Similarity Algorithm Based on Minimum Tilt Outer Rectangle Clustering of UAV Videos

Wei Ding[1,2,*], Luyao Wang[1], Wei Zhang[1,2], Myung Jin Lee[3], Ling Wang[4], Kwang Woo Nam[5,*]

[1] *Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China.*

[2] *Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China.*

[3] *Industry-University Cooperstion Foundation, Chungbuk National University, Cheongju, 28644, Korea.*

[4]*School of Computer Science, Northeast Electric Power University, Tai'an, 132011, China.*

[5]*School of Software, Kunsan National University, Kunsan, 54150, Korea.*

*Corresponding Author：dingw@sdas.org , kwnam@kunsan.ac.kr

***Abstract*—In this paper, we propose a new UAV video similarity measurement algorithm to cluster and analyze UAV videos by measuring the similarity between UAV videos and train them in sub-groups, which in turn improves the efficiency of UAV video data mining. Our preliminary work introduces the maximum common view subsequence (LCVS) to compute the similarity between UAV videos, but LCVS needs to frequently compute the common visible region of video frames generating extremely high time cost. In order to solve the problem of high computational cost of LCVS, we propose a similarity algorithm based on minimum tilted outer rectangle clustering (MTORC) for UAV videos. The MTORC algorithm adopts minimum tilted outer rectangle to cluster the field of view of the marked points in each frame, which reduces the time complexity of calculating the common visible region. By comparing experiments with LCSS and LCVS, the experimental results prove that MTPRC is better than the present previous research in terms of algorithm accuracy and time cost.**

Keywords*—*UAV video, Video clustering, Video similarity measurement, Minimum Tilt outer rectangle.

I. INTRODUCTION

With the rapid development of UAVs targeting industry applications, UAV videos are endowed with spatial information, making the videos with scalable GIS data to provide new types of data support for information acquisition, feature extraction, etc.[1]. This type of UAV video with spatial information can be displayed overlaid with 2D and 3D geospatial data to achieve the effect of GIS-enhanced video. The video data is superimposed with geographic data such as points (GPS), lines (moving trajectories), surfaces (video shooting areas), and text, which accurately depict the geometric characteristics of the UAV flight process in the video, and these geometric characteristics can be extracted from the visual scenes of individual UAV videos [2], which are used to calculate the similar scenes of UAVs and cluster similar UAV videos.

Among the algorithms for UAV video clustering, the cluster analysis algorithm based on historical trajectory data is most widely used. This class of methods mainly studies the spatial distance problem of UAV video trajectories, and adopts the trajectory similarity metric based on GPS information to calculate the Euclidean distance between UAV video trajectories[2]. The computational objectives are: to measure the distance of the spatial positions of UAV videos, and to quickly cluster UAVs with similar flight trajectories. Currently known methods mainly use UAV video clustering algorithms based on image similarity, which measure the similarity between videos by the degree of overlap between video images, and the accuracy has been greatly improved. However, the clustering task of high-precision videos creates higher requirements for time efficiency, and the current image similarity-based UAV video clustering algorithms still face the following major problems: image similarity-based UAV video clustering algorithms[3] use video frames as the data representation model, and calculate the similarity between video frames by comparing them frame by frame. However, as the number of frames increases, the computation volume of video frame comparison increases exponentially, which is prone to cause "dimensionality disaster" and high time cost.

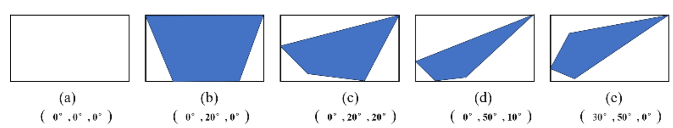


Fig. 1. Visible area of UAV imagery[4].

Visible area of UAV images in different azimuth, pitch, and roll angles, with blank area as error index space.

Therefore, to address the above pending scientific issues, this project proposes to carry out research on UAV video clustering methods based on adaptive scene similarity, summarizing the main bases as follows:

1) Construct a novel adaptive UAV video scene data model, adopt "multi-frame fusion" to construct UAV video scene model, and optimize the data representation of UAV video.

2) Proposed UAV video similarity calculation method: similarity algorithm based on UAV video minimum tilt outer rectangle clustering (MTORC).

3) We conducted a comparison experiment between the new algorithm and the previous algorithm, and the results show that our algorithm is better than the previous algorithm in terms of accuracy and computational cost, and further solves the problem that the previous algorithm is more sensitive to the unwanted noise and uneven sampling rate.

The rest of the paper is distributed as follows: Chapter II describes the work related to UAV video similarity measurements. Chapter III briefly summarizes our prior work. Chapter IV provides similarity algorithm based on UAV video minimum tilt outer rectangle clustering (MTORC) in detail. Chapter V describes our experimental design scheme and gives experimental results. Chapter VI summarizes the full paper.

II. Related Work

In this chapter, we summarize related work on UAV video. we describe point-based trajectory similarity computation methods along the timeline. There are several similarity metrics in previous studies aiming to measure the similarity between any two trajectories, which are mainly categorized into the following two categories[5]:

A. *Point-based trajectory similarity calculation methods.*

Point-based trajectory similarity calculation methods usually include DTW[6], LCSS[7], EDR and so on. Currently DTW has become a widely used point-based trajectory similarity algorithm. The similarity calculation based on DTW inherently has the problems of over-stretching and over-compressing, in order to avoid the appearance of over-phenomenon and thus ensure the reliability of similarity classification and clustering, Li et al. proposed an adaptive constrained DTW algorithm[8]. Besides, the extension of DTW never stops, Jeong et al. proposed the concept of weighted DTW[9] and Hong et al. followed by elaborating the concept of shape segmented DTW[10]. Liang Maohan et al. proposed an unsupervised CAE-based trajectory similarity measure [23], which automatically extracts the similarity of trajectories by a convolutional autoencoder (CAE) automatically extracts the low-dimensional features and converts the high-dimensional spatial features into low-level spatial features, which can improve the efficiency and effectiveness of similarity calculation. Considering the point set differences between two trajectories, Liu et al. proposed a Contrastive Self-Supervised Trajectory Representation Model (CSTRM)[11], which learns the trajectory representation by distinguishing the trajectory-level and point-level differences between trajectories. However, in general, they still need to bear great computational costs when facing larger UAV video datasets.

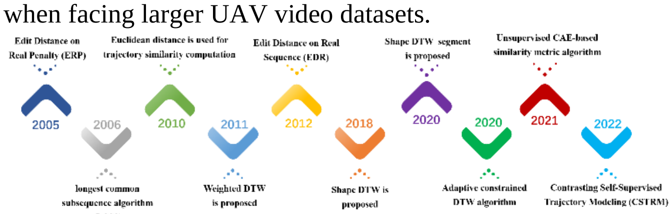


Fig. 2. Point-based similarity study timeline.

B. *Shape-based trajectory similarity calculation methods.*

Shape-based methods for calculating trajectory similarity usually include Hausdorff distance, Fréchet distance, etc. Arguedas et al. suggested calculating Hausdorff distance to cluster similar behaviors together[12]. Fréchet distance is commonly used to measure the distance of a pair of trajectories. However, for a large number of trajectories, the Fréchet distance is computationally very time-consuming. To further reduce the time cost of calculating trajectory similarity, Jinkwan Park et al. proposed a fast heuristic algorithm based on discrete Fréchet distance measurements[13], which quickly distinguishes trajectory adjacencies by discrete Fréchet distances. Cao et al. proposed to apply Fréchet distances to achieve adaptive trajectory clustering[14] for Roberts et al. developed a shape-based local spatial correlation measurement algorithm[15], which laid the foundation for the development of a shape-based local spatial correlation measurement algorithm by Roberts et al. Xu et al. proposed a metric based on the tight representation to calculate the spatio-temporal trajectory similarity metrics of the moving targets , which compresses the spatio-temporal trajectories of the two moving targets, and provides a new way of thinking for the proposal of the LCVS. Ding et al. proposed a maximal common view sub-sequence (LCVS)[16], but the LCVS The bottleneck of LCVS is the high time cost of frequently calculating the common visible region. Ding et al. proposed Maximum View Vector Subsequence (VVS)[2] on the basis of LCVS, which solves the problem of high computational cost of LCVS by comparing the frame-by-frame and calculating the viewpoint vector distances to get the maximum view subsequence, and then obtaining the trajectory similarity. Yu et al. proposed a method of mixing the LCVS algorithm with the VVS algorithm to compute the candidate videos mixed with the new K-SSMV algorithm[17]. For better learning of trajectories, Luo et al. designed a graph-based comparative learning framework for trajectory similarity computation (CLAIS)[18], which adopts an unsupervised approach to train the model for trajectory similarity. However, these shape-based methods are usually more sensitive to unwanted noise and inhomogeneous sampling rate, resulting in the associated trajectory data mining results will be degraded in the actual computation.

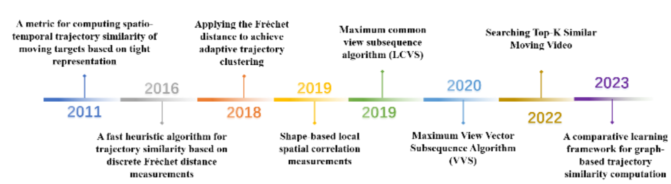


Fig. 3. Shape-based similarity study timeline.

We found that although these methods and their extensions help to compute the similarity between trajectories, these traditional methods usually have some drawbacks such as high computational cost, high sensitivity to unwanted noise and uneven sampling rate. Pre-processing of the dataset has been done in the previous work, but there are still problems such as high computational cost and high sensitivity of similarity calculation results to certain factors. In order to further solve the existing problem, we have worked out a new similarity metric algorithm.

III. PRELIMINARIES

In this chapter, we review prior work that can measure the similarity of UAV videos and discuss the limitations of the prior work. We present prior work on the LCVS algorithm[15].

The field of view is defined as the visual range of a video frame in the UAV video to be measured. The spatial attributes of a UAV video frame with scalable GIS data are defined as $f_i = (p_i, r, \theta_i, \delta_i)$, where i is the timestamp, p_i is the position of the UAV, θ_i is the angle between the direction from due north and the direction of the UAV camera, and δ_i is the maximum visible range of the UAV camera. The visual area of the UAV video frame f_i is defined as $view(f_i)$. The UAV video with frame rate m can be represented as:

$$V = \left\{ f_i \,|\, p_i, r, \theta_i, \delta_i, 1 < i \le m \right\}$$

The intersection region of two UAV video frames can be denoted as $|view(f_i) \cap view(f_j)|$, and the concatenation region of two UAV video frames can be denoted as $|view(f_i) \cup view(f_j)|$. The similarity of two UAV video frames can be reflected by the common view weight (cvw), which is defined as:

$$cvw(f_i, f_j) = \frac{|view(f_i) \cap view(f_j)|}{|view(f_i) \cup view(f_j)|} \tag{1}$$

Therefore, we can use the LCSS core algorithm to calculate the similarity of UAV videos. Define $V(A)$ as the sequence of fields of view of A from 1 to $m-1$, i.e., $V(A) = f_1, \dots, f_i, \dots, f_{m-1}$, Define σ as the threshold of the maximum time interval.

The LCVS core algorithm is defined as:

$$\begin{cases} 0, \ if \ A \ or \ B \ is \ empty \\ cvw(f_i, f_j) + LCVS(V(A), V(B)) \ if \ cvw(f_i, f_j) > 0 \ and \ |i-j| < \sigma \\ \max(LCVS_\sigma(V(A), B), LCVS(V(B), A)) \ otherwise \end{cases} \tag{2}$$

The result returned by the LCVS algorithm is the sum of the common view weights between the two UAV videos, and we need to normalize the returned result. The final similarity of A, B is defined as:

$$simLCVS_\sigma(A, B) = \frac{LCVS_\sigma(A, B)}{\min(m, n)} \tag{3}$$

The LCVS algorithm has the limitation of approximating the field of view region as triangles for processing, which is in error with the real perspective. Secondly, the LCVS algorithm needs to calculate the intersection and concatenation of the field-of-view regions in large quantities, which leads to a great time cost of calculation.

To solve these existing problems, we propose a new algorithm, i.e., similarity algorithm based on UAV video minimum tilt outer rectangle clustering (MTORC).

We formally define the problem of measuring similarity between UAV videos and give the notational definitions commonly used in this paper.

In the problem definition, the spatial attributes of a UAV video frame with scalable GIS data are defined as: $f_i = (p_i, r, \theta_i, \delta_i)$, where i is the timestamp, p_i is the position of the UAV, θ_i is the angle between the direction from due north and the direction of the UAV camera, δ_i is the maximum visible range of the UAV camera. The UAV video with frame rate m can be represented as:

$$V = \left\{ f_i \,|\, p_i, r, \theta_i, \delta_i, 1 < i \le m \right\}$$

Given two UAV videos A, B with scalable GIS data, $MTORC(A, B, r, \sigma, fps, \delta_i)$ problem is defined as follows:

Input: Two UAV videos with scalable GIS data, where UAV videos A and B, include:

- Visual distance of the videos r.
- Maximum time interval σ.
- Frames per second of the video fps.
- Maximum range of viewable angle at marker points δ_i.

Output: Similarity value of the two UAV videos s.

Purpose: The new algorithm utilizes the spatial location of sample points in UAV videos to calculate the similarity of UAV videos using the minimum skew outer rectangle, video clustering method.

The following TABLE I shows the notation definitions commonly used in this paper.

TABLE I. FREQUENTLY USED NOTATIONS.

Notation	Description
$V(A)$	A sequence of fields of view from 1 to m-1.
f	Frames in a UAV video segment.
i	The timestamp of an UAV video.
p_i	The spatial location of f_i.
r	Visual range of field of view marker points.
θ_i	The angle between the direction from due north and the direction of the UAV's camera.
lat_i	Latitude of the field-of-view marker point at time point i.
lon_i	Accuracy of field-of-view marker points at time point i.
δ_i	Maximum viewing angle of field of view marker points.
fps	The frame rate of this UAV video.
$corner_i$	Minimum tilted rectangle after field-of-view clustering.
σ	Maximum time threshold.

IV. SIMILARITY ALGORITHM BASED ON UAV VIDEO MINIMUM TILT OUTER RECTANGLE CLUSTERING.

The similarity of UAV video frames is related to the field of view range and we give the following four definitions:

A. Field of view[19]

The similarity of UAV video frames is related to the field of view range and we give the following four definitions:

Definition 1 (Field of view): Given a video frame f_i in a UAV labeled video A, $vertex_i$ is the set consisting of the coordinates of the vertices of the field of view range of video framef_i. $vertex_i$ is defined as:

$$\begin{cases} vertex1 = (lat_i, lon_j) \\ vertex2 = (lat_i + r * \cos(\theta_i + \delta_i), lon_i + r * \sin(\theta_i + \delta_i)) \\ vertex3 = (lat_i + r * \cos(\theta_i - \delta_i), lon_i + r * \sin(\theta_i - \delta_i)) \end{cases} \tag{4}$$

Definition 2 (Maximum Tilt External Rectangle): Given the field of view range point set $vertex_i$ for a unit time video frame, define box to include the coordinates of the center point of the smallest tilted external rectangle within the field of view range in the unit time $(center_x, center_y)$, the rectangle's length and width w and $\hbar$ and the tilt angle μ.

$$box = (center_x, center_y, w, \hbar, \mu) \tag{5}$$

Definition 3 (Common view weight after clustering process): f_i and f_j are two video frames in a given UAV video A and B, then the common attempted weight per unit of time labeled is $mvw(f_i, f_j)$. The public view weight (mvw) after clustering process is:

$$mvw(f_i, f_j) = \frac{|view(f_i) \cap view(f_j)|}{|view(f_i) \cup view(f_j)|} \tag{6}$$

Where $|view(f_i) \cap view(f_j)|$ is the intersection of the field of view after clustering of two UAV video frames f_i and f_j per unit of time, and $|view(f_i) \cup view(f_j)|$ is is the intersection of the field of view after clustering of two UAV video frames f_i and f_j per unit of time, and the value of mvw is in the range 0 to 1.

B. Definition of basic algorithms

In order to obtain the intersection area of the minimum tilted outer rectangle of the field of view range after clustering of video frames, we give three related algorithms. In calculating the intersection area of the minimum tilted outer rectangle of the field of view range after clustering of video frames it is necessary to obtain the set of vertex coordinates of the intersection area, so it is necessary to make a judgment on whether a particular line segment intersects or not. The details of the algorithms are as follows:

Algorithm 1: Intersection_polygon(corner1, corner2)

Input: **corner1:** The set of coordinates of the vertices of the smallest tilted outer rectangle of the range of the field of view per unit time of the UAV video A
corner2: The set of coordinates of the vertices of the smallest tilted outer rectangle of the range of the field of view per unit time of the UAV video B
Output: Get the set of coordinates of the vertices of the intersection of rectangles corner1 and corner2.

1 $corner1$ ← Obtain the coordinates of the four vertices of the rectangle from box_A
2 $corner2$ ← Obtain the coordinates of the four vertices of the rectangle from box_B
3 **for** i = 0 → 4 **do**
4   **for** j = 0 → 4 **do**
5   m, temp ← ***Line_ntersection***(***corner*1**, ***corner*2**, ***i***, ***j***)
6   **if** m **then**
7   interpoint [inter*2] ← temp [0]
8   interpoint [inter*2 + 1]← temp [1]
9   inter ++
10   **end**
11 **end**
12 **return interpoint**

Line_ntersection(***corner*1**, ***corner*2**, ***i***, ***j***) in the fifth line of Algorithm 1 is used to determine if the lines intersect and return the coordinates of the intersection point.

Algorithm 2: Calculate_intersection (coords)

Input: **coords:** Set of coordinates of the vertices of the intersection area of the two smallest inclined exterior rectangles
Output: Area of intersection of the two smallest inclined external rectangles.

coords ← Intersection_polygon(corner1, corner2)
n ← length(coords)
j ← n − 1
for i = 0 → n **do**
    area ← area + (coords [j][0] + coords [i][0]) * (coords [j][1] - coords [i][1])
    j ← i
end
return area/2

C. Similarity Algorithm Based on Minimum Tilt Outer Rectangle Clustering of UAV Videos (MTORC)

In this section, we further describe *MTORC*. Since the UAV video consists of countless consecutive video frames, we define *MTORC*. Let A be a sequence of band by UAV video frames of length m and B be a sequence of frames of band by UAV video of length n. Let V(A) be a subsequence frame of A of length m-1, i.e., $V(A) = f_1, \ldots, f_j, \ldots, f_{m-1}$. And V(B) is a B subsequence frame of length n-1, i.e., $V(B) = f_1, \ldots, f_j, \ldots, f_{n-1}$.

The core algorithm of $MTORC_\sigma(A, B)$ is defined as:

$$\begin{cases} 0, \ if \ A \ or \ B \ is \ empty \\ mvw(f_i, f_j) + MTORC(V(A), V(B)) \ if \ mvw(f_i, f_j) > 0 \ and \ |i-j| < \sigma \\ \max(MTORC_\sigma(V(A), B), MTORC(V(B), A)) \ otherwise \end{cases} \tag{7}$$

V. Experimental Evaluation

In this chapter, we give the experimental design and analyze the experimental results. We experimentally evaluated the overall performance of the three algorithms (LCSS, LCVS and MTORC).

A. Experiment setup

For this experiment we used 5000 UAV videos in New York, USA from BDD100K. The 5000 datasets include elements such as timestamp, altitude, longitude, latitude, horizontal acceleration, vertical acceleration, and velocity of the UAV videos.

The dataset for this experiment includes:

(1) 5000 raw UAV videos: all these data have been collected by UAVs.

(2) 5000 datasets are generated when the video frame rate is 24.

(3) When the video frame rate is 30, 120,000 datasets are generated.

(4) When the video frame rate is 60, 150,000 datasets are generated.

In the dataset used for the experiments, the orientation of the viewing angle was randomly varied and not necessarily aligned with the orientation of the moving object, obtained from the camera recordings of the UAV. Due to some systematic errors in the remote positioning of the geographically clustered observation points in the video using GPS, there is an error of ±0.06 in the measured data.

All experiments were performed on an Intel (R) Core (TM) i5-1135G7 CPU machine with 16 GB of RAM and 20 TB of hard disk space. The algorithms were implemented based on Python 3.9.7.

The Fig.4 shows our experimental setup. We chose two different evaluation aspects, data size and frames per second, to compare and analyze the three different algorithms (LCSS, LCVS, MBTR).In addition, we set the default threshold *cvw* to 0.35. In general, since the viewing angle of the UAV camera is between 90° and 150°. Therefore the default threshold of visual angle is set to 120° in the experiment. The maximum clear visibility distance of the UAV is 100 meters, so we set the distance threshold to 50 meters. Finally, we measured the similarity of these trajectories and calculated the computation time, accuracy to verify the efficiency and effectiveness of MTORC in measuring the similarity of geographically clustered videos.

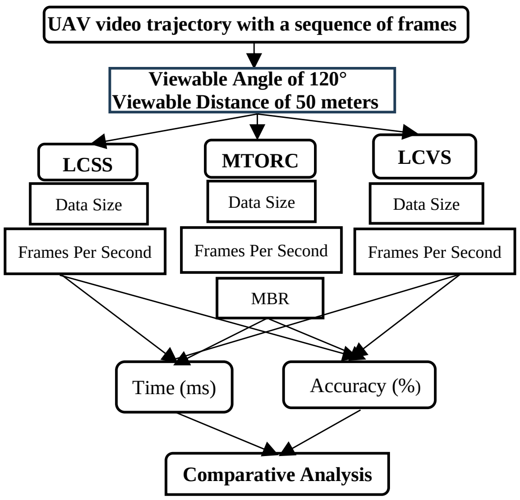


Fig. 4. Experiment layout.

B. *Effect of Data size*

The first experiment evaluated the effect of data size on the performance of the algorithms. The performance metric is the time cost of the algorithm. The number of data takes values ranging from 1000 to 5000. The frame per second is set to 30.

The Fig.5(a) gives the effect of data size on the running time of LCSS, LCVS and MTORC. The running time of all the algorithms increases with the increase in the number of UAV videos. We can clearly see that the time performance of MTORC and LCSS is significantly better than that of LCVS. This is because LCVS needs to utilize geometric computation frequently to compute the common visible area of the UAV video viewpoint.

The second experiment evaluated the effect of data size on the performance of the algorithm. The performance metric is the accuracy of the algorithm. The amount of data ranged from 1000 to 5000, and the frames per second is set to 30.

The Fig.5(b) shows that the accuracy of LCSS, LCVS and MTORC is relatively stable for different amount of data. The accuracy of MTORC is close to 94% and the accuracy of LCVS is more than 90%. Since LCSS relies only on spatial location to compute the similarity of UAV videos, LCSS performs poorly in terms of accuracy at around 40%. This is due to the fact that in both LCVS and MTORC algorithms dynamically determine the common view weights by calculating the intersection and concatenation between viewpoints, which possesses a higher accuracy compared to LCSS which only considers point distances to measure inter-trajectory similarity.

C. *Effect of Frames per second*

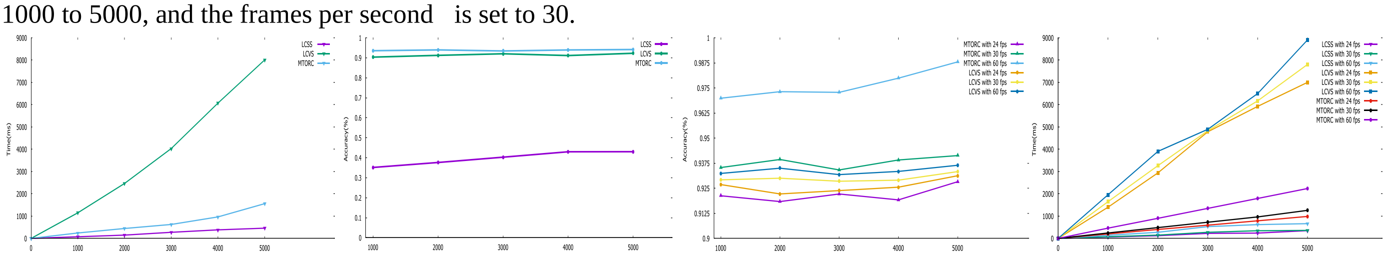
The third experiment evaluates the effect of frames per second on the performance of the algorithm. The performance metric is the accuracy of the algorithm. Common frame rates 24, 30, and 60 were selected for comparison experiments. The number of data is set to 5000.

Since LCSS utilizes spatial location for similarity calculation and the increase in frame rate per second only serves to increase the number of UAV video frame identification points, the accuracy of LCSS is not ideal and the increase in frame rate has no significant effect on the improvement of LCSS accuracy, so it is no longer shown in the above Fig.5(c). As the frame rate increases, the trend of MTORC accuracy improvement is more significant than LCSS and LCVS. This is because MTORC relies on clustering the range of viewpoints per second before calculating the video similarity, the higher the frame rate, the denser the range of field of view per unit time, and the smaller the clustering error using the minimum tilted outer rectangle.

D. *Effect of Frames per second and Data size*

The performance metric is the computational cost of the algorithm. The amount of data varies from 1000 to 5000. The visual distance threshold is set to 50 meters.

The Fig. 5(d) gives the effect of data size and frame rate on the running time. The runtime of all the algorithms increases with the increase in the amount of data. When the frame rate is the same, the time performance of MTORC and LCSS is significantly better than that of LCVS. This is because LCVS needs to calculate the common field of view area of the viewpoint frequently. When the number of data is the same, the running time of all algorithms increases with the increase of frame rate, but the computing time of LCVS is most affected by the frame rate and LCSS is least affected by the frame rate.



(a) (b) (c) (d)

Fig. 5. Experiment result.

E. Comparison of MBR and MTORC Accuracy

The fifth experiment evaluated the comparison of the performance of the MBR algorithm and the MTORC algorithm. The performance metric was the accuracy of the algorithm. The number of data takes values ranging from 1000 to 5000. The frame per second is set to 30.

TABLE II. COMPARISON OF MBR AND MTORC ACCURACY

Data size	MBR	MTORC
1000	69.96%	93.53%
2000	72.95%	93.93%
3000	70.95%	93.41%
4000	75.27%	93.91%
5000	78.78%	94.13%

The TABLE II shows that as the number of data increases, the change in the accuracy of MTORC stabilizes and is above and below 94%. While the accuracy change of MBR fluctuates more, fluctuating in the range of 70% to 74%. This is because MBR clusters the range of the field of view per unit time and takes the smallest outer rectangle after clustering, while the direction of the viewing angle of the dataset used in the experiments varies randomly, so if the direction of the viewing angle differs greatly from the direction of the moving object, the smallest outer rectangle obtained has a large error. MTORC solves this problem. This is the main reason why both the accuracy and robustness of MTORC are better than MBR.

VI. CONCLUSION

We propose a UAV video similarity measure to improve the efficiency of UAV video data mining by clustering and analyzing UAV videos and training their subclasses based on the measured UAV video similarity. However, calculating the UAV video similarity is challenging due to the high computational cost of calculating the common visual region between two UAV video frames. In this paper, we propose a simplified similarity measurement method based on field-of-view clustering for quickly measuring the similarity between UAV videos, namely, the similarity algorithm based on the minimum tilt outward rectangular clustering of UAV videos (MTORC). The core idea of the MTORC algorithm is to utilize the field-of-view clustering method to solve the problem of the previous algorithms with high computational cost. Experimental results show that the algorithmic accuracy and time performance of MTORC are better than previous studies.

ACKNOWLEDGEMENT

This work was supported by the One Belt One Road Innovative Talent Exchange with Foreign Experts under Grant No.DL2022024004L with the part of the Pilot International Cooperation Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant No.2022GH007.

REFERENCES

[1] Wang T, "The Literature Review of Unmanned Aerial Vehicle (UAV) Industry[J].The Frontiers of Society, " Science and Technology,2023,5(4)

[2] W. Ding, J. Tian, Y. Lee, K. Yang, and Kwang Woo Nam, "VVS: Fast Similarity Measuring of FoV-Tagged Videos," IEEE Access, vol. 8, pp. 190734–190745, Jan. 2020, doi: https://doi.org/10.1109/access.2020.3031485.

[3] M. Liang, R. W. Liu, S. Li, Z. Xiao, X. Liu, and F. Lu, "An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation," Ocean Engineering, vol. 225, p. 108803, Apr. 2021, doi: https://doi.org/10.1016/j.oceaneng.2021.108803.

[4] Y. Lu and C. Shahabi, "Efficient Indexing and Querying of Geo-tagged Aerial Videos," Nov. 2017, doi: https://doi.org/10.1145/3139958.3140046.

[5] W. Ding, C. Wang, S. Zhao, and X. Wang, "A Moving Object Query Method Based on the Cache Hit Rate in the Domestic Platform," 2023.

[6] Z. Zhang, K. Huang, and T. Tan, "Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes," Jan. 2006, doi: https://doi.org/10.1109/icpr.2006.392.

[7] H. Li, J. Liu, R. Liu, N. Xiong, K. Wu, and T. Kim, "A Dimensionality Reduction-Based Multi-Step Clustering Method for Robust Vessel Trajectory Analysis," Sensors, vol. 17, no. 8, p. 1792, Aug. 2017, doi: https://doi.org/10.3390/s17081792.

[8] H. Li, J. Liu, Z. Yang, R. W. Liu, K. Wu, and Y. Wan, "Adaptively constrained dynamic time warping for time series classification and clustering," Information Sciences, vol. 534, pp. 97–116, Sep. 2020, doi: https://doi.org/10.1016/j.ins.2020.04.009.

[9] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," Pattern Recognition, vol. 44, no. 9, pp. 2231–2240, Sep. 2011, doi: https://doi.org/10.1016/j.patcog.2010.09.022.

[10] Jae Yeol Hong, Seung Hwan Park, and J.-G. Baek, "SSDTW: Shape segment dynamic time warping," Expert Systems with Applications, vol. 150, pp. 113291–113291, Jul. 2020, doi: https://doi.org/10.1016/j.eswa.2020.113291.

[11] X. Liu, X. Tan, Y. Guo, Y. Chen, and Z. Zhu, "CSTRM: Contrastive Self-Supervised Trajectory Representation Model for trajectory similarity computation," Computer Communications, vol. 185, pp. 159–167, Mar. 2022, doi: https://doi.org/10.1016/j.comcom.2022.01.001.

[12] V. Fernandez Arguedas, G. Pallotta, and M. Vespe, "Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 722–732, Mar. 2018, doi: https://doi.org/10.1109/tits.2017.2699635.

[13] J. Park, T. Kim, B. Park, and H.-G. Cho, "Fast Heuristic Algorithm for Similarity of Trajectories Using Discrete Fréchet Distance Measure," KIISE Transactions on Computing Practices, vol. 22, no. 4, pp. 189–194, Apr. 2016, doi: https://doi.org/10.5626/ktcp.2016.22.4.189.

[14] J. Cao et al., "PCA-based hierarchical clustering of AIS trajectories with automatic extraction of clusters," Mar. 2018, doi: https://doi.org/10.1109/icbda.2018.8367725.

[15] S. A. Roberts, "A Shape‐Based Local Spatial Association Measure (LISShA): A Case Study in Maritime Anomaly Detection," Geographical Analysis, vol. 51, no. 4, pp. 403‐425, Nov. 2018, doi: https://doi.org/10.1111/gean.12178.

[16] W. Ding, K. Yang, and Kwang Woo Nam, "Measuring similarity between geo‐tagged videos using largest common view," Electronics Letters, vol. 55, no. 8, pp. 450‐452, Apr. 2019, doi: https://doi.org/10.1049/el.2018.7499.

[17] W. Ding, Q. Yu, and Z. Du, "Searching Top-K Similar Moving Videos," Computing and Communications, pp. 168–174, 2022.

[18] S. Luo, W. Zeng, and B. Sun, "Contrastive Learning for Graph-Based Vessel Trajectory Similarity Computation," Journal of Marine Science and Engineering, vol. 11, no. 9, pp. 1840–1840, Sep. 2023, doi: https://doi.org/10.3390/jmse11091840.

[19] Sakire Arslan Ay, R. Zimmermann, and Seon Ho Kim, "Viewable scene modeling for geospatial video search," Oct. 2008, doi: https://doi.org/10.1145/1459359.1459401.