

学号 2013302580047

密级 公 开

武汉大学本科毕业论文

基于运动轨迹数据的室内人群识别

院（系）名 称：国际软件学院

专 业 名 称 ： 软件工程

学 生 姓 名 ： 许琳

指 导 教 师 ： 朱卫平 副教授

二〇一七年四月

BACHELOR'S DEGREE THESIS OF WUHAN UNIVERSITY

Indoor Human Flock Recognition Based on Motion Trajectory

College : International School of Software
Major : Software Engineering
Name : Lin Xu
Supervisor : Dr. Weiping Zhu

April 2017

郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名：_____

日期：_____

摘 要

在公共场合中,大部分人群都会以群组的形式行走活动,这些人群可能是家人、朋友、情侣、同事等等。随着人口数量的增加,公共场合中的人群管理是个非常重要的问题。群组内具有运动高度一致性,空间位置凝聚性,信息自传播性和自组织性等性质。紧急情况下人群疏散,团队小组工作控制,商场针对性营销都可以依靠对人群的分组,以组的单位进行。因为这样可以实现群组内信息流通和配合工作完成的最高效化。但是目前关于群组识别的研究还只是停留在研究阶段,本文通过对现有研究的学习,提出了并实现了针对群组识别的实时图形化监控算法和轨迹插值算法。

论文开始对于群组识别的相关研究做了比较,综合算法的性能和实现的可能性选择了基于轨迹数据的时空聚类算法。对于空间-时序聚类这个十分经典的方法和概念进行了详细介绍,并提出了自己对时空聚类算法的实时化改进。然后提出了图形化的需求分析,针对需求详细阐述了图形实时化颜色选取的算法。最后提出了基于 Catmull-Rom 的轨迹插值算法,并通过实验得出最优的插值算法的开始时刻偏移量。算法的 Frechet 误差度平均在 1.2 左右,采样插值后的轨迹得到了 97.8% 的空间-时序群组识别的准确度,说明了插值算法的有效性。

关键词: 群组; 密度聚类; 空间-时序群组识别; 实时图形化; 轨迹插值; 轨迹相似度

ABSTRACT

Most people in public tend to walk in group, which is composed by family, friends, couples and colleagues. As the rapid increasing of population, the management of crowd in public has raised great attention, such as crowd evacuation in shopping mall, group work like fireman's needing team coordination, large group of people advertising. All these activities need an effective division of groups due to that group possesses properties of consistency, cohesion, self-propagation and self-organization which contribute to most efficient information flow and work completion of coordination. Unfortunately, current research to this topic has been still only stay in the research stage. Our paper proposed the implementation of real time graphical group and trajectory interpolation algorithms to make the group recognition more executable and visible

First, the paper did some comparison of current research method of group recognition. With both algorithm efficiency and possibility to realize considered we chose Spatial-Temporal clustering method. The paper has explained Spatial-Temporal clustering carefully and proposed my improvement on this to realize real-time recognition. Then we presented graphical group method by introducing requirements and corresponding color choosing algorithm. In the end we elaborate the trajectory interpolation algorithm based on Catmull-Rom and calculated the optimal starting time offset parameter of this interpolation algorithm. According to analysis of this algorithm, the deviation on average is about 1.2 and spatial-temporal group recognition accuracy is about 97.8% comparing to original data, which explains the efficiency of our interpolation algorithm.

Key words: Flock, Density Clustering, Spatial-Temporal Group Recognition, Trajectory Interpolation, Trajectory Similarity.

目 录

第一章 绪论	1
1.1 研究背景与研究意义	1
1.1.1 研究背景	1
1.1.2 研究意义	2
1.2 群组识别技术在国内外研究现状	3
1.2.1.国外研究现状	3
1.2.2.国内研究现状	4
1.3 论文结构	5
第二章 时空聚类算法	6
2.1 “群组”概念	6
2.2 密度聚类算法	7
2.2.1 密度聚类的基本概念	7
2.2.2 DBSCAN 和 DJ-CLuster	7
2.3 实时时序群组识别	8
2.3.1 基本概念	9
2.3.2 实时时序群组识别算法	9
第三章 群组实时图形监控	12
3.1 群组图形化功能分析	12
3.2 颜色选择算法	13
3.3 实时图形化的实现	15
第四章 运动轨迹插值算法	17
4.1 Catmull-Rom 样条算法	17
4.2 对齐轨迹插值	18
4.2.1 不对齐数据特征	18
4.2.2 对齐插值算法	19
4.3 对齐轨迹插值算法评估	20
4.3.1 对齐插值算法误差模型	20

4.3.2 参数选取影响	21
第五章 实验过程与结果分析	24
5.1 数据集	24
5.2 空间-时序群组识别结果	24
5.2.1 群组识别结果	24
5.2.2 图形化处理结果	25
5.2.3 群组识别算法准确度	27
5.3 插值算法结果	28
第六章 结语	30
6.1 结论	30
6.2 问题与展望	30
参考文献	32
致谢	35

第一章 绪论

本章主要介绍群组识别的研究背景和研究意义，国内外研究现状，还有本项目的总体结构。

1.1 研究背景与研究意义

本课题关于行走人群群组识别的研究背景和研究意义如下。

1.1.1 研究背景

在公共场合中，人群的群组化是一个非常普遍的现象，例如家人，朋友，同事，情侣等在商场中一起进行的消费活动，还有一些需要分组合作的任务等。据调查显示，约有 74% 的人会结成群组进行社会活动。同时，在不同的环境下约有 50% 到 70% 的人是以群组的形式在进行随意的走动^[4]。随着人口数量的增加，公共场合中的人群管理是个非常重要的问题。有调查^[24]显示，人们在危险情况下更倾向依附于群组。特别是在疏散过程中，人们愿意和熟悉的人成组而不是单独行动。因此在此种情况下，组内社会关系（家人，朋友）约束较强而且合作意识占主导地位，不存在自私不合作等行为。所以如果能够通过监测每个人的运动情况将人群进行分组，并针对特定的组发送特定的引导信息而不是所有人的都接受到一样的引导信息，这样就减少了组内分歧以及人群的盲目行动从而加速了疏散。这是因为群组内具有一致性，凝聚性，自传播性和自组织性等性质，能够实现信息流通和配合工作完成的最高效化。所以以群组化的形式研究人群的行为有着重要的意义。

此外，随着移动智能设备技术的发展，在城市中几乎所有人都有具备定位功能的移动智能设备。智能手机的应用而且越来越多的商业活动都已经依赖智能手机定位的功能进行，例如阿里公司推出的喵街 APP，它的商场内导航功能就是典型的例子。而且国内外关于室内定位的研究也一直在深入，精度也在不断提高。因为智能设备使用的普遍性和定位技术的发展，所以通过人群运动轨迹进行群组识别是完全可行的。

尽管如此，当前不管是商业活动，社会活动，生产作业活动还是紧急疏散等几乎都没有考虑到空间中人群群组化这样一个特点。造成这个结果的原因是由于人群中的群组识别技术上还存在着很多的不完善之处，包括识别的设备成本，技术难

度以及算法的健壮性和实用程度等等。例如，定位轨迹数据通常是使用手机等智能设备进行的采集的。这种采集的数据存在两个问题，一方面，每个加入群组识别范围内的人员开始时间不同，在频率一致的情况下，不同人员的数据存在时刻上不对齐的情况；另一方面考虑到用电和流量的因素，数据采集的频率存在限制。所以如何对齐不同人员的数据时刻和轨迹还原都是亟待解决的问题。又如，群组识别更多还是处于理论阶段，如何将群组识别的结果实用化也是十分重要的研究。

所以本毕业设计的主要内容是设计一个基于室内人员运动轨迹数据的群组识别算法，分析人员的相似性从而对商场中消费人群进行分组。在现有研究的基础上提高群组识别的准确度。解决轨迹群组识别中轨迹不对齐和低频率的问题，并将群组识别的结果实用化。

1.1.2 研究意义

从消费层面上来看，识别出正确的群组，商家可以针对已识别的群组进行有针对性的促销。人是社会性的动物，单独的个人也许会因为他们是社会角色一部分进行托管式的群组行为，这种消费会因为群组的吸引或者群组成员之间的相似性受到影响。例如，作为一个球队的球迷，他们会因为同球队人的行为一同购买门票等^[8]。譬如在看比赛时，同一个球队的人群会做出相似的行为，通过识别他们为一个群组之后，将商家的促销宣传缩小到群组中影响力较大的一个人，从而大大减少商家的宣传开销。从日常生活方面来说，如果能够准确地识别出室内运动人员的群组，并能够利用图形化形式实时展现分组变化的情况，对于需要小组活动协调的场景，譬如文献[9]监控分析消防小组在救火过程中的合作行为，这种对群组变化的监控更有利于团队指挥，提高小组的合作能力，在更短的时间内完成任务。而针对于公共场合中紧急情况下的大量人群疏散问题。而对于有较多小分组的大量人群，例如导游带领游客以及商场中紧急情况地疏散工作。由于各个小分组存在行为的相似性和内部的凝聚性和自组织性，识别出各个小分组能更有效地针对小分组采取措施，从而有效处理紧急事件和更加方便管理。

所以本项目的实现，有利于生产生活各方面的人群的群组性行为分析，在人群疏散还有群组性的行为预测等方面有着重要意义。此外，在今后的对于群组内人员交互行为，性质，影响力等的研究以及群组间关系研究方面都有着一定的意义。

1.2 群组识别技术在国内外研究现状

人群或者群组并不是一个十分新的词汇。由于人的社会性，公共场合中的人群的运动总会呈现群组化。所以近年来，国内外关于群组识别这个话题的研究也十分的火热。

1.2.1. 国外研究现状

群组识别使用到的数据形式非常多。第一种是基于视频流数据进行的识别。基于视频流的一般处理步骤是先处理原始视频的每一帧得到轨迹数据（即位置数据，运动方向等），然后根据位置和运动的方向以及衍生的速度这几种数据进行的亲密度计算从而分组，[13]基于 PCA 提出了一种监督式的学习算法来确定运动人群的是否存在亲密关系，[14]使用视频流数据得到的人员运动行为来评估每个人与其他人互动的意图。视频数据的好处在与能够相对准确地得到人群的位置、人的动作以及人与人之间地交互情况，但是由于视频中存在人员重叠，光线等问题会使识别困难。另外考虑到当前公共场合中监控的分布具体情况，很多地方都会是盲区，如果使用视频流数据无法得到完整的运动路径。另一种则是基于传感器数据等的识别。Sebastian 提出了使用智能手机中的 ANT radio 服务和气压计来判断消防员之间的距离和当前所在的楼层，从而对正在灭火任务中的消防队进行任务分组^[9]。ANT radio 可以根据人员之间的距离进行分组，但是不能知道当前人群的具体的方位，这样对于人群疏散几乎没有任何意义。Dawud 提出了基于 DBAD 的群组识别算法，通过对加速度传感器和方向传感器数据使用高斯混合模型以及冯米塞斯混合模型进行拟合并计算参数的 jeffery divergence 来判断人与人之间运动的相似性，从而进行分组^[12]。此方法使用的是佩戴在腿上加速度和方向传感器数据，在实际生活中并不适合在人员身上佩戴传感器。如果是使用手机的传感器数据，手机的摆放的任意性都会影响到传感器数据，同一组内人员的数据完全会呈现不同的状态。另外，此方法依赖严格的群组内运动的一致性，而正常情况下即使是同一群组内的人员也不一定是完全一致的运动，适用的场景有限。Brscic^[22]设计了一个基于三维测距传感器的人员位置，运动方向以及高度跟踪技术，可以得到较准确的人员运动轨迹，这为基于运动轨迹的群组奠定了基础。Wirz 提出了“群组”这样一个时空概念^[15]，他使用 DJ-cluster 进行聚类避免了类似于磁盘式的聚类，同时还借用了 Kalnis^[11]在论文中提出的基于空间和时间的聚类方法对人的运动轨迹做了时空上“群组”识

别。此后，由 Kjærsgaard 团队做出了一系列针对行“群组”识别的研究^[16-18]。首先他们做了基于 wifi 定位数据的室内“群组”识别，紧接着研究了基于混合传感器群组识别的模型，并对各种混合的检测准确度进行了评估，最后研究了一种检测人群的跟从和影响力的时延算法。他们这一系列研究得到的群组识别准确度达到 85%。这一系列基于运动轨迹的群组识别的研究的前提都是在每一个时刻数据都是对齐的，因此在每个时刻都有所有人员的空间位置的快照。这样才可以使用空间聚类算法，再在时序上进行“群组”识别。

1.2.2.国内研究现状

而从国内的研究来看，目前关于群组识别类的研究相对较少，更多的是关于人群的定位跟踪以及人群密度的监测等等。针对本毕业设计的题目来说，国内的相关中文研究包括人群运动轨迹的识别跟踪^[1]，提高室内定位精度的研究^[3,5]等一些可以用于群组识别中的算法改进。[5]使用了基于 LMS 修正算法的定位数据修正，平均误差为 0.83m，这相对于以前的 2-3m 的定位误差取得了巨大的进步。具体到群组识别这一块，[7]做了关于运动人群的特征的研究，包括一个群组内人的数量的分布，运动速度和运动速度差异的分布，运动方向和运动方向差异的分布，总结了行人群组的运动特征，为群组性运动数据的模拟仿真提供了理论基础。而清华大学的研究^[19]则是在 Kjærsgaard^[16-18]的基础上做出了一个基于 wifi 定位，陀螺仪加速度传感器识别群组追踪群组和检测群组事件的系统。通过实际的实验数据，得到群组识别的准确度平均为 77%-85%，检测群组事件（例如群组的融合形成，离开等事件）的准确率达到 85%-96%，这是群组识别实用化的一个重要表现。Yang^[26]设计了基于增加性的轨迹数据的实时群组识别算法。使用的轨迹数据包括了位置，速度，方向及加速度。通过相关系数判断人员之间两两的相似性，使用滑动窗口的方式实现几乎实时的群组识别。这与 Kjærsgaard 的区别在于，首先 Kjærsgaard 可以每个时刻进行一次群组识别而 Yang 的方法还是需要窗口化的数据，群组识别程序至少需要一个窗口长度的数据，因为仅仅一个位置数据没有办法计算相关系数；再者，Kjærsgaard 是使用前后两个时刻的组的相似性确定组是否在时序上延续，而 Yang 是每次刷新不同人员之间的相似度。

通过对国内外研究现状的总结，背景中提出的基于运动轨迹的人群识别的几

个问题依然存在，本文将根据 Kjærsgaard 团队提出的空间-时序”群组“算法进行群组识别算法的改进和实用化应用。

1.3 论文结构

第一章是论文的绪论部分。主要介绍论文的研究背景和研究意义，以及关于群组识别的国内外发展现状。

第二章是对本文的中提到的空间-时序群组识别算法进行详细的阐述，并且介绍改进后的实时群组识别算法。

第三章描述了实时图形化监控的需求分析，实时图形化中用到的算法步骤的介绍以及实时图形化监控实现的方法

第四章描述了基于 Catmull-Rom 的轨迹插值算法，一方面阐述 Catmull-Rom 算法的原理，另一方面描述对齐的轨迹插值算法以及参数选取对算法性能的影响。

第五章介绍了论文的数据集，实验过程，以及各个实验的结果分析。

第六章是论文的结论以及展望。

第二章 时空聚类算法

本章主要详细介绍时空聚类算法的内容。

2.1 “群组”概念

为了更好的介绍时空聚类算法。首先我们以数学的方式系统化的描述一下群组这样一个概念。这个概念最早源自 Kalnis^[11]提出的运动的组，此后由 Kjærsgaard^[16-18]团队做出了更具体的定义和描述。下面就介绍一下群组的概念。

假设空间中有这样一组运动的实体的集合， $E = \{e_1, e_2, \dots, e_n\}$ 。每个实体都有一个运动轨迹 $T_i = \{T_i^1, T_i^2, \dots, T_i^m\}$ ，轨迹中每个点的数据是这样一个时空的元组 $T_i^k = (t_k, L_i^{t_k})$ 。其中 t_k 是当前点的时间，而 $L_i^{t_k}$ 是在 t_k 时间的位置信息。此外，如果假设这些实体中有同时在 t_k 时刻有位置信息的，我们就有了这样一个快照的概念^[11]。 $S(t_k) = \{e_i \in E, L_i^{t_k} \neq \emptyset\}$ ，即在 t_k 时刻实体以及实体位置信息的一个集合。对于每个这样的快照，我们可以用空间聚类的方法得到一个组的划分 $\Gamma(t_k) = \{C_1^{t_k}, C_2^{t_k}, \dots, C_m^{t_k}\}$ 。

这样我们可以认为运动的组的概念是

定义 1：假设 $G = \{C_\alpha^{t_1}, C_\alpha^{t_2}, \dots, C_\alpha^{t_p}\}$ 是一列时序上组的集合，其中 $i(1 \leq i \leq p)$ ， $t_{i+1} - t_i \leq \delta_t$ ， $\alpha \in \Gamma(t_i)$ ，同时连续的两个组满足 $f(C_\alpha^{t_i}, C_\alpha^{t_{i+1}}) \geq \theta, 0 \leq \theta \leq 1$ ， f 是两个聚类相似性度量的函数，这样 G 就是一个运动的组。

由这样一个运动组的概念得到一个群组的概念：

定义 2：一个群组是一个存在时间满 $t \geq \tau$ 的运动的组，其中组的人数要满足 $n \geq v$ 。

如果要得出这样一个群组，对于每个实体对应的运动轨迹有个要求，即要求在大多数时刻下每个实体的运动轨迹点要求是对齐的，这样才可以得到更多的快照并得到更多的组的划分，从而在时序上对这些划分的连续性进行判断，但是在日常生活中我们的数据往往不是完全对应的，不同实体加入位置采集的时刻并不相同，在相同的采样周期下，开始时间稍有偏移就会导致之后的数据一直都不对齐，这也是我们将要解决的问题，在之后的章节里会详细介绍。

2.2 密度聚类算法

聚类是指将相似的对象分成一组的一个过程，空间聚类则是聚类是基于空间位置数据的聚类。聚类算法多种多样，划分型的算法包括 k-means, 层次聚类等等，这类算法将数据分为 k (k 通常提前确定) 类以确保离中心点的总误差最小，还有基于密度的聚类算法，包括 DJ-cluster 和 DBSCAN, 这类算法将分布密集的区域分为一类，与划分型算法的比较，密度聚类可以识别出任意形状的聚类，部分低密度区域的值通常会被认为是噪声点。人群行走存在随意性，群组的形状并不固定，所以本文对于空间聚类使用了密度聚类的算法。

2.2.1 密度聚类的基本概念

ϵ -邻域：聚类点 P 的距离小于 ϵ 的点的集合

minPts 密集邻域：包含至少 minPts 个点的 ϵ -邻域

在使用聚类方法之前首先要了解这样两个概念。

密度聚类目前有两种算法，DBSCAN 和 DJ-Cluster，下面就针对两方面做一些比较。

2.2.2 DBSCAN 和 DJ-Cluster

DBSCAN (Density-based Spatial Clustering of Applications with Noise) 是由 Ester^[25] 在 1996 年提出的基于密度聚类的算法。算法将 minPts 密集邻域这样一个中心的点称为核心点，任何一个对象只有包括在核心点的 ϵ -邻域才能属于一个组，核心点之间互为 ϵ -邻域或者具有连通性则成为一个组，如图 2.1 右图所示。属于多个核心点 ϵ -邻域的非核心点必须属于其中一个核心点的组，非核心点也不属于任何核心点 ϵ -邻域的点被认为是噪声点。 算法的流程：

1. 选取一个未标记的点 p 计算 ϵ -邻域，若为 minPts 密集邻域，以 p 为中心形成一个组，并将 p 邻域中所有未标记点加入 seeds 集合，反之为噪声点
2. 从 seeds 集合中选取点 q 计算 ϵ -邻域，若为 minPts 密集邻域，将其 ϵ -邻域未标记点加入 seeds 集合，q 从 seeds 集合删除，重复此过程直到 seeds 集合为空
3. 重复 1-2 直到所有点都被标记。

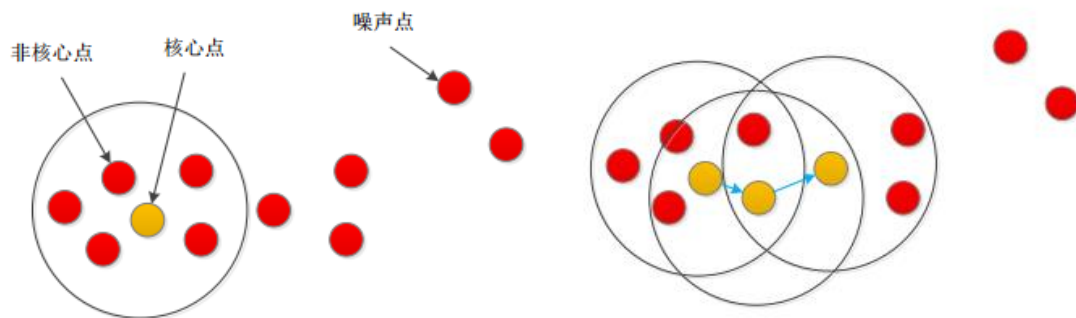


图 2.1 DBSCAN 过程演示图

DJ-Cluster^[27] (Density-Joinable Cluster) 是 DBSCAN 的另一种变体, 此种方法提出了一个概念叫密度可连接, 其实比较简单, 两个 ϵ -邻域如果有公共的元素则认为其密度可连接。循环选取一个点计算其 ϵ -邻域, 如果满足 minPts 密集邻域则成为一个组, 每次将当前的组与其所有密度可连接的组合并成一个新的组, 如此迭代进行得到最终的分组。

DJ-Cluster 与 DBSCAN 相比较在于 DJ-Cluster 的组的扩张是一个迭代性的过程, 而 DBSCAN 组的扩张更像是一个递归过程。因此在使用过程中, DBSCAN 由于需要存储大量的 seeds 点, 在一次组的扩张中需要更多的内存, 而 DJ-Cluster 则需要与点个数相等的迭代过程, 并且组的合并无论使用什么数据结构都是一个比较耗时的过程。在相对点数较少的计算中 DBSCAN 使用空间减少了时间的消耗, 所以在运行速度上 DBSCAN 的优势更明显。对于大量的数据, DBSCAN 可能对于内存的消耗可能会非常巨大, 但是针对于我们室内人群的识别, 并希望可以实时化地得到分组地结果, 使用 DBSCAN 可能更高效。Kjærsgaard 团队使用的 DJ-Cluster 空间 u 类算法, 在这里针对我毕业设计的初衷, 我将使用 DBSCAN。下面就是我们空间群组划分算法。

2.3 实时时序群组识别

在 2.1 中我们提到群组这个概念, 所谓群组是指在时间上有连续性的组。这个概念与空间上的分组不同, 时序上的群组是建立在空间的分组之上的。为了通过人的运动轨迹数据得到群组, 我采用了来自 Kjærsgaard 团队^[15]于 2011 年根据 Kalnis^[11]提出的时序聚类的改进的群组检测算法原理。在此基础上将算法进行了实时化的

修改。这个团队研究了室外运动行人在街道上运动情况，并通过收集 GPS 数据计算行人的分组以及组的变化状态，他们的算法假定了所有的运动行人的在 GPS 的收集上是周期性的并且是同步的，以确保能够在时刻上对齐，得到每个时刻的人员位置分布快照，并能更好的应用时序群组识别算法。

2.3.1 基本概念

在我们先介绍时序算法本身之前，有以下几个概念：

组的连续，在前面介绍群组概念的同时我们提到了运动的组的概念，要求 $f(C_{\alpha}^{t_i}, C_{\alpha}^{t_{i+1}}) \geq \theta, 0 \leq \theta \leq 1$, 此处我们使用的 f 称为 Jaccard Similarity, 即两个集合的交集个数除以集合的并集个数。

$$f(C_{\alpha}^{t_i}, C_{\alpha}^{t_{i+1}}) = \frac{|C_{\alpha}^{t_i} \cap C_{\alpha}^{t_{i+1}}|}{|C_{\alpha}^{t_i} \cup C_{\alpha}^{t_{i+1}}|}$$

活跃的群组：到目前时刻已经连续存在了超过时间 τ 的群组

潜在的群组：到目前时刻已经连续存在的时间小于 τ 的群组

解散的群组：一个之前存在时间至少 τ 但是在目前时刻已经不存在群组

2.3.2 实时时序群组识别算法

现在我们介绍下实时的群组检测算法，实时化有个特点，因为数据的不连续性，我们需要根据之前的算法的结果来得到新到来数据的结果。实时时序群组识别算法的流程如图 2.2 所示。我们对每个时刻人员的位置数据进行空间聚类得到分组后，结合上一时刻的群组识别情况，判断群组是否存在延续，解散和新生成的情况，从而得到当前时刻的群组集合。这个集合同样会用于下一时刻的群组识别。

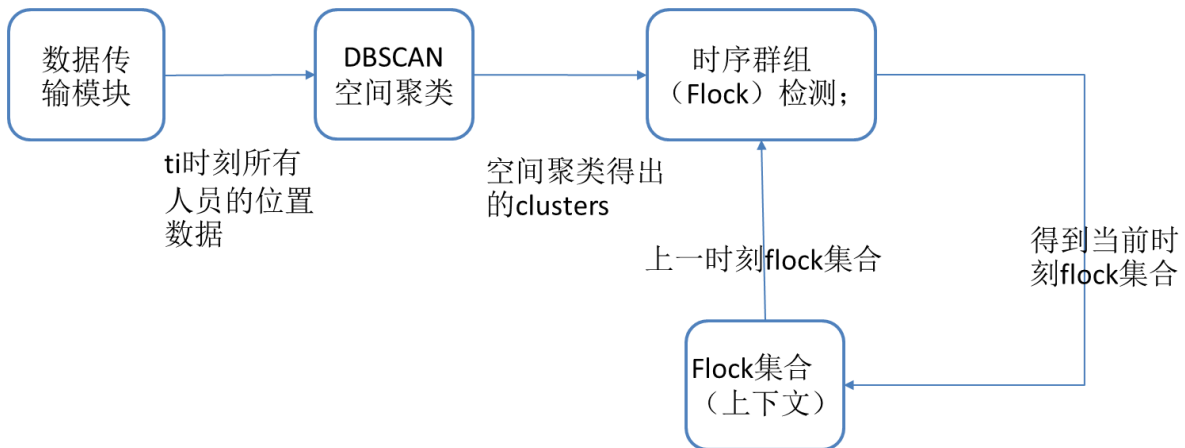


图 2.2 实时时序群组识别算法流程图

实时时序群组识别算法的详细步骤如表 2.1 所示。

为了实现实时数据的连续，每次调用此算法，都需要上次调用后得到的活跃群组，潜在群组和解散群组，以及上次已经用到的 flock 的 id。Flock 本身是一个类，它的属性包括，id，延续时间 duration，以及自从上次没有延续的时间 lastseen。这些是就像是衔接上下文的现场保存。

算法内部的内容包括：

1. 首先根据现场数据判断是否上一步的三种群组是否还满足要求，具体来说如果活跃群组没有延续的时间即 lastseen 超过提前预定好的值 delta 则认为其已经解散，而对于潜在的群组如果有群组延续时间超过 gama 则认为其已经是活跃群组，如果潜在群组延续时间没超过 gama 但是未延续时间已经超过 delta，则将其从潜在群组中剔除。
2. 然后使用 DBSCAN 算法计算当前时刻快照的空间聚类分组。循环将分组中的每个小组与现有群组比较计算 Jaccard Similarity，若其是现有群组的延续，即存在相似度大于 θ 的群组，则将其该分组的值赋给相似度最大的群组，反之，直接将其设为一个新的群组，并赋予一个 id，id 从开始是不断递增。
3. 为了更好的表现群组的划分和群组动态变化的一个过程，我们需要记录每个时刻的一个群组的划分，这个记录最好的方法是记录当前实验人员所属的一个群组的 id。所以 draw_array 是记录当前传输过来数据的每个时刻每个人对应的群组 id，是一个时间长度乘以人员个数乘以 1 的三维矩阵。在最后图形化的展示的时候，我们会将上次的群组的分组情况与当前计算出来的群组情况进行合并，确保在图形上上下文的连贯性。
4. 当然函数 rtFlockDetect 函数的参数的都是这函数的返回值，进行现场的一个保留。

表 2.1 实时时序群组检测算法

算法 1 实时群组检测
rtFlockDetect(gama, delta, pointMatrix, act_flocks, pot_flocks, dis_flocks, curid): for all t in timestamps do #draw_array 记录当前传输过来数据每个时刻的群组分组情况

```

draw_array := array(len(timestamp),len(users))
# 判断当前存在群组的状态
for pot_flock in pot_flocks do
    if pot_flock.duration()>= gama then
        act_flocks := act_flocks+pot_flock
        pot_flocks := pot_flocks\pot_flock
    end if
    if pot_flock.lastseen()>=delta then
        pot_flocks := pot_flocks\pot_flock
    end if
end for
for act_flock in act_flocks do:
    if act_flock.lastseen()>=delta then
        act_flocks := act_flocks\act_flock
        dis_flocks := dis_flocks+act_flock
    end for
clusters := DBSCAN(t,eps,minPts,pointMatrix)
for cluster in clusters do
    计算 cluster 与 act_flocks 和 pot_flocks 中所有 flock 的相似
    度, 如果 cluster 与所有 flock 不满足大于 delta 的相似度, 则新
    生成一个 flock 并给其指定 id , curid += 1,id= curid; 反之
    取相似度最大的进行群组的延续, 延续工作包括将可延续的 flock
    的 duration 加 1, 此外, 对于没有延续的群组 lastseen 加 1
end for
for flock in act_flocks+pot_flocks do
    将当前时刻的每个用户所属 flock 的 id 记录在 draw_array 数组
    中, 其中未属于任何 flock 的用户则为-1, 而数据丢失的用户则为
    -2
end for
end for

```

第三章 群组实时图形监控

所谓群组实时图形监控就是能够通过图形化的方式对识别出的群组进行实时展示。由第二章的内容可知，通过我们的一个空间-时间的群组识别算法可以实时地得到过去已有数据和当前的群组分组情况，并且可以得到这些组在时间上的延续情况。这为我们对分组结果进行实时性地图形化展示提供了重要地基础。实时化有这样的一个要求。新的数据的加入不会影响过去的结果。这个要求对于我们的实时时空算法来说十分符合，因为实时的群组识别算法的结果只受当前数据以及过去已有结果的影响，而现在的结果并不会影响过去的结果。正如绪论中所说，将分组的结果图形化展示更方便工作人员对群组进行监控从而实时做出判断或者指挥工作。这对于我们真正将人群群组化这样一个性质用到实处。所以实时性的图像化展示是一个十分有意义的尝试。接下来我就介绍以下如何进行实时图形化。

3.1 群组图形化功能分析

在实现实时图形化之前，很有必要分析一下这种图形化展示的一些功能要求。

第一，从组的划分来说，为了更形象的展示出当前的分组情况，图形的展示在同一时刻需要用明显不同的标识表示不同的组，目前我们知道表示组的划分的方法可能包括用不同颜色方框或者圈将群组圈出来，或者给每个人员一个颜色标记，从而确定分组。

第二，从时间上来说，我们一方面要将图形实时表现出来，确保图形是根据数据的更新在变化，另一方面，我们还需要记录过去一段时间的分组情况，这样我们才会知道在时间上群组划分的一个变化情况。例如什么时候群组分开了，什么时候哪几个群组融合到一起以及这些人在一起走了多久。这种要求仅仅使用在人群分布图上用圈圈出分组的方法就不够了。

针对以上分析的两种功能需求，并结合我们在第二章的实时群组识别算法。每个人员在每个时刻都有一个分组的群组的序号，没有的就为-1，-2 等。这样我们得到的是一个三维的矩阵，这让我想到一个十分相似的数据结构，那就是一幅图像的 RGB 值。不过我们这里的横坐标是时间而纵坐标是人员 id，而这一对坐标对应的数据是他们当前所属的群组 id。所以我们可以把这个三维的矩阵画成一幅类似于热度图的图像。如果群组在时间上有很强的连续性，我们就可以看到这个热度图

是一个个连续的长条。

这样一方面可以在一个时刻区分分组的情况，另一方面还可以观察这个群组在时间上的连续情况和群组的分裂和融合等变化。如果涉及到画热度图，颜色选取不可避免。下面就讲一下颜色选择算法。

3.2 颜色选择算法

与实时空间-时序群组识别的要求相同。

- 1) 图像在实时展示的时候不能因为新的数据的加入而发生变化。这里的要求就是不能因为新的数据加入而改变每个之前每个数值对应的颜色。
- 2) 另外随着时间的推移，新的群组不断加入，旧的群组离开，群组的 id 会不断增大，即热度图的值域不断增大。如果使用热度图常见的渐变的颜色，id 数值差距不大的不同组的颜色区别会非常小。而且在当前情况下，往往 id 数值差距不大的组会在同一时间段出现，呈现在页面上的就是因此我们必须设计一个能够重复利用颜色的方法。

下面就介绍一下颜色选择算法：

因为群组的进入和离开，所以在一段时间之后某个 id 的群组会不再出现，这为有限的颜色重复利用提供了可能。首先需要有一个颜色指定算法，我们有一个有限的颜色集合，里面的颜色相对差别较大。另外，每次新数据进来的时候都有 assigned 这个集合表示已经被使用的颜色，再取新的颜色需要在未被使用的集合里面随机选取一种颜色。

表 3.1 颜色指定算法

算法 2 颜色指定算法
<pre>assignColor(assigned): notAssign := colors \ assigned return Random(notAssign)</pre>

接下来就是颜色选取的总算法，每次计算记录最大的群组的 id 的数值。通过遍历当前计算的 draw_array 的值，对比上次的最大值大的 id 进行新的颜色的赋值。为了对颜色重复利用，我们需要对不需要的颜色进行回收。为了与群组识别算法相

对应，这里同样使用了 `delta` 的值。如果颜色 `id` 超过 `delta` 时间没有出现，那么其对应的颜色就可以被重新回收。下面是颜色选取算法的步骤伪代码。

表 3.2 颜色选取算法

算法 3 颜色选取算法
<pre>chooseColor(draw_array,lastflockids,assigned,maxid): newcmap := [] for ti in draw_array.width do #对新出现的 id 赋予新的颜色 for ui in draw_array.height do x = draw_array[ti][ui] if x > newmaxid then newmaxid = x end if flockids.append(x) end for for id in flockids do if id > maxid then newcmap[id] := assiggColor(assigned) end if end for #如果 id 超过 delta 时间没有出现，则该 id 对应的颜色被释放可以重新利用 for id in lastflockids do: if id not in flockids then if id.lastseen > delta then assigned[id] release else id.lastseen += 1 end if else id.lastseen = 0 lastflockids.append(id) end if end for</pre>

```
end for
```

3.3 实时图形监控的实现

实时图形监控展示有十分多的方法和工具，最常见的莫过于网页的图表展示，类似与股票涨势图等实时动态的图形不胜枚举。目前网页能够展示的图形也不仅限于基础的图标像折线，柱状，饼状图等，热度图也可以展示在网页上。例如 Highchart 这个基于 JavaScript 的图标库，但是对于实现我们需要的动态的热度图，实现方面有一定的难度。为了方便数据控制，我使用的是 python 的一个画图的库 matplotlib。这个库可以方便的绘出网页上可以绘出的图。这里主要用的函数就是 pcolormesh() 这个函数。Pcolormesh 函数的作用是画出一个 2d 的数组的值。

每次新的数据加入的时候，我们将前一部分数据与新的数据合并，包括颜色的选取。每个 flock 的 id 对应一个颜色（由于重复利用，颜色会有重复，但是在同一段时间里不会有同一颜色出现，因为同一种颜色第二次出现时前面使用这个颜色的群组已经离开）。整个绘图过程中，每次新的数据传输过来时，对数据进行群组识别，并分配颜色，这样的新的图会接到已有的图后面，当达到一定长度之后图会像左移动，图形表现出滑动的形式。图 3.1 是实时图形监控系统的模块图。首先是有用户操作界面，界面负责选择参数和开启服务。界面图如图 3.2 所示。底层数据监控实现了基于 TCP 的数据传输监听端口，接收远程特定格式的数据。接收数据格式为：

(1) 参与人员 ID 序列：“%id1, id2, id3, ……, idn%”

(2) 每个时刻所有人员的位置数据：

“#时间戳；

人员 id1, 位置 x, 位置 y；

人员 id2, 位置 x, 位置 y；

… … …

人员 idn, 位置 x, 位置 y；#”

接收到的数据为字节符类型，而且以上两种形式的数据会在传输中连接到一起，因此在数据 TCP 服务器端还实现了基于对于这种数据解析。

接着数据会传输到实时群组识别模块，结合上个时刻的识别结果，得到当前时刻的群组识别结果。然后根据当前结果更新颜色列表。最后将当前识别结果和颜色列表传送到用户界面，用来刷新当前的图形。从而可以实现图形的实时化展示。

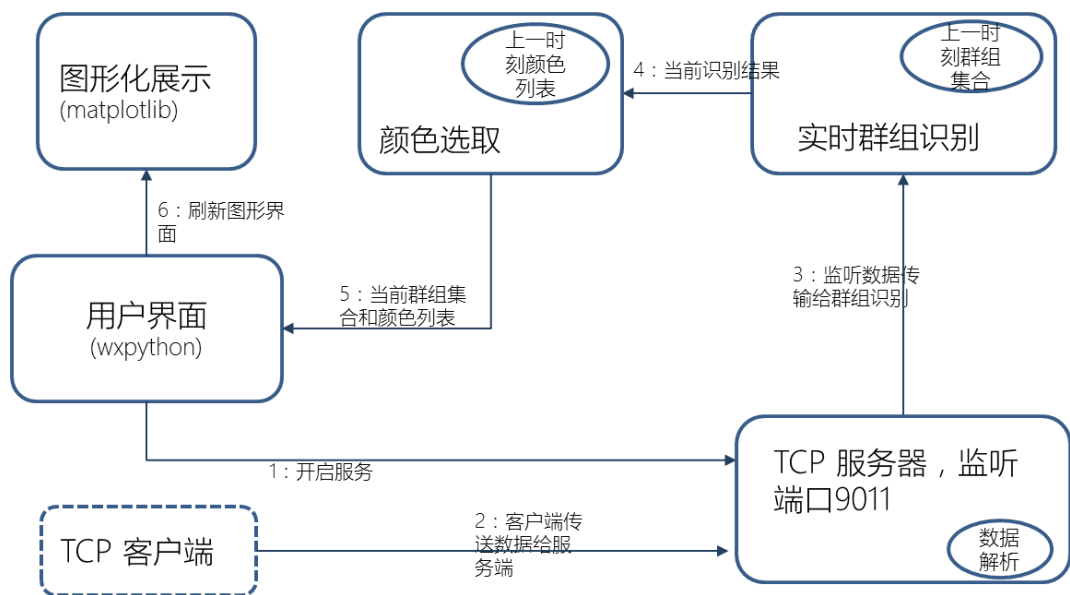


图 3.1 实时图形监控系统模块图



图 3.2 实时图形监控操作界面

第四章 运动轨迹插值算法

正如在前面多次提起的一样，空间-时序的群组识别算法有一个苛刻的要求就是所有参与人员的采集数据在时刻上是对齐的，这样才能在每个时刻都得到位置数据快照。但是在实际生活中，对于人的运动轨迹的采集往往开始时间都是不同的，而且如果人的轨迹开始时间不是相差一个采样周期的话，所有采集的数据都不会是同步对齐的。除此之外，考虑到耗电量和设备性能的因素，采集的频率不会特别高，这对群组识别的准确影响比较大。当然还有一方面就是室内的定位方法到目前为止，除非使用特殊设备，定位误差都比较大，因为 wifi 信号分布的不均，不同地点的精度也不同。如果将低频率的点（即采集周期比较大的点）或者在定位中选取精度较高的点作为样点，能够使用样本点根据人的运动特点对人的运动轨迹进行精准地还原。这些问题就会迎刃而解。

本章就会提到一种轨迹插值的算法，并提出了怎样解决数据不对齐的方法。

4.1 Catmull-Rom 样条算法

就像我们生活中根据几个点勾勒出一条曲线一样，Catmull-Rom 样条算法就是这样一个能够将这条曲线勾勒得平滑的方法。在游戏开发中此算法常用来产生人物的一个行走的路径，同样对于机器人仿真人的行走，这种方法也运用的十分广泛。Catmull-Rom 算法是在 1974 年提出的综合了拉格朗日插值和 B-Spline 算法的一种插值算法，此后学者将此算法分解因子得到如图 4.1 的金字塔型的计算方法，近年来也由不少研究者对其进行了改进和参数选取探讨^[21]。首先介绍一下 Catmull-Rom 样条算法^[21]本身。

算法基本原理是根据 4 个控制点得到中间两个点之间的估计点的坐标。它得出的是一条 Bezier 曲线，它的定义大致如下：

$P_i = [x_i, y_i]^T$ 表示一个点，对于点 P_0, P_1, P_2, P_3 定义的一条曲线 C （具体来说是点的序列），和其对应的结点序列 t_0, t_1, t_2, t_3 ，假设 t 为 t_1 和 t_2 之间的一个时刻的序列， $t = \{t^0, t^1, \dots, t^n\}, t_1 < t^k < t_2, k = 0, 1, \dots, n$ ，则 Catmull-Rom 生成曲线的过程如下：

$$C = \frac{t_2 - t}{t_2 - t_1} B_1 + \frac{t - t_1}{t_2 - t_1} B_2$$

$$\begin{aligned}
B_1 &= \frac{t_2 - t}{t_2 - t_0} A_1 + \frac{t - t_0}{t_2 - t_0} A_2 \\
B_2 &= \frac{t_3 - t}{t_3 - t_1} A_2 + \frac{t - t_1}{t_3 - t_1} A_3 \\
A_1 &= \frac{t_1 - t}{t_1 - t_0} P_0 + \frac{t - t_0}{t_1 - t_0} P_1 \\
A_2 &= \frac{t_2 - t}{t_2 - t_1} P_1 + \frac{t - t_1}{t_2 - t_1} P_2 \\
A_3 &= \frac{t_3 - t}{t_3 - t_2} P_2 + \frac{t - t_2}{t_3 - t_2} P_3
\end{aligned}$$

这里

$$t_{i+1} = \left[\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \right]^\alpha + t_i$$

当我们令 $t=t_1$ 和 $t=t_2$ 时我们得到, P_1, P_2 , 这说明这个曲线确保了控制点在曲线上。此外, 文献[21]使用数学方法证明了只有当参数 α 等于 0.5 时, 曲线不会形成回路, 不会有尖点也不会自己交叉^[21]。大多数情况下人的正常行走轨迹不会存在回路和交叉等, 所以使用我们使用 $\alpha=0.5$, 以确保在任意两个样本点之间是平滑的曲线。

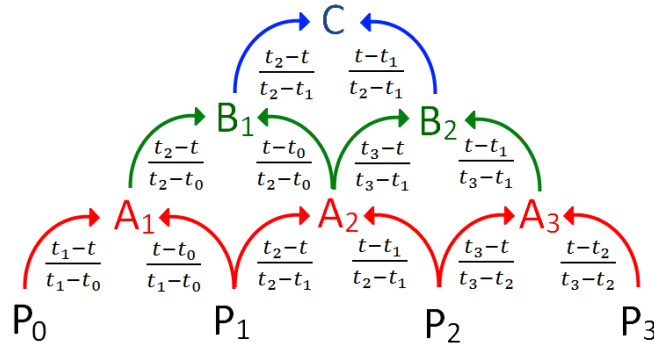


图 4.1 Catmull-Rom 算法原理

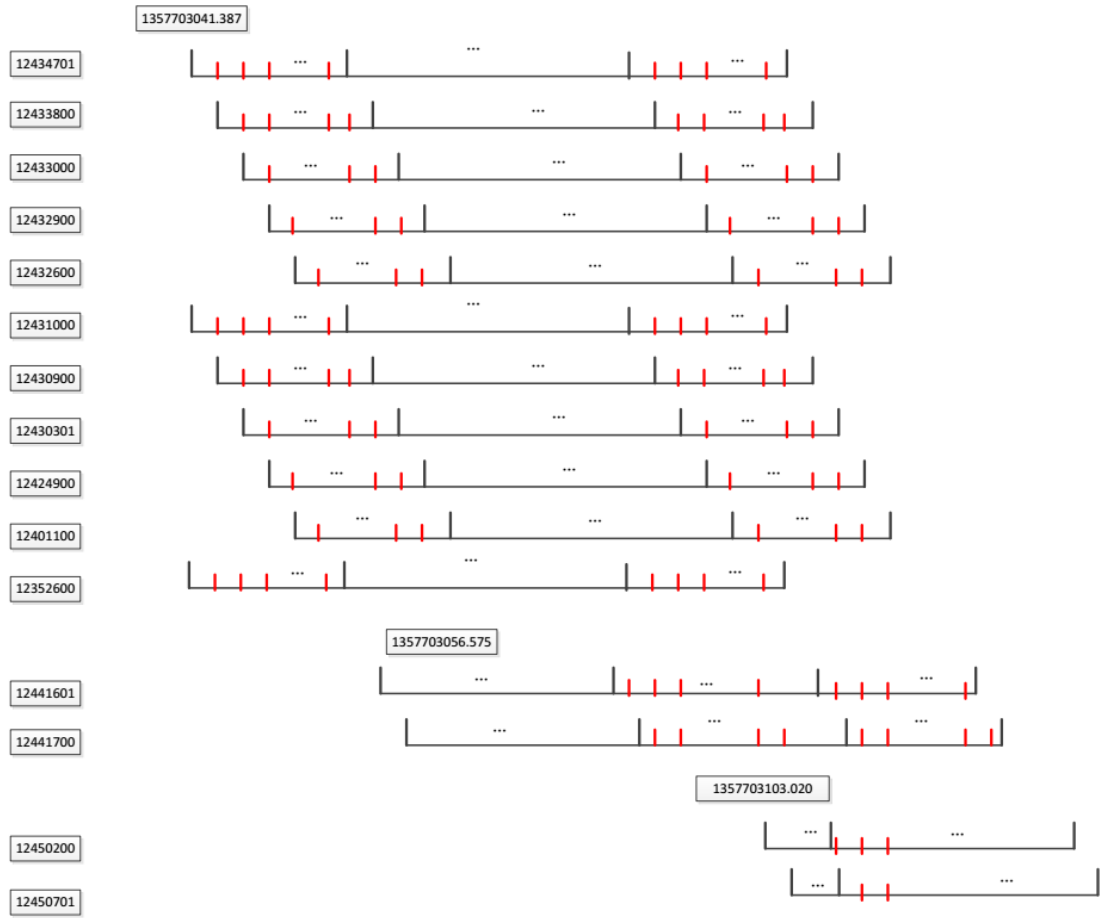
4.2 对齐轨迹插值

因为实际生活中的数据采集时间不对齐的原因, 导致空间-时序群组识别无法使用, 所以接下来介绍是如何对不对齐轨迹数据进行插值的。

4.2.1 不对齐数据特征

针对开始时间不对齐的数据, 如图 4.2 所示, 左边表示人员的 id 号, 每行表示

不同人员数据对应时刻的特征。其中线段上的黑色较长断点为采样时刻点，红色较短的断点表示插值的时刻点。此外，第 1 条线段的起始点时间为 1357703041.387，后面的 10 位人员起始时间分别以第 1 条往后偏移一个 0.04s。然后 12441601 对应的开始时间为 1357703056.575，后面一位偏移 0.04s。12450200 开始时间为 1357703103.020，后面一位偏移 0.04s。图中 15 位人员的数据的采集时刻从开始就不对齐，由于采样周期相同，之后的采样点一直不会对齐。为了解决这个问题，我们可以使用对其进行插值，使原本采样周期下不对齐的数据周期变小，从而实现数



据的对齐，如图中红色断点所示，所有用户的红色断点在时刻上一致对齐。

图 4.2 人员不对齐数据特征

4.2.2 对齐插值算法

假设我们的数据采样的周期为 F ，而插值后的数据要求的周期为 f 。算法的解释如下：

由于算法每次需要选取四个点，但是每次只会插入中间两个点之间，为了不丢失开始值和结束值，为原始数据首尾拼接两个点，这两个点的值为 $[2P_0-P_1]$ ， $[2P_n-P_{n-1}]$ ，这样插值后的曲线会通过所有样本点。

滑动式选取四个点向中间两个点使用 Catmull-Rom 算法按 f 的间隔插值。

原始数据样点不同人员在时间上存在错位。假设我们按照间隔 f 为每个插值点分配时刻，如果两个人员样点的时间差是 f 的整数倍，那么插值后的数据两者数据在时刻上就会对齐。而如果两个人员样点的时间差不是 f 的整数倍，插值后时刻仍旧不是对齐的。譬如 $[1,2,3]$ ， $[1.3,2.3,3.3]$ 两个样点时刻序列，如果插值的周期是 0.1，插值得到，

$[1,1.1,1.2,1.3,1.4,\dots,2.3,2.4,\dots,2.9,3]$ ， $[1.3,1.4,\dots,2.3,2.4,\dots,3,3.1,3.2,3.3]$

数据存在对齐部分，如果插值周期 0.4，插值后得到，

$[1,1.4,1.8,2.2,2.6,3.0]$ ， $[1.3,1.7,2.1,2.5,2.9,3.3]$

数据完全没有对齐时刻，所以插值算法需要考虑到这方面的问题，算法设计的是选取一个开始参考时间 t_0 ， $0 \leq t_0 < f$ 。然后以开始时间为准，后面的插值的时刻全部是 $t_0 + n * f$ 得到（这种计算方法说明了为什么 $0 \leq t_0 < f$ ），对于原始数据点的时刻保留。因此，如果使用 1 作为开始的参考时间 t_0 ，上面的例子 0.4 的插值时刻结果就是

$[1,1.4,1.8,2.2,2.6,3.0]$ ， $[1.3,1.4,1.8,2.2,2.6,3.0,3.3]$

这样时刻就可以对齐了。

4.3 对齐轨迹插值算法评估

虽然实现了一个插值算法，我们可以得到插值的结果，但是如何评估这个算法的有效性才是更重要的部分。算法中设计到几个部分的参数设置。第一，假设插值的最终周期均为 f ，插值前的 F 对于插值可能有重要的影响。第二，开始时间 t_0 的设置。

4.3.1 对齐插值算法误差模型

经过插值后，原始样点成为了一条运动轨迹，如何去判断这条轨迹的误差是评判算法性能的重要部分。由于得到的轨迹是一列在时序上离散的点，我们可以将这个序列看成一条曲线，一个有效的计算曲线相似性的方法是 Fréchet 距离^[23]。这个

理论起源于狗-人距离测量模型。一个人和一只狗由一条绳子牵引，分别同时从起点走到终点，在此过程中人和狗的运动速度任意，但是运动轨迹事先定好，而最后两条轨迹之间的距离就是所需的最短的绳子的长度。

Fréchet 距离的计算，简单的来说就是对两条曲线进行采样，得到其采样区间的内两条曲线同一时刻上对应距离的最大值，Fréchet 距离则是使最大距离最小化采样方法下的值。

因为插值算法本身比较复杂，难以从数学角度对算法进行分析，所以为了计算差值算法的误差，我从原本高频率的数据集中按一定频率(周期为 F) 采样得到原始控制点序列，然后使用插值算法进行轨迹的插值还原，得到与原始数据的频率近似相等的序列，然后使用 Fréchet 距离计算插值后的数据与原始数据的差距，以此判断误差。

4.3.2 参数选取影响

插值必然会受到原始数据周期和插值精度的影响，首先我们需要研究的是随着样本数据的周期的增大，即样本数据的减少对于插值结果的影响。从一般经验来看，原始值减少，必然意味着插值误差的增大，因此我也对不同采样周期的样本数据进行实验得到结果，其中 $t_0 = 0$ ，即为数据集开始的最早时间。结果如图 4.3 所示：

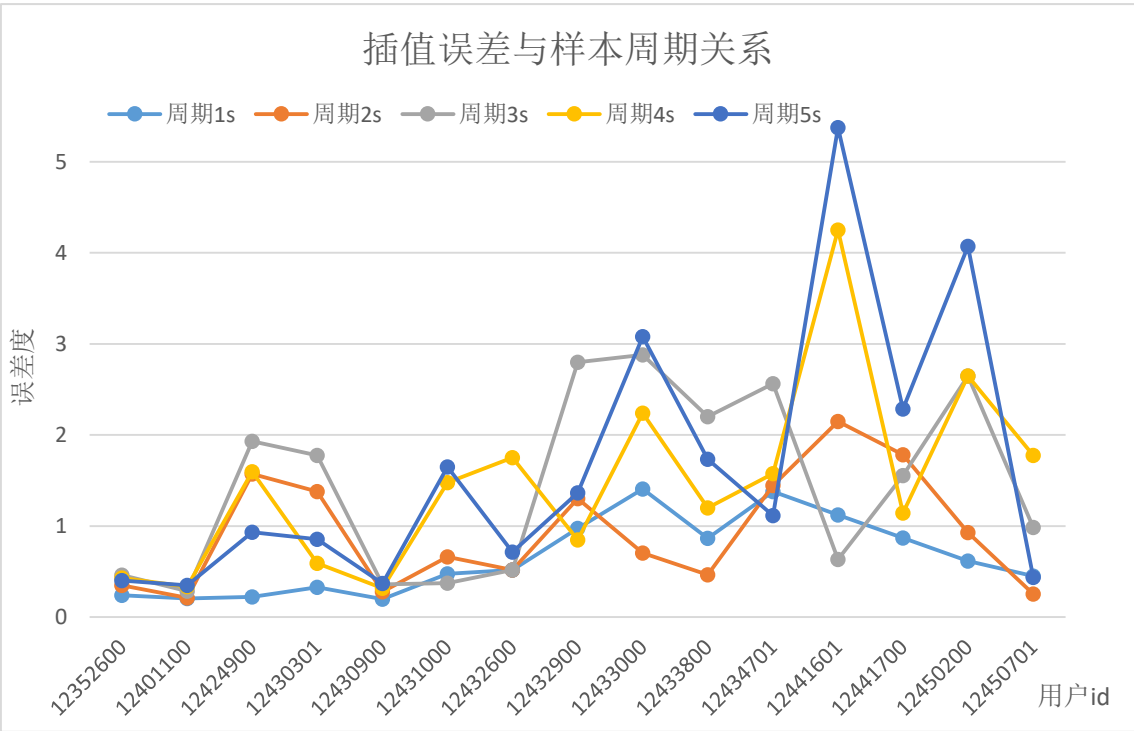


图 4.3 插值误差与周期关系

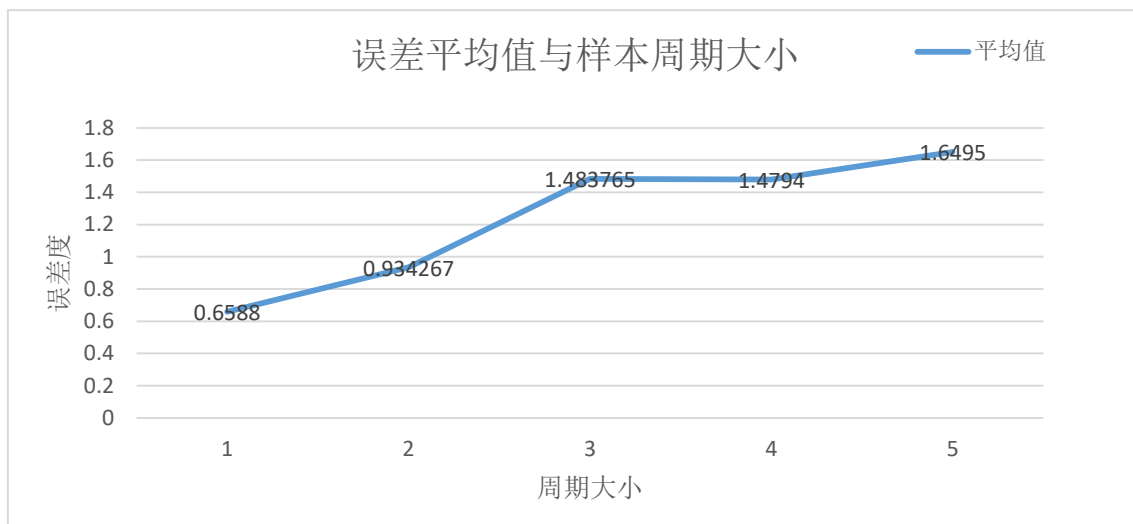


图 4.4 插值平均误差与周期关系

图 4.3 表示的是参与实验的所有人员在不同采样周期下插值后的结果与原数据的误差度, 五条折线分别表示不同周期 1s, 2s, 3s, 4s, 5s。图 4.4 中每个采样周期对应的误差值对应的是综合了所有参与人员在不同周期下的误差值的平均。由图像可知, 随着样本数据的周期的增大, 插值结果与原数据的误差度逐渐增大, 这与预猜想的结果一致。

插值算法选择使用了一个参数 t_0 , 这个参数是用来决定所有插值时刻的, 其中 $0 \leq t_0 < f$ 。不同的 t_0 会对插值后数据产生一定的影响, 为了研究这个参数对插值结果的影响, 同样我们需要通过实验的方式来计算。前面我们研究了不同周期下, 插值结果的误差度, 有上图可以看出, 3-4s 之间误差度处于一个较为平稳的状态, 所以此处我们控制采样周期 F 的大小为 3s, 然后计算不同开始时间下误差度的变化。由于我们插值的 $f = 0.04$, 因此 t_0 的取值范围为 $[0, 0.04]$, 步长为 0.005. 计算结果如图 4.5 所示。

图 4.6 更清晰地表现了误差度随开始时刻 t_0 变化的趋势。由图可以看出, 与预期相同, t_0 的取值会使误差度呈周期性变化。另外, 数据在 $0 \leq t_0 < f$ 区间内, 误差度存在一定的起伏, 变化呈 \cup 状, 先减小再增大, 在 0.01-0.02 和 0.035-0.04 能够取到较小的误差度值。

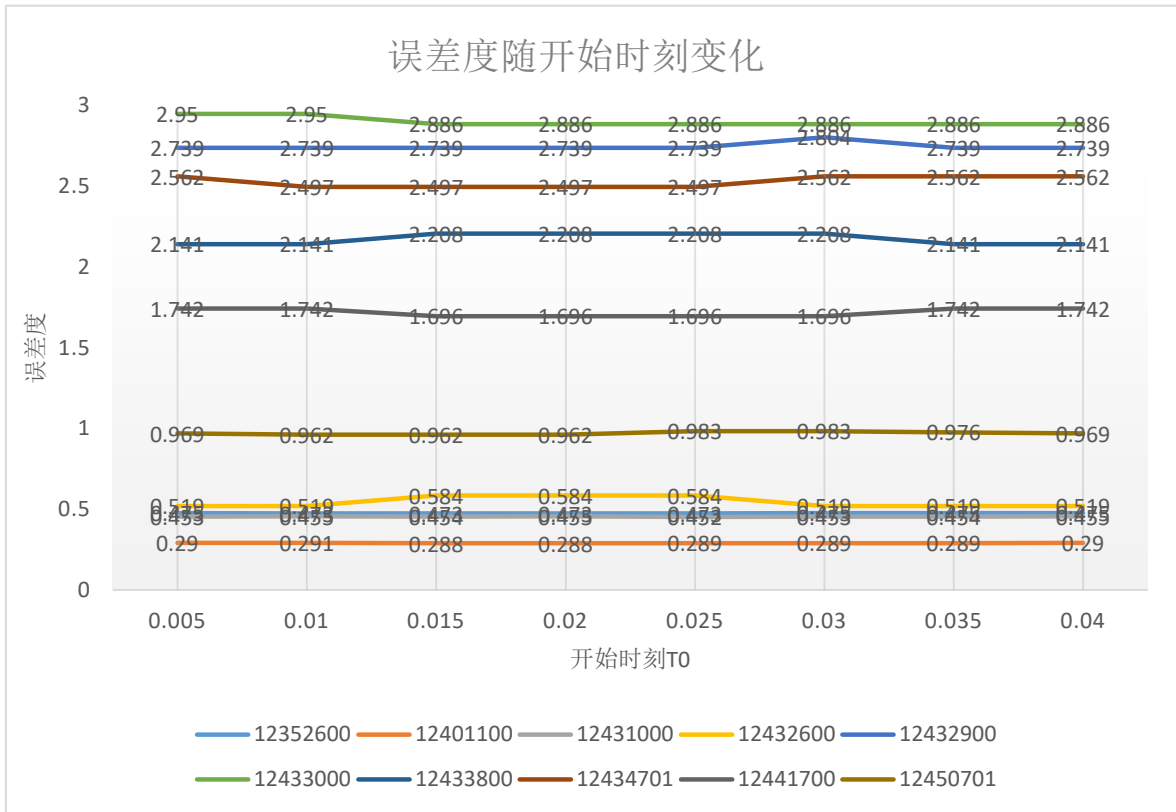


图 4.5 误差度随开始时刻变化

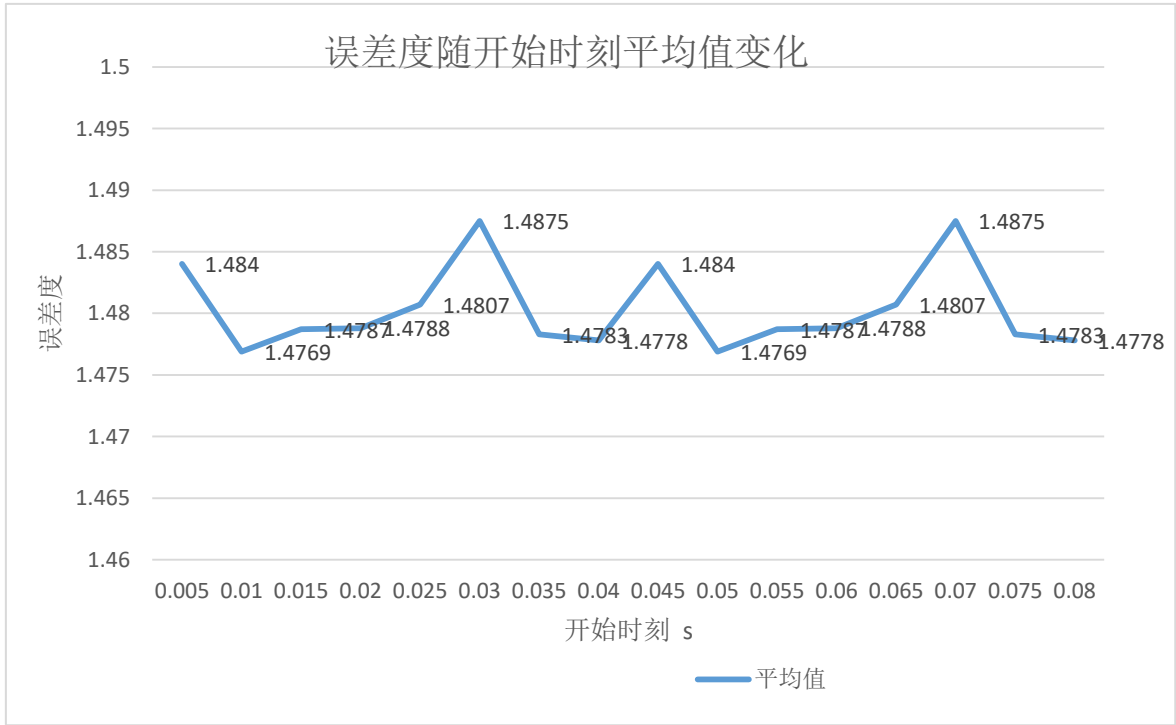


图 4.6 误差度随开始时刻平均值变化

第五章 实验过程与结果分析

本章主要介绍了实验使用的数据集，空间-时序群组识别的结果以及轨迹插值算法结果的群组识别准确度。

5.1 数据集

为了测试算法的运行结果，这里我们使用了 ATC^[22] (*Asia and Pacific Trade Center*) 的行人群组数据集。这个数据集在日本大阪的购物商业中心采集，使用的 3D 距离传感器。此数据集采集过程持续了 6 天，每天 4 个时间段，每段一个小时。得到的人的运动轨迹数据格式是：

[时间戳, 人员 id, 位置 x, 位置 y, 位置 z, 速度, 运动方向, 面部朝向]

此外每个对于所有采集范围内的人员做了 id 标识，并且对所有参与人员的分组和组内的交互情况做了记录。因此，此数据集非常适合用于空间-时序聚类算法的验证。

5.2 空间-时序群组识别结果

关于空间-时序群组识别结果分为三个部分，分别为群组识别结果数据形式，群组识别结果实时图形化展示和群组识别结果准确度。

5.2.1 群组识别结果

对于每个时刻的位置数据，我用了一个类 Position 来表示。Position 的属性包括原数据的所有栏。通过运行空间-时序群组识别的算法，得到的结果如图 5.1，得到每个时刻 flock 的结果。

其中第一栏为时间戳，剩下的栏，举个例子 [0 3 0 (12434701, 12432900)]，第一个 0 表示 flock 的 id，3 表示 flock 延续的时刻数。第二个 0 表示自从上一次延续有多久了，0 表示在时间上一直延续，如果为 1 表示已经有 1 个时刻没有延续了，如果超过 2 个时刻没有延续则这个 flock 被删除，因为当前程序设置的 delta 值是 2。我们设置 delta 值的目的是如果存在数据丢失或者一两个时刻的误差能够保证群组的连续，但是如果群组的确从此时刻离开了，或者群组从此时刻变化了分组，的确不属于再连续情况，那么在这两个时刻的就相当于延迟了群组的变化。

化和解散。所以 δ 的选值不能过大，这样可能会导致某些群组的变化被延迟。

```

1357703041.387000, [0 1 0 (12434701,12432900)], [1 1 0 (12433800,12430301)], [2 1 0 (12431000,12424900)]
1357703041.427000, [0 2 0 (12434701,12432900)], [1 2 0 (12433800,12430301)], [2 2 0 (12431000,12424900)]
1357703041.468000, [0 3 0 (12434701,12432900)], [1 3 0 (12433800,12430301)], [2 3 0 (12431000,12424900)]
1357703041.509000, [0 4 0 (12434701,12432900)], [1 4 0 (12433800,12430301)], [2 4 0 (12431000,12424900)]
1357703041.553000, [0 5 0 (12434701,12432900)], [1 5 0 (12433800,12430301)], [2 5 0 (12431000,12424900)]
1357703041.591000, [0 6 0 (12434701,12432900)], [1 6 0 (12433800,12430301)], [2 6 0 (12431000,12424900)]
1357703041.629000, [0 7 0 (12434701,12432900)], [1 7 0 (12433800,12430301)], [2 7 0 (12431000,12424900)]
1357703041.670000, [0 8 0 (12434701,12432900)], [1 8 0 (12433800,12430301)], [2 8 0 (12431000,12424900)]
1357703041.710000, [0 9 0 (12434701,12432900)], [1 9 0 (12433800,12430301)], [2 9 0 (12431000,12424900)]
1357703041.752000, [0 10 0 (12434701,12432900)], [1 10 0 (12433800,12430301)], [2 10 0 (12431000,12424900)]
1357703041.794000, [0 11 0 (12434701,12432900)], [1 11 0 (12433800,12430301)], [2 11 0 (12431000,12424900)]
1357703041.835000, [0 12 0 (12434701,12432900)], [1 12 0 (12433800,12430301)], [2 12 0 (12431000,12424900)]
1357703041.878000, [0 13 0 (12434701,12432900)], [1 13 0 (12433800,12430301)], [2 13 0 (12431000,12424900)]
1357703041.918000, [0 14 0 (12434701,12432900)], [1 14 0 (12433800,12430301)], [2 14 0 (12431000,12424900)]
1357703041.960000, [0 15 0 (12434701,12432900)], [1 15 0 (12433800,12430301)], [2 15 0 (12431000,12424900)]
1357703042.001000, [0 16 0 (12434701,12432900)], [1 16 0 (12433800,12430301)], [2 16 0 (12431000,12424900)]
1357703042.046000, [0 17 0 (12434701,12432900)], [1 17 0 (12433800,12430301)], [2 17 0 (12431000,12424900)]
1357703042.086000, [0 18 0 (12434701,12432900)], [1 18 0 (12433800,12430301)], [2 18 0 (12431000,12424900)]
1357703042.128000, [0 19 0 (12434701,12432900)], [1 19 0 (12433800,12430301)], [2 19 0 (12431000,12424900)]
1357703042.173000, [0 20 0 (12434701,12432900)], [1 20 0 (12433800,12430301)], [2 20 0 (12431000,12424900)]
1357703042.213000, [0 21 0 (12434701,12432900)], [1 21 0 (12433800,12430301)], [2 21 0 (12431000,12424900)]
1357703042.253000, [0 22 0 (12434701,12432900)], [1 22 0 (12433800,12430301)], [2 22 0 (12431000,12424900)]
1357703042.295000, [0 23 0 (12434701,12432900)], [1 23 0 (12433800,12430301)], [2 23 0 (12431000,12424900)]
1357703042.333000, [0 24 0 (12434701,12432900)], [1 24 0 (12433800,12430301)], [2 24 0 (12431000,12424900)]
1357703042.375000, [0 25 0 (12434701,12432900)], [1 25 0 (12433800,12430301)], [2 25 0 (12431000,12424900)]
1357703042.416000, [0 26 0 (12434701,12432900)], [1 26 0 (12433800,12430301)], [2 26 0 (12431000,12424900)]
1357703042.454000, [0 27 0 (12434701,12432900)], [1 27 0 (12433800,12430301)], [2 27 0 (12431000,12424900)]
1357703042.491000, [0 27 1 (12434701,12432900)], [1 28 0 (12433800,12430301)], [2 28 0 (12431000,12424900)]
1357703042.531000, [0 27 2 (12434701,12432900)], [1 29 0 (12433800,12430301)], [2 29 0 (12431000,12424900)]
1357703042.568000, [1 30 0 (12433800,12432600,12430301)], [2 30 0 (12431000,12424900)]
1357703042.610000, [1 31 0 (12433800,12430301)], [2 31 0 (12431000,12424900)], [3 1 0 (12434701,12432900)]
1357703042.650000, [1 32 0 (12433800,12430301)], [2 32 0 (12431000,12424900)], [3 2 0 (12434701,12432900)]
1357703042.689000, [1 33 0 (12433800,12430301)], [2 33 0 (12431000,12424900)], [3 3 0 (12434701,12432900)]
1357703042.725000, [1 34 0 (12433800,12430301)], [2 34 0 (12431000,12424900)], [3 4 0 (12434701,12432900)]
1357703042.762000, [1 35 0 (12433800,12430301)], [2 35 0 (12431000,12424900)], [3 5 0 (12434701,12432900)]
1357703042.805000, [1 36 0 (12433800,12430301)], [2 36 0 (12431000,12424900)], [3 6 0 (12434701,12432900)]
1357703042.843000, [1 37 0 (12433800,12430301)], [2 37 0 (12431000,12424900)], [3 7 0 (12434701,12432900)]

```

图 5.1 群组识别结果

5.2.2 图形化处理结果

通过这样的群组识别结果，再转换我们可以得到所有时刻数据每个人员 id 属于的群组 id ，注意当某个 id 因为 δ 的延续既属于活跃的群组也属于潜在的群组的时，我保留的是它所属的那个活跃的群组的 id ，这体现了我们使用 δ 放宽群组延续条件的思想。选取 ATC 数据集当中的一段数据运行后得到的结果如图 5.2 所示，其中 $\epsilon = 1.17$ ：

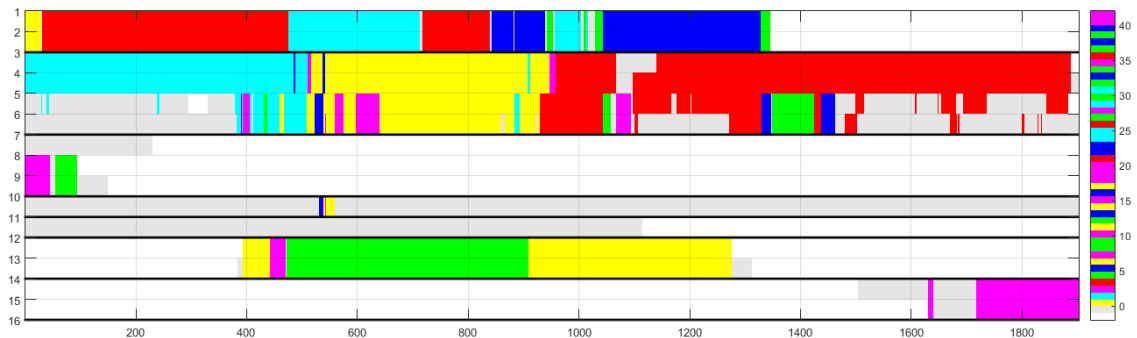


图 5.2 测试图形化处理结果

如图 5.2，其中白色区域标识该时刻该人员的数据丢失，浅灰色区域表示该时刻该人员并不属于任何群组。为了方面展示分析，图中的黑色直线表示的是分组的一个标记值。我们很清楚的了解到分组情况是[1, 2], [3, 4, 5, 6], [7, 8, 9], [12, 13], [14, 15]。其中 10, 11 很明显除了在 560 左右的时刻 10 被分入了[3, 4, 5, 6]的分组，其他时候都是单独成组，这符合最初的分组情况。此外[7, 8, 9]数据较少，其中 7 与 8, 9 可能距离比较大，在当前的阈值情况下一直被分为单独的一个组，在时刻 100 左右就离开了监控区域。与[7, 8, 9]相似，[14, 15]是最后加入的监控区域，大部分时候 14, 15 被分到了一起，部分时候两者都无分组，而[12, 13]也是大部分被分到了一起，但是在 910-940 的时候被分到了[3, 4, 5, 6]一组。观察[1, 2]和[3, 4, 5, 6]这两个分组，开始 0-480 左右，1 和 2 分组正常，3, 4, 5, 6 开始 5, 6 无分组，而到了 390-480 之间 3, 4, 5, 6 时而分成一组，时而分为 3, 4 和 5, 6 两组。再往后一点从 480-500 左右，这六个人被分为了一组，之后又分开。值得一提的是 1 和 2 的图形上出现了多次的颜色变化，其中由红色变成浅蓝色是因为被分到了[3, 4, 5, 6]的一组，而其他的颜色的变化的原因都是因为 1, 2 在经过了 delta 时候没有延续后，再次出现的时候群组的 id 发生了变化。我们可以看到中间没有延续的时刻图形是浅灰色的，所以表明在这个时候 1, 2 没有分组导致了群组没有延续。

上面使用的是一张静态图片，包含 1900 个时刻的数据，这张图片几乎揽括了所有的图形变化形式。接下来是一个动态展示图的截图，eps 的值取为 3.72。我们可以看到，图形上与图 5.2 相似，横轴的范围是 500，开始可以看到图像在不停地增长后面就可以发现是呈一个向左滑动的状态。此外，热度图的 colorbar 也随着数据的加入而不断改变，但是只是改变的最上面的值，下面的值即过去的群组 id 对应的颜色并不会改变，这样确保了我们已有图形的颜色不会改变。

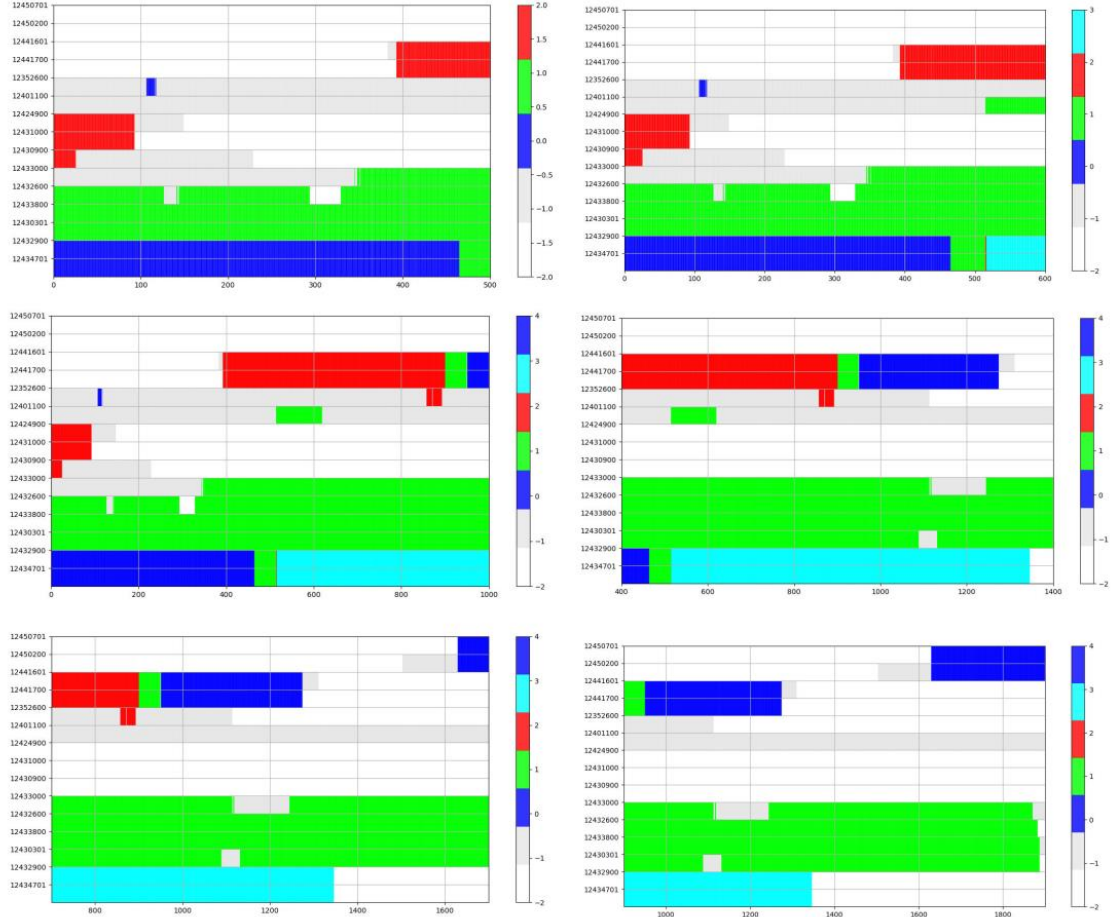


图 5.3 实时群组变化展示图

5.2.3 群组识别算法准确度

群组识别算法最终需要的是准确度计算，目前用到比较多的群组识别准确度计算方法是 F-measure。F1-measure 是对准确率 Precision 和召回率 Recall 的综合评价指标：

$$F1 - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

对于每个时刻都有一个 F1-Measure，所以使用每个时刻的 F1 的平均值。Wirz^[15]将这个值命名为 FAA(Flock Assignment Accuracy)，这里我们也这样使用。另一个 Wirz 提出的评价指标是 NFDA (Number of Flocks Detection Accuracy)，即每个时刻分组的个数与最初的分组个数是否相等，如果相等就为 1，反之为 0，最后计算所有时刻的平均值。如表 5.1 所示,eps 选值与准确度计算的关系，其中 FAA 和

NFDA 的变化趋势基本相同，两者都在 ϵ 约等于 3.5 时取到较大值。

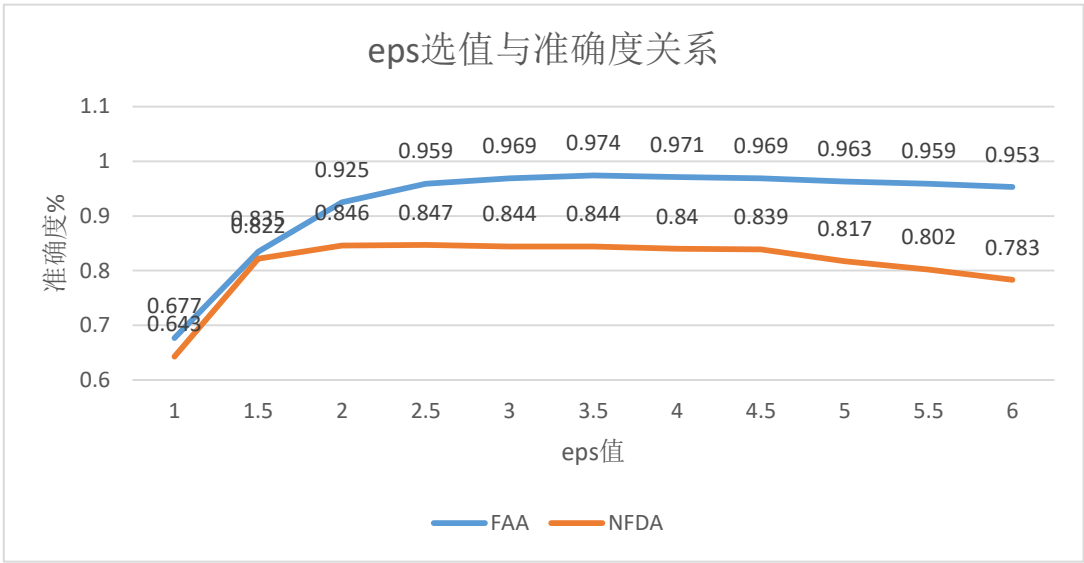


图 5.4 ϵ 选值与准确度关系

5.3 插值算法结果

设计轨迹插值算法的目的就是为了针对不对齐的采样周期较大的数据，使此类数据成为一个在时刻上对齐同时频率较大的平滑的轨迹，从而有利于对人群进行空间-时序的群组识别。首先我们来看一下插值后轨迹的变化。如图 5.5 所示，这是数据集中一个用户的轨迹图，由图可见插值后的轨迹较原始数据更加平滑，没有毛糙的尖角，这是 Catmull-Rom 样条算法的特性，它是通过对样本数据模拟经过控制点的平滑的曲线来插值的。另外，图形的运动趋势与原始数据基本一致，这说明了插值算法对轨迹还原的准确性。

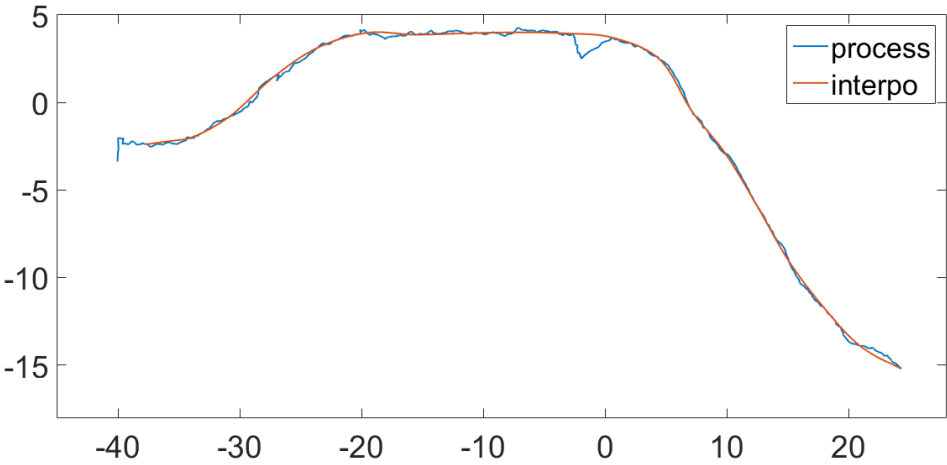


图 5.5 插值轨迹展示

当然从肉眼上也许无法观察出实际效果，因此我们将插值后的轨迹用于群组识别，通过实验结果检验出插值轨迹的有效程度，这里我们使用的阈值 $\text{eps}=3$ ，迭代结果也表明 3 的时候可以取到较好的分组。结果如图 5.6 所示，在采样周期为 1-5 之间，群组识别的准确度变化幅度不大，准确度都十分高。但是我们可以看到在采样周期为 2, 3 的时候准确度相对其他周期有了一定的上升，这说明插值后的轨迹比原始轨迹分组的效果更好。背后的原因我们可以推测是因为原始数据存在一定的噪点值。图 5.5 中看出，原始数据更加毛糙。而轨迹插值更像是做了对原始轨迹的平滑，排除了噪点的误差。在采样周期为 1 时，仍旧有不少噪点进入样本数据。在采样周期为 4, 5 时，原始数据丢失严重，对于轨迹的还原存在不可避免的硬性误差，所以群组识别准确度相对开始下降。而当采样周期达到 2, 3 左右时，采样的数据和插值算法的搭配对于轨迹还原度最高，对群组识别最有利。此外对比未插值数据的群组识别准确度，插值后得到的准确度明显高于未插值前，我们可以判断插值算法的确有效地进行了轨迹的插值。

日常生活中，人的行走速度基本保持在 0.5m/s - 2m/s ，以 3 秒的采样周期也许够用，如果采样周期到了 5 秒，人的行走距离很可能达到 5m 左右，这种频率的定位数据在室内对于群组识别是不够的。本章提出的插值算法可以达到 97.5% 的群组准确度，并且识别群组个数的准确达到 89%，这说明了即使是在 5 秒的采样周期通过使用本章的轨迹插值算法得到的轨迹依旧可以很好的进行群组识别。

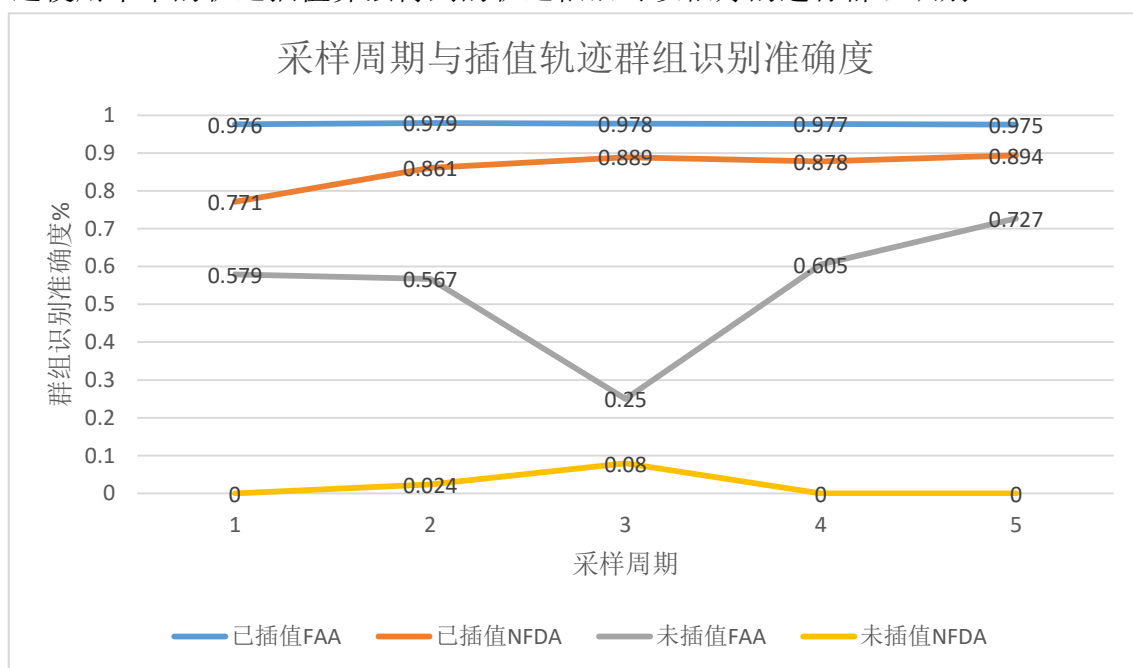


图 5.6 采样周期与插值轨迹群组识别准确度

第六章 结语

本章对前文各个模块的算法和结果进行了总结，并针对研究细节提出了问题，针对问题介绍下一步的研究方向。

6.1 结论

本文的内容主要包括四个部分，空间-时序群组识别算法，实时图形化展示，以及基于 Catmull-Rom 的插值算法。其中主要介绍了群组的概念，不同空间聚类算法的区别，实时化的空间-时序识别算法，图形化展示需求的分析，实时化图形展示所需的实时颜色选取算法，以及 Catmull-Rom 轨迹插值算法原理和实现。通过空间-时序识别算法我们得到了 97.4% 的一个分组准确度。通过对群组识别算法的实时化计算，和颜色选取算法，实现了对于群组识别的实时图形化监控，使用既能看到当前的分组情况，又能看到群组在时序上的变化状态，满足了对图形化的需求。通过 Catmull-Rom 的轨迹差值算法，解决了空间-时序群组识别算法所需时刻对齐的问题，即使原始数据并不是时刻上对齐的，而且数据本身采集频率过低，或者只有部分准确的定位数据，我们都可以通过轨迹插值的方法得到一条相对还原度不错的轨迹数据，轨迹的平均 Frechet 误差度在 1.24 左右，其中在开始时间 $t_0 = 0$ 和采样周期等于 3 的时候得到一个最佳的误差度和群组识别准确度。还原后的轨迹得到的群组识别准确度与原数据相差无几，说明插值算法的有效性。本文通过提出实时图形化和轨迹插值算法真正促进了基于运动轨迹的群组识别的研究，一方面有群组变化的可视化给我们对群组更好的监控，在难以现场指控的情况下加速群组性工作的指挥调度，另一方面轨迹插值算法减轻了对于采样数据的高要求，譬如减少高频率的手机定位数据从而减少了耗电量和大量数据传输所需的带宽流量等等。

6.2 问题与展望

当然通过实验对算法的验证，虽然取得了一定的好的结果，但是本毕业设计还存在以下问题。

首先，正如在图形化那章所提到的，对于我们使用 δ 时间来对群组做时间上的延续，我们并没有一个比较好的衡量标准来确定 δ 值的选取对于群组识

别的影响。另外实时的群组识别图形化的是由 python 的 matplotlib 库实现的。由于还没有进行大量数据的测试，图形化目前还属于 demo 阶段，如果要真正用于实际生活中的群组监测还需要更多的工作完成，包括算法的性能的优化，逻辑性测试等等。最后，轨迹插值算法还存在很多局限性，譬如参数值 t_0 的选值对于不同的数据影响规律性不强，还不能科学的给出最优的 t_0 。

所以接下来的工作也十分明确，为了使实时图形化方法真正有效，需要实现从手机端到服务器端的数据传输，实时群组识别，和实时图形化展示整个系统，让实时图形化可以用到实际工作中。这三个部分一定程度上是耦合性不强的，因此我们可以替换实时群组识别算法，和不同的数据传输，以比较不同群组识别算法的准确度和性能。

插值算法的背后的数学原理本身比较复杂，在插值开始时刻偏移量上的研究还需要更深入一步，从原理上给出最优的 t_0 。此外计算插值算法只使用了 ATC 数据进行验证，我们可以使用更多数据进行验证检验其鲁棒性。当然插值不仅仅局限于轨迹插值，很多时候我们进行群组识别还会用到传感器还有无线电等等设备的数据，这些数据较轨迹有所不同，插值算法是否可以用于这个数据也是值得我们探讨的一个问题。

参考文献

- [1]崔颖. 基于 Kinect 的人群识别与跟踪系统[D]. 复旦大学, 2013.
- [2]刘悦韡. 基于轨迹还原的智能手机室内定位系统的设计与实现[D]. 西安电子科技大学, 2014.
- [3]王福伟, 黄智刚. 室内定位中移动目标轨迹跟踪研究[J]. 导航定位学报, 2016, (01):33-37.
- [4] Liu Y S, Fang Y, Ramani K. Using least median of squares for structural superposition of flexible proteins[J]. BMC Bioinformatics, 2009, 10(1):29.
- [5] Li S, Qin Z, Wu C, et al. TraIL: Pinpoint Trajectory for Indoor Localization[J]. International Journal of Distributed Sensor Networks, 2015, 2015(1):1-8.
- [6] Sochman J, Hogg D C. Who knows who - Inverting the Social Force Model for finding groups[C]// IEEE International Conference on Computer Vision Workshops. IEEE, 2011:830-837.
- [7] Li X, Xiong S, Duan P, et al. A Study on the Dynamic Spatial-Temporal Trajectory Features of Pedestrian Small Group[C]// International Symposium on Dependable Computing and Internet of Things. 2015.
- [8] Fisher R J. Group-Derived Consumption: The Role of Similarity and Attractiveness in Identification with a Favorite Sports Team[J]. Advances in Consumer Research, 1998.
- [9] Feese S, Arnrich B, Burtscher M, et al. Sensing group proximity dynamics of firefighting teams using smartphones[C]// International Symposium on Wearable Computers. 2013:97-104.
- [10] Helbing P, Molnár T. Social Force Model for Pedestrian Dynamics[J]. Physical Review E Statistical Physics Plasmas Fluids & Related Interdisciplinary Topics, 1998, 51(5):4282.

- [11] Kalnis P, Mamoulis N, Bakiras S. On Discovering Moving Clusters in Spatio-temporal Data[M]// Advances in Spatial and Temporal Databases. Springer Berlin Heidelberg, 2005:364-381.
- [12] Gordon D, Wirz M, Roggen D, et al. Group affiliation detection using model divergence for wearable devices[C]// ACM International Symposium on Wearable Computers. ACM, 2014:19-26.
- [13] Masum A A, Rafy M H, Rahman S M M. Video-based affinity group detection using trajectories of multiple subjects[C]// International Conference on Electrical and Computer Engineering. IEEE, 2015:120-123.
- [14] Chen F, Cavallaro A. Detecting group interactions by online association of trajectory data[C]// International Conference on Acoustics, Speech, and Signal Processing. 2013:1754-1758.
- [15] Wirz M, Schl, Pfer P, et al. Towards an online detection of pedestrian flocks in urban canyons by smoothed spatio-temporal clustering of GPS trajectories[C]// ACM Sigspatial International Workshop on Location-Based Social Networks. ACM, 2011:17-24.
- [16] Kjærgaard M B, Wirz M, Roggen D, et al. Mobile sensing of pedestrian flocks in indoor environments using WiFi signals[C]// IEEE International Conference on Pervasive Computing and Communications. IEEE, 2012:95-102.
- [17] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, "Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones, " in Proc. ACM Conf. Ubiquitous Comput., 2012, pp. 240 - 249.
- [18] Kjargaard M B, Blunck H, Wustenberg M, et al. Time-lag method for detecting following and leadership behavior of pedestrians from mobile sensing data[C]// IEEE International Conference on Pervasive Computing and Communications. IEEE, 2013:56-64.
- [19] Li S, Qin Z, Song H. A Temporal-Spatial Method for Group Detection, Locating and Tracking[J]. IEEE Access, 2016, 4:1-1.

- [20] Kang J H, Welbourne W, Stewart B, et al. Extracting places from traces of locations[C]// ACM International Workshop on Wireless Mobile Applications and Services on Wlan Hotspots. ACM, 2004:110-118.
- [21] Yuksel C, Schaefer S, Keyser J. Parameterization and applications of Catmull-Rom curves. [J]. 2011, 43(7):747-755.
- [22] Brscic D, Kanda T, Ikeda T, et al. Person Tracking in Large Public Spaces Using 3-D Range Sensors[J]. IEEE Transactions on Human-Machine Systems, 2013, 43(6):522-534.
- [23] Fan X, Lei Y, Wang Y, et al. Long-term intuitionistic fuzzy time series forecasting model based on vector quantisation and curve similarity measure[J]. Iet Signal Processing, 2016, 10(7):805-814.
- [24] Mawson A R. Understanding mass panic and other collective responses to threat and disaster. [J]. Psychiatry, 2005, 68(2):95-113.
- [25] Ester M, Kriegel H, Jiirg S, et al. A density based algorithm for discovering clusters in large spatial databases[J]. 1996.
- [26] Yang Q, Huang B, Zeng X. A Group Member Search Method Based on Incremental Trajectories Data[C]// International Symposium on Computational Intelligence & Design. IEEE, 2015:529-533.
- [27] Zhou C, Frankowski D, Ludford P, et al. Discovering personal gazetteers: an interactive clustering approach[C]// ACM International Workshop on Geographic Information Systems. ACM, 2004:266-273.

致谢

很庆幸自己能够顺利地完成此次毕业论文，在这里我要对我的帮助过我的老师同学表示衷心的感谢。

首先需要感谢我的指导老师，朱卫平老师，在我不知道如何去开始毕业论文选题的时候，他悉心指导，帮助我在众多论文信息中理清思路，确定了选题。在毕业论文完成的过程中，朱老师不断关注我们的项目进展，定期让我们进行汇报，根据我们研究的结果提出建设性的意见，让我们将毕业论文内容完成得更加充实而有说服力。朱老师这种严谨认真的治学态度和深深地影响了我，让我学会要一丝不苟地对待科研，任何一个细节都不放过。另外也感谢老师对我的监督，让我能在规定的时间内完成论文的撰写。

其次我还要感谢学院，感谢对我言传身教的老师们，是您们的教学和训练，让我拥有了扎实的专业基础和初步科研的能力，这样才能顺利地完成任务，读论文，提出课题，研究并得出结果的这样一个过程。

最后感谢武汉大学这样一个优秀环境，让我与很多优秀的同学交流学习，开阔了我的视野，增长了见识，同时也感谢学校提供的丰富的资料书籍期刊的使用权，让我可以无障碍地学习。感谢大学四年的时光让我成为了更好的自己。