

Incremental trajectory aggregation in video sequences

Ionel Pop^{1,2}, Mihaela Scuturici¹, Serge Miguet¹

¹LIRIS - Lyon 2 University

²FOXSTREAM - <http://www.foxstream.fr>

{ionel.pop, mihaela.scuturici, serge.miguet}@univ-lyon2.fr

Abstract

This article introduces new similarity measures between trajectories, in order to detect uncommon behaviors. These measures are used to find the most common trajectories in a sequence, using an implicit aggregation method. They may be applied to trajectories of objects tracked in real time. Moreover, by combining one or more measures, it is possible to vary the impact of the temporal dimension — velocity along a trajectory. Our experiments show that the measures are able to properly identify rare trajectories in a video, as well as to detect the most frequent ones.

1 Introduction

Trajectory classification is a relative recent field of study. The results are used in the domain of video surveillance, mostly to detect uncommon behaviors. After a observation period of a scene, the system is able to learn the common moving patterns of objects in the scene. Then, any object that does not follow one of the learned pattern is detected as abnormal.

The current study presents some modalities to construct the trajectory patterns along with their frequencies. This approach is able to integrate new data and adapt the patterns continuously, so there is no need for a *learning phase*.

2 Related work

Most of the studied works involving trajectory classification follow a common pattern. The trajectories are either acquired using various tracking algorithms or they are simulated. In the case of simulated trajectories, they are randomly generated, under a certain predefined model. The trajectories are then normalized (temporally, spatially or both). Some authors split them in

sub-trajectories. The resulting segments may have fixed or variable length. In case of variable length, the division is done upon certain criteria, like discontinuity in speed, acceleration, etc. The trajectories (or the sub-trajectories) are then clustered. A metric is required for this step. There are a lot of propositions for the clustering method (spectral clustering, k-means, vector quantization, etc.) and the metric used.

In [5] and [1], trajectories are divided on points of high curvature, respectively discontinuities of the 1st and 2nd derivative. Each point and segment is then labeled with a symbol and comparing trajectories is transformed into a string matching problem.

Mixtures of Von Mises distributions are used in [3] to model trajectories. Similarities between trajectories are estimated with the Bhattacharyya distance and for clustering, k-medoids algorithm is used. The principal trajectories and an equivalent of “variance” are obtained in [7] using Altruistic Vector Quantization (AVQ), while [6] use a two-step clustering on histograms of pairs location-direction.

In [2], the similarity of two trajectories is estimated with a modified version of the LCSS (Longest Common Subsequence) algorithm applied on geometrical information. The clustering is done with agglomerative hierarchical clustering.

Every trajectory is modeled as a polynomial curve in [4]. Semantic information is then extracted based on the variation of speed and direction. This information is analyzed by a time delayed neural network, which was previously trained according to the user’s query.

3 Proposed solution

The solution proposed in the current study is composed of two steps. First, a measure of similarity is defined between two trajectories. Based upon this measure, similar trajectories are aggregated into *models of trajectories*. The solution estimates the frequency of each model.

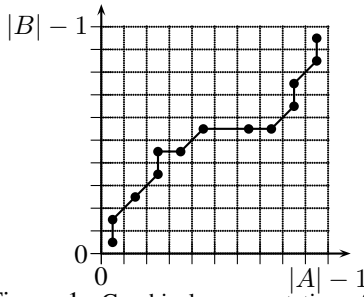


Figure 1: Graphical representation of DTW. The two axes represent the temporal dimension in the two temporal sequences. Each cell (i, j) contains the distance between the points $a_i \in A$ and $b_j \in B$. The cells are selected so that the sum of the distances is minimized.

For the first step, several measures are proposed and compared. They are based on Dynamic Time Warping (DTW), which has mainly been used in speech recognition for word comparison [8]. The main advantage of the method resides in its simplicity. The input data does not need to be transformed. DTW is used to align two temporal sequences, by warping them.

3.1 Dynamic Time Warping

Let $A = (a_t)_{t=0, S_a}$ and $B = (b_t)_{t=0, S_b}$ be two temporal sequences. DTW finds the optimum sequence $O = \{(i_k, j_k) | i_k \in \{0..S_a\}, j_k \in \{0..S_b\}, k \in \mathbb{N}\}$ so that the sum $D = \sum_{(i,j) \in O} \delta(a_i, b_j)$ is optimized. O is an ordered sequence of pairs of points of two sequences. The function δ is a distance function. i_k and j_k are non-decreasing with respect to k . Graphically, the problem may be assimilated to finding the shortest path in a 2D array, each cell containing the cost of traversing it (see figure 1). The condition on monotony of i_k and j_k indicates that the shortest path should be orientated from left-bottom to right-upper. In this study the similarity between A and B is D normalised by division by the number of steps, i.e. number of elements in O .

Following notations are used for trajectory specific data. T_i and S_i are two formalisms designating the same trajectory. T_i is composed of a set of points $T_i = (p_t^i)_{t=0, size_i}$, where $size_i$ is the duration of the trajectory. The point p_t^i contains information acquired at time t , such as the position, the size of the object, orientation, etc. In this study, only the position of the gravity center and the size of the object are considered. S_i is the equivalent sequence of segments: $S_i = (s_t^i)_{t=1, size_i}$.

3.2 New similarity measures

The first measure p -DTW based on DTW is applied directly to the points of the trajectories. In this case, the

distance $\delta = d(p_i, p_j)$ between two points is defined as the euclidean distance between the position of the points. The warping uses all points of two trajectories, i.e., $\forall x \in N, x < size_1, y \in N, y < size_2, \exists k, l$ such that $(x, k) \in O, (l, y) \in O$. The first point is $(i_0, j_0) = (0, 0)$ and the last one $(i_k, j_k) = (size_1, size_2)$. The set of all sequences O satisfying these conditions is designated with Ω . The similarity between the two trajectories is defined as

$$p\text{-DTW}(T_1, T_2) = \min_{O \in \Omega} \frac{1}{|O|} \sum_{(i,j) \in O} d_{euclidian}(p_i^1, p_j^2),$$

where $p_i^1 \in T_1, p_j^2 \in T_2$.

Next measure, g - dtw , is mostly geometrical. It is applied to a point-trajectory T_1 and a segment-trajectory S_2 . The distance δ between a point $p_i^1 \in T_1$ and a segment $s_j^2 \in S_2$ is defined as the spatial distance between the point p_i^1 and the segment s_j^2 . The condition on O is different from p -DTW: if $(i_k, j_k) \in O$, i_k is strictly increasing, while some points from S may not be used. The set of all sequences O satisfying these conditions is designated with Ω' . The first point has $i_0 = 0$ and the last one $i_k = size_1 - 1$. There are no supplementary requirements for j_k . Moreover, the trajectories do not need to be sampled uniformly.

The third measure t - dtw , is mostly temporal. It is similar to the second one, except for the distance δ that is defined as temporal distance between the p_i^1 and s_j^2 :

$$t\text{-dtw}(T_1, S_2) = \min_{O \in \Omega'} \frac{1}{|O|} \sum_{(i,j) \in O} |i - proj_{1,2}(j, i)|$$

where $p_i^1 \in T_1, s_j^2 \in S_2$. The function $proj_{1,2}(j, i)$ returns the time when the object on the segment s_j^2 is geometrically closest to the point p_i^1 . Its value is relative to the T_2 time line.

p -DTW is a simple and fast method to compare trajectories. Its value considers both the temporal and spatial dimensions. The temporal dimension is implicitly taken into account, by explicitly requiring the presence of each point and by the constant sample rate. The drawback is the difficulty to precisely delimit the influence of temporal information. g - dtw solves this problem, its estimation being the mean of the geometrical distance between the two trajectories. t - dtw is a complement of g - dtw , adding information about the temporal similarity of two trajectories.

g - dtw and t - dtw are asymmetric. In order to make them symmetric, they are applied twice to the same trajectories and the maximum of two measures is kept. The symmetric measures will be designated by g -DTW and t -DTW.

3.3 Data aggregation

The current study proposes a method to aggregate two trajectories into a third one, in order to construct representations of *common* trajectories. The aggregation is based on the results of DTW measures previously introduced. By successively applying it on acquired trajectories, the trajectories are clustered in an incremental on-line approach into trajectory models. The weight of a model is defined as the number of trajectories used for its building. This number, compared to other weights, gives information about the rarity of a model or trajectory. It is also used as weighting factor in aggregation.

After the estimation of DTW-based distances, the sequence O contains information about the associations of time-points of the two trajectories. In the case of p -DTW, these pairs are composed of two points (p_i^1, p_j^2) , while for g -DTW and t -DTW the pairs contain a point and a segment (p_i^1, s_j^2) . Nevertheless, it is possible to reduce the problem to the precedent case by replacing the segment s_j^2 with the point $p_j^2 \in s_j^2$ closest to p_i^1 . Its characteristics are estimated by linear interpolation.

For each such pair of points, their weighted *mean* is estimated by linear interpolation (including for the timestamp). The weights of the two points are the weights of the two trajectories or models of trajectory. The new point is added to the aggregated trajectory.

4 Experiments

Four types of experiments were made. First, the distances were tested on artificially generated regular trajectories, in order to verify some of their proprieties. Then, the aggregation mechanism was tested on random generated trajectories. The whole system was tested on a video sequence. At the end, the system was compared with the one presented in [2].

4.1 Artificially generated regular trajectories

Artificially generated trajectories are based on a straight line which was sampled in 13 different ways (see figure 2). They are rotated with $\pi/2$, π and $3 * \pi/2$, so that at the end there are 52 trajectories. The tests made on these trajectories confirm some properties of the similarity measures. The measure p -DTW is too sensitive to velocity and shifting of the trajectories, due to its implicit notion of time. g -DTW is sensitive to the direction of the trajectory, the difference between the normal and reversed trajectories being important, while the similarities between trajectories with same orientation, but different speeds stays low. t -DTW differentiates the various velocities, without being sensitive to

shifting. A $\pi/2$ rotation of the trajectory has little impact on its value.

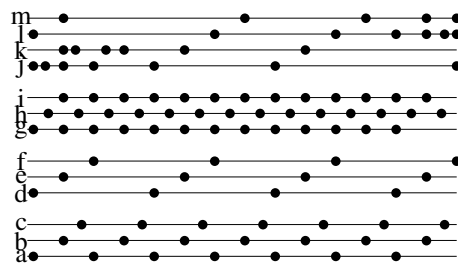


Figure 2: Section of the hand-generated trajectories: normal spaced trajectories (a-c), double spaced (d-f), half spaced (g-i), accelerated(j,k) and decelerated (l,m).

4.2 Random trajectories

To test the aggregation of trajectories, a simple algorithm was developed. A base of model trajectories is maintained. If the current trajectory is close enough to one of the model (given a threshold), the two trajectories are aggregated considering the weight of the model. After aggregation, the model trajectory is resampled with unitary step and its weight is increased. Otherwise, a new model trajectory with initial weight equal to one is created.

The second set of experiments is made on randomly generated trajectories. A few random models are chosen, with constant number of curvature changes. Trajectories are generated based upon these models by simulating a walk along these models, perturbed with Gaussian noise. The results show that all 500 trajectories generated with one curvature point on 320x240 images are well classified in one of the 15 models, using p -DTW and g -DTW with a threshold of 20. As previously said, g -DTW returns the *mean* distance between two trajectories. Therefore, a value of threshold of 20 constructs a model of 20 pixels *thick*, which is reasonable given the image size. This value was used for all our tests.

4.3 Video sequence

The same approach was used on a 320x240 video sequence, recorded during 5h with a frame rate of 5fps. The video shows the activity at one of the exits of the campus.

The objects were detected by background subtraction and tracked using feature matching (position, contour and histogram). The data for each point contains the position and the size of the object. Although the size is estimated during aggregation, it is not accounted for when estimating similarities between trajectories.

The most frequent trajectories are the ones showing cars entering and leaving, as well as pedestrians walking along the street. There are some rare detected trajectories, such as a car driving the wrong way, a bike turning around or a pedestrian walking across the street (see Figure 3). Some false trajectories are detected, mainly because of the poor performance of the tracker in case of occlusions and light reflections. The results are similar when using $p\text{-dtw}$ and $g\text{-dtw}$. There are about 220 detected objects, which were classified in 27 classes, most of them (17) with less than 3 trajectories.

The second experiment runs in about 5mn ($g\text{-DTW}$), respectively 1 minute ($p\text{-DTW}$), while the last experiment takes about 7 minutes ($p\text{-DTW}$) and 20 minutes ($g\text{-DTW}$) on a Intel 2.4GHz. The difference between ($p\text{-DTW}$) and ($g\text{-DTW}$) is partially explained by the fact that ($g\text{-dtw}$) must be estimated twice to obtain ($p\text{-DTW}$). In ($p\text{-DTW}$), δ estimates a distance between two points, while in ($g\text{-dtw}$) δ is a distance between a point and a segment, more time consuming.



Figure 3: Results of $p\text{-dtw}$ on the video. The first two pictures show common trajectories, while the last ones are detected as being rare, with only one occurrence for each model.

4.4 Comparison with LCSS

The solution presented in this study was compared with LCSS [2], based on the results of the first two experiments.

In the first experiment, it is noticed that the rotation of the trajectory with Π has the same effect as doubling the sampling rate in the case of linear trajectories, which makes LCSS less reliable than $g\text{-DTW}$.

In the case of random generated trajectories, LCSS correctly classifies all the 500 trajectories, but it requires more computing time (20 mn). This large difference is partially explained by the chosen clustering

algorithm. The advantage of our approach is the on-line clustering, which uses elements computed during distance estimation, other than the distance itself. This justifies partially the difference in speed.

5 Conclusions and future directions

The current study presented several similarity measures used to compare trajectories. In addition, these measures offer an implicit aggregation method. This method is adapted for on-line trajectory analysis, due to its iterative aggregation. Moreover, it is possible to view a graphical representation of trajectory models.

By linearizing input trajectories, it is possible to significantly reduce the computation time for $g\text{-DTW}$ and $t\text{-DTW}$. The similarity measures detect also if a sub-trajectory is included or not in another trajectory. This property may be used to predict the most probable trajectories of an object present in the scene.

References

- [1] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Transactions on Multimedia*, 9(1):58–65, Jan. 2007.
- [2] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Proc. 17th International Conference on Pattern Recognition ICPR 2004*, volume 2, pages 521–524, 23–26 Aug. 2004.
- [3] S. Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS 2007*, pages 141–146, 5–7 Sept. 2007.
- [4] X. Chen and C. Zhang. An interactive semantic video mining and retrieval platform—application in transportation surveillance video for incident detection. In *Proc. Sixth International Conference on Data Mining ICDM '06*, pages 129–138, Dec. 2006.
- [5] J.-W. Hsieh, S.-L. Yu, and Y.-S. Chen. Motion-based video retrieval by trajectory matching. *IEEE Trans. Circuits Syst. Video Technol.*, 16(3):396–409, March 2006.
- [6] X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Proc. 18th International Conference on Pattern Recognition ICPR 2006*, volume 1, pages 591–594, 2006.
- [7] A. Mecocci and M. Pannozzo. A completely autonomous system that learns anomalous movements in advanced videosurveillance applications. In *Proc. IEEE International Conference on Image Processing ICIP 2005*, volume 2, pages II–586–9, 11–14 Sept. 2005.
- [8] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust., Speech, Signal Process.*, 26(1):43–49, 1978.