# Watermark Hiding in Video Object Based on Complex Shape Trajectory

[1]Prof. Dr. Israa Hadi Ali, [2]Safa S. AL-Murieb
[1]*IT College - Babylon University, Iraq, israa_hadi1968@yahoo.com*
[2]*Ph.D. postgraduate - IT college - Babylon University - Iraq, safasaad1981@yahoo.com*

***Abstract***

*In order to protect and secure a video, watermark is added within data of video, to avoid conceal the watermark in fixed location of the frames of video, this paper indicates an approach for selecting an active object that characterizes with motion activity this means its corresponding trajectory has complex shape in distribution of x and y coordinates without gradual and sequential of points distribution, whereas this object's trajectory has unexpected distribution of points for its motion and position changing throughout frames of an input video. The approach of this paper depends on the concept of metric space with allocation of epsilon parameter which implements the maximum distance that can aid to determine the way or shape of distribution a trajectory's point, and specify which a trajectory with the complex shape and there are more curvatures in it in order to conceal the watermark in an object that corresponds to a trajectory that is selected by the approach that was indicated in details in this paper.*

**Keywords:** *Information Hiding, Video Object Trajectory, Video Watermarking, Complex and Curvature Trajectory, Metric Space.*

## 1. Introduction

In communications ways when the sender sends his/her information to other parity, it may be stolen or modified, so that it needs a protection and security way in order to ensure preservation of original information. To secure the data of multimedia which is in digital form, information is added within it; multimedia can be image, video, audio, or text [1]. There are many methods of data securing and protecting such as cryptography, watermarking, and steganography. Each one has its characteristics and appropriate applications in wide range of data transitions, Whereas a method of converting the source data to another form which is non-readable if the foreign user gets it, except the authorized receiver user can transform it to the original form and can reads it, this protection schema is called cryptography [2]. Another different preserving way is by conceal certain information which is in a digital form as text or picture to preserve the absentee of information, this is named as watermarking, while a schema of writing a data such as any digital signal (audio file, image, text, and video) in another signal that is called a cover, named as steganography [3]. To avoid concealing the watermark in fixed location of the frames of video, so that video objects are tracked, whereas video object tracking refers to estimating object's position over time, the video tracking aims to locate target objects in consecutive frames of video. A trajectory can be defied as the path that an object moving on it, it is a set of (x-position, y-position) points. Trajectories are the moving objects' traveling during time such as motion of a person, a vehicle, etc. [4].

There were many researches worked with trajectory's treatment, Thi-Lan [5] segmented the trajectory into subtrajectories according to movement's characteristics using control points that are selected along the trajectory, these points have high curvature. With extraction from trajectories the control points, using distance ratio and sequences of direction of movement to represent the trajectories, with similarity evaluation using Edit Distance. Thanh [6] proposed new descriptor of spatio temporal to recognize an action by modelling it from many trajectories, and extracted the features of dominant points that were detected from these trajectories. Soleymani [7] used movement's features (cross scale analysis) in classification of objects, whereas parameters of movement (glance, velocity and distance traveled) were computed from trajectories of GPS. Ryoo [8] presented a representation of feature for activity recognition of first person videos depends on time series pooling, by keeping track the changes in values of descriptor over time and representing the motion in activity video. Luca [9] discovered patterns that are consistent in bottom-up strategy, whereas every trajectory was attached to a different moved part of an object. Since the state of the art features employs single trajectories, and to get more discriminative action motions, pairs of trajectories descriptor were used that are entirely relied on the motion. Bingbing [10] recognized and labeled an action using local features of a trajectory which are related to motion parts, in the first step candidates of motion parts are generated, and in the second step an objective function was used to select an appropriate term or label for each motion action.

In this paper after tracking the video objects and depending on an algorithm of watermark concealing, either selecting the most two active objects in its motion over the frames of video, or the most active object and hide the watermark in this object that has the motion activity depending on its trajectory over the frames.

## 2. Video Tracking

In computer vision there are many important areas such as segmentation, detection, tracking, and classification of moved objects, one of the problems that is studied and has many applications is tracking of an object, which depends on many factors like the availability of a prior knowledge about an object and what is the parameters type that will be tracked such as contour, location, etc.

A typical system of target tracking composes of three components, a model of object's appearance at particular location, a motion model (e.g. the relation of locations of an object over time, and a search mechanism to find the most likely location of an object in the current frame of video. Because of the dynamicity of the visual world and finitely number of objects that are moved across scenes of a video, there is assumption of immense importance. The cameras of video capture information of interested objects, the information are represented as sets of images whereas the pixels construct an image. To track the moving objects, it is necessary to segment these objects in the first frame (image), then detect and track them in the image sequences of a video, whereas in the tracking operation the location of each moving object is estimated over time by modelling the relationship between the appearance of target and values of its corresponding pixel.

The three steps in video analyzing are detecting moving objects, keeping track of objects, and analyzing the tracks of objects for recognizing their behavior, whereas a tracker provides object's information like area, shape, orientation, or location [4][11].

object tracking is relevant with many tasks such as:

- Recognition based on motion (for example when identify a human basing on gait, detecting an object automatically).
- Video indexing (e.g. video annotation and retrieval),
- Surveillance systems which means detecting events that are unlikely and suspected activities.
- Monitoring the traffic.
- Path planning and avoiding of obstacles to track the movement of vehicle.

Tracking of objects can be complicated because:

- There is noise in images.
- Some of information can be loss.
- The motion of an object is complex.
- There are partial or full occlusions in objects.
- Some objects are articulated.
- Requiring of real time processing.
- Complex shapes of objects.
- Changing in illumination of scene.

Video tracking refers to the locating process of an object or many objects that is moving throughout video frames. The trajectory's generation requires locating at time (t) the coordinates x, y of a moving object that is referred to as $\{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_N, y_N, t_N)\}$, whereas $(x_i, y_i)$ refers to the coordinates of moving object at time $t_i$. Object tracking aims to track a moving object by establishing correspondence between objects in series of frames, then to extract features information like speed, trajectory, direction, behavior, velocity, directional distance, glance, and others. The tracking operation requires detection of an object either in the first appearance in video or in each frame over time [11][12].

## 3. Metric Space

Metric space identifies the term of distance in real line, let S be a non-empty set of points (where as the groups of points is called a curve that its coordinates are continuous functions with time variable), where a pair (S, d) represents a metric space whereas d is a metric (e.g. distance function) that defined on set S [13].

The metric d(a, b) means a distance from point a ($a_1$, $a_2$) to point b ($b_1$, $b_2$), which is defined to be:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (1)$$

This distance function assigns a real number; d(a, b); to each pair of points (a and b) in set S if it has the following characteristics for all a, b, and c $\in$ S [14]:

- Positivity : It means that a distance is greater than zero, such that d(a, b) > 0.
- Non-degenerated : It means that a distance be a zero if a=b then d(a, b) = 0.
- Symmetry : It means that a distance does not base on direction (e.g. a distance from a to b point is the same as it from b to a point) such that d(a, b) = d(b, a).
- Triangle disparity : It means that d(a, c) <= d(a, b) + d(b, c).

## 4. Proposed Method of Active Object's Selection

To determine the active object which has complex shape trajectory, the first step is splitting input video to set of frames, and getting the objects in the first frame, then tracking all the object throughout all the frames of an input video. The tracking operation results the path (trajectory) for each video object, whereas the trajectory consists of point, each point is represented in x and y coordinates which implements the position of a center of an object.

### 4.1. First Approach

This approach depends on selecting each time three points sequentially, and computing the distance between the second point of selected three points and the middle point that is in the line between first and third points of selected three points as in the following algorithm.

**Algorithm 1 : Determining Complex Trajectory (Based on mid-Point).**
**Input**: Video Object Trajectories (Sets of n points, each point as (x,y)).
**Output**: More Complex Trajectory.
**Begin**
Step 1: For each video object trajectory
Step 1-1: Reduce the trajectory's points (make each point has one appearance).
Step 1-2: Set accumulator variable to zero.
Step 1-3: For each point *i* in the current trajectory do (except the last two points).
Step 1-3-1: Take the two subsequent points of current point, this forms three points.
Step 1-3-2: Compute the distance of line between the first and third points as:
$line = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ where j=i+2.
Step 1-3-3: Find the middle point *mid* that lies in the line between first and third points.
Step 1-3-4: Compute the distance *dist* between the second point and computed middle point as: $dist = \sqrt{(x_{i+1} - x_{mid})^2 + (y_{i+1} - y_{mid})^2}$
Step 1-3-5: Increment the accumulator by the calculated *dist*.
Step 1-3-6: Get the next point from the current trajectory and go to Step 1-3-1.
Step 2: Store the resulted accumulator value with its current trajectory number.
Step 3: Go to Step 1-1 to work with next trajectory.
Step 4: Retrieve the stored accumulator of each trajectory and sorted them decreasingly according to accumulated count value.
Step 5: Return the trajectory that has max value of accumulator that belongs to the more active motion of object , and all trajectories are sorted according to their complexity.
**End**

## 4.2. Second Approach

After tracking all the objects in an input video, their corresponding trajectories are get, and this approach depends on computing the distance between any two points of each trajectory and all its consequence points with dependence on determining the allowed maximum distance. The following figure (Figure 1) explains the block diagram of second proposed approach.

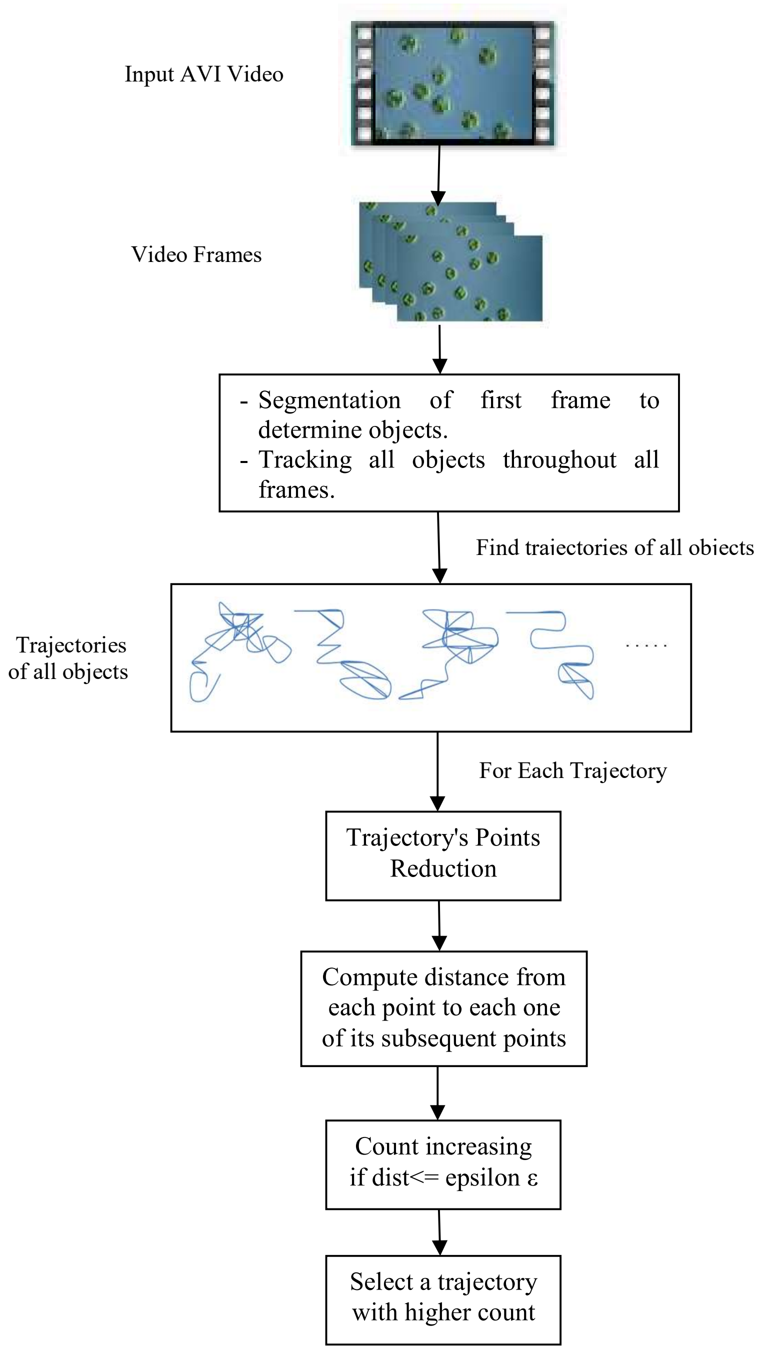


**Figure 1.** Block diagram of proposed method.

To determine the trajectory with complex shape that has curvatures which corresponds to the more active object, the distance *dist* between each two pixels is computed, compared with the value of epsilon variable $\varepsilon$, and compute the times when a distance *dist* is less or equal to $\varepsilon$.

For example if the trajectory consists of points a, b, c, d, e, and f as shown in figure 2 that indicates the distances from each point to all other points.

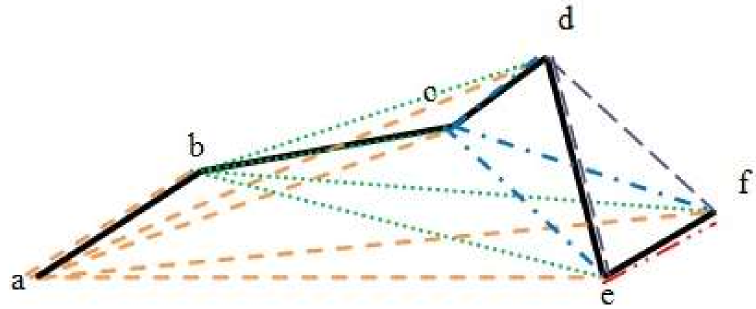


**Figure 2.** Distances between each point of trajectory and the others.

Sometimes a video object going through a point that it went through in previous frames, or in certain frames an object stays in its position without any motion, this means that the corresponding trajectory of that object has reduplicated point, and they will be considered more than once while they appear once in the trajectory's graphic, this affects the accounts of distances, so that the points of each object's trajectory are reduced to ensure that each point appears once as in a trajectory's drawing. If a trajectory consists of five points for example a, b, c, d, e, and f points as in figure 2, the dotted lines explain the distances from each point to the others.

The following algorithm indicates how to determine which trajectory is complex to select it for hiding the watermark in its corresponding object.

**Algorithm 2: Determining Complex Trajectory (Based on Epsilon Parameter) .**
**Input**: Epsilon Variable $\varepsilon$, Video Object Trajectories (Sets of n points, each point as (x,y)).
**Output**: More Complex Trajectory.
**Begin**
Step 1: For each video object trajectory
Step 1-1: Reduce the trajectory's points (make each point has one appearance).
Step 1-2: Set count variable to zero.
Step 1-3: For each point in the current trajectory do (except the last point).
Step 1-3-1: Compute the distance between the current point and each one of subsequent points as: $dist = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ where $i< j<=n$.
Step 1-3-2: Test the resulted *dist* value according to the $\varepsilon$ value.
Step 1-3-3: Increment the count by one if $dist <= \varepsilon$.
Step 1-3-4: Get the next point from the current trajectory and go to Step 1-3-1.
Step 2: Store the resulted count value with its current trajectory number.
Step 3: Go to Step 1-1 to work with next trajectory.
Step 4: Retrieve the stored count of each trajectory and sorted them decreasingly according to accumulated count value.
Step 5: Return the trajectory that has max value of count that belongs to the more active motion of object , and all trajectories are sorted according to their complexity.
**End**

## 5. Experimental Results

After splitting the input video into frames, tracking all the object in a video throughout video's time, as the result the paths of all video object's motion are get (i.e. object's trajectory through frames). The experimental result was get with the input video of one kind of bacteria and its movement, figure 3 displays samples of the resulted data of trajectories with some points for each one, whereas each group represents x and y position of an object, each group is for a different trajectory of an object.

| Object ID 1 | | Object ID 2 | | Object ID 3 | | Object ID 4 | | Object ID 5 | |
|---|---|---|---|---|---|---|---|---|---|
| x | y | x | y | x | y | x | y | x | y |
| 920 | 057 | 353 | 068 | 103 | 171 | 541 | 230 | 265 | 272 |
| 920 | 057 | 351 | 069 | 103 | 170 | 541 | 230 | 265 | 272 |
| 920 | 057 | 351 | 069 | 103 | 169 | 541 | 230 | 265 | 272 |
| 921 | 055 | 353 | 068 | 103 | 169 | 542 | 230 | 265 | 271 |
| 922 | 055 | 352 | 068 | 104 | 169 | 541 | 230 | 265 | 271 |
| 922 | 057 | 352 | 068 | 103 | 169 | 542 | 230 | 265 | 271 |
| 921 | 057 | 352 | 069 | 104 | 168 | 553 | 231 | 265 | 271 |
| 921 | 056 | 351 | 069 | 104 | 168 | 541 | 230 | 265 | 271 |
| 921 | 057 | 351 | 070 | 104 | 167 | 541 | 230 | 265 | 271 |
| 922 | 058 | 350 | 071 | 104 | 166 | 542 | 230 | 265 | 271 |
| 921 | 058 | 350 | 071 | 104 | 165 | 541 | 230 | 265 | 271 |
| 922 | 058 | 350 | 071 | 105 | 164 | 541 | 230 | 265 | 272 |
| 921 | 059 | 351 | 071 | 106 | 164 | 542 | 229 | 265 | 272 |
| 919 | 061 | 351 | 071 | 105 | 164 | 542 | 228 | 265 | 272 |
| 919 | 060 | 352 | 071 | 104 | 164 | 540 | 228 | 265 | 271 |
| 920 | 059 | 350 | 072 | 105 | 163 | 540 | 228 | 264 | 271 |
| 920 | 059 | 350 | 072 | 105 | 162 | 541 | 228 | 266 | 271 |
| 921 | 060 | 350 | 072 | 105 | 163 | 550 | 229 | 265 | 269 |
| 918 | 061 | 351 | 072 | 105 | 163 | 540 | 227 | 264 | 269 |
| 917 | 059 | 350 | 072 | 107 | 162 | 552 | 229 | 264 | 269 |
| 918 | 060 | 350 | 072 | 106 | 162 | 541 | 228 | 265 | 269 |
| 918 | 062 | 351 | 072 | 106 | 161 | 540 | 228 | 263 | 269 |
| 918 | 061 | 351 | 072 | 109 | 161 | 542 | 228 | 264 | 268 |
| 918 | 062 | 351 | 072 | 109 | 161 | 541 | 227 | 265 | 268 |
| 918 | 062 | 351 | 071 | 107 | 161 | 540 | 226 | 265 | 269 |
| 918 | 062 | 351 | 072 | 109 | 160 | 540 | 227 | 264 | 269 |
| 918 | 062 | 351 | 072 | 109 | 160 | 541 | 227 | 265 | 268 |
| 919 | 060 | 351 | 072 | 108 | 159 | 540 | 226 | 264 | 268 |
| 919 | 061 | 351 | 072 | 109 | 160 | 540 | 226 | 263 | 267 |
| 920 | 062 | 351 | 073 | 111 | 160 | 551 | 228 | 263 | 268 |
| 919 | 061 | 351 | 073 | 110 | 159 | 551 | 228 | 265 | 269 |

| Object ID 6 | | Object ID 7 | | Object ID 8 | | Object ID 9 | | Object ID 10 | |
|---|---|---|---|---|---|---|---|---|---|
| x | y | x | y | x | y | x | y | x | y |
| 466 | 400 | 677 | 480 | 697 | 463 | 830 | 531 | 350 | 545 |
| 466 | 400 | 671 | 477 | 695 | 464 | 830 | 531 | 353 | 543 |
| 466 | 400 | 671 | 474 | 691 | 473 | 830 | 531 | 353 | 543 |
| 462 | 394 | 670 | 474 | 687 | 478 | 830 | 531 | 354 | 543 |
| 463 | 390 | 669 | 474 | 683 | 480 | 830 | 531 | 356 | 542 |
| 461 | 387 | 667 | 472 | 681 | 484 | 830 | 531 | 355 | 546 |
| 459 | 384 | 665 | 469 | 679 | 486 | 830 | 531 | 355 | 550 |
| 455 | 383 | 661 | 466 | 675 | 491 | 829 | 531 | 356 | 553 |
| 453 | 379 | 664 | 464 | 673 | 497 | 830 | 531 | 357 | 554 |
| 451 | 378 | 667 | 459 | 671 | 501 | 830 | 531 | 359 | 558 |
| 451 | 375 | 667 | 452 | 665 | 505 | 829 | 531 | 361 | 562 |
| 450 | 368 | 670 | 449 | 659 | 506 | 829 | 530 | 364 | 563 |
| 449 | 365 | 672 | 447 | 648 | 508 | 830 | 530 | 365 | 565 |
| 448 | 363 | 673 | 446 | 641 | 508 | 830 | 529 | 366 | 566 |
| 449 | 360 | 672 | 447 | 633 | 508 | 829 | 529 | 369 | 567 |
| 448 | 358 | 670 | 444 | 628 | 510 | 829 | 529 | 371 | 568 |
| 450 | 356 | 671 | 445 | 623 | 507 | 829 | 529 | 374 | 567 |
| 450 | 352 | 672 | 447 | 612 | 503 | 828 | 529 | 375 | 567 |
| 449 | 352 | 675 | 450 | 609 | 500 | 828 | 528 | 375 | 566 |
| 449 | 352 | 679 | 453 | 607 | 495 | 829 | 528 | 376 | 565 |
| 449 | 352 | 688 | 458 | 602 | 492 | 829 | 528 | 377 | 564 |
| 449 | 353 | 690 | 459 | 597 | 487 | 828 | 528 | 380 | 563 |
| 449 | 354 | 691 | 459 | 588 | 477 | 829 | 528 | 382 | 561 |
| 458 | 351 | 693 | 459 | 583 | 471 | 829 | 527 | 381 | 561 |
| 452 | 348 | 695 | 460 | 577 | 463 | 828 | 527 | 381 | 559 |
| 453 | 347 | 698 | 458 | 573 | 459 | 828 | 527 | 379 | 557 |
| 454 | 346 | 698 | 460 | 571 | 456 | 829 | 528 | 376 | 557 |
| 453 | 348 | 697 | 463 | 564 | 449 | 828 | 526 | 372 | 556 |
| 454 | 347 | 695 | 464 | 561 | 449 | 828 | 527 | 370 | 556 |
| 455 | 348 | 691 | 473 | 559 | 446 | 829 | 527 | 363 | 556 |
| 453 | 348 | 687 | 478 | 557 | 444 | 829 | 527 | 362 | 557 |

**Figure 3.** Samples of points of trajectories for some video objects through some frames.

The trajectories of some objects are as shown in figure 4 that indicates the motion of each object throughout the frames of video.

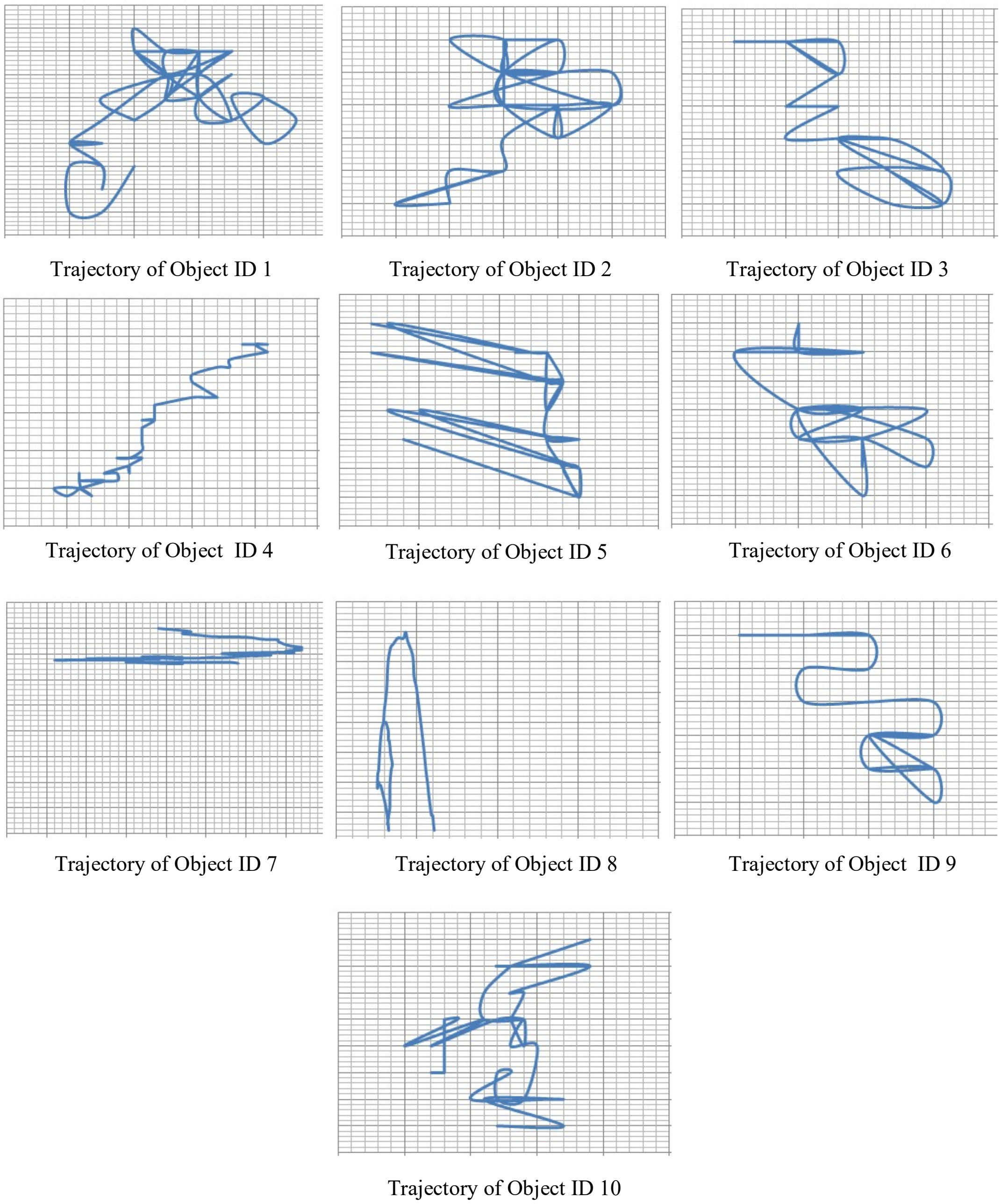


**Figure 4.** Trajectories of video objects.

It is clear in figure 3 that there are points equal in its x and y coordinates, this means that an object is stable without any movement in a period of time, and there are points that an object goes to them although of its movement passing these point before, so that the points of each trajectory needs to reduce them to exceed such cases.

When applying the first algorithm, it returns the trajectories that are not more complex in their shapes of points distribution. Table 1 shows the results of first approach.

**Table 1.** Results of first algorithm with sorting trajectories according to their shape complexity.

| | **Object Trajectories IDs According to Shape Complexity** |
|---|---|
| ↑ | Trajectory of Object ID 10 |
| | Trajectory of Object ID 5 |
| | Trajectory of Object ID 3 |
| | Trajectory of Object ID 8 |
| | Trajectory of Object ID 9 |
| | Trajectory of Object ID 2 |
| | Trajectory of Object ID 6 |
| | Trajectory of Object ID 1 |
| Less | Trajectory of Object ID 7 |
| Complexity | Trajectory of Object ID 4 |

It is clear that trajectories of objects IDs 1, 2, 6, 9, 10 are more five complex trajectories with more curvature.

When applying the second algorithm and compute distance from each point from each trajectory (except the last point) to each one of its subsequent point in the same trajectory. With every computed distance, comparing it with the entered epsilon $\varepsilon$ value increasing the value of count variable which implements the times of distances that less or equal to maximum determined distance between any two point.

The $\varepsilon$ implements the maximum distance which can aid to determine the way or shape of distribution a trajectory's point, thence specify which a trajectory with the complex shape of its point's allocation, which means selection a trajectory of an active object that has unexpected motion without gradual and sequential distribution of motion. Table 2 shows the results of the most complex trajectories (for example the more four complex, or as determining of number of resulted trajectories) with different values of $\varepsilon$ variable.

**Table 2.** Results of the most complex trajectories.

| **Epsilon $\varepsilon$ Value** | **More Four Complex Trajectories** | |
|---|---|---|
| | **Object trajectory No.** | **Count value** |
| 1 to 1.3 | Trajectory of Object ID 1 | 66 |
| | Trajectory of Object ID 10 | 64 |
| | Trajectory of Object ID 9 | 61 |
| | Trajectory of Object ID 2 | 58 |
| 1.5 to 1.8 | Trajectory of Object ID 1 | 119 |
| | Trajectory of Object ID 2 | 115 |
| | Trajectory of Object ID 9 | 115 |
| | Trajectory of Object ID 10 | 110 |
| 2 to 2.3 | Trajectory of Object ID 1 | 168 |
| | Trajectory of Object ID 9 | 163 |
| | Trajectory of Object ID 2 | 154 |
| | Trajectory of Object ID 10 | 145 |
| 2.5 | Trajectory of Object ID 1 | 246 |
| | Trajectory of Object ID 9 | 226 |
| | Trajectory of Object ID 2 | 222 |
| | Trajectory of Object ID 10 | 207 |

## 6. Conclusions

In order to increase the powerful and the characterized hiding algorithm, it doesn't depend on hiding the watermark in fixed place in the frames of video. To avoid the weakness of watermark concealing in the same place through the video frames, so that this paper tends to determine which one of the objects trajectories is more complex with more curvatures in order to conceal the watermark in different locations within the frames depending on hiding within a selected object that has unexpected motion without gradual and sequential points distribution of motion.

The first approach does not give more complex trajectories and does not sort them according to complexity and more curvature. While the result of second approach was get with different values of epsilon $\varepsilon$, which represents the maximum distance between any two point. The results indicates the more complex trajectories (for example four trajectories) that are the same with $\varepsilon$ value 1 to 1.3 with the same count for each, while when $\varepsilon$ value 1.5 to 1.8 also the same complex trajectories are get with a difference in their sequence according to their complexity and their counts are also changed, when $\varepsilon$ value 2 to 2.5, the same sequence of resulted trajectories is get with a difference their counts according to $\varepsilon$ value change (maximum distance). Whenever increase $\varepsilon$ value more, the approach begins leaving the more complex trajectories toward selects trajectories which have less complexity in their distribution of points. It is concludes that second approach gives more accurate results (trajectories with more complex shape).

It is possible to add features of trajectories such as the total distance that an object was moved throughout a video (this feature is also means the length of a trajectory, and the speed of object's motion through all frames. If an algorithm of watermark hiding depends on more two active objects and the approach that is shown in this paper can return more than one complex trajectory as you determine, It is possible to select the most two trajectories that have complex shape, that correspond to their objects for hiding in them, from the resulted complex trajectories, depending on computing the features of length and speed for the returned trajectories with determining a weight for each feature, and the trajectories that have higher result of accumulation of multiplication each feature with its weight, will be returned as the selected trajectories.

## 7. References

[1] Komal Patel, Sumit Utareja, Hitesh Gupta, "A Survey of Information Hiding Techniques", International Journal of Emerging Technology and Advanced Engineering, vol.3, Issue 1, pp.347-350, 2013.

[2] Richa Gupta, "Information Hiding and Attacks : Review", International Journal of Computer Trends and Technology (IJCTT), vol.10, no.1, pp.21-24, 2014.

[3] Richa Gupta, Sunny Gupta, Anuradha Singhal, "Importance and Techniques of Information Hiding : A Review", International Journal of Computer Trends and Technology (IJCTT), vol.9, no.5, pp.260-265, 2014.

[4] Emilio Maggio, Andrea Cavallaro, "Video Tracking: Theory and Practice", John Wiley & Sons, Ltd., United Kingdom, 2011.

[5] Thi-Lan Le, Alain Boucher, and Monique Thonnat, "Subtrajectory-Based Video Indexing and Retrieval ", In Proc. of MMM conference, pp. 418–427, 2007.

[6] Thanh Phuong Nguyen, Antoine Manzanera, "Action recognition using bag of features extracted from a beam of trajectories ", In Proc. of International Conference on Image Processing ICIP, melbourne, Australia. pp.4354 - 4357, 2013.

[7] Ali soleymani, E. Emiel van Loon, Robert Weibel, "Capability of Movement Features Extracted from GPS Trajectories for the Classification of fine-grained behaviors ", In Proceedings of the AGILE, International Conference on Geographic Information Science, Castellon, Spain, 2014.

[8] M. S. Ryoo, Brandon Rothrock, and Larry Matthies, "Pooled Motion Features for First-Person Videos", Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition, pp. 896-904, 2015.

[9] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, Vittorio Ferrari, "Articulated Motion Discovery using Pairs of Trajectories", Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition, pp. 2151-2160, 2015.

[10] Bingbing Ni, Pierre Moulin, Xiaokang Yang, Shuicheng Yan, "Motion Part Regularization: Improving Action Recognition via Trajectory Group Selection", Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition, pp. 3698-3706, 2015.

[11] Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2010.

[12] Subhash Challa, Mark R. Morelande, Darko Musicki, Robin J. Evans, "Fundamentals of Object Tracking", Cambridge University Press, United States of America, 2011.

[13] Satish Shirali, Harkrishan Lal Vasudeva, "Metric Spaces", Springer Science & Business Media, London, 2006.

[14] Mícheál O'Searcoid, "Metric Spaces", Springer Undergraduate Mathematics Series, London, 2007.