these tools by offering a specialized feature for path extraction.

Figure 2a demonstrates the successful extraction of the intricate venation pattern from a complex dragonfly wing represented by red color. Notably, when dealing with a dragonfly wing featuring a Pterostigma, Figure 2a.2 showcases the effectiveness of the distance transform in detecting veins around the Pterostigma. In contrast, Figure 2a.3 illustrates the detection of a vein within the Pterostigma when this option is not utilized. This function enhances vein detection and leads to more accurate junction detection. Furthermore, Figure 2c provides insight into the distribution of veins based on their length.

Additionally, this article presents two approaches for detecting venation patterns: Sequential vein detection (SVD) (Video S5, Supporting Information) and parallel vein detection (PVD) (Video S4, Supporting Information).<sup>[53]</sup> While PVD offers faster processing due to its parallel nature, it is essential to note that accurately extracting venation patterns is a highly delicate process. Initially, we developed SVD and subsequently enhanced its speed. Following that, we developed PVD. Our observations revealed that while both methods generally detect all veins within the wing, there are instances where one of the methods may miss some veins. Consequently, we incorporated both methods into WingSegment to ensure comprehensive vein detection.

### 3.4. FreeCAD Macro Export: Seamless Transfer of the Wing Geometry to FreeCAD

In terms of finite element modeling (FEM), WingGram surpasses WingSegment with its array of tools for modeling insect wings. WingGram offers functionalities such as marking discontinuities to exclude specific areas from the model and an innovative tool for corrugation assignment. Furthermore, WingGram can export the final model as an Abaqus JNL file. In contrast, WingSegment generates a FreeCAD macro file from extracted cell boundaries, allowing for the import of a wing cell sketch into FreeCAD. However, it is essential to consider that Abaqus is not open access, limiting accessibility for many users. In contrast, the FC macro exporting tool in WingSegment, although lacking corrugation assignment capabilities, generates a file compatible with FreeCAD—an open-source and accessible FEM software package. This aspect enhances the usability and practicality of WingSegment, particularly for users without access to specialized software like Abaqus. Furthermore, this seamless integration empowers users to easily create FE or 3D printable models within the FreeCAD environment. The practical utility of WingSegment is exemplified in Figure 2e, showcasing a successful application of 3D printing. More examples of dragonfly, damselfly, and locust 3D printed wings with complex geometries are also available in Figure S4, Supporting Information.<sup>[53]</sup> The physical wing was produced using a Prusa MK3S 3D printer, employing a 0.25 mm nozzle size. Prusament PLA Jet black filament was used to fabricate the wing in Figure 2e, and Prusament PLA Galaxy Silver was used in producing wings in Figure S4,

Supporting Information. The nozzle temperature was set at a constant value of 225 °C, while the bed temperature was maintained at 65 °C. To ensure precise replication of all features, the wing model was scaled  $\approx 4.5$  times its original size, resulting in a 3D-printed wing that accurately represents the morphology of the original insect wing. The 3D printable model of the wing is stored as Model S2, Supporting Information.<sup>[53]</sup>

It is essential to highlight that, like WingGram, WingSegment extracts all points along the boundaries of cells, which can lead to many unnecessary points. These plenty of points significantly increase the volume of the macro file, consequently increasing the runtime during the importing process in FreeCAD. Therefore, it is crucial to recognize the importance of employing line simplification techniques before generating the macro file. This is exemplified in Figure 2d, where the density of points before and after line simplification is showcased in Figure 2d.1,d.2, respectively.

### 3.5. Robustness of WingSegment

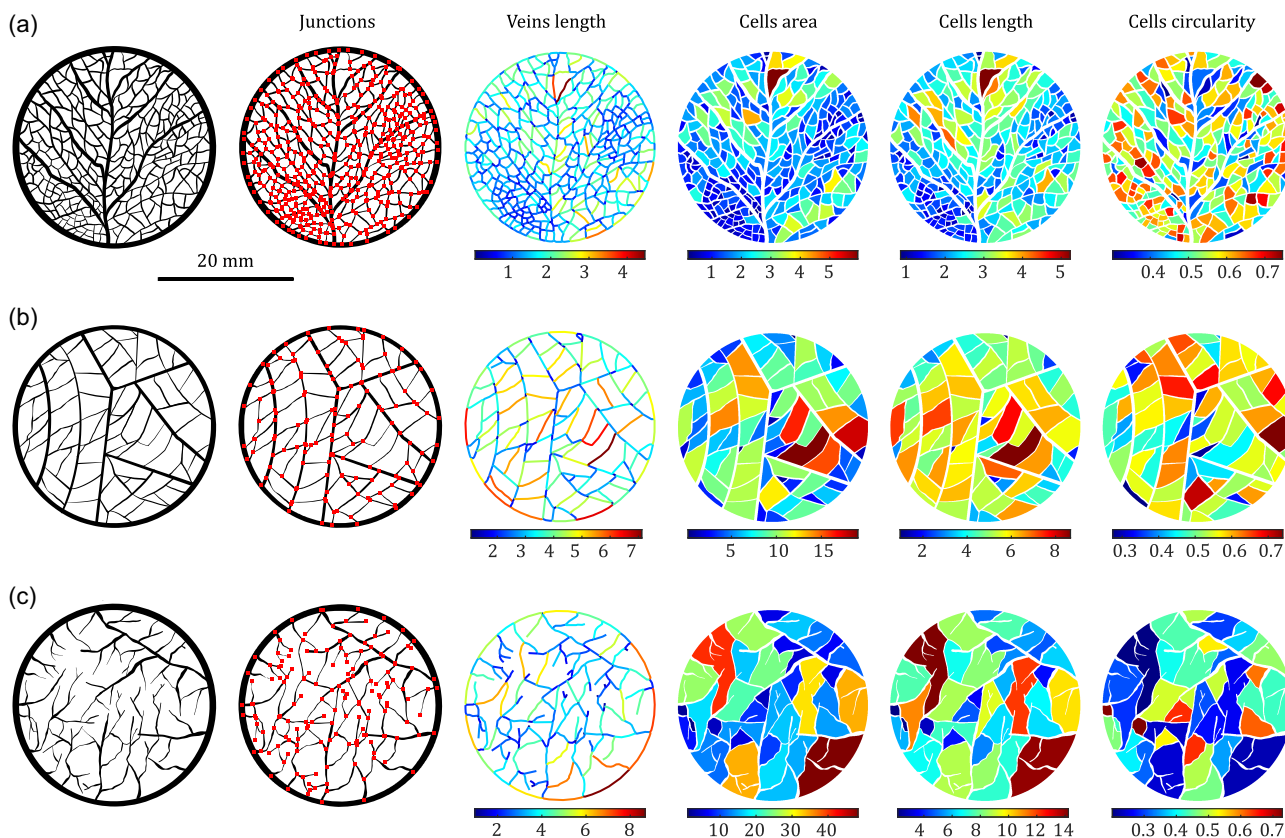
WingSegment demonstrates exceptional versatility in segmenting wings with both simple and complex geometries. This capability is exemplified through the segmentation of a straightforward honeybee wing (Figure 1) and the intricate hind wing of a Scarlet dragonfly (Figure 2). Furthermore, to validate the robustness of cell and junction extraction, we utilized a Scarlet dragonfly forewing, showcasing the software's

performance (Figure S1, Supporting Information).<sup>[53]</sup> We have also presented additional 3D printed models of wings with a high number of cells and geometric complexity in Figure S4, Supporting Information.<sup>[53]</sup> These wings are from the damselfly, locust, and Scarlet dragonfly forewings.

In practice, WingSegment adeptly handles various wing complexities but is more sensitive to image quality. While very high-resolution images may impact runtime, excessively low-resolution images can affect software performance. In our assessments, images with a resolution of 300 dpi were found optimal.

It is important to note that WingSegment excels when the wing image is clear. The software successfully extracts features, but the user must ensure that the image quality meets certain criteria. Images with thin veins, faded details, or noise may not yield optimal results. WingSegment distinguishes between black and white pixels, which is crucial in cell and vein detection tasks. However, this feature has limitations; for instance, black pixels are considered veins in the region-growing method. Therefore, WingSegment is particularly well-suited for wings with transparent membranes and dark veins. Challenges arise when dealing with transparent veins, necessitating user intervention to enhance image contrast.

It's worth mentioning that WingSegment may not be suitable for wings with dark colors or black spots due to its pixel differentiation method. Nonetheless, any wing image can be pre-processed to meet the software's requirements. To address these limitations, we are actively developing filters and incorporating



**Figure 3.** Illustration of WingSegment performance beyond insect wings. a) Leaf vein pattern. b,c) Soil crack pattern.



deep learning and machine learning methodologies to enhance software flexibility, minimizing the need for manual editing or pre-processing. Our ongoing efforts include integrating venation sections and corrugations into final 3D models generated with WingSegment.

### 3.6. Applications of WingSegment

WingSegment may play a pivotal role in advancing the study of insect wings by offering automated solutions, significantly reducing the effort required for analyzing mass samples. The software's ability to rely solely on wing images without additional specialized information enhances its accessibility and applicability. WingSegment may contribute substantially to understanding the asymmetry of insect wings, extracting various geometric features from images.<sup>[55]</sup> Moreover, it will facilitate the design and development of bio-inspired and biomimetic wings by generating intricate 3D models.<sup>[16,56]</sup> The tool also may aid in studying the functional morphology of wing venation patterns, cells, and junctions through automated extraction of these elements from wing images. Beyond entomology, scientists can harness WingSegment for feature extraction in machine learning and deep learning applications, enabling the development of automated identification, modeling, and analysis methods for insect wings.<sup>[57–60]</sup> Notably, WingSegment's versatility extends beyond insect wings, as it can extract features from 2D images of other cellular structures such as leaves, skins, and foam-like formations<sup>[61]</sup> like that illustrated in **Figure 3**. Additionally, as depicted in **Figure 3b,c**, it can contribute to the investigation of soil and the development of clay cracks.<sup>[62]</sup> This broadens its impact on diverse fields, including developing and designing cellular thin films, geocells, and modeling and analyzing leaves and plant species determination.<sup>[63]</sup> We envision WingSegment as an invaluable tool in entomology and the comprehensive study of various planar biological structures. WingSegment showcases advancements in designing and developing 3D modeling and printing of wings and exhibits their potential applicability to a wide array of planar structures. This can contribute to advancing 3D printing technologies.<sup>[64]</sup>

## 4. Conclusion

In conclusion, this article introduces WingSegment, a novel computer vision-based hybrid method for insect wing segmentation, venation extraction, and FreeCAD modeling. By leveraging the advantages of computer vision, graph theory, and image analysis, WingSegment offers a powerful solution for studying and analyzing complex insect wings. The proposed method demonstrates remarkable accuracy in extracting cell boundaries, precisely detecting venations, and capturing junctions. Additionally, WingSegment provides various geometric statistical features, such as cell area, length, and circularity. Moreover, the integration of a user-friendly graphical user interface (GUI) and the generation of FreeCAD macro files enhance the accessibility and usability of the software.

The experimental results, showcasing the application of WingSegment on a honeybee wing, highlight the successful extraction of cells and veins and the subsequent creation of a

detailed 3D model. Validation using ImageJ software further confirms the reliability and effectiveness of WingSegment. Furthermore, WingSegment and the previously developed tools, WingGram and WingMesh, are comparatively discussed (**Table 3**). While WingSegment surpasses WingGram in venation pattern extraction and circularity measurement, WingGram retains its own advantages for FE modeling of insect wings. Notably, the accessibility and affordability of WingSegment, which employs FreeCAD as a platform, make it an attractive option for students and researchers. Our development simplifies the study of insect wings and opens avenues for future developments. With its versatility, the presented software is exemplified by exploring a more complex dragonfly wing, further demonstrating the method's robustness—intricate insect wing structures, empowering researchers and students with valuable study tools.

## 5. Experimental Section

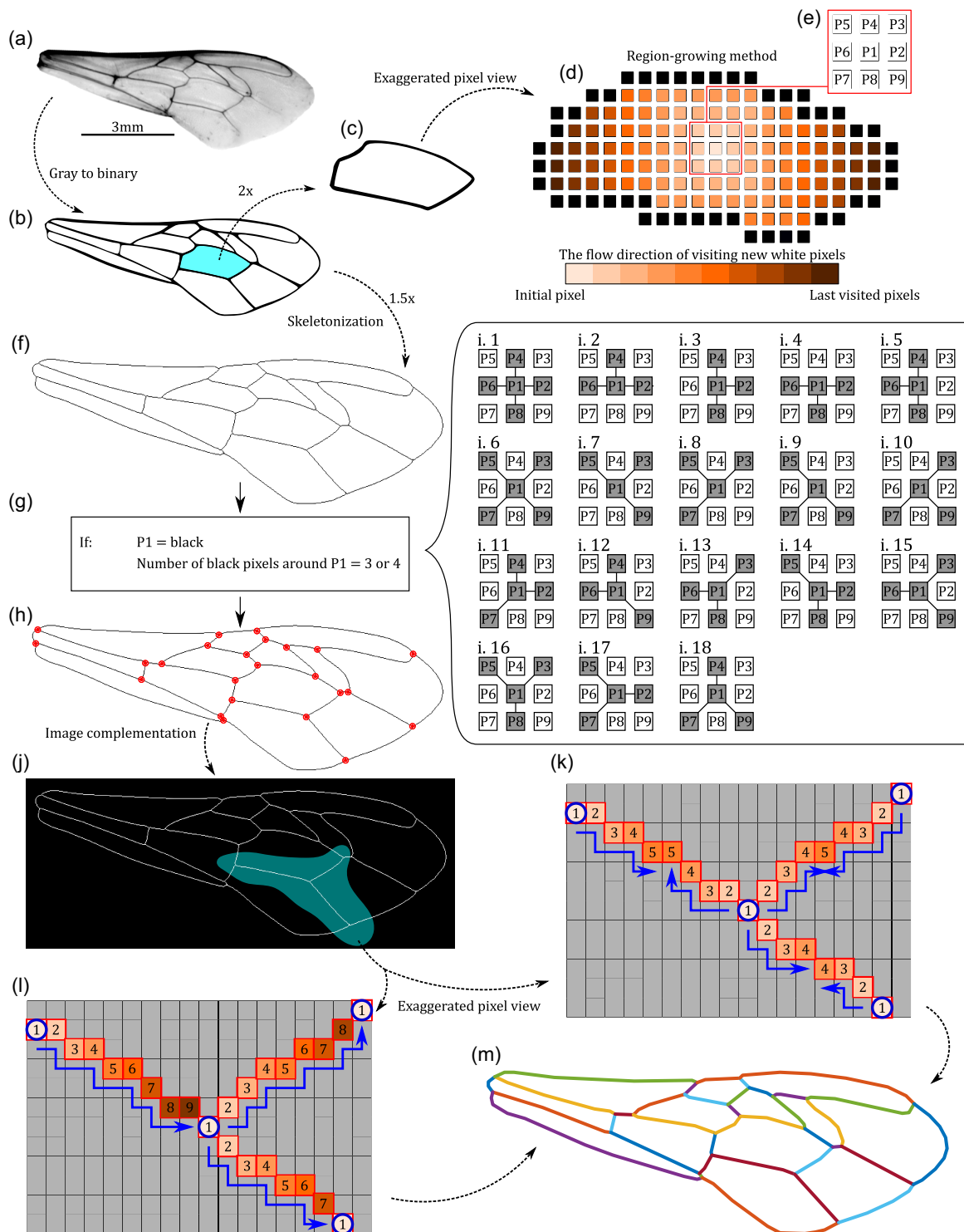
This study proposes a hybrid approach, WingSegment, for segmenting insect wings by integrating various algorithms in image analysis and graph theory. The algorithm is accompanied by a user-friendly graphical interface designed in MATLAB App-designer. WingSegment can detect cells, junctions, and vein paths in the insect wing by taking an image of the wing as input. A region-growing algorithm is incorporated into WingSegment to detect the boundary of cells and the wing. The thinning algorithm in image processing is utilized to locate the junctions. A combination of Dijkstra's and thinning algorithms is employed to detect the venation patterns in the wing. In addition, a distance transform function is incorporated to identify the presence of the Pterostigma in the wing. Lastly, a line simplification method compresses the detected cells and venation patterns.

**Image Preparation:** A high-quality image is required since we intend to present this method to segment the insect wing image in detail. A camera, scanner, or microscope can be used for taking images. **Figure 4a** is a sample image of a honeybee taken by a high-resolution scanner (Epson Perfection V850 Pro). Since the method turns the image into a binary image with only black and white pixels with a thresholding of 0.54, the user should ensure that all veins and membranes remain apparent after binarizing the image. **Figure 4b** represents an example of an appropriate image for use in WingSegment. Code S2, Supporting Information,<sup>[53]</sup> is designed for importing the image and performing the binarization process.

**Cell Domain Extraction:** To segment cell domains, a region-growing method is utilized in this study.<sup>[40–42,65]</sup> **Figure 4d** provides an exaggerated pixel view of the domain in **Figure 4c** and illustrates how the method finds a region from a starting point. Five sets are defined to describe the method, including visited white pixels, unvisited white pixels, visited black pixels, unvisited black pixels, and current pixels. The algorithm involves five steps, starting with marking all white pixels as unvisited. Then, a random initial white pixel within a cell is selected as the current pixel, marked as visited and removed from the unvisited white pixels set. Then, all neighbors' unvisited pixels around the current pixels are checked. If they are white, they are marked as visited white pixels, replaced with the current pixel, and removed from the unvisited white pixels. If they are black pixels, they are marked as visited black pixels and removed from the unvisited black pixels. The algorithm continues while the unvisited white pixels set is not empty, and when it becomes empty, the boundary of that domain is extracted. Video S3, Supporting Information, illustrates this procedure on a honeybee wing. In ref. [53], Code S3, Supporting Information, depicts the function for implementing the region-growing method.

**Cell Domain Extraction: Generating Voronoi Pattern:** Incorporating a method for extracting the cell boundaries into WingSegment facilitates the calculation of cell centroids. This has been integrated into the WingSegment algorithm, allowing for generating a Voronoi pattern by





**Figure 4.** Illustration of the implemented methodology in WingSegment. a) Image of a honeybee wing, b) binary representation of the wing image, c) visualization of a single cell, d) exaggerated pixel view of a single cell, e) representation of a pixel and its surrounding pixels, f) Skeletonized image of the wing, g) criteria for identifying junction locations, h) detected junctions, i.1–i.18) pixel-level depiction of the formula shown in panel, g,i) complemented skeletonized image, k) sequential vein detection (SVD) method representation, l) Parallel vein detection (PVD) method representation, m) detected veins in the wing.

utilizing the centroids of the segmented cells. By accurately determining the centroids, WingSegment enables the visualization of a Voronoi pattern that represents the spatial distribution of insect wing cells.

**Cell Domain Extraction: Evaluating Geometric Features:** During the process of the boundary extraction, WingSegment simultaneously detects the number of pixels in each boundary. This information is utilized to measure the area of individual cells accurately. Moreover, WingSegment measures the maximum distance inside each boundary, corresponding to the cell's length. Additionally, since the boundary of each cell is already extracted, WingSegment can measure the perimeter of each cell. By utilizing these calculated values and employing Equation (1), WingSegment determines the circularity of each cell. The circularity ranges between 0 and 1, where 1 indicates the highest level of circularity, representing how closely a shape resembles a perfect circle.<sup>[66]</sup> The circularity is quantified using Equation (1).

$$C = (4\pi * \text{area}) / (\text{perimeter}^2) \quad (1)$$

where, the area represents the measured area of the shape, and the perimeter represents the measured perimeter of the shape. In ref. [53], Code S4, Supporting Information, represents the function responsible for quantifying all geometric features through the utilization of the region-growing method.

**Cell Domain Extraction: Generating FC Macro File of Detected Boundaries:** In FreeCAD, a powerful 3D modeling software, the term "FC macro" refers to a macro script written in the FreeCAD scripting language.<sup>[50,51]</sup> FreeCAD is an open-source parametric 3D modeling software that allows users to create and modify 3D designs using a graphical interface. However, it also provides a scripting interface allowing users to automate tasks and create custom Python tools. FC macro in FreeCAD is a Python script specifically written to extend the functionality of FreeCAD. Users can create their macros to perform repetitive tasks, automate design processes, or add new features to the software. These macros can be executed within the FreeCAD environment, providing additional functionality beyond what is available through the graphical interface alone.

In addition to automatically detecting cells from the insect wing image, WingSegment incorporates a valuable tool that allows users to export the detected boundaries in an FC macro file format. This feature enables easy transfer of the wing geometry, specifically the boundaries of the detected cells, into FreeCAD. By exporting the boundaries as an FC macro file, users can conveniently import the wing geometry into FreeCAD and perform various operations such as extrusion, generating finite element (FE) models, or creating 3D printable models. This functionality enhances the usability of WingSegment, providing users with a seamless pathway to further analyze and utilize the detected cell boundaries in their design and simulation processes. In ref. [53], Code S8, Supporting Information, corresponds to the code developed for generating the FCMacro file. Video S7, Supporting Information, shows how the FC Macro files generated by WingSegment can be imported in FreeCAD.<sup>[53]</sup>

**Junction Detection:** In insect wings, a junction refers to a point where two or more wing veins meet or intersect. The method for detecting wing junctions involves a series of steps. First, the width of all veins is reduced to one pixel using the Zhang-Suen thinning algorithm.<sup>[47]</sup> This process is called skeletonization, commonly used in image processing to reduce an object to its core components. In this case, skeletonization is used to reduce the width of the veins to a single pixel, making it easier to detect intersections.<sup>[29,35,37]</sup> The skeletonized image is shown in Figure 4f, consisting of a network of thin lines representing the veins in the wing. To detect the junctions, the skeletonized image is examined for black pixels that are surrounded by three or four black pixels (Figure 4g), as shown in the conditions illustrated in Figure 4i.1–i.18. These conditions take into account the position of the black pixels relative to their neighbors, and ensure that they meet specific geometric criteria.<sup>[29,35]</sup> Any black pixel that meets these conditions is considered a junction. The junctions are then marked and identified in the image, as shown in Figure 4h. By locating the junctions in this way, it is possible to accurately map the venation patterns in the wing and obtain a detailed understanding of its structure. Code S5,

Supporting Information, represents the function developed for detecting junctions in the Supporting Information.<sup>[53]</sup>

**Vein Path Extraction:** We use path-finding, also known as the pathing method, to extract venation patterns here. This technique involves using a computer program to draw the most direct path between two points and is a more useful version of solving mazes. Path-finding methods rely heavily on Dijkstra's algorithm,<sup>[67]</sup> which is used to find the shortest possible route on a graph with weights assigned to its edges. We have adapted this algorithm to find every path between the junctions in a wing using the following approaches.

**Vein Path Extraction: Sequential Vein Detection (SVD):** This approach involves defining three sets: initial, ending, and front pixels. Each pixel in a set has a so-called parent-child relationship, where the parent indicates the origin of the child from the previous iteration. To initiate the process, one of the identified junctions is designated as the initial pixel, serving as the starting pixel for a generation (Figure 4l, Pixel 1). The initial pixel designates the surrounding pixels as front pixels. Since all veins in the skeletonized image are represented as white pixels, the method transfers these white pixels from the front set to the set of detected pixels. The initial pixel is considered the parent of all pixels detected afterward. Subsequently, the initial pixels are replaced by newly detected pixels, and the same procedure is repeated for each newly generated initial pixel. This process continues until a new junction is among the front pixels, indicating the detection of a vein's ending pixel. In such cases, the pixel that detects the junction and its parent begin to trace its way back to the starting pixel of its generation, with each parent returning to its own parent. This iterative process continues until they reach the initial pixel, resulting in the respective detection of all pixels associated with that vein. This procedure is then repeated for each junction, one at a time, until all veins are detected. Figure 4l visually illustrates this entire process. Furthermore, Video S5, Supporting Information, demonstrates the detection of veins on honeybee wings using the SVD method. The code designed for implementing this method is provided and explained in detail under Code S7, Supporting Information.<sup>[53]</sup>

**Vein Path Extraction: PVD:** The parallel method considers all junctions simultaneously as initial pixels and begins finding paths concurrently. Consequently, as depicted in Figure 4k, the front pixels converge at the center of the connecting vein where two junctions are connected. Video S4, Supporting Information, showcases the application of the PVD method for detecting veins on honeybee wings. The code created for this approach is presented and described under Code S6, Supporting Information.<sup>[53]</sup>

**Pterostigma Detection:** The Pterostigma is a feature commonly found in the wings of certain insect orders including Odonata,<sup>[1,68,69]</sup> Hymenoptera,<sup>[70]</sup> Orthoptera,<sup>[71]</sup> and Hemiptera.<sup>[72]</sup> However, the size, shape, and coloration of Pterostigma can vary among different species. Image processing operations are applied to detect the Pterostigma in the wing. 1) Distance transform.<sup>[73]</sup> The first step involves calculating the distance transform of the wing image. The distance transform assigns a value to each pixel in the image, representing its distance to the nearest object. The Pterostigma, being a concentrated region within the wing, causes the distance transform to assign the highest value at its central location. 2) Normalization. The resulting distance transform image is then normalized in a range of 0–1 using the max-min normalization method. This normalization step ensures that the maximum distance has a value of 1, facilitating subsequent thresholding. 3) Thresholding. We have observed that a threshold of 0.7 applies to the normalized distance transform image. Pixels with values above this threshold are set to 1, while those below or equal to the threshold are set to 0. This thresholding operation helped isolate regions in the image that are likely to contain the Pterostigma. 4) Overlaying. Finally, the threshold image is added to the original image to enhance the detected Pterostigma visually. This overlaying operation helped highlight the location of the Pterostigma by visually merging the threshold regions with the original wing image.

Following these steps, the code successfully detects and localizes the Pterostigma in the wing image.

**Line Simplification:** The Ramer–Douglas–Peucker algorithm, also known as the Douglas–Peucker algorithm and iterative end-point fit algorithm, simplifies a curve composed of line segments.<sup>[48,49]</sup> Its purpose is to

reduce the number of points while maintaining the overall shape. Originally developed for cartographic generalization, this algorithm aims to create a similar curve using fewer points. In the context of WingSegment, where both the region-growing and path-finding methods detect all pixels, including many unnecessary ones, the Ramer–Douglas–Peucker algorithm is employed. By integrating this algorithm, WingSegment can eliminate redundant points and retain the critical ones that contribute to preserving the entire shape of the detected boundaries.

**GUI:** WingSegment is accompanied by an accessible, user-friendly GUI. The GUI is developed using MATLAB App-Designer and can be accessed in two formats via the Supplementary Files: an executable file (EXE) for installation as standalone software on Windows desktops and an MLAPPINSTALL file for installation as an app within MATLAB. The developed GUI enables users to import wing images and utilize various tools such as cell and vein detection, junction identification, FreeCAD export functionality, and statistical analysis visualizations. Figure S3, Supporting Information, shows a detailed visual depiction of WingSegment's GUI. Moreover, in Video S6, Supporting Information, we demonstrate how to use WingSegment from start to finish.<sup>[53]</sup>

**Validation:** The performance of WingSegment was rigorously validated through comparisons with established software tools and manual measurements. First, we utilized ImageJ, a widely recognized software, to measure cells' area, length, and circularity. Additionally, ImageJ was employed to measure the length of veins. However, ImageJ automatically detects the cell boundaries and measures the area of the domain. Nevertheless, vein length, cell length, and cell circularity were measured manually. These measurements were then compared to the corresponding results obtained from WingSegment, ensuring the accuracy and reliability of our method. Moreover, we manually measured the number of cells and junctions in a Scarlet dragonfly wing, which served as a reference for comparison. To achieve this, we developed a custom code (available in Code S1, Supporting Information) that enables users to import a wing image and mark junction or cell locations with a simple click. This code facilitated the quantification of cells and junctions in a Scarlet dragonfly. Video S1 and S2, Supporting Information, respectively show our manual counting of cells and junctions using Code S1, Supporting Information.<sup>[53]</sup> This allowed us to evaluate the automation capabilities of WingSegment, as it automatically extracted and quantified these features.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

We thank Mr. Mohsen Jafarpour (Kiel University), Dr. Chuchu Li (Kiel University), and Dr. Thies Büscher (Kiel University) for their invaluable insights during the paper's preparation. Special recognition is extended to Ms. Atieh Mirtorabi (atieh.mirtorabi@gmail.com) for her exceptional work in illustrating the drawings in Figure 4. Furthermore, we acknowledge the invaluable assistance of Hannah Ramien (Kiel University) in preparing the wing images. Gratitude is also extended to the German Academic Exchange Service (DAAD) for their generous financial support under the "Research Grants - Doctoral Programmes in Germany, 2019/20 (57440921)" awarded to Shahab Eshghi.

Open Access funding enabled and organized by Projekt DEAL.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

S.E., H.R., and S.G.: conceptualization; S.E. and J.P.: data curation; S.E. and S.G.: funding acquisition; S.E.: methodology; H.R. and S.G.: project administration; S.G.: resources; S.E.: software; H.R. and S.G.: supervision; S.E. and J.P.: validation; S.E.: visualization; S.E.: writing—original draft preparation; S.E., H.R., J.P., and S.G.: writing—review and editing.

## Data Availability Statement

The data that support the findings of this study are openly available in [Zenodo] at [https://zenodo.org/doi/10.5281/zenodo.8014587], reference number [53].

## Keywords

entomology, FreeCAD, maze-solving, morphology, network extraction

Received: October 31, 2023

Revised: February 6, 2024

Published online:

- [1] H. Rajabi, S. Gorb, *Int. J. Odonatol.* **2020**, 23, 23.
- [2] S. Combes, T. Daniel, *J. Exp. Biol.* **2003**, 206, 2979.
- [3] F. Johansson, M. Söderquist, F. Bokma, *Biol. J. Linn. Soc.* **2009**, 97, 362.
- [4] S. Combes, T. Daniel, *J. Exp. Biol.* **2003**, 206, 2989.
- [5] S. Krishna, M. Cho, H. Wehmann, T. Engels, F. Lehmann, *Insects* **2020**, 11, 466.
- [6] J. Rudolf, L. Wang, S. Gorb, H. Rajabi, *J. Mech. Behav. Biomed. Mater.* **2019**, 99, 127.
- [7] J. Hasan, A. Roy, K. Chatterjee, P. Yarlagadda, *ACS Biomater. Sci. Eng.* **2019**, 5, 3139.
- [8] A. Khaheshi, S. Gorb, H. Rajabi, *Adv. Sci.* **2021**, 8, 2004388.
- [9] A. Khaheshi, S. Gorb, H. Rajabi, *Appl. Phys. A* **2021**, 127, 181.
- [10] A. Khaheshi, H. Tramsen, S. Gorb, H. Rajabi, *Mater. Des.* **2021**, 198, 109354.
- [11] S. H. Eraghi, A. Toofani, A. Khaheshi, M. Khorsandi, A. Darvizeh, S. Gorb, H. Rajabi, *Adv. Sci.* **2021**, 8, 2004383.
- [12] A. Toofani, S. H. Eraghi, M. Khorsandi, A. Khaheshi, A. Darvizeh, S. Gorb, H. Rajabi, *Acta Biomater.* **2020**, 110, 188.
- [13] M. Jafarpour, S. N. Gorb, H. Rajabi, *Adv. Eng. Mater.* **2023**, 25, 13.
- [14] M. Jafarpour, S. Gorb, H. Rajabi, *J. R. Soc. Interface* **2023**, 20, 20220757.
- [15] H. Rajabi, S. H. Eraghi, A. Khaheshi, A. Toofani, C. Hunt, R. Wootton, *Proc. Natl. Acad. Sci.* **2022**, 119, e2211861119.
- [16] K. Saito, H. Nagai, K. Suto, N. Ogawa, Y. Seong, T. Tachi, R. Niiyama, Y. Kawahara, *Sci. Rep.* **2021**, 11, 18631.
- [17] L. Dufour, K. Owen, S. Mintchev, D. Floreano, in *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, NJ **2016**, pp. 1576–1581.
- [18] X. Bao, A. Bontemps, S. Grondel, E. Cattani, *J. Micromech. Microeng.* **2011**, 21, 125020.
- [19] C. Li, S. Gorb, H. Rajabi, *Acta Biomater.* **2021**, 134, 490.
- [20] L. Mishnaevsky Jr, M. Jafarpour, J. Krüger, S. Gorb, *Biomimetics* **2023**, 8, 448.
- [21] A. Preuss, T. Büscher, I. Herzog, P. Wohlsein, K. Lehnert, S. Gorb, *Commun. Biol.* **2024**, 7, 36.
- [22] T. Büscher, S. Gorb, *Biomimetics* **2022**, 7, 173.
- [23] J. Winand, S. Gorb, T. Büscher, *J. Comp. Physiol. A* **2023**, 209, 313.
- [24] N. Chitsaz, R. Marian, J. Chahl, *PLoS One* **2020**, 15, e0232193.

- [25] N. Chitsaz, R. Marian, A. Chitsaz, J. Chahl, *Appl. Sci.* **2020**, *10*, 5389.
- [26] T. Doan, C. Nguyen, *Information Processing in Agriculture*, Elsevier, China **2023**.
- [27] J. Hoffmann, S. Donoughe, K. Li, M. Salcedo, C. Rycroft, *Proc. Natl. Acad. Sci.* **2018**, *115*, 9905.
- [28] M. Salcedo, J. Hoffmann, S. Donoughe, L. Mahadevan, *Biol. Open* **2019**, *8*, bio040774.
- [29] S. Eshghi, F. Nabati, S. Shafaghi, V. Nooraefar, A. Darvizeh, S. Gorb, H. Rajabi, *Sci. Rep.* **2022**, *12*, 13917.
- [30] H. Rajabi, V. Schroeter, S. Eshghi, S. Gorb, *Biol. Open* **2017**, *6*, 1290.
- [31] R. Korkmaz, H. Rajabi, S. Eshghi, S. N. Gorb, T. H. Büscher, *Insect Sci.* **2022**, *30*, 1507.
- [32] S. Eshghi, V. Nooraefar, A. Darvizeh, S. Gorb, H. Rajabi, *Insects* **2020**, *11*, 546.
- [33] S. Eshghi, H. Rajabi, A. Darvizeh, V. Nooraefar, A. Shafiei, T. Mirzababaei Mostofi, M. Monsef, *Sci. Iran.* **2016**, *23*, 2194.
- [34] J. Lasser, E. Katifori, *Source Code Biol. Med.* **2017**, *12*, 4.
- [35] M. Dirnberger, T. Kehl, A. Neumann, *Sci. Rep.* **2015**, *5*, 15669.
- [36] A. Dobens, L. Dobens, *G3: Genes, Genomes, Genet.* **2013**, *3*, 1443.
- [37] A. Tofilski, *J. Insect Sci.* **2004**, *4*, 17.
- [38] P. Persson, G. Strang, *SIAM Rev.* **2004**, *46*, 329.
- [39] T. Barrett, D. Savage, M. Ardeljan, M. Knezevic, *Comput. Mater. Sci.* **2018**, *141*, 269.
- [40] D. Chopp, *SIAM J. Sci. Comput.* **2001**, *23*, 230.
- [41] W. Lindquist, S. Lee, D. Coker, K. Jones, P. Spanne, *J. Geophys. Res.: Solid Earth* **1996**, *101*, 8297.
- [42] Y. Chang, X. Li, *IEEE Trans. Image Process.* **1994**, *3*, 868.
- [43] Y. Murata, Y. Mitani, in *SICE Annual Conf. 2011*, IEEE, Piscataway, NJ **2011**, pp. 32–33.
- [44] Y. Murata, Y. Mitani, *J. Image Graphics* **2014**, *2*, 89.
- [45] N. Kaur, *Int. J. Sci. Res. Dev.* **2019**, *6*, 431.
- [46] O. Kathe, V. Turkar, A. Jagtap, G. Gidaye, in *2015 IEEE Bombay Section Symp. (IBSS)*, IEEE, Piscataway, NJ **2015**, pp. 1–5.
- [47] T. Zhang, C. Suen, *Commun. ACM* **1984**, *27*, 236.
- [48] D. Douglas, T. Peucker, *Cartogr.: Int. J. Geogr. Inf. Geovisualization* **1973**, *10*, 112.
- [49] A. Saalfeld, *Cartogr. Geogr. Inf. Sci.* **1999**, *26*, 7.
- [50] B. Falck, D. Falck, B. Collette, *FreeCAD [How-To]*, Packt Publishing Ltd., Birmingham, UK **2012**.
- [51] J. Riegel, W. Mayer, Y. van Havre, *FreeCAD, FreeCADspec2002* **2016**, <https://wiki.opensourceecology.org/images/3/36/FreeCADspec2002.pdf>.
- [52] A. Okabe, B. Boots, K. Sugihara, S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester, UK **2009**.
- [53] S. Eshghi, H. Rajabi, J. Poser, S. Gorb, <https://zenodo.org/doi/10.5281/zenodo.8014587> (accessed: February 2024).
- [54] Z. Liu, *Ph.D. Thesis*, Dartmouth College, **2021**.
- [55] E. Kokko, K. Floate, D. Colwell, B. Lee, *Ann. Entomol. Soc. Am.* **1996**, *89*, 398.
- [56] Z. Zhihao, I. Makoto, W. Xishu, L. Jinsheng, *J. Tribol.* **2021**, *143*, 051103.
- [57] A. Cannet, C. Simon-Chane, M. Akhoundi, A. Histace, O. Romain, M. Souchaud, P. Jacob, P. Delaunay, D. Sereno, P. Bousses, P. Grebaut, A. Geiger, C. Beer, D. Kaba, D. Sereno, *Sci. Rep.* **2022**, *12*, 20086.
- [58] A. Cannet, C. Simon-Chane, A. Histace, M. Akhoundi, O. Romain, M. Souchaud, P. Jacob, D. Sereno, L. Gouagna, P. Bousses, F. Mathieu-Daude, D. Sereno, *Sci. Rep.* **2023**, *13*, 17628.
- [59] A. Cannet, C. Simon-Chane, A. Histace, M. Akhoundi, O. Romain, M. Souchaud, P. Jacob, D. Sereno, P. Bousses, D. Sereno, *Sci. Data* **2024**, *11*, 4.
- [60] D. Geldenhuys, S. Josias, W. Brink, M. Makhubele, C. Hui, P. Landi, J. Bingham, J. Hargrove, M. Hazelbag, *PLoS Comput. Biol.* **2023**, *19*, 1011194.
- [61] Y. Kong, S. Wang, C. Ma, B. Li, Y. Yao, in *Computer and Computing Technologies in Agriculture, Volume II: First IFIP TC 12 Int. Conf. Computer and Computing Technologies in Agriculture (CCTA 2007)*, August 2007, Springer, Wuyishan, China **2008**, pp. 755–760.
- [62] D. Li, B. Yang, C. Yang, Z. Zhang, M. Hu, *Environ. Earth Sci.* **2021**, *80*, 671.
- [63] P. Bhambri, S. Singh, S. Sangwan, J. Devi, S. Jain, *J. Surv. Fish. Sci.* **2023**, *10*, 5216.
- [64] S. Verykokou, C. Ioannidis, *Sensors* **2023**, *23*, 596.
- [65] F. Shih, S. Cheng, *Image Vision Comput.* **2005**, *23*, 877.
- [66] Y. Takashimizu, M. Iiyoshi, *Prog. Earth Planet. Sci.* **2016**, *3*, 2.
- [67] A. Goyal, P. Mogha, R. Luthra, N. Sangwan, *Int. J. IT Eng.* **2014**, *2*, 1.
- [68] C. Suárez-Tovar, R. Guillermo-Ferreira, I. Cooper, R. Cezário, A. Córdoba-Aguilar, *J. Zool.* **2022**, *317*, 9.
- [69] R. Åke Norberg, *J. Comp. Physiol.* **1972**, *81*, 9.
- [70] I. Mikó, C. Trietsch, T. Van De Kamp, L. Masner, J. Ulmer, M. Yoder, M. Zuber, E. Sandall, T. Baumbach, A. Deans, *Insect Syst. Diversity* **2018**, *2*, 4.
- [71] Y. Fang, A. Muscente, S. Heads, B. Wang, S. Xiao, *J. Paleontol.* **2018**, *92*, 1028.
- [72] D. Burckhardt, G. Poinar, *Cretaceous Res.* **2020**, *106*, 104240.
- [73] A. Farag, J. Suri, G. Grevera, *Deformable Models: Biomedical and Clinical Applications*, Springer, New York, NY **2007**, pp. 33–60.