Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 129 (2018) 468–474

Procedia Computer Science

www.elsevier.com/locate/procedia

2017 International Conference on Identification, Information and Knowledge in the Internet of Things

The Synchronization of Data Collection for Real-time Group Recognition

Lin Xu, Weiping Zhu*

School of Computer Science, Wuhan University, China

Abstract

It is commonplace for people to perform various kinds of activities in groups. The recognition of groups is of importance in many applications including crowd evacuation, teamwork coordination, and advertising for groups. Existing group recognition approaches require snapshots of human trajectories, which is often impossible in real life situation due to the differences of data collection start time and frequency. This paper proposes an approach to synchronize the trajectory data of people by interpolation based on Catmull-Rom Spline. The optimal interpolating points are computed based on our proposed error function. Moreover, we propose an approach to assign the groups proper colors and then uses the hot map to show the dynamic changes of groups graphically. A real-life data set is used to validate the effectiveness of the proposed approach. The results show that 97.9% accuracy of group recognition can be achieved and the dynamic changes of groups is well shown.

Keywords: Group recognition; Synchronization; Real-time; Trajectory Interpolation

1. Introduction

It is commonplace for people to perform various kinds of activities in groups. For example, family members, friends, and colleagues often go shopping together, and people under an earthquake often escape with familiar people rather than stay alone [1]. High cohesion of group enables more effective information dissemination and management for people. For example, if we send escape instructions to people groups rather than individuals in dangerous places, the redundant message transmission are not required and thus leads to a faster evacuation. In many such applications, real-time recognition of people groups is highly demanded.

Existing works for real-time recognition of people groups use the spatial-temporal clustering approach based on the trajectory of people [2, 3, 4]. The snapshots of the locations of people at specified time are obtained firstly.

* Corresponding author.
E-mail address: wpzhu@whu.edu.cn

10.1016/j.procs.2018.03.026

Such snapshots are feeded into a central server to perform clustering based on the neighbouring of people using density-based clustering algorithm like DBSCAN [5] or DJ-Cluster [6]. The identified groups at different snapshots are correlated to determine whether two groups in different snapshots are the same. If the similarity of a group at the current snapshot and a group at the last snapshot exceeds a threshold, we consider that these groups are the same group.

The snapshot of the locations of people is critical to the whole processing of the group recognition. The challenge is that the data of different people are required to be collected at the same time. We call this *the synchronization of data collection*. Actually, there are many factors affecting it: First, the sensing devices may start their data collection at different time. Second, the frequencies of data collection at different devices are different because there are different parameter settings in different devices. Due to such reasons, it is difficult to collect the data from people at given time. More data processing is needed to improve this situation.

This paper focuses on the synchronization of data for real time group recognition. First, we design a trajectory interpolation algorithm based on Centripetal Catmull-Rom Spline algorithm, and an error evaluation approach for the interpolation algorithm. Based on them, we synchronize the data of different people by interpolating data at a period f from starting time t_0. Detailed algorithm to compute the optimal t_0 is also illustrated. After that, we propose an approach for real-time graphical group monitoring and illustrate the detailed color selection algorithm used in it. Finally we use the real-life data to evaluate proposed approach and report the results. In summary, this paper makes the following contributions.

- We identified the unsynchronized data collection problem in the group recognition, and proposed an trajectory interpolation algorithm to solve it.
- We designed a reasonable error function for the trajectory interpolation, and proposed an algorithm to optimize the interpolation based on the error function.
- We designed a framework for the real-time graphical group recognition, and proposed a color selection algorithm to display the detected groups in the real-time way.

The rest of the paper is organized as follows: Section 2 summarizes the related works of this study. Section 3 introduces the system model. Section 4 describes our solution for synchronizing the collecting data. Section 5 shows the experimental evaluation and Section 6 concludes this paper.

2. Related Works

In recent years, many researchers have investigated the group recognizing based on sensors. Wirz [2] et al. proposed a pedestrian flock detection algorithm by using the spatial-temporal clustering [7] of a series of snapshots of GPS data of humans. Kjærgaard et al. further investigated the group recognizing using WiFi positioning data and the fusing of data from different kinds of sensors [3, 4, 8]. Feese [9, 10] et al. proposed to detect groups of firefighters by using built-in ANT radio and atmospheric pressure sensor of mobile phones. The ANT radio is used to determine the distance between two firefighters by their communications, and the atmospheric pressure sensor is used to determine the current floor in a building of a firefighter. These data are further clustered to detect the groups of moving people. In stead of using raw sensing data, Gordon et al. [11] proposed a distributed group affiliation detection algorithm based on the data distribution, i.e., a mixture of Gaussian for acceleration and a mixture of von Mises distributions for orientation. This method greatly reduces the amount of data to be transmitted and hence gains benefits in the real-time processing and energy consumption.

All these works are based on a series of snapshots of sensing data. It means that data need to be collected in a synchronized way. However, it is not easy to achieve this due to the different stating time and frequency of data collecting in different devices. We will solve this problem in this paper.

3. System Model

There are a number of n people in an region (e.g., a shopping mall) required to be monitored. Data collection starts when these people come into the region. The time duration for the data collection is T. For person i, the data is

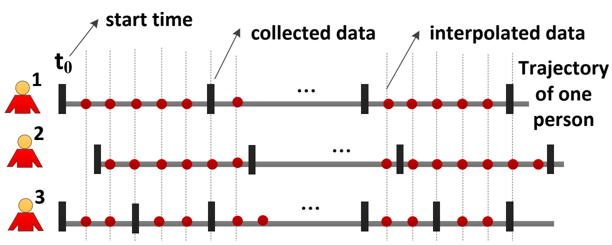

Fig. 1. Each user's Fréchet distance with different F

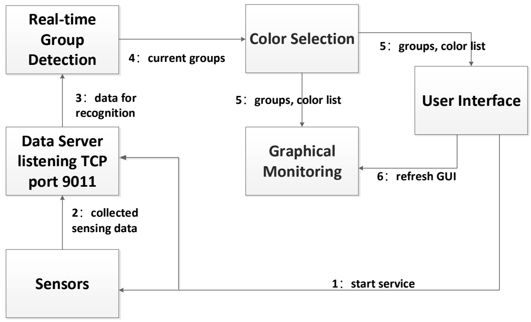

Fig. 2. Real-time graphical group monitoring system

collected periodically with the period of F_i. The data collection of the persons have not been synchronized, i.e., the starting times and the frequencies of their data collections are not necessary the same. Then we obtain a sequence of the persons' location data $P_0, P_1, ..., P_n$.

In this scenario, people may come into the region at different time and causes different start time of data collection. This further causes unaligned collected data. Moreover, different data collection frequencies also lead to unaligned collected data. We use Figure 1 to demonstrate data collection process in this scenario. The rectangles represent the timestamps of data collection. According to this figure, it can be seen that person 1 and person 2 start their data collections at different time, and thus the collected data are not aligned. Person 1 and person 3 have different data collection frequencies and thus also have unaligned collected data. The unaligned data make the group recognition difficult, because the group recognition is based on the snapshot the states of the persons.

We aim to synchronize the data collection by using interpolation technique. As displayed in Figure 1, the dots represents the interpolated data based on the collected data. We can see that the dots are aligned for all persons. After that, the group recognition can be performed normally. We assumed that after the interpolation, all the collected data used for group recognition has the period of f where $f < \min F_i, (i = 1, 2...n)$.

4. Aligned Interpolation of Trajectory

In the section, we introduce the aligned trajectory interpolating algorithm for synchronizing the collected data.

4.1. Aligned Trajectory Interpolation

Our interpolation is based on Catmull-Rom Spline[12]. The principle of the algorithm is based on four consecutive data (called control points) to interpolate data. Centripetal Catmull-Rom Spline holds the property that the control points are in the Bézier curve and also no cusp and intersection on the curve determined by the control points and interpolated data [12], which is consistent with the walking trajectory.

The interpolation process include two steps, which are illustrated as follows:

a) Add one datum (i.e., $2P_0 - P_1$) before the first collected data, and another one (i.e., $2P_n - P_{n-1}$) after the last collected data.

b) Go through the whole data set, get four data each time as control points and interpolate one datum between the middle two control points. The interpolation should be done at $t_n = t_0 + n{*}f$ $(0 \leq t_0 < f, n = 1, 2.....)$

The value of starting time t_0 affects the accuracy of the interpolation. We use a error function to measure the difference between the interpolated data and the real data. To define an effective error function is a challenging problem.

Empirically, we consider that interpolation error of one datum becomes larger when it is farther away from the control points. Based on this idea, we propose the error functions as follows:

$$g(t) = -10(1-t)^{10} + 10(1-t)^{9} \tag{1}$$

where $t \in [0, 1]$ represents the normalized distance of the interpolated data from the second control point to the third control point.

With the error functions being determined, given a t_0, we can compute the error of each interpolated datum and accumulate it to the total errors. By iterating all $t \in [0, f)$, we can get the optimized t_0. The detailed algorithm is illustrated in Algorithm 1. We assume that all the data of users are stored in *userData*, and t_0 is checked with a time unit *unitTime*. We first obtain all the interpolated data as in lines 4-6. Then for each interpolated data, we find the two control data (d_0 and d_1) that generates it, and compute the normalized distance of it from d_0 to d_1(lines 8-10). Using the error function, we can compute the error of this interpolated data, and accumulate it into *errors* (line 11). We iterate all possible values of t_0 in $[0, f)$ and set the value of it to the one that achieves the minimal errors (line 14-15).

Algorithm 1 Starting Time Selection Algorithm

Input: T, f, *unitTime*, *userData*
Output: t_0

```
1:  minErrors = ∞
2:  for t = 0; t < f ; t = t + unitTime do
3:      errors = 0
4:      for i = 0; i < T/f; i = i+1 do
5:          interpolatedData.add(t + i*f)
6:      end for
7:      for each uData ∈ userData do
8:          for each d ∈ interpolatedData do
9:              find the nearest two data in uData, say d0 and d1
10:             x = (d − d0)/(d1 − d0)
11:             errors = errors + g(x)
12:         end for
13:     end for
14:     minErrors= min(minErrors, errors)
15:     t0 is the t that achieves minErrors
16: end for
```

Algorithm 2 Color Assignment Algorithm

Input: *curGroups*, *preGroups*, *colors*, *delta*
Output: *colors*

```
retainGroups = ∅
for each g ∈ preGroups do
    if g ∉ curGroups then
        if g.lastseen > delta then
            colors[g.color] = 0
        else
            g.lastseen=g.lastseen+1
            retainGroups.append(g)
        end if
    else
        g.lastseen = 0
    end if
end for
for each g ∈ curGroups do
    if g.color = null then
        g.color=colors[i] (i = min_j colors[j] = 0)
        colors[i]=1
    end if
end for
curGroups = curGroups ⋃ retainGroups
```

4.2. Spatial-Temporal Clustering for Group Reconginition

We use Wirz's spatial-temporal flock detection algorithm [2] for group recognition. In each timestamp, the algorithm divides all the people into groups by DJ-Cluster [6], a density-based spatial clustering method, according to people's locations. After that, the deduced groups are compared with the ones in previous timestamps. If one group is sufficiently similar with another one in previous timestamps, they are regarded as the same group. The groups that does not exit in previous timestamps are regarded as new groups.

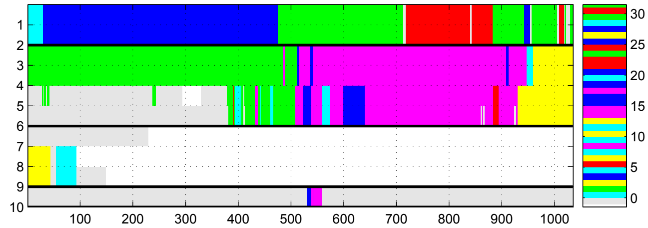

Fig. 3. Graphical display of groups. Blank space means no data, gray space means the person currently does not belong to any group.

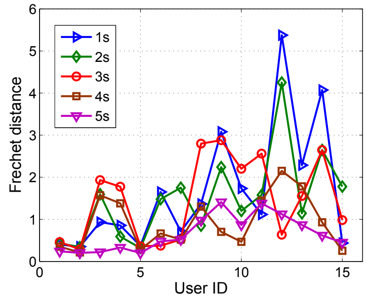

Fig. 4. Each user's Fréchet distance with different F

In this algorithm, the groups are determined by past and present trajectory data, which provides feasibility of real-time recognition of groups. We make a change to this algorithm, using DBSCAN [5] rather than DJ-Cluster in the clustering process. This because DBSCAN can provide higher computational speed for real-time processing.

4.3. Real-time Graphical Group Monitoring

In our system, the groups are monitored graphically in the real time. There are two requirements for that. First, distinctly display the groups at each timestamp. Second, show the dynamic changes of groups such as group continuing, group fusion, and group splitting in real time.

Similar with [13], we use a hot map to implement it and assign each group one kind of color. At each timestamp, people of the same group are displayed in the same color while people of different groups are displayed in different colors. Moreover, a group is displayed using the same color in different timestamps. Therefore, if a person always belongs to one group, there is a long bar in same color. If a group is split into two groups, a bar turns into two bars of different colors. Figure 3 shows an example of hot map.

Figure 2 is the system architecture diagram of our real-time group recognition system. First, there is a user interface from which the users can config parameters and clustering methods (e.g., DBSCAN or DJ-Cluster), and start the service to listen to remote data.

The underlying protocol of data receiving is TCP. Specific data format is as following:

a. Participating people ids: "id_1,id_2,...,id_n"

b. People's locations at each timestamp:

"#Timestamp;
id_1, location x, location y;
id_2, location x, location y;
......
id_n, location x, location y;#"

The received data are transmitted to the real-time group recognition module and color selection module. They compute a set of groups and the colors corresponding to them. Finally the groups are refreshed on graphics by the graphical monitoring module.

As the time goes by, new groups are constantly added in, and more colors are needed to distinguish them. On the other hand, some groups may disappear and their colors can be reused for later groups. Therefor, a proper color assignment approach is required in the real-time group monitoring.

The color assignment algorithm is illustrated in Algorithm 2. *curGroups* records the results of group recognition in current timestamp, and *preGroups* records those in the last timestamp. *colors* is a bit vector indicating whether a color is used, where each color is assigned a unique ID. *delta* is a time-tolerance parameter in the algorithm. From line 2 to 13, each group in *preGroups* is checked. If it is not recognized for a sufficient long time (*lastseen* is used to record the times not recognized, and compared with the threshold *delta*), it is regarded as disappeared, and its color is recycled (line 4-5). If it is not recognized but *lastseen* is less than *delta*, we just increase its *lastseen* by 1, and add it into the

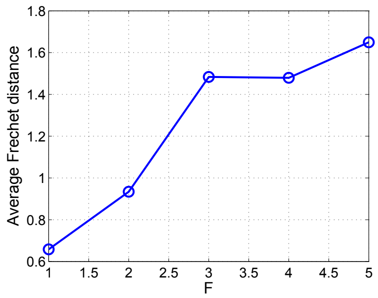

Fig. 5. The average Fréchet distance with different F

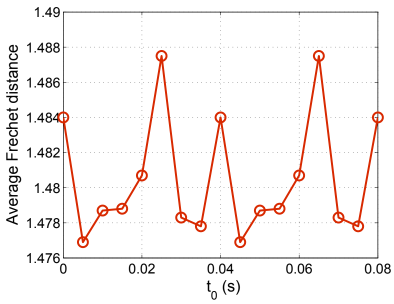

Fig. 6. The average Fréchet distance under different t_0

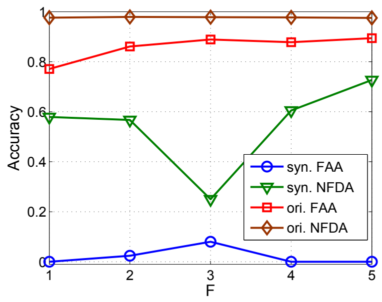

Fig. 7. FAA and NFDA of synchronized data and original data at different F

current groups (line 7-8, 20). If the group is also recognized in current timestamp, its *lastseen* is renewed to 0 (line 11). For any new group in *curGroups*(i.e., its color has not been determined), a new color is assigned for distinguish purpose. In this paper, we assign the first unassigned color to it (line 16-17), and more complex strategies can also be used.

5. Evaluation Results

We use the ATC (Asia and Pacific Trade Center) pedestrian group data set [14] to evaluate the effectiveness of the proposed approach. This data set was collected in a shopping center of Osaka, Japan using 3D-range sensors. It includes the ID of users in the shopping center and their 3D locations, speeds, directions of movement, and face orientations at different timestamps. We sampled the data from this data set with a period of F, and assumes this to be the collected sensing data for group recognition. The proposed approach is used to synchronize the collected sensing data. We mainly check the accuracy of synchronized data to the real data, and the accuracy of group recognition.

Fréchet distance [15, 16] is used to measure how accurate that the synchronized data match the real data. This measurement is based on dog-man distance measurement model where a person holds a dog by a rope in an arbitrary speed and the distance between them is measured by the length of the rope. It is commonly used to measure the distance of two curves.

We first check the Fréchet distance between the synchronized data and the real data with the change of F. Figure 4 shows the result of individuals persons, and Figure 5 shows the average value of them. It can be seen that as F increases, almost all users' Fréchet distances are increased. The Fréchet distance when $F = 1$s is less than that when $F = 5$s. The average Fréchet distance shows the similar trend. This is because sparser location data in the trajectory can be obtained at a larger F.

After that, we check the Fréchet distance between the synchronized data and the real data with different t_0. The result is shown in 6. F is set to 3s and f is set to 0.04s. It can be seen that Fréchet distance changes periodically with the change of t_0 and the period is f. Different values of t_0 leads to different Fréchet distances. The minimum value is achieved when t_0 equals 0.005, which matches the result of our approach.

Finally, we measure the effect of data synchronization on the group recognition. Following the literature [2], FAA(Flock Assignment Accuracy) and NFDA(Flock Assignment Accuracy) are used as the measurements. The former denotes the average F1-measure of all timestamps and the latter denotes the proportion of timestamps that the groups are correctly affiliated. When F changes from 1 to 5, the result is shown in Figure 7. First, it can be seen that both FAA and NFDA [2] of synchronized data are higher than the original ones. Second, FAA of synchronized data is quite stable and is up to 97.9%. This value is even higher than using the original ATC data set, which is 97.4%. This can be explained by that the interpolated trajectory is smoother than original trajectory by ruling out some location noises. FAA of synchronized data increases at first and then decreases a little later. When F is 1s, there are a lot of noises in the collected data. When F is 4s or 5s, a large portion of data are lost, causing difficulty to deduce the trajectory and the groups.

6. Conclusion

This paper proposes a data collection synchronization approach for real-time group recognition. It includes the aligned trajectory interpolation algorithm and real-time graphical group display algorithm. The aligned trajectory interpolation algorithm solves the problem of unsynchronized data by interpolating data periodically for all people from a starting time. We propose the error function for interpolation and compute the optimal starting time to minimize the errors between the interpolated data and the real data. The real-time graphical group display algorithm assigns proper colors to groups and display the dynamic changes of them in the real time. A real-life data set is used to validate the effectiveness of the proposed approach. The results show that 97.9% accuracy of group recognition can be achieved and the dynamic changes of groups is well shown.

Acknowledgements

This research is supported in part by National Natural Science Foundation of China No. 61502351, Natural Science Foundation of Hubei, China No. 2015CFB340, National Undergraduate Training Programs for Innovation and Entrepreneurship No.201610486094, Luojia Young Scholar Funds of Wuhan University No. 1503/600400001, and Chutian Scholars Program of Hubei, China.

References

[1] A. R. Mawson, "Understanding mass panic and other collective responses to threat and disaster," *Psychiatry: Interpersonal and Biological Processes*, vol. 68, no. 2, pp. 95–113, 2005.

[2] M. Wirz, P. Schläpfer, M. B. Kjærgaard, D. Roggen, S. Feese, and G. Tröster, "Towards an online detection of pedestrian flocks in urban canyons by smoothed spatio-temporal clustering of gps trajectories," in *Proc. of ACM SIGSPATIAL International Workshop on Location-based Social Networks*, 2011, pp. 17–24.

[3] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, , 2012, pp. 95–102.

[4] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, "Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones," in *Proc. of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 240–249.

[5] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A density based algorithm for discovering density varied clusters in large spatial databases," *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1–4, 2010.

[6] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personal gazetteers: an interactive clustering approach," in *Proc. of the 12th Annual ACM International Workshop on Geographic Information Systems*, 2004, pp. 266–273.

[7] P. Kalnis, N. Mamoulis, and S. Bakiras, "On discovering moving clusters in spatio-temporal data," in *Proc. of the 9th international conference on Advances in Spatial and Temporal Databases (SSTD)*, 2005, pp. 364–381.

[8] M. B. Kjærgaard, H. Blunck, M. Wirz, D. Roggen, and G. Tröster "Time-lag method for detecting following and leadership behavior of pedestrians from mobile sensing data," in *Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom),* 2013, pp. 56–64.

[9] S. Feese, B. Arnrich, G. Tröster, M. Burtscher, B. Meyer, and K. Jonas, "Sensing group proximity dynamics of firefighting teams using smartphones," in *Proc. of the International Symposium on Wearable Computers*, 2013, pp. 97–104.

[10] S. Feese, B. Arnrich, G. Troster, M. Burtscher, B. Meyer, and K. Jonas, "Coenofire: monitoring performance indicators of firefighters in real-world missions using smartphones," in *Proc. of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (Ubicomp)*, 2013, pp. 83–92.

[11] D. Gordon, M. Wirz, D. Roggen, G. Tröster, and M. Beigl, "Group affiliation detection using model divergence for wearable devices," in *Proc. of the ACM International Symposium on Wearable Computers*, 2014, pp. 19–26.

[12] C. Yuksel, S. Schaefer, and J. Keyser, "Parameterization and applications of catmull–rom curves," *Computer-Aided Design*, vol. 43, no. 7, pp. 747–755, 2011.

[13] S. Feese, B. Arnrich, G. Troster, M. Burtscher, B. Meyer, and K. Jonas, "Coenofire: Monitoring performance indicators of firefighters in real-world missions using smartphones," in *Proc. of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2013, pp. 83–92.

[14] D. Brscic, T. Kanda, T. Ikeda, and T. Miyashita, "Person tracking in large public spaces using 3-d range sensors," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 6, pp. 522–534, 2013.

[15] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, vol. 5, no. 01n02, pp. 75–91, 1995.

[16] K. Toohey and M. Duckham, "Trajectory similarity measures," *ACM SIGSPATIAL Special*, vol. 7, no. 1, pp. 43–50, 2015.