

AntVis: A web-based visual analytics tool for exploring ant movement data

Tianxiao Hu^a, Hao Zheng^{b,*}, Chen Liang^c, Sirou Zhu^c, Natalie Imirzian^d, Yizhe Zhang^b, Chaoli Wang^b, David P. Hughes^d, Danny Z. Chen^b

^a Facebook, Inc., Menlo Park, CA 94025, United States

^b University of Notre Dame, Notre Dame, IN 46556, United States

^c Carnegie Mellon University, Pittsburgh, PA 15213, United States

^d Pennsylvania State University, University Park, PA 16802, United States

ARTICLE INFO

Article history:

Received 22 December 2019

Received in revised form 12 February 2020

Accepted 12 February 2020

Available online 25 February 2020

Keywords:

Ant movement

Object detection

Image segmentation

Visual analytics

Knowledge discovery

ABSTRACT

We present AntVis, a web-based visual analytics tool for exploring ant movement data collected from the video recording of ants moving on tree branches. Our goal is to enable domain experts to visually explore massive ant movement data and gain valuable insights via effective visualization, filtering, and comparison. This is achieved through a deep learning framework for automatic detection, segmentation, and labeling of ants, ant movement clustering based on their trace similarity, and the design and development of five coordinated views (the movement, similarity, timeline, statistical, and attribute views) for user interaction and exploration. We demonstrate the effectiveness of AntVis with several case studies developed in close collaboration with domain experts. Finally, we report the expert evaluation conducted by an entomologist and point out future directions of this study.

© 2020 Zhejiang University and Zhejiang University Press. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Many living organisms take part in flow-like collective movements. Accurate movement or trajectory data collection and analysis are essential for understanding these behavior patterns such as migration (Wang et al., 2016) and foraging (Imirzian et al., 2019). Unlike most species in which the flow is unidirectional, ant movements are predominately bidirectional (Fourcassié et al., 2010). The emerging research of ant movements has provided interesting insights into ant traffic rules, revealing that an ant colony as a whole can be considered as a typical self-organized adaptive system. Such studies can shed light on the basic principles of behavioral ecology and evolution of ants. For instance, Imirzian et al. (2019) recently discovered that most foragers walk in nearly straight lines in the same areas as other ants, but a subset of foragers are more explorative.

An interactive tool is highly desired to better help domain experts analyze a large amount of ant surveillance data. Several challenges remain for designing and developing such a tool. The

first challenge is that we need to understand well domain experts' requirements and expectations so that we can design effective functions. We present AntVis, a tool that takes the segmented and tracked ant movement data as input and presents an interactive visual interface that consists of coordinated multiple views for users to visually explore the ant movement data. Working with the domain experts, we perform task analysis and determine design requirements. We collected the data from seven sequences of separate video recordings. The video clips show carpenter ants (a particular species named *Camponotus rufipes*) moving on a tree's branch so their movements can be abstracted as one-dimensional traces or trajectories on a two-dimensional plane. The videos were recorded at the same night and nearly all of the ants were from the same nest. In the last three video clips, the background is different from that recorded before due to a camera rotation. To keep the movement data consistent, we performed image rotation to correct the shooting angle.

The second challenge is that accurate ant detection is a crucial prerequisite for movement analysis and behavior study. However, manual annotation of surveillance videos is tedious and time-consuming due to the sheer amount of data. Automatic ant detection and segmentation from surveillance videos are, therefore, essential. Recently, convolutional neural networks (ConvNets) significantly improve the performance of object segmentation and detection for images (Ronneberger et al., 2015; Shin et al., 2016). We design an automatic segmentation-and-detection framework to attain object (i.e., ant) segments, from which we

* Corresponding author.

E-mail addresses: tianxiaohu@fb.com (T. Hu), hzheng3@nd.edu (H. Zheng), liangchendj@gmail.com (C. Liang), srzhu97@gmail.com (S. Zhu), nsi2@psu.edu (N. Imirzian), yzhang29@nd.edu (Y. Zhang), chaoli.wang@nd.edu (C. Wang), dhughes@psu.edu (D.P. Hughes), dchen@nd.edu (D.Z. Chen).

Peer review under responsibility of Zhejiang University and Zhejiang University Press.

obtain the structured movement data from the video clips. For each frame of the video clips, the structured data are composed of three parts: an ant's ID, the frame's ID, and the (x, y) pixel position of the ant. The data extracted for each video clip are saved into a single file. We also calculated statistical features and saved them into separate JSON files to make the data easier to read and manipulate for our visualization tool.

The third challenge is that we need to carefully design different views of the tool to help domain experts explore the ant trajectory data from multiple perspectives for comprehensive understanding. The AntVis interface includes the core movement view, the related similarity view, timeline view, statistical view, and attribute view, along with the control panel for interaction. These views are connected together via brushing and linking. Users can interactively filter the data in a variety of ways to gain valuable insights. We record the experts' workflow and observations and present several case studies to demonstrate the effectiveness of AntVis.

2. Related work

2.1. Object detection and segmentation in videos

Object detection is a task that requires distinguishing location, class, and the number of objects in images. It has attracted great attention in image processing and computer vision. As a complex problem hardly be well solved by traditional algorithms, new approaches based on deep learning have been applied. For example, the series of region-based convolutional neural networks (R-CNN) (Girshick, 2015; Ren et al., 2015; He et al., 2017) first generated region proposals and then conducted regression and classification within each extracted bounding box simultaneously. The most recent advances have achieved better detection accuracy and higher frame rate (Lin et al., 2017). Another line of works directly predicted bounding boxes in a single feed-forward pass without reusing any components of the neural network or generating proposals of any kind, thus speeding up the detector. The YOLO family (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018) reframed object detection as a single regression problem and output the bounding box coordinates and class probability for each pixel. In this way, images can be processed in real time at 67 FPS with good accuracy. The SSD family (Liu et al., 2016; Fu et al., 2017) used reference boxes of various sizes and aspect ratios to predict object instances by making the whole network as a regressor as well as a classifier. Recently, RefineDet (Zhang et al., 2018) tried to combine the advantages of double-stage methods and single-stage methods by incorporating two new modules in the single-stage classic architecture. CornerNet (Law and Deng, 2018) offered a new approach for object detection by predicting bounding boxes as paired top-left and bottom-right keypoints. Zhou et al. (2019) proposed a new proposal-free method by grouping five detected keypoints (top-most, left-most, bottom-most, right-most extreme points and one center point) into a bounding box if they are geometrically aligned.

However, labeling images for detection is far more expensive. At most of the time, real-world videos are weakly or barely labeled. Besides, manually labeled data often include limited categories of objects. Models trained on such data sets have limited generalization power on unseen categories. Hartmann et al. (2012) formulated image labeling as learning the weakly supervised classifier for recognizing a set of independent spatial-temporal segments. They utilized the graph cuts to refine the obtained objects to generate the final object masks. Zhang et al. (2015) proposed a segmentation-by-detection framework in which object and region detectors pre-trained on still images were used to generate the detection and segmentation proposals. Then object trackers were refined by inferring shape

likelihoods to suppress background noise while preserving the spatial-temporal consistency of foreground objects. More recently, Zhang et al. (2017) integrated the self-paced learning regime and the deep neural network into a unified and compatible framework, i.e., a novel self-paced fine-tuning network (SPFTN), to learn to explore the context information within the video frames and capture the adequate object semantics without using the negative videos. Pathak et al. (2017) used unsupervised motion-based segmentation on videos to obtain segments, which were used as "pseudo ground truth" to train a ConvNet to segment objects from a single frame.

Generally, the existing works usually first train segmentation-level classifiers or inference models under the weak supervision to identify the segments related to the given object categories in the video. Then post-processing methods are applied to refine the object segmentation masks. Although this pipeline has achieved good performance in various cases, there might still be some limitations that could be addressed for further improvement in our specific case. First, it usually needs costly pre-processing methods to generate auxiliary segment proposals or pseudo masks (e.g., optical flow images), or even a large amount of manual annotations. Second, in open-source data sets, there are no weakly labels for ant surveillance videos. In our scenario, there is only one class of object in the video, which is easier to obtain pseudo masks for the supervised training process. Most errors can be detected and corrected, and limited manual correction is needed.

2.2. Visualization of movement data

In recent years, movement or trajectory data visualization has been an active research topic in information visualization and visual analytics. Many works on movement data visualization focus on traffic data, which are generated and collected on moving vehicles and objects. For example, Wang et al. (2013) utilized GPS trajectories to visualize traffic jams in a city and explored the propagation of traffic jams. Wang et al. (2014) used sparse urban traffic data recorded by transportation cells to help users find macro traffic patterns at the city scale. Huang et al. (2016) converted taxi trajectories into a graph and divided the graph into chunks to visualize urban traffic data. Al-Dohuki et al. (2017) integrated map trajectory data and taxi documents to help non-professional users explore and query urban traffic data more easily using semantic information.

The increasing availability of spatiotemporal data from various sources provides new opportunities to discover moving patterns for traffic control and city planning. For instance, Tominski et al. (2012) studied a novel approach that integrates space, time, and attributes by stacked color-coded trajectory bands, enabling users to view spatiotemporal attributes clearly. Doraiswamy et al. (2014) used spatiotemporal urban data to enable users to explore and query event patterns in cities. von Landesberger et al. (2016) developed a graph-based method that combines spatial and temporal simplifications to show the movement patterns of the urban population. In addition, social media data also provide various information and contexts about urban moving patterns. Chen et al. (2016a) developed an interface to help users explore and detect moving patterns from sparsely geo-tagged social media data. Cao et al. (2018) used a heat map to visualize the spatiotemporal data from a streaming source and proposed a method to detect the abnormal moving patterns.

Given massive movement or trajectory data, data aggregation and clustering must be performed before visualization in order to achieve cost-effective viewing, navigation, and understanding. Andrienko and Andrienko (2008) introduced various possible ways to aggregate massive movement data, using abstraction and generalization to support information visualization. Andrienko

et al. (2009) presented a method for clustering very large data sets which cannot be loaded into the main computer memory. The proposed workflow solves the problem by taking a subset – clustering – building a classifier – classifying. The workflow can be applied to other types of structurally complex spatial and non-spatial objects. To solve the problem of heavy intersection and overlapping in the visualization of the vast amount of trajectory data, Andrienko and Andrienko (2011) proposed a method for data aggregation that hides details of data while maintaining their essential characteristics.

Apart from urban traffic data, the more closely related works are those that focus on trajectory data in human society. Stein et al. (2018) utilized data extracted from soccer video to improve the accuracy and effectiveness of soccer analysis. Andrienko et al. (2018) proposed a method for clustering massive airline trajectory data. Their distance measure between trajectories ignores irrelevant elements. For ant movement data, Imirzian et al. (2019) recently proposed to use the exploration index (EI) to measure how explorative an individual trajectory is by calculating how many “different” places are covered by the specific trajectory compared to other trajectories. They also presented a metric for evaluating the straightness of trajectories, which is used to figure out whether some ants are engaged in forage. In this work, we study the rarely-explored ant movement data from data visualization perspective. We first extract ant trajectory data from recorded video clips using deep learning, then perform hierarchical clustering and dimensionality reduction, and finally, present a visual interface called AntVis for users to explore ant movement patterns and identify abnormalities.

3. Task analysis and design requirements

3.1. Task analysis

Domain experts collected the data at a research station in southeast Brazil (Imirzian et al., 2019). We held regular meetings with the experts to discuss the processing and visualization of the surveillance data collected. The experts wanted to know the ant movement patterns given the persistent occurrence of disease threats in the wild. Detecting group movement patterns and identifying outliers of individual ants are of interest. Accordingly, we recognized the following tasks.

T1. Overview of the movement data. This task aims to answer the questions such as “What are the general patterns of ant movement over space and time?”, “Are there any areas of the branch that are passed by ants more frequently than other areas?”, and “How does the ant count vary over time?” In our scenario, a typical video clip is around 20 min, with nearly 30,000 frames recorded. The majorities of ant movements are across the branch (moving from one end to the other end) but some ants just “wander around” (starting and finishing at the same end, or zigzagging or circling). Our visualization should allow users to obtain an overview of the movement data in the first place before they examine the details to gain more insights.

T2. Investigation of individual movements. This task aims to answer the questions such as “Could we select individual movements and examine or compare them in detail?” and “What are the similarities and differences among individual ant movements?” There are hundreds of movements in a video clip and each of them corresponds to an ant’s trace or trajectory. When moving along the branch, different ants may have distinct goals or destinations, which leads to different moving patterns. It is important for users to select individual movements of interest for further examination or comparison. This also helps to verify whether or not a group of ants has the similar moving pattern, which may suggest that those ants are moving toward the same destination (e.g., from nest to food or vice versa). Given a large

number of movements, examining the similarities and differences among them would provide valuable insights into their patterns, allowing better summarization and differentiation of ant behaviors.

T3. Exploration of movement attributes and statistics. This task aims to answer the questions such as “Can we focus on specific movements and examine their related attribute or statistical information?” and “Is one attribute correlated with another attribute?” It is helpful for users to examine various movement attributes and high-level statistical information such as the average speed and the total length or duration of a movement. Selecting specific movements and digging into their attributes would allow users to gain a more detailed understanding of the data and further explore a subset of the movement data that share similar attribute value ranges or statistics. Furthermore, users are often eager to investigate whether or not certain relations exist among different attributes. For example, a longer movement may be associated with a higher or lower moving speed, and vice versa. Enabling the detection of correlation among multiple attributes would allow users to discover attribute correlations.

T4. Exploration of cluster-level movement patterns. This task aims to answer the questions such as “Can we automatically group similar ant movements to identify movement patterns?”, “What do different moving patterns look like?”, and “How different are the statistics of multiple moving patterns?” If two ants are spatially close to each other along their respective traces, we say that their corresponding movements are “similar”. Users are interested in finding similar movements given a movement of interest. Besides, since movements alike reflect the same moving pattern, grouping similar movements into clusters would allow users to explore cluster-level moving patterns. After getting clusters of movement, users would like to compare different moving patterns. For instance, do the ants move across the branch or turn around halfway? Comparing attributes among multiple clusters would allow users to gain further understanding about the distributions of high-level attributes. For example, users would be able to investigate if the movements on the upper part of the branch are usually associated with a higher speed than those on the lower part. All these insights would enable users to better discover the similarities and differences among multiple moving patterns.

T5. Identification of abnormal movements. This task aims to answer the questions such as “Are there any special or abnormal ant movements?” and “How different are these abnormal movements compared with the regular or common movements?” It is important to study the abnormal movements, e.g., those with a remarkably high speed or long path. Detecting and visualizing abnormal ant movements would draw user attention to these particular movements or outliers. Detailed exploration of abnormal movements would also help users get a more comprehensive understanding of the entire ant movement data.

3.2. Design requirements

In order to allow users to perform T1 to T5, our visualization tool should meet the following design requirements.

R1. Display the context. This requirement corresponds to T1 and T5. In order to depict the original ant movements, the visualization system should replicate their traces. For efficiency, the context can be shown in a pseudo video format, with the background image (showing the tree branch) extracted from video and ant movements displayed as traces in a static or animated way. Such visualizations would help users gain a firsthand view of the movement data (T1) and spot possible abnormal movements (T5).

R2. Examine movement distributions. This requirement corresponds to T1. Visualizing spatiotemporal movement distributions would help users gain a general impression and identify

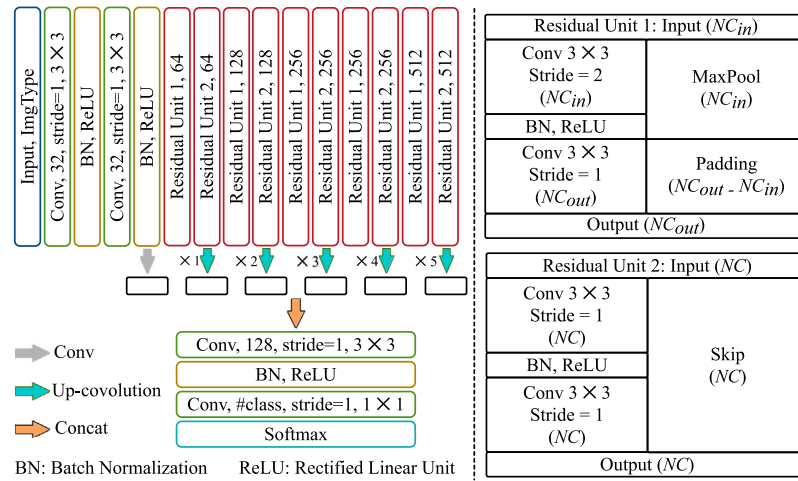


Fig. 1. The FCN structure for automatic ant segmentation from video. The kernel size and number of channels for convolution (Conv) operation are labeled. The up-convolutional layers upsample the size of feature maps by the backwards strided convolution (4×4 w/ stride=2).

the regular pattern of ant movements. Providing a heat map of movement data can detect whether there are popular traces on the branch and how many ants have particular moving patterns. In addition, showing the temporal distribution of ants over time allows users to pinpoint the moments when ants move out of the nest for foraging and carry food back to the nest.

R3. Filter the movements. This requirement corresponds to **T2** and **T5**. Providing various filters can help users observe the movement data from multiple perspectives and select ants they find interesting for a closer exploration. After narrowing down to selected movements, users would be able to compare multiple movements (**T2**) in terms of their similarities and differences. In addition, these filters would also enable users to detect abnormal movements more easily (**T5**) and investigate or analyze the possible causes.

R4. Compare statistical attributes. This requirement corresponds to **T3** and **T4**. After calculating the statistical attributes of ant movements and grouping them into clusters, comparing these statistical attributes over different clusters could lead to interesting findings. Users might want to figure out, for example, whether or not similar movements have similar statistical attributes (**T3**), or what are the differences between clusters from the perspective of statistical attributes (**T4**).

R5. Discover attribute correlations. This requirement corresponds to **T3**. Various movement attributes are not independent, instead, they may be highly interrelated to each other. Our system should provide users an overview of the distribution of different attributes and show users their trends and correlations (e.g., the average speed of movement and the length of the trace). Meanwhile, detection of whether certain attributes have an ascending or descending trend over time could be essential for users to discover findings or draw conclusions.

R6. Visualize clusters and (dis)similarities. This requirement corresponds to **T4** and **T5**. Clustering similar movements would help users observe multiple ant traces sharing the same moving pattern, and study the internal regularities. Once similar movements can be displayed, the next logical step is to observe the dissimilarities among movements with different moving patterns. Providing visualization of moving patterns would help users figure out why these movements are clustered into different groups and what are the key properties of moving patterns (**T4**). Furthermore, through clustering, movements sharing little in common with others would be isolated out, which can help to shed light on abnormal movements (**T5**).

4. Data analysis

4.1. Automatic segmentation via deep learning

Although supervised segmentation based on deep neural networks (DNNs) has been extensively studied in the domain of computer vision in both natural scene images (Long et al., 2015) and biomedical images (Ronneberger et al., 2015; Chen et al., 2016b; Liang et al., 2019; Zheng et al., 2019), few studies have explored in the field of biological image computing, where data are usually of large volume and lack of adequate annotation. We apply a modified 2D fully convolutional network (FCN) (Long et al., 2015) to obtain segments in every frame, which follows the structure of the deep contextual network (DCN) (Chen et al., 2016b). Based on recent advances of DNN structures such as residual learning (He et al., 2016) and batch normalization (Ioffe and Szegedy, 2015), we use the ResNet block and batch normalization layer to alleviate the problem of gradient vanishment and significantly reduce the number of parameters (for better generality) while maintaining reasonable results.

Fig. 1 shows the detailed structure of our FCN model. Compared with the original DCN, our FCN model is much deeper and constructed mainly based on two types of residual modules. Therefore, our model has a much larger field of view to accommodate the size of input images. Specifically, it includes the main encoding path that extracts multi-level information of the input image (i.e., low-level local/contextual information from shallower layers and high-level global/semantic information from deeper layers). The global information helps to locate the areas-of-interest and the local information helps to improve segmentation accuracy. The multi-level contextual information is fused with a concatenation operation followed by two convolution operations. Finally, the probability maps are generated by inputting the fused map into a softmax classification layer.

Fig. 2 outlines the main idea and steps of our deep learning framework. At the first step, given a group of unannotated videos, our approach prepares the training data by decomposing these videos into frames and generating pseudo-labels for these frames. Once the surveillance camera is set, the background is determined and ants (and few flying bugs) are the only moving objects in videos. Therefore, by averaging consecutive frames, we can remove moving objects and noise. In practice, we calculate the average images in 50 consecutive frames and obtain pseudo-labels by subtracting the background image from the original frame. However, as we can see in Fig. 2(a), these initial pseudo-labels are

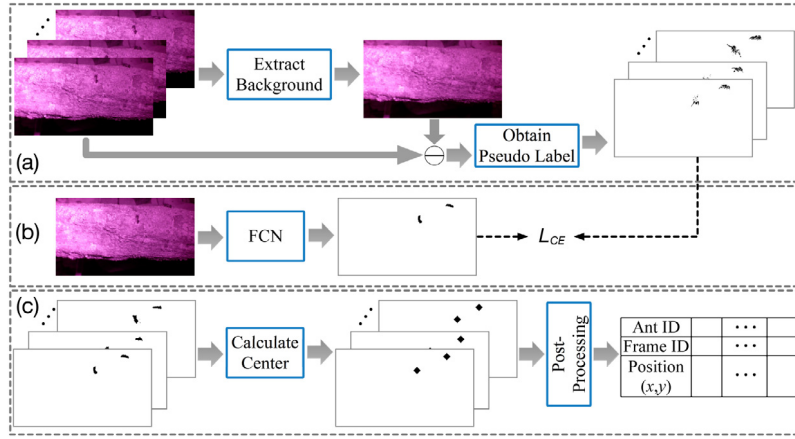


Fig. 2. The ant detection framework consists of (a) background image extraction and pseudo-label generation, (b) semantic object segmentation, and (c) trajectory data generation.

pretty noisy and broken and could be entirely missed in quite a few frames. Therefore, more accurate detection of ants is needed. Motivated by recent works on training deep learning models with pseudo-labels (Pathak et al., 2017), we further propose to train a deep learning model to refine these ant segmentation masks and automatically propagate to other poorly segmented frames. As shown in Fig. 2 (b), at the second step, we utilize the automatically generated pseudo-labels as “ground truth” to train the aforementioned 2D FCN to segment semantic objects (i.e., ants) in each frame. Although the training data is noisy, the learned FCN could generate smoother and more accurate segmentation results. The position of ants is the most essential for building the trajectory of ants. Therefore, at the final step (Fig. 2(c)), we calculate the center point of each detected ant and conduct a post-processing (e.g., merging two detected ants that are extremely small and close) to eliminate mis-segmented components and generate structured ant trajectory data.

4.2. Labeling of segmentation results

Having obtained pixel-level segmentation results of ants, we first eliminate some major mis-segmented components, such as few flying objects and other noisy dots in the background beyond the branch, which can be found easily by the position. Then we apply a series of image processing techniques to obtain refined segmentation results. Specifically, we calculate the connected component (CC) to fill tying holes in segmented “ants”. Next, we compute the center position of each CC to represent an ant. Lastly, we associate ants in consecutive video frames using the Earth Mover’s Distance (EMD) matching model (Chen et al., 2016a) and get structured trajectory data. Each ant is indexed by its ID and we record its appearing and disappearing frame numbers and physical positions in each frame.

Due to the limitation of the automatic segmentation and tracking algorithm, there still exist some errors so we need to manually inspect and correct the tracking results. There are mainly three kinds of errors: (1) When an ant goes behind leaves or the branch and then reappears (usually not far away from the disappearing position), its ID changes in the process. (2) Due to the complex morphology of ant, the linking part between alitrunk and gaster (i.e., body and rear) is hard to segment. If an ant is segmented into two parts in a middle frame, it would be regarded as two separate ants and our algorithm would assign a new ID to this ant. (3) When two ants meet and touch each other with antennae and if they are extremely close, they would be treated as one ant. The other ant’s trajectory becomes interrupted in several frames and a new ID would be assigned when these two

ants move apart. In all these scenarios, there are ant IDs appearing or disappearing in strange positions (most of the time, in the middle of the tree branch). According to our prior knowledge, however, ants usually appear in particular positions, i.e., the left or right side of the frame. We can detect and correct abnormal tracking results by drawing a plot with the x axis representing the x -coordinate of ant appearing and the y axis representing the x -coordinate of ant disappearing. In such a plot, normal ants should fall into the top-left and bottom-right corners and abnormal ants would fall into other areas.

4.3. Ant movement clustering

Similarity Measure. In our ant movement data, each trace P consists of a list of discrete points, which are represented by the (x, y) pixel positions of the ant. Given two traces P_i and P_j , we use the Euclidean distance to measure their similarity. Point pairs are formed by mapping each point on a trace to the *closest* point on the other trace. Specifically, we apply the *mean of closest point distances* (MCP), which is defined as follows

$$d_M(P_i, P_j) = \frac{d_m(P_i, P_j) + d_m(P_j, P_i)}{2}$$

where

$$d_m(P_i, P_j) = \frac{\sum_{p_k \in P_i} \min_{p_l \in P_j} \|p_k - p_l\|}{N}$$

and N is the number of points in P_i , $d_m(P_j, P_i)$ is defined similarly. The smaller $d_M(P_i, P_j)$ is, the more similar P_i and P_j are.

Hierarchical Clustering. We use a bottom-up agglomerative hierarchical clustering method to cluster ant traces. This method performs well on the data containing distances between pairs of points. Besides, it allows us to conveniently identify the appropriate number of clusters. The clustering begins with each trace in a separate cluster. It then merges the two most similar clusters into one in an iterative manner until a stopping criterion is satisfied. We use the *average linkage* hierarchical clustering, that is, we use the average MCP distance of all pairs of traces (one from the first cluster and one from the second cluster) to measure the similarity between two clusters. The smaller the average MCP distance is, the more similar the two clusters are.

For each video clip, the left side is where the food is and the right side is where the nest is. Besides considering all ant movements, we also group them by their moving directions: from right to left (begin foraging), from left to right (return to the nest), from right to right, and from left to left. For each of these five groups (all, left \rightarrow right, right \rightarrow left, right \rightarrow right, and left

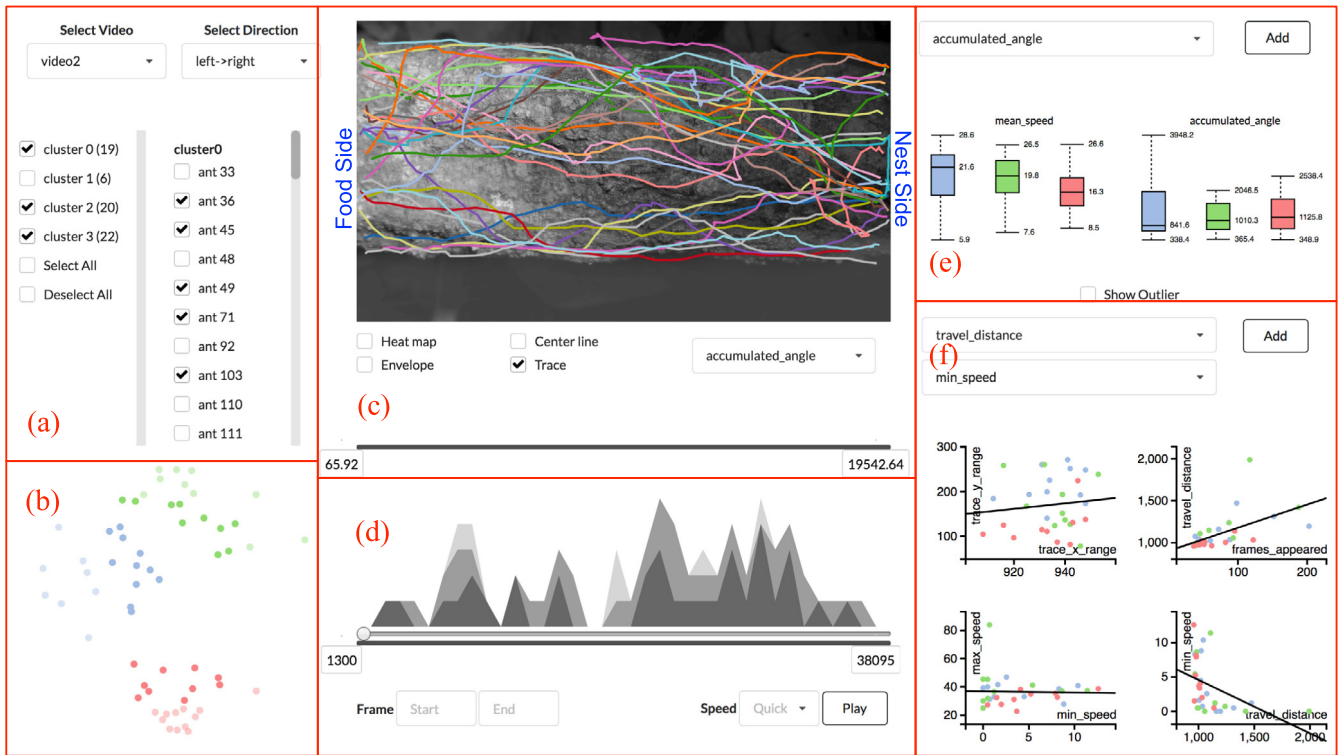


Fig. 3. The AntVis interface. (a) to (f) are the control panel, similarity view, movement view, timeline view, statistical view, and attribute view, respectively.

→ left), we then perform ant movement clustering separately. The desired number of clusters is manually selected based on our empirical observation. The clustering results are used in our visualization system for user exploration.

Dimensionality Reduction. As a popular dimensionality reduction technique that projects high-dimensional data into a lower-dimension (typically 2D) for visualization and exploration, *multidimensional scaling* (MDS) attempts to model data similarity as distances in a geometric space. There exist two types of MDS algorithms: *metric* and *non metric*. In our implementation we choose the metric MDS. Since we have defined the MCP distance for pairs of ant traces, the distance matrix of ant traces is exactly the similarity matrix required by the MDS algorithm. After the MDS is performed, each trace is assigned coordinates in a 2D plane. In this 2D projected plane, each point represents the trace of an ant. Points closer to each other in the plane indicate that their corresponding traces are more similar.

5. AntVis tool

We develop our visualization system called AntVis that enables users to visually explore ant movement data. As shown in Fig. 3, our AntVis system has six components: the control panel, movement view, timeline view, similarity view, statistical view, and attribute view. The current system includes multiple video clips. Ants in each video clip are divided into five different groups based on the directions of movement (all, left → left, left → right, right → left, and right → right). Users can select a video clip and a movement direction on the control panel to quickly narrow down to the selected group of movements. After that, the clusters within the selected group and ants in each cluster are displayed in the two separate lists for further interaction, as shown in Fig. 3(a).

The movement view (Fig. 3(c)) displays the actual traces of the ant movements, enabling users to examine the data in the original spatial domain. The tree branch is extracted from video

and displayed in this view as the context. The timeline view (Fig. 3(d)) provides users the temporal distribution information of ant movements. Users can playback the frames as a pseudo video. The similarity view (Fig. 3(b)) visualizes the ant clusters after the MDS. With each trace represented by a point in the 2D projected plane, this view shows the similarity between clusters by the distance between the points. Finally, the statistical view (Fig. 3(e)) and attribute view (Fig. 3(f)) offer the comparison between quantitative attributes both for clusters and individual traces. All these views are connected via brushing and linking. In the following, we describe these five views in detail.

5.1. Movement view

As the main view of our AntVis system, the movement view mainly displays the original ant movement data, providing users a direct view of ant paths and their distribution on the tree branch. This view corresponds to the design requirements **R1**, **R2**, **R3**, and **R6**. Through this view, users can not only examine the traces of ant but also select and filter ant traces for detailed exploration. The video clip and movement direction can be selected on the top left corner of the interface. The clusters and individual ants displayed in the movement view are linked with the checkboxes shown on the control panel.

Pseudo Video. When users click on the “Play” button at the bottom right corner of the timeline view, the currently selected ant traces will be displayed as a pseudo video. We update the pixel positions of ants frame by frame in the movement view. Traces are colored randomly. As shown in Fig. 4(a), for a trace, when a frame is updated (played), a new step (represented by a triangle icon) appears in the movement view while previous steps gradually fade out. Steps are connected by line segments to form the movement path. The triangle boundary of the current step is highlighted in yellow. The speed information is encoded by the size of the triangle: a larger triangle corresponds to a higher

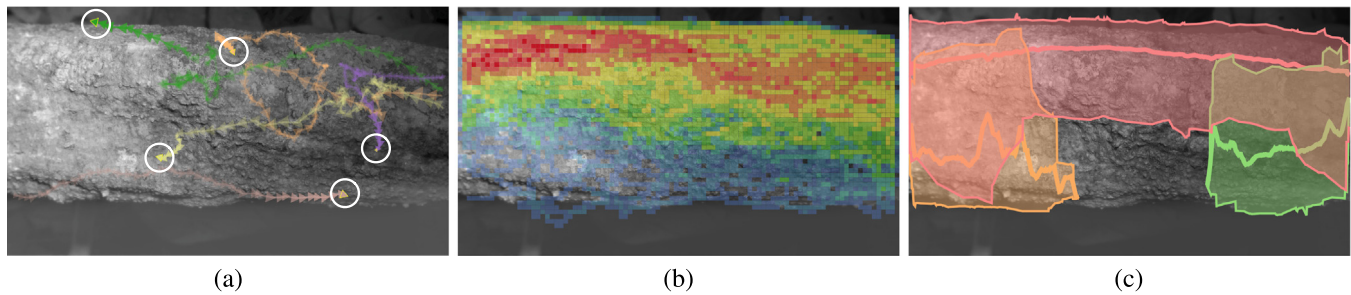


Fig. 4. (a) The visualization of the traces of five ants shows their movements along the tree branch. Their current positions are marked with white circles. (b) The heat map of all 278 ant traces corresponding to a cluster shows a certain “band of paths” favored by these ants, as indicated by the reddish squares. (c) The center lines and envelopes of three selected clusters show their respective summary information of ant traces. The red cluster having ants crossing the entire branch while the orange and green clusters do not have ants crossing the entire branch.

velocity. In addition, the orientation of the triangle indicates the velocity direction. The playback speed can be adjusted by the drop-down list located to the left of the “Play” button. We provide three playback speeds: quick, medium, and slow. The min–max frame bar above will show the progress of the pseudo video being played. Users can also drag the progress indicator to fast forward or backward.

Heat Map. The original size of our video clips is 960×540 pixels. We divide the movement view into squares of 10×10 pixels and yield 5184 squares. For each of the squares, we count the ants that have passed through. If the “Heat map” checkbox is on, the heat map of ant movements will be shown right on top of the background image. To color the heat map, we use a rainbow color map where blue (red) corresponds to less (more) ants. In this way, the popular areas that are passed by ants will be shown in saturated red and the opposites will be shown in light blue. Fig. 4(b) shows an example of the heat map for all ant traces of a cluster, which clearly shows that these ants favor a certain “band of paths” along the branch.

Center Line and Envelope. For each cluster selected on the control panel, we compute its center line and envelope as a summary of the ant traces in the cluster. We generate the center line along the x axis. For each $x_p \in [0, 959]$ on the background, we first select ant traces which have points (x_p, y) , where $y \in [0, 539]$, on the trace and assign the average of all y values to the y coordinate for x_p on the center line. Therefore for clusters that having ants crossing the background image (all, left \rightarrow right, right \rightarrow left), the center line will also cross the background image. Otherwise, the center line will end somewhere in the middle (i.e., left \rightarrow left, right \rightarrow right). In our implementation, different clusters are assigned to different colors. If the “Center Line” checkbox is on, the bold curves of the corresponding colors will be displayed in the movement view, representing the center lines for the selected clusters.

For each envelope, in a similar way, we generate its upper bound and lower bound along the y axis for each $x_p \in [0, 959]$. Then if the “Envelope” checkbox is on, we show all envelope boundaries and shade the corresponding areas for the currently selected clusters. Both the upper and lower bounds will be drawn in the movement view and the envelope will be drawn in a semitransparent manner using the corresponding color of the cluster. Fig. 4(c) shows an example of the center lines and their corresponding envelopes of three selected clusters.

Traces. If the “Trace” checkbox is on, we draw all selected ants’ traces in the movement view. Ant positions extracted from video frames are connected by line segments to form the ant’s trace. We randomly assigned different colors to different traces for clear observation. Fig. 3(c) shows such an example.

Users can select or deselect the checkboxes of ants on the control panel to filter the traces displayed. They can also brush a

rectangular area in the movement view to narrow down to those traces that pass through the selected area. The filtering results will be linked to the checkboxes of ants on the control panel and traces that are not selected will be hidden in the movement view. We note that each brushing in the movement view is a new selection (i.e., the filtering does not accumulate). Besides, users can click on the background image to undo the filtering.

Attribute Filter. We provide a filter for users to narrow down to traces within a specific range of a given attribute. The attributes are arranged in alphabetical order. Users can select any of these attributes in the drop-down list and drag the min–max attribute bar to set the range. Only traces whose attribute values are in the selected range will be displayed in the movement view and the filtering results will be linked to the control panel. We provided the following attributes for users:

- **accumulated angle:** Along an ant’s trace, an intermediate point p_i has two neighbors: p_{i-1} and p_{i+1} . A winding angle is formed between segments $p_{i-1}p_i$ and p_ip_{i+1} , indicating how much the ant has changed its direction. The “accumulated angle” is the sum of all the winding angles along the trace. A trace with a large accumulated angle either has a long or winding path on the tree branch.
- **max angle:** The maximum winding angle along an ant’s trace. This attribute would help users identify ants that make any abruptly sharp turn along its trace.
- **mean angle:** The average winding angle of an ant’s trace. A trace with a small “mean angle” indicates that the ant moves almost straightly.
- **max/mean/min speed:** We calculate the speed at each point p along an ant’s trace using the length of two neighboring points and the time between the two corresponding video frames. These three attributes show the maximum/average/minimum speed of the ant’s trace.
- **frames appeared:** We count the number of frames between an ant’s first and last appearances of each trace, which measures how long an ant has appeared in the video clip. A large value indicates that the ant may either have a long and winding path or frequently stop or stay for a while on the tree branch.
- **travel distance:** We sum up the lengths of all segments along an ant’s trace as its “travel distance”. A long distance means the ant moves back and forth on the branch.
- **trace $x(y)$ range:** We identify the minimum and maximum $x(y)$ coordinate values of an ant’s trace and assign their absolute difference as “trace $x(y)$ range”. This attribute reflects how a trace spreads along the $x(y)$ axis.

In addition, we provide a “No Filter” option at the bottom of the attribute filter drop-down list, allowing users to quickly go back

to the selected ant traces before any attributed-based filtering is applied.

Interaction. When the “Trace” checkbox is on, users can hover over any trace to highlight it for detailed observation. Other traces will fade out and the triangle icons representing the speed will be automatically drawn on the highlighted trace. A tooltip will also appear showing the attributes of the corresponding ant. The “Heat map”, “Center line”, “Envelope”, and “Trace” checkboxes can be applied in any order. For “Center line”, “Envelope”, and “Trace” checkboxes, they will be drawn to the movement view in the order of the corresponding checkboxes being turned on. However, to provide the context and avoid covering other lines and segments, we always draw the heat map at the bottom layer (i.e., right on top of the background image).

5.2. Timeline view

The timeline view mainly provides users an overview of the temporal distribution of ants over time. It allows users to figure out the moments when ants appear most frequently in the video clip, and when they move out of the nest to forage and carry food back to the nest. This view corresponds to the design requirements **R2**, **R3**, and **R6**. The direction, clusters, and individual ants displayed in the timeline view are linked with the drop-down list and checkboxes on the control panel. Through this view, users can examine the temporal distribution of ants as well as select and filter ants by their time of appearance.

Timeline. Each of our video clips consists of nearly 30,000 frames and is divided into time intervals, and each interval includes 1000 frames. Then we calculate the number of ants appearing in each of the time intervals. As shown in Fig. 3(d), we place a stacked histogram showing ant counts over time on top of the frame bar. This allows users to quickly drag the min-max frame bar to examine the time interval of interest in the movement view. Three histograms are stacked in the display. The first with the lightest gray color shows the number of ants in the selected direction (all, left → left, left → right, right → left, and right → right), the second with the medium gray color corresponds to the number of ants in the selected clusters as shown on the control panel, and the third with the strongest gray color is linked with the checkboxes of ants on the control panel. By turning on and off the checkboxes of ants, users can get a highly customized view of temporal distribution for selected ants.

Users can set a range to filter ants that appeared in a specific range of frames by dragging the min-max frame bar or typing in the two input boxes given. Accordingly, only ants whose frames of the first appearance fall into the range are displayed in the movement view. The filtering results are linked to the checkboxes on the control panel.

5.3. Similarity view

As discussed in Section 4.3, we perform MDS on ant movement data to present a projected view of ant traces in 2D. This view corresponds to the design requirements **R3** and **R6**. Each trace will be represented by a point on a 2D projected plane. If two traces are similar, their corresponding points are close to each other.

MDS Similarity. The points displayed on the similarity view is linked to selected clusters on the control panel. For better observation, we assigned different colors to different clusters (the color assignment is consistent with that of the center line or envelope as shown in the movement view).

Similarity Filter. We provide users a filter to select traces close to each other for careful examination. Users can brush a rectangular area in the similarity view to make the selection.

Points in selected area will be marked in more saturated colors. The filtering results will be linked to the checkboxes of ants on the control panel and ant traces displayed in the movement view. Besides, users can undo the filtering by a single click on the similarity view. Like the brushing in the movement view, each brushing in the similarity view is a new selection, i.e., the filtering does not accumulate.

5.4. Statistical view

Corresponding to the design requirement **R4**, this view aims to help users compare statistical attributes of different clusters. Besides the attributes provided in the attribute filter, we add seven more attributes for users to explore:

- *frame start/end id*: The frame ID of ants' first/last appearance.
- *speed range/std*: We identify the minimum and maximum speeds of an ant's trace. We assign their absolute difference as “speed range” and calculate the standard deviation of the ant's speed as “speed std”.
- *max pos/neg acceleration*: We calculate the acceleration for each point along an ant's trace using the speed change between two neighboring points and the time elapsed between the two corresponding video frames. We record its maximum positive acceleration and minimum negative acceleration as these two attributes.
- *acceleration range*: This attribute is the absolute difference between “max pos acceleration” and “max neg acceleration”.

Box Charts. The attributes are arranged in alphabetical order. Users can select the attribute they want to compare in the drop-down list and click on the “Add” button. Then a box chart will be displayed showing the attribute distributions of the clusters selected on the control panel. Each box corresponds to one cluster and is filled with the corresponding cluster's center line or envelope color as shown in the movement view. When the checkbox of a cluster is turned on or off, the corresponding box will be updated in the statistical view. Users can create two box charts to view two attributes at the same time. Besides, by double-clicking a box chart, users can remove it and make space for a new box chart. Users can hover over any box to bring out a tooltip showing the ID of the corresponding cluster. If the “Show Outlier” checkbox is turned on, outliers (if any) are shown as circles. This checkbox is turned off by default as adding outliers in the display will squeeze the boxes and make it more difficult for users to read the boxes.

5.5. Attribute view

This view corresponds to the design requirements **R3**, **R5**, and **R6**. Because attributes are often interrelated, this view is mainly designed for users to quickly discover correlations between two selected attributes. This view can also serve as a filter for users to narrow down to ants with specific attribute value ranges.

Scatterplots. The attributes provided here are the same as those in the statistical view. The difference is that users can select two attributes they want to explore at a time from the two drop-down lists. After clicking on the “Add” button, a scatterplot will be displayed showing the joint distribution of the two attributes, helping users detect whether the selected two attributes exhibit a positive or negative correlation. Besides, users can select “frame start id” as one of the attributes to find whether another attribute has an ascending or descending trend over time. Each point in the scatterplot represents an ant and is filled with the corresponding cluster's color, as shown in the movement view, thus helping users compare among different clusters. Users can create up to

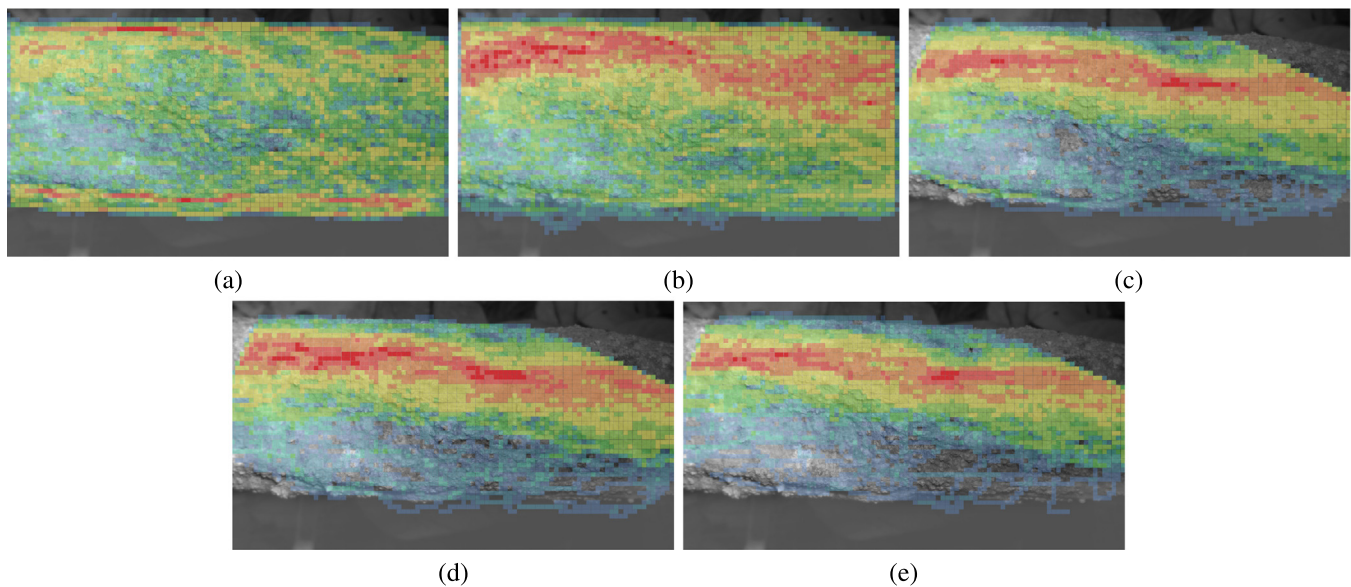


Fig. 5. (a) to (e) The heat maps of all ant traces for five video clips: “video 2”, “video 3”, “video 5”, “video 6”, and “video 7”, respectively. For (c) to (e), the camera positions were shifted and therefore, we calibrate the images via matching feature points extracted from the background tree branch. The calibration shows that the top-right corner of the tree branch was not captured in “video 5”, “video 6”, and “video 7”.

four scatterplots to view different pairwise attribute distributions and correlations simultaneously. Double-clicking on a scatterplot will remove it.

We provide users a filter to select ants with specific attribute value ranges. Users can brush a rectangular area on one scatterplot to make the selection. Points in the selected area will remain in all the scatterplots and all scatterplots will be automatically zoomed in to let users see the distribution of selected ants more clearly. The filtering results will be linked to the checkboxes of ants on the control panel, ant traces displayed in the movement view, and points shown in more saturated colors in the similarity view. Besides, users can undo the filtering by a single click on the brushed scatterplot. However, unlike the brushing in the movement view and similarity view, each brushing in the attribute view will take the previous filtering results as the input, i.e., the filtering accumulates.

6. Results and evaluation

AntVis is a web-based tool implemented with D3.js and jQuery for dynamic and interactive visualization. The tool is released online at: <https://sites.nd.edu/chaoli-wang/demos/>. To avoid any compatibility issues, we recommend users to use the Google Chrome browser to explore our AntVis system. In the following, we present three case studies and highlight the insights gleaned from the ant movement data. The three case studies jointly cover all tasks described in Section 3.1.

6.1. Case studies

Case Study 1: Overview of Ant Movement Data. This case study aims to provide users an overview of the data, especially the spatiotemporal distribution of ant traces. Our data consist of several video clips, so it is essential for users to gain a high-level cross-video comparison before they narrow down to a specific video clip of interest. This case study corresponds to Task T1. After the overview, users also perform some basic selections in the similarity view to explore ant clusters.

Users begin with the exploration of the spatial distribution. We should have seven video clips in all. However, due to the small number of ants appeared in the first clip and the missing

of the fifth clip, our system currently contains five video clips. Take the second video clip (video 2) as an example. Users first select “video 2” and “all” from the drop-down lists on the top of the control panel. By turning on the “Select All” checkbox and the “Heat map” checkbox in the movement view, they can get a rough overview of the spatial distribution of ant movements in the current video clip. Afterward, users can switch between the video clips and indirectly compare the specific areas of the tree branch that ants passed more or less frequently. Fig. 5 clearly shows that as time went by (the video clips were taken in sequence), ants gradually formed a “band of paths” on the branch.

After gaining the overview of the spatial distribution in all video clips, users can compare the temporal distributions among video clips via the timeline view. Take the second video clip (video 2) as an example. Users can select “video 2” and “all” from the drop-down lists on the top of the control panel. The histogram of the lightest gray color shows the appearance frequency of ants in the current video clip. Besides, we know that the left side is where the food is and the right side is where the nest is. So by switching the direction to right→left, we can observe the distribution of ants that move out of the nest to forage. Similarly, the direction left→right corresponds to ants carrying food back to the nest. Users can switch between the video clips and directions to observe more detailed temporal distributions of different video clips. As shown in Fig. 6, we can find that in video 2, both groups of ants appeared more frequently during the 2nd half of the clip. However, in video 3, ants moving out of the nest appeared more frequently during the 1st half of the clip while ants carrying food back appeared more frequently during the 2nd half of the clip.

In this case study, users can perform some basic selections in the similarity view and get a direct understanding of ant traces and their MDS distances. Users can select “video 3” and “all” from the drop-down lists on the top of the control panel. Then turn on the “Traces” checkbox in the movement view for further interaction. If users make a selection that covers all orange points on the similarity view, the filtering results are linked to the movement view so only traces on the top part of the branch are displayed. On the contrary, if users brush all pink points, only traces on the bottom part are displayed. This basic selection directly allows users to gain an overview of ant traces and clusters.

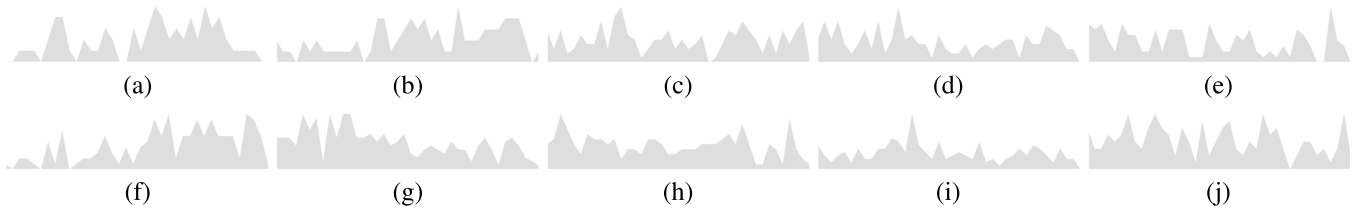


Fig. 6. The timeline views of ant frequency for the five video clips. Left to right are “video 2”, “video 3”, “video 5”, “video 6”, and “video 7”, respectively. Top and bottom rows are left → right and right → left ant groups, respectively.

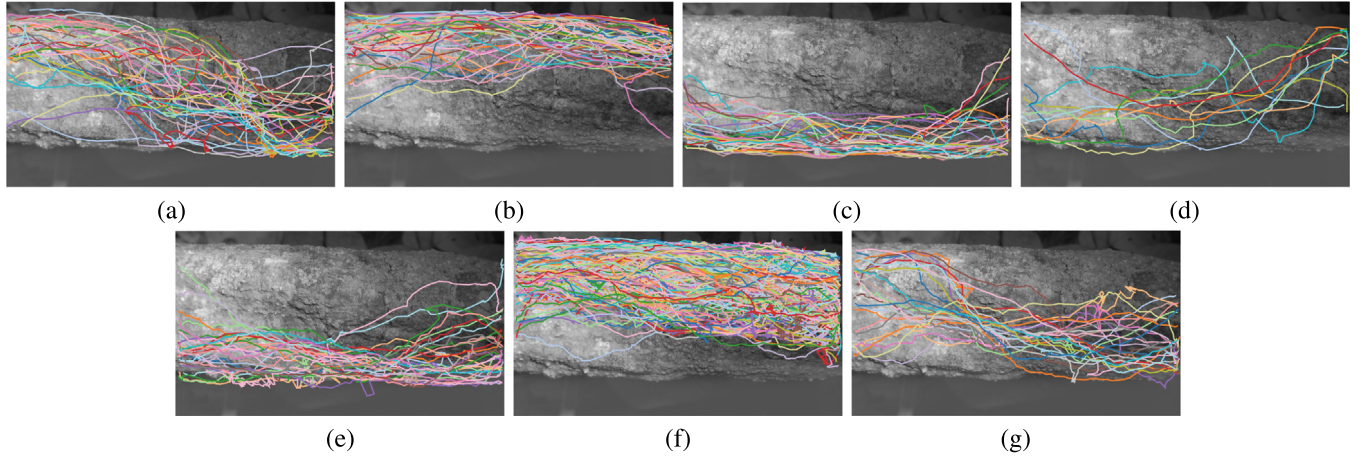


Fig. 7. The traces of clusters (right→left) in video 2 (top row) and video 3 (bottom row). (a) to (d) are clusters 0 to 3 in video 2, respectively. (e) to (g) are clusters 0 to 2 in video 3, respectively. The result of cluster matching is: (a) to (g), (b) to (f), (c) to (g), and (d) to (e).

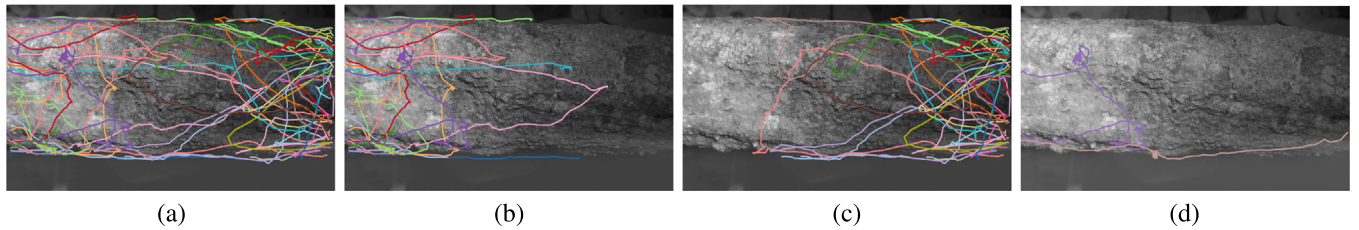


Fig. 8. (a) Using the attribute filter of “travel x range” to show only traces starting and ending in the same direction. (b) and (c) Further spatial filtering of the result shown in (a). After performing a selection covering the left or right side of the movement view, users can observe the group of ants in the left→left or right→right direction only. (d) The traces of the two ants which stayed for a long time on the tree branch.

For high-level cross-video comparison, we performed cluster matching across different video clips via a post-analysis which is not included in the AntVis visual interface. We calculated the average MCP distance between clusters and selected the cluster with minimum average MCP distance for matching. Taking clusters in video 2 and video 3 as an example, we can observe the changes of moving pattern at the cluster level by matching clusters, as shown in Fig. 7. Note that according to our computation, (c) and (g) form a better match than (c) and (e) as the average MCP distance between (c) and (g) is 98.0 and that between (c) and (e) is 100.4.

Case Study 2: View Interaction and Attribute Exploration.

This case study aims to show users the various interactions among multiple AntVis views, and how those interactions can lead to deeper insights gained from the ant movement data. Tasks T2, T3, and T4 are covered here.

Users can begin this study by selecting “video 2” and “all” from the drop-down lists on the top of the control panel. They then turn on the “Traces” checkbox in the movement view for view interaction. We add a scatterplot of “mean speed” vs. “travel distance” to explore their correlation. When all ants are taken into account, the regression line’s slope is almost zero thus no

obvious trend is shown. However, users can add an attribute filter of “travel x range” and set the min and max values to 27 and 750, respectively. Since the “travel x range” attribute reflects how a trace spreads along the x axis and the entire x range is 960, only traces starting and ending in the same direction (e.g., left→left and right→right) remain in the movement view, as shown in Fig. 8(a). Users can identify that attributes “mean speed” and “travel distance” have a positive correlation.

Afterward, users can apply spatial and temporal filters to seek further insights. As shown in Fig. 8(b) and (c), by performing a selection covering the left side of the movement view, users can observe the group of ants in the left→left direction only. The filtering results are linked to the scatterplot and users can similarly apply the filter and observe ants in the right→right direction. The results show that the positive correlation of “mean speed” and “travel distance” is much more obvious among ants in the right→right direction. Furthermore, users can drag the min-max frame bar of the timeline filter (e.g., gradually changing the max value from 20,000 to 10,000) and find that the positive correlation is stronger among ants appeared in the early part of the video, as shown in Fig. 9.

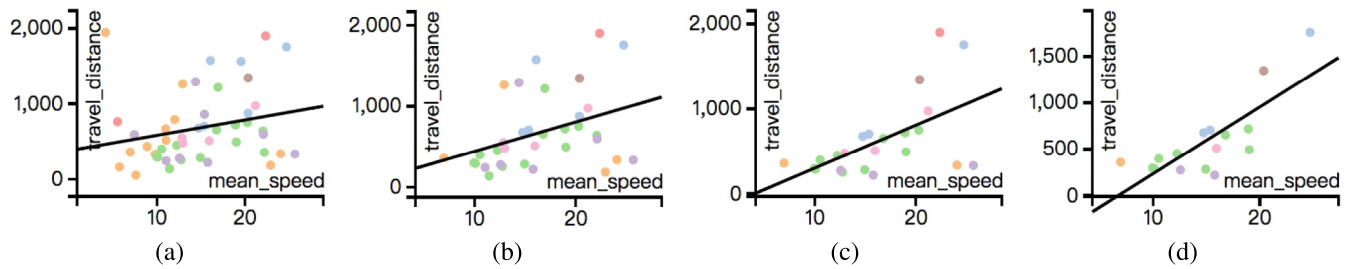


Fig. 9. Temporal filtering results. (a) to (d) are scatterplots and regression lines with the max-value of the timeline filter set to 38095, 26910, 16456, and 11558, respectively. We can observe that the slope of the regression line gets steeper as the max-value decreases.

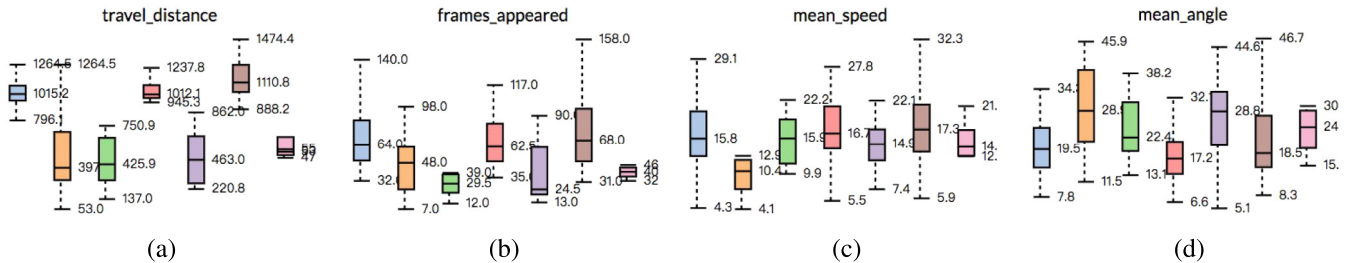


Fig. 10. Using the box charts to show attribute variations among clusters. (a) to (d) show the result of attribute “travel distance”, “frames appeared”, “mean speed”, and “mean angle”, respectively. There are seven boxes in each box chart and they correspond to clusters 0 to 6, respectively. Ants in clusters 1, 2, 4, and 6 appeared and disappeared from the same side (i.e., left→left, right→right) and ants in clusters 0, 3, and 5 appeared and disappeared from different sides (i.e., left→right, right→left).

Case Study 3: Cluster Comparison and Abnormal Movement Detection. In this last case study, we show users how to compare attribute distributions between different clusters and detect abnormal movements. Users first perform cluster-level exploration on the movement data, then narrow down to specific ants and detect abnormal traces. Tasks **T3**, **T4**, and **T5** are covered here.

First, users can begin this study by selecting “video 2” and “all” from the drop-down lists on the top of the control panel. Then by turning on the “Select All” checkbox and the “Center line” checkbox in the movement view, they are able to get a rough overview of the clusters’ directions (e.g., left→left, left→right, right→right, or right→left). Afterward, users can add several box charts to compare the attribute distributions of selected clusters. By adding box charts of “travel distance”, “frames appeared”, “mean speed”, and “mean angle” as shown in Fig. 10, users can observe the difference between groups whose ants appeared and disappeared from the same side and groups whose ants appeared and disappeared from different sides. On average, groups whose ants appeared and disappeared from the same side tend to have shorter “travel distance”, fewer “frames appeared”, smaller “mean speed”, and larger “mean angle”.

Abnormal movements can be detected using the scatterplot and its filter. Users can turn off the “Center line” checkbox, turn on the “Traces” checkbox, and add a scatterplot of “mean speed” and “accumulated angle”. They can identify two abnormal points which are far away from other points but close to the axis of “accumulated angle”. By brushing the two abnormal points, the filtering results are linked to the movement view and users can find out that these two ants stayed for a long time on the tree branch, as shown in Fig. 8(d).

Insights Using Our Tool. The entomologists used the ant movement data to investigate if how the patterns of ant movement impacted the colony’s susceptibility to infection. AntVis provided their first impression on the ant movement data and allowed them to conduct an initial qualitative investigation before a more thorough analysis of the data (Imirzian et al., 2019). Being able to quickly visualize the trajectories of different ants stimulated questions on the variation in the trajectories observed.

Here is a summary of the initial insights that overlap with and support their final discoveries.

- Observing the heat map of the trail indicated that the majority of ants walk directly across the trail and cover similar areas, limiting the exposure of most ants to environmental threats. Meanwhile, some individual trajectories showed evidence of searching behavior and covered unexplored areas.
- Looking at the trajectories over time using the envelope in the movement view indicated that at different time periods, ants show different preferences in foraging patterns. For example, ants would more actively explore the area at the beginning of a foraging period, as this is when the pheromone trail would be the weakest. Additionally, outgoing and incoming ants vary in how directly they walk across the trail.
- The clustering and center line capability of the AntVis tool allowed the authors to investigate different groups of ants, and they found that a group of ants wander across the trail more (case study 3), and these ants are more likely to explore a unique area. This has biological implications since it could increase infection risk while also assisting the colony with food discovery.

6.2. Expert evaluation

An entomologist studying ant behaviors assessed the AntVis tool and in the context of the five tasks **T1–T5** and gave a detailed analysis of each view’s benefits and limitations as follows.

Movement View. A first step in analyzing the movement data is visualizing where the individuals are moving. The movement view provides users an interface to not just see where the individuals are moving (i.e., individual traces) but also to investigate potentially interesting patterns (i.e., the center line and envelope of each group of ants). Most importantly, the heat map is especially informative since a question regarding this data is whether or not ants use this tree space evenly. If the heat map reveals the areas of the trace that are used more frequently, then

users could ask about the average speed of the individuals of that area and whether it creates traffic jams. Furthermore, users can select different attributes and see whether trajectories with similar attributes also have similar trajectories. The entomologist found that the heat map attracted users most at the beginning, and the attributes and filter functions were more frequently used by users to exam a specific group of ants or an individual ant.

Timeline View. This view allows researchers to investigate ant patterns over time. For example, studies on ant foraging behavior suggest the presence of different types of foragers such as patrollers that leave the nest first at the beginning of a foraging period and other ants that retrieve the food after (Greene and Gordon, 2007). Researchers can answer such questions by comparing the movement of ants over time. Furthermore, this view provides them a play function to watch a clip of the pseudo video and allows them to analyze the movement data dynamically. The filter allows users to choose a frame of interest to explore the data further. The main drawback the entomologist found is that it can only show the ant count data over time for the chosen video; users need to save the screenshots and conduct the cross-video comparison.

Similarity View. The clustering results according to trace similarity allows researchers to analyze different groups that might be represented in the ant foraging groups. By filtering different groups, researchers can see when they appear (using the timeline view) or what area of the trace they cover (using the heat map view), thus investigating what behavioral groups the clustered trajectories belong to. The entomologist found that it could be difficult to filter a complete cluster because only a rectangular area can be selected and some outliers could be selected inevitably.

Statistical View. This view allows researchers to analyze the statistical attributes of possible different behavioral groups. Knowing whether the group has a higher or lower speed, and the time that they start allows researchers to relate the behavior to the biology of the ants. Besides, it also makes cross-cluster comparison much easier and intuitively, enabling researchers to look into a specific group of ant of interest. The entomologist found that users may not be equally interested in every attribute and not all attributed are entirely independent of each other. Grouping similar attributes into several groups could help users explore the most important ones first.

Attribute View. While traditional analyses would require researchers to analyze these features one at a time, with this tool, researchers can quickly examine between different attributes, then relate a particular attribute view to the actual movement on the trace. Furthermore, this view is useful for determining the relationship between different groups of ants. If possible, it would be better if multiple regression lines can be computed for each group of ants.

The entomologist concluded that AntVis is a helpful tool that can assist entomologists and data scientists in discovering the overall patterns of foraging traces of carpenter ants (i.e., *C. rufipes*), the detailed statistical attributes of each ant, and the correlation between attributes of interest. It is a critical step to inspire researchers to raise more meaningful questions on representative and peculiar movements, leading to a comprehensive analysis of ant moving behaviors.

To further develop AntVis into a tool that can benefit entomologists or biologists, the entomologist made the following suggestions:

- develop an aggregated view that summarizes the movement data of the same colony over the whole night so experts can more conveniently investigate the variation of trace patterns over time.
- provide a cross-colony comparison view so that experts can cross-validate whether or not different groups of ants share similar trace patterns and measure how similar they are.

- allow users to define customized attribute measurements and conduct the computation using the trajectory data directly, and add to the attribute view as required.
- show the robustness of the tool by manipulating with other types of trajectory data, such as mammal immigration GPS signal.

7. Conclusions and future work

We have presented AntVis, a web-based visual analytics tool for exploring ant movement data. Our work automatically segments ants from video frames via deep learning techniques and labels ants to generate their traces. This allows us to store only ant movement data with a single background image to provide the context for the subsequent analysis and visualization. Hierarchical clustering and dimensionality reduction are then applied to ant traces to support interactive visual exploration. Through interacting with a visual interface, we enable users to gain an overview of the movement data, detailed explore trace attributes, and identify common patterns and detect abnormal movements using five coordinated views, namely, the movement, similarity, timeline, statistical, and attribute views. We demonstrate the effectiveness of AntVis with selected case studies and an expert evaluation.

In the future, we would like to collect and process more video recordings to analyze ant movement data at larger spatial and temporal ranges. Cross-video visual summarization and comparison will be implemented and integrated into the AntVis tool. Ultimately, we would release this tool to domain experts, enabling them to streamline the analysis and visualization of massive ant movement data in their daily workflow.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRedit authorship contribution statement

Tianxiao Hu: Methodology, Software, Visualization, Writing - original draft. **Hao Zheng:** Methodology, Software, Visualization, Writing - original draft. **Chen Liang:** Methodology, Software. **Sirou Zhu:** Methodology, Software. **Natalie Imirzian:** Investigation, Writing - review & editing. **Yizhe Zhang:** Methodology, Software. **Chaoli Wang:** Conceptualization, Methodology, Writing - review & editing. **David P. Hughes:** Supervision. **Danny Z. Chen:** Supervision.

Acknowledgments

This research was supported in part by the US National Science Foundation through grants IIS-1456763, IIS-1455886, CNS-1629914, CCF-1617735, and DUE-1833129, and by the US National Institutes of Health through grant R01 GM116927. T. Hu, S. Zhu, and C. Liang conducted this work as iSURE (International Summer Undergraduate Research Experience) students at the University of Notre Dame during Summer 2017.

References

- Al-Dohuki, S., Kamw, F., Zhao, Y., Ma, C., Wu, Y., Yang, J., Ye, X., Wang, F., Li, X., Chen, W., 2017. SemanticTraj: A new approach to interacting with massive taxi trajectories. *IEEE Trans. Vis. Comput. Graphics* 23 (1), 11–20.
- Andrienko, G.L., Andrienko, N.V., 2008. Spatio-temporal aggregation for visual analysis of movements. In: *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*. pp. 51–58.

- Andrienko, N.V., Andrienko, G.L., 2011. Spatial generalization and aggregation of massive movement data. *IEEE Trans. Vis. Comput. Graphics* 17 (2), 205–219.
- Andrienko, G.L., Andrienko, N.V., Fuchs, G., Garcia, J.M.C., 2018. Clustering trajectories by relevant parts for air traffic analysis. *IEEE Trans. Vis. Comput. Graphics* 24 (1), 34–44.
- Andrienko, G.L., Andrienko, N.V., Rinzivillo, S., Nanni, M., Pedreschi, D., Giannotti, F., 2009. Interactive visual clustering of large collections of trajectories. In: *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*. pp. 3–10.
- Cao, N., Lin, C., Zhu, Q., Lin, Y.-R., Teng, X., Wen, X., 2018. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE Trans. Vis. Comput. Graphics* 24 (1), 23–33.
- Chen, J., Cai, Y., Wei, C., Yang, L., Alber, M.S., Chen, D.Z., 2016. Segmentation and tracking of *Pseudomonas aeruginosa* for cell dynamics analysis in time-lapse images. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*. pp. 968–971.
- Chen, H., Qi, X., Yu, L., Heng, P.-A., 2016. DCAN: Deep contour-aware networks for accurate gland segmentation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2487–2496.
- Chen, S., Yuan, X., Wang, Z., Guo, C., Liang, J., Wang, Z., Zhang, X., Zhang, J., 2016a. Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data. *IEEE Trans. Vis. Comput. Graphics* 22 (1), 270–279.
- Doraiswamy, H., Ferreira, N., Damoulas, T., Freire, J., Silva, C.T., 2014. Using topological analysis to support event-guided exploration in urban data. *IEEE Trans. Vis. Comput. Graphics* 20 (12), 2634–2643.
- Fourcassié, V., Dussutour, A., Deneubourg, J.-L., 2010. Ant traffic rules. *J. Exp. Biol.* 213, 2357–2363.
- Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C., 2017. DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*.
- Girshick, R., 2015. Fast R-CNN. In: *Proceedings of IEEE International Conference on Computer Vision*. pp. 1440–1448.
- Greene, M.J., Gordon, D.M., 2007. How patrollers set foraging direction in harvester ants. *Amer. Nat.* 170 (6), 943–948.
- Hartmann, G., Grundmann, M., Hoffman, J., Tsai, D., Kwatra, V., Madani, O., Vijayanarasimhan, S., Essa, I., Rehg, J., Sukthankar, R., 2012. Weakly supervised learning of object segmentations from web-scale video. In: *Proceedings of European Conference on Computer Vision Workshops and Demonstrations*. pp. 198–208.
- He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. In: *Proceedings of IEEE International Conference on Computer Vision*. pp. 2961–2969.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- Huang, X., Zhao, Y., Yang, J., Ma, C., Zhang, C., Ye, X., 2016. TrajGraph: A graph-based visual analytics approach to studying urban network centrality using taxi trajectory data. *IEEE Trans. Vis. Comput. Graphics* 22 (1), 160–169.
- Imirzian, N., Zhang, Y., Kurze, C., Loreto, R.G., Chen, D.Z., Hughes, D.P., 2019. Automated tracking and analysis of ant trajectories shows variation in forager exploration. *Sci. Rep.* 9 (13246).
- Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of International Conference on Machine Learning*. pp. 448–456.
- von Landesberger, T., Brodtkorb, F., Roskosch, P., Andrienko, G.L., Andrienko, N.V., Kerren, A., 2016. MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Trans. Vis. Comput. Graphics* 22 (1), 11–20.
- Law, H., Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. In: *Proceedings of European Conference on Computer Vision*. pp. 734–750.
- Liang, P., Chen, J., Zheng, H., Yang, L., Zhang, Y., Chen, D.Z., 2019. Cascade decoder: A universal decoding method for biomedical image segmentation. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*. pp. 339–342.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: *Proceedings of IEEE International Conference on Computer Vision*. pp. 2980–2988.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: *Proceedings of European Conference on Computer Vision*. pp. 21–37.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3431–3440.
- Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B., 2017. Learning features by watching objects move. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 6024–6033.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 779–788.
- Redmon, J., Farhadi, A., 2017. YOLO9000: Better, faster, stronger. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 6517–6525.
- Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Proceedings of Annual Conference on Neural Information Processing Systems*. pp. 91–99.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 234–241.
- Shin, H.-C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* 35 (5), 1285–1298.
- Stein, M., Janetzko, H., Lamprecht, A., Breikreutz, T., Zimmermann, P., Goldlücke, B., Schreck, T., Andrienko, G.L., Grossniklaus, M., Keim, D.A., 2018. Bring it to the pitch: Combining video and movement data to enhance team sport analysis. *IEEE Trans. Vis. Comput. Graphics* 24 (1), 13–22.
- Tominski, C., Schumann, H., Andrienko, G.L., Andrienko, N.V., 2012. Stacking-based visualization of trajectory attribute data. *IEEE Trans. Vis. Comput. Graphics* 18 (12), 2565–2574.
- Wang, Z., Lu, M., Yuan, X., Zhang, J., van de Wetering, H., 2013. Visual traffic jam analysis based on trajectory data. *IEEE Trans. Vis. Comput. Graphics* 19 (12), 2159–2168.
- Wang, Y., Luo, Z., Takekawa, J., Prosser, D., Xiong, Y., Newman, S., Xiao, X., Batbayar, N., Spragens, K., Balachandran, S., Yan, B., 2016. A new method for discovering behavior patterns among animal movements. *Int. J. Geogr. Inf. Sci.* 30 (5), 929–947.
- Wang, Z., Ye, T., Lu, M., Yuan, X., Qu, H., Yuan, J., Wu, Q., 2014. Visual exploration of sparse traffic trajectory data. *IEEE Trans. Vis. Comput. Graphics* 20 (12), 1813–1822.
- Zhang, Y., Chen, X., Li, J., Wang, C., Xia, C., 2015. Semantic object segmentation via detection in weakly labeled video. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3641–3649.
- Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z., 2018. Single-shot refinement neural network for object detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.
- Zhang, D., Yang, L., Meng, D., Xu, D., Han, J., 2017. SPFTN: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4429–4437.
- Zheng, H., Yang, L., Han, J., Zhang, Y., Liang, P., Zhao, Z., Wang, C., Chen, D.Z., 2019. HFA-Net: 3D Cardiovascular Image segmentation with asymmetrical pooling and content-aware fusion. In: *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 759–767.
- Zhou, X., Zhuo, J., Krahenbuhl, P., 2019. Bottom-up object detection by grouping extreme and center points. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 850–859.