
清 华 大 学

综 合 论 文 训 练

题目： 三维运动捕捉数据的
关键帧提取和恢复

系 别： 自动化系
专 业： 自动化
姓 名： 潘佳
指导教师： 徐文立 杨青
辅导教师： 赵明国

2005 年 5 月 24 日

清华大学综合论文训练

清华大学综合论文训练

摘要

运动捕捉数据的关键帧提取是很多应用如运动捕捉数据库的建立、运动序列的比较、运动编辑、网络传输的基础，在实际中非常重要。论文针对建立运动捕捉数据库和在网络上传播运动捕捉的需要，使用计算机视觉、计算机图形学和模式识别的基本方法，提出了完整的捕捉数据关键帧提取算法，并和已有的算法进行了比较，证明了算法的优越性。

主要工作如下：

1. 建立了一个基本的运动编辑的平台，可以显示和简单处理运动捕捉数据；
2. 搜集总结了描述运动捕捉数据必要的数学工具，并详细讲述了实现中的问题，着重介绍了数值计算的细节，例如奇异值问题；
3. 在已有的算法的基础上，对如何完整的表示运动进行了分析，提出了使用全局整体运动和局部的细节运动综合表示运动信息的方法. 提出使用微分几何中的挠度和曲率的加权和来描述全局整体运动的方法；
4. 对已有的动态聚类算法进行了改进，在保持时序性的前提下，尽量保持了动态聚类的优越性，从而提高了算法的性能；
5. 进行了仔细的实验设计，对实验数据给予了详细的分析，对实验数据的选择原则给予了说明。并根据实验数据说明了实验结果中的一些问题；
6. 对算法在某些实验数据上的反常结果给予了一定的解释。

关键词

运动捕捉，关键帧，聚类，挠度，曲率

清华大学综合论文训练

Abstract

Key frame extraction technology, which is of great importance in real world, is the basis of many applications of motion capture data, such as the establishment of motion capture library, the comparison among motion sequences, motion editing, transmission of motion capture data via internet, etc. According to the request of motion capture library and transferring of motion capture data via web, a complete key frame extraction algorithm is proposed using principles of computer vision, computer graphics and pattern recognition, which is compared with existing algorithms and shows great advantages over them.

The major achievements and contributions can be summarized as follows:

1. A toolbox of motion editing is built, which can now show the capture data and do some simple procession of the data;
2. I summarize the math knowledge and principles which are necessary to describe the motion capture data, and give a detailed description of problems coming during programming. Great emphasis is put on the details of numerical computation, for example, the singularity during finite-length computation;
3. Based on existing algorithms, I give a discussion on how to represent motion perfectly. A method using global motion and local motion is proposed. Torsion and curvature, two important quantity of the 3D curve, are used to give a mathematical description of the global motion;
4. The existing dynamic clustering algorithm is improved. Advantages of clustering algorithm are kept as more as possible while keeping the scheduling information of the key-frame sequence;
5. A careful experiment design is done. The data for experiment are analyzed in detail. I give a clear description to the criteria to choose data, and give some explanations to the experiment result from the perspective of data.
6. Some explanations are given, attempted to explain the failure of algorithm on some data set.

Keyword

Motion capture, key frame, cluster, torsion, curvature

清华大学综合论文训练

目录

第 1 章 引言	1
1.1 背景.....	1
1.2 相关工作.....	1
1.2.1 动作捕捉数据的重用技术.....	2
1.2.2 关键帧和关键帧技术.....	4
1.2.3 关键帧和关键帧技术的发展历史.....	5
1.3 本文结构.....	6
第 2 章 运动捕捉系统简介	8
2.1 运动捕捉系统分类.....	8
2.2 光学运动捕捉系统.....	8
2.2.1 光学运动捕捉系统的组成.....	8
2.2.2 光学运动捕捉系统的特点.....	10
2.3 电磁式运动捕捉设备.....	11
2.4 机械式运动捕捉设备.....	12
第 3 章 四元数简介	13
3.1 四元数的定义和性质.....	13
3.2 四元数和三维旋转的对应关系.....	14
3.3 使用四元数表示旋转变量之间的距离.....	17
3.4 插值特性.....	18
3.5 单位四元数的指数表示.....	20
3.6 四元数相对于欧拉角在表示旋转中的优越性.....	20
3.7 结论.....	23
第 4 章 微分几何简介	24
4.1 曲线的参数方程表示.....	24
4.2 Frenet 标架下的几何量：切矢，挠度和曲率.....	25
4.3 曲线几何量计算方法.....	28
第 5 章 数值微分简介	30
5.1 数值微分基础.....	30
5.2 常用数值微分公式.....	31
5.3 数值微分计算挠度和曲率的细节问题.....	33
第 6 章 运动捕捉数据	35
6.1 运动捕捉数据的结构化.....	35
6.2 三维旋转的表示方法.....	39
6.3 树形结构的优点和缺点.....	41
6.4 伪码数据结构.....	42

清华大学综合论文训练

第 7 章 运动捕捉序列的关键帧提取	44
7.1 关键帧提取算法.....	44
7.1.1 聚类算法整体框架.....	44
7.1.2 给定关键帧率时的调整算法.....	45
7.1.3 关键帧属性度量—传统方法.....	48
7.1.4 聚类算法改进—考虑时序约束.....	50
7.1.5 关键帧属性度量—考虑整体几何性质.....	52
7.1.6 遗留的问题.....	54
7.2 插值恢复算法.....	55
第 8 章 数据文件结构	56
8.1 TRC 文件简介.....	56
8.2 BHV 文件格式和 HTR 文件格式简介.....	57
第 9 章 算法实现	61
9.1 映射.....	61
9.2 三维世界坐标转换为四元数.....	62
9.3 奇异值的解决思路.....	63
第 10 章 实验结果	65
10.1 实验数据和实验设计.....	65
10.2 阈值、关键帧率及恢复误差.....	66
10.3 数据分析.....	68
10.4 算法比较.....	71
10.5 结论.....	76
第 11 章 将来工作.....	77
参考文献	78
致 谢.....	81

清华大学综合论文训练

图目录

图 1	三维形变一例	2
图 2	三维形变又例	3
图 3	三维形变再例	3
图 4	光学运动捕捉系统	9
图 5	电磁式运动捕捉设备	11
图 6	机械式运动捕捉设备	12
图 7	公式证明简图	16
图 8	四元数之间的距离	17
图 9	Euler 角	20
图 10	运动捕捉对象上的 Frenet 标架	27
图 11	人体和运动捕捉数据的对应	36
图 12	OpenGL 变换堆栈在手建模中的应用	39
图 13	人体运动捕捉数据的树形结构	39
图 14	旋转的矢量表示	40
图 15	聚类算法 1: 常规的动态聚类算法	45
图 16	聚类算法 2: 保持时序周期信息的动态聚类算法	51
图 17	聚类算法 3: 保持时序周期信息的动态聚类算法改进	52
图 18	聚类算法 4: 考虑了几何性质	54
图 19	四元数旋转表示示意	62
图 20	传统算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图	67
图 21	本文算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图	68
图 22	传统算法得到的平均误差在四种数据上的结果	69
图 23	本文提出算法得到的平均误差在四种数据的结果	69
图 24	传统算法得到的最大误差在四种数据上的结果	70
图 25	本文提出的算法得到的最大误差在四种数据上的结果	70
图 26	算法在典型数据 Stand 上的比较	72
图 27	算法在典型数据 Walk 上的比较	73
图 28	算法在典型数据 Turn Over 上的比较	74
图 29	算法在典型数据 Play golf 上的比较	75

清华大学综合论文训练

第1章 引言

2005年4月至6月,我参加了‘基于计算机视觉的三维运动捕捉系统’的项目。这个项目主要目的是使用三维计算机视觉的原理和方法,建立实用的光学运动捕捉系统。整个项目基本上属于计算机视觉领域。我主要进行了关键帧提取的部分,建立了一个运动编辑平台。一方面可以为三维运动捕捉系统提供一个后期处理的工具,同时也为以后进行运动编辑的研究打下比较扎实的基础。

1.1 背景

运动捕捉技术可以在较短的时间内产生高质量的真实感动画,这是传统的动画技术无法做到的。当前我国蓬勃发展的动画产业极需这样的技术。但是这种技术基本上被 ViconPeak 和 Motion Analysis 两家公司垄断。这个项目的目的主要就是为了打破它们的垄断,扶持民族动画产业。

所谓运动编辑研究的就是如何对已有的运动数据进行处理,产生满足指定约束运动的技术。运动编辑是近期图形学、图像学的研究热点之一。虽然运动捕捉可以得到非常真实的运动,但是一方面它的成本非常高昂,必须要通过适当的方法对已有的运动捕捉数据进行重用;另一方面,运动捕捉产生数据都是真实的,无法满足一些特技动作的要求。因此运动编辑是产业界非常需要的技术。但是由于它的困难性,从80年代末提出后发展很缓慢,直到最近这方面的研究才多起来。

1.2 相关工作

关键帧提取算法已经大量应用在二维视频序列上。基于聚类的方法也已经用于基于内容的视频压缩中。对于运动捕捉数据的关键帧,[20]中已经提出了使用聚类来提取的方法。但是该文只解决了应用于建立运动捕捉数据库的关键帧提取问题,而且没有区分特征提取和插值恢复两种不同应用要求下的关键帧提取问题。该文没有利用整体特性在关键帧提取中的作用,只利用了局部细节的信息。运动捕捉数据表示的一些数学工具和传统的方法在[1, 8, 19, 22]进

清华大学综合论文训练

行了介绍。数据的分层的思想也来自上述的文献。

1.2.1 动作捕捉数据的重用技术

运动捕捉技术能够实时的捕捉和记录人体运动的数据，并可将其直接映射到计算机动画中人物角色上并驱动角色的表演。它的优点是能够真实地、自然地获取现实世界中人体运动特征，并能够有效地表达演员地个性化表演特征。当前，运动捕捉方法在三维动画制作和游戏开发中十分流行，也被业界认为是获取高质量运动数据的标准方法之一。但运动捕捉数据很难被重用，这是因为其底层的关节点细节描述缺乏足够的结构化信息，导致了对它的编辑和修改都很容易引起运动要素的变形，进而使得改动后的运动数据在质量上急剧退化。如何有效地和令人信服地编辑和改变三维运动数据，使得有关操作能在改动用户所指定的运动特性的同时，保持运动数据中的其余运动特性基本保持不变。这已经成为目前计算机动画的研究热点之一。例如，沈向洋（Heung-Yeung Shum）等近期的工作就是如何在三维形变的同时保持运动曲面的自然过渡和衔接。他们的方法主要是利用偏微分方程中的理论，即通过边值条件的约束强制三维实体的表面三角形之间保持拓扑上的自然过渡和表面纹理的自然衔接。



图 1 三维形变一例



图 2 三维形变又例



图 3 三维形变再例

但是，运动捕捉方法又有它自身的不足，运动捕捉系统一般都非常昂贵。同时，运动捕捉的过程需要表演者和技术人员消耗相当的时间和精力才能完成。为了使得运动捕捉技术得到更加广泛的应用，运动捕捉数据就有必要进行重用。运动数据的重用通常指的是通过对存储在运动捕捉数据库中的运动片断进行适当的运动转换，生成满足一定编辑要求的新的运作数据的过程。它实际上是拓展了运动捕捉技术，并提供了一种相对便宜的方法来快速的得到高质量的运动数据。上述沈向洋等的工作实际上也为比较完美的三维重用提供了一种可能的方向。

运动捕捉数据重用的核心技术主要是两个方面：运动数据库构建技术，运动编辑和合成技术。运动数据库的构建是重用的基础，运动捕捉数据库中包括运动数据的类型、运动数据的内容、进行重用的必要辅助信息（表演者的几何尺寸和骨架结构）。运动编辑和合成技术是运动数据重用的质量保证，不论对源数据进行怎样的变换（操作），都需要在最大程度上保持原始运动的真实感、美感和自然性，否则运动捕捉数据的重用就失去了意义。

清华大学综合论文训练

当前的运动编辑和合成技术的研究才刚刚开始。在这些探索性的研究工作中采用的方法主要是[4~6, 9~14]:

信号处理方法。运动特性参数被作为采样后的信号处理,并借助图像处理和信号处理领域的相关技术来编辑和修改有关联的运动参数,从而产生新的运动序列。由于用户的处理要求大多基于高层的语义和运动特征描述,而信号处理方法不能显示的描述运动特征,难以对其编辑要求进行之间的支持。

约束优化方法。它将用户要保留或改变的运动特征的需求显式的表述为约束条件,然后基于约束的优化求解方法来产生符合要求的运动序列。这些约束条件主要包括空间(几何)约束。如时空约束、物理学定律、动力学定律和逆向运动学求解等。约束优化方法受限于约束条件范围和约束优化问题求解的效率。这是当前的主流方法之一。

基于统计学模型的运动合成方法。运动捕捉数据往往缺少结构化的描述,统计学模型如隐马尔可夫模型可以用于学习和提取运动的统计学特征,即有意义的属性,如运动模式等,然后再对其进行统计学意义上的运动合成。基于统计学模型的运动合成往往是基于实际运动数据的统计学模型特性进行,而非实际运动特性本身,因此有可能使得一些看起来是细节但是实际上非常重要的运动特征丢失,从而导致合成出来的运动数据失真。

基于插值的运动编辑方法。插值法是基于运动序列的参数化描述,来插值产生新的运动,例如线性插值方法被用于生成不同行走姿势的运动编辑。这是一类传统的方法,但是和二维的插值不同,三维的曲线和曲面的插值大量的使用微分几何的知识和技术。这也是当前动画领域流行的方法。

1.2.2 关键帧和关键帧技术

计算机动画是指用计算机绘制程序生成一系列的画,每张画面称为一帧,其中当前帧画面是对前一帧的部分修改。关键帧技术直接来源于传统的动画制作。出现在动画片中的一段连续画面实际上是由一系列静止的画面来表现的,制作过程中并不需要逐帧绘制,只需要从这些静止的画面中选出少数的几帧加以绘制。被选出的画面一般都出现在动作变化的转折点(拐点),对这段连续动作起着关键的控制作用,因此称为关键帧(Key Frame)。而关键帧之间的画面就是中间帧,计算机动画中的中间帧是由计算机自动生成的。插值(这儿是广

清华大学综合论文训练

义的，包括上节中运动编辑的一切方法，即所有的从关键帧光滑的过渡到新的一帧的方法都可以认为是插值算法）代替了设计中间动画帧的动画师。所有影响画面图像的参数都可以成为关键帧的参数，如位置、旋转角、纹理的参数等。几乎所有的动画软件如 Maya, SoftImage, WaveFront、3DSMAX 等都提供对运动参数插值实现对动画的运动控制，如物体的位置、空间取向、颜色等的变化，以及对多个运动参数进行组合插值的相关软件和硬件技术。

计算机动画是由计算机生成的虚拟运动效果，画面中的物体不需要去真正的建造，物体、虚拟摄影机的运动没有受到什么限制。动画师可以随心所欲的创造他的虚幻世界，这都离不开关键帧技术在设计中的作用。关键帧技术是计算机动画设计的重要工具，它包括创造计算机动画运动和变化效果的计算机技术和控制方法。随着计算机技术的发展，关键帧技术包含的功能也越来越丰富，对这些功能的创造性运用可以产生新的视听效果和增加计算机动画设计的表现形式和手段。

从原理上，关键帧插值问题可以归结为参数插值问题。传统的插值方法都可以运用到关键帧方法中。但关键帧插值又与纯数学的插值不同，有其特殊性。一个好的关键帧插值方法必须能够产生逼真的运动效果，并能给用户方便有效的控制手段，即良好的交互手段。一个特定的运动从空间轨迹上看（微分几何或射影几何）可能是正确的，但是从运动学或计算机动画设计上看可能是错误的或是不合适的。用户必须能够控制运动的运动学特性，即通过调整插值的算法和参数来改变运动的速度和加速度，使得动画的效果与物理世界中的效果基本一致。当然，非真实感动画例外，如超人效果就可能违反物理定律。

1.2.3 关键帧和关键帧技术的发展历史

传统的动画是由动画师在画纸上手绘真人的动作，然后再复制到卡通人物之上。20 世纪 70 年代后期，计算机技术迅速发展，纽约技术学院计算机绘图实验室的 Rebecca Allen 教授将录像带上的舞蹈演员的影像投射到计算机的显示器上，利用计算机绘图记录影像的动作，然后描摹轮廓。1982 年，MIT 和纽约技术学院同时利用光学追踪技术记录人体动作。1975 年，日本计算机动画的先驱河口洋一郎（Yoichiro Kawaguchi）运用计算机创作出早期的动画作品。利用由计算机生成的点、线的自转以及公转所产生的运动空间，作出运动的造型物体。

清华大学综合论文训练

美国是最早发展计算机动画的地方，1970 左右开始利用计算机模拟人物活动的关键帧效果。1982 年，Disney 推出第一套电脑动画的电影 Tron(电脑争霸)。在三维计算机动画中，中间帧的生成由计算机来完成，关键帧技术开始得到广泛的应用。

1983 年，MIT 的 Ginsberg 和 Maxwell 发展了 Graphical Marionette，利用计算机语言控制卡通的关键帧动作。但受到当时计算机硬件速度的限制，一个简单的动画的生成需要惊人的时间。

Tantanic 的成功很大程度上得益于其对关键帧技术的应用。Digital Domain 使用了一年半的时间，使用了 300 多台 SGI 图形工作站。中国的动画片<西游记>使用了 100 万张原画、近 2 万张背景，耗纸 30 吨，耗时 5 年。Disney<花木兰>中，一场匈奴大军的厮杀只用了 5 张手绘士兵的原画，利用计算机生成的关键帧效果，就变化出三、四千个不同表情的士兵作战的模样。

1.3 本文结构

首先，介绍了运动捕捉系统的原理及其分类。

然后，为了下面讨论问题的方便，对算法中使用的、但是在大学数学中并不讲授的数学知识做了介绍，主要包括四元数、微分几何和数值微分的基础。大部分经典的推导已经略去。保留证明的部分都是由笔者作出的证明，这些证明或者不见于文献，或文献中的证明用到的知识过于艰深，由笔者用初等方法重证。此部分原拟归入附录，但夜读 Donald E.Knuth 的经典《The Art of Computer Programming》，对他在介绍正题前先介绍数学基础的写法尤其赞赏，遂改变顺序，将此三章加入正文。

在此基础上，详细介绍了关键帧提取的算法。本算法在已有工作的基础上考虑了整体运动轨迹的几何特性，对关键帧提取算法做了较大的改进。

紧接着，讨论了实现中的一些问题。它们虽然是细节，但是不得不注意。

最后，进行了实验，将我们提出的算法和已有的算法做了比较。并对存在的问题和改进的方向做了一定的讨论。

尤其注意的是，第 3、4、5 章虽然是以数学基础为主，但是还是有相当的篇幅讨论实现中的问题。不是为了介绍数学而介绍数学，而是结合关键帧提取

清华大学综合论文训练

的问题来介绍，这三章讨论的问题都是非常重要的，决不是多余。

第2章 运动捕捉系统简介

2.1 运动捕捉系统分类

运动捕捉系统按传感器和感应源的不同基本上分为三类：

1) Outside-in system

使用外部的传感器收集放置在身体上的感应源。使用专用相机设备的光学运动捕捉设备即属于此种。这里，相机就是传感器，反射标记点（Marker）就是感应源，它反射外部光源的光线以供相机接收。

2) Inside-out system

将传感器放置在身体上，收集外部的感应源。基于电磁系统的运动捕捉设备就属于这一类。它的感应源是外部的一个电磁场，电磁传感器随着人体在这个电磁场中运动。

3) Inside-in system

这儿的传感器和感应源都放置在身体上。这种运动捕捉装置类似于一个电动机械的衣服。其中的传感器主要是电位计和角度仪，感应源是身体中的骨骼。

当前的运动捕捉系统主要就是基于上面的三种不同的原理，对应的分别为光学式、电磁式和机械式的运动捕捉系统。

我们的系统是基于光学设备的，所以主要介绍一下光学运动捕捉系统，对其他两种设备也将给予简单的介绍。

2.2 光学运动捕捉系统

2.2.1 光学运动捕捉系统的组成

光学运动捕捉系统（Optical Motion Capture Systems）是当前运动捕捉设备中的主流。

清华大学综合论文训练

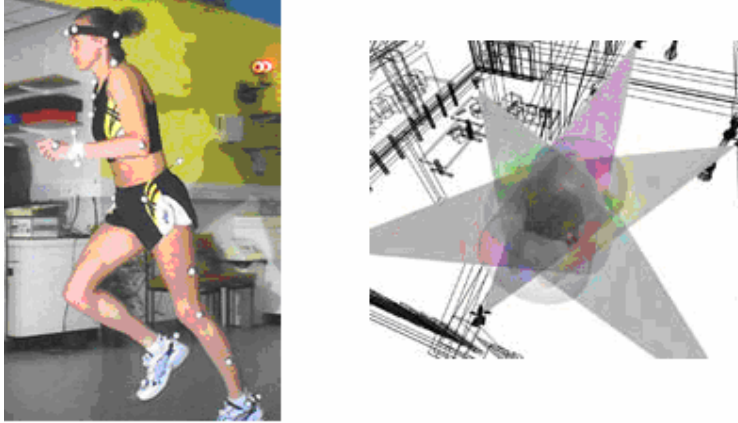


图 4 光学运动捕捉系统

当我们使用先进的计算机视觉系统去捕捉动作时，光学运动捕捉毫无疑问的是最精确的一种。但是它实际上不是一个实时性的系统，对于目标角色的实时反馈从理论上来说是无法做到的，因为通过光学运动捕捉系统得到的三维数据，必须经过大量的后期处理才能够作为计算机图形学后期算法的输入。

一个典型的光学运动捕捉系统使用一台高性能计算机作工作站，控制和处理从一些 CCD 数字摄像机组成的阵列输入的大量数据。CCD 是感光装置，它使用一些排列的光电感应元（像素）捕捉到光，然后每一个感应元会测量出光的强度，产生出三维物体的二维投影的数字表达。一个 CCD 摄像机像素排列是一个矩阵。显然，像素的数目越多，图像的解析度就越好，分辨率就越高。但是其中也有相应的平衡。每秒钟产生样本的速率，或者叫做帧率，要足够快，可以跟随人体的运动速度，才能捕捉到高速运动中的细微的差别。分辨率和帧率一般是相关的。例如为了提高帧率可以适当的降低分辨率，为了提高分辨率也可以适当的降低帧率。

运动捕捉系统中另一个很重要的地方就是摄像机阵列的同步。一般先进的光学运动捕捉设备上都有对应的同步装置。

通常应用于运动捕捉的摄像机阵列的相机数量不少于 4 个，不多于 32 个。它们捕捉 Marker 点的速率无论在什么地方都是每秒 30 帧到 1000 帧，应该注意的是帧率在比较高的时候很难保证恒定，给同步造成很大的困难。因此在系统应用允许的情况下，应该适当的降低帧率。

相机应该与它们自身的光源相匹配。

清华大学综合论文训练

光源可以使 Marker 点产生直线反射。Marker 点是一种涂有特殊材料的球状的反光材料。红外光源可以使得捕捉的画面几乎不失真。在小范围内进行运动捕捉时，标记点小球的直径一般为几毫米。

光学系统必须校准，使得所有的相机都能跟踪到目标，使计算机可以辨认出三维物体，如带有反射标记点的定标块(立方体和棒状支架)。为了得到 Marker 点在空间的准确位置，必须使得所有相机对标志点的三维透视关系上相互结合。一个单独的点在三维空间中的位置至少需要两个视角的跟踪。但是并不是说相机的数目越多越好，因为每增多一个相机就会延长后期处理的时间。实际上，Marker 点三维位置的计算是通过类似共轭梯度方法迭代计算，所以相机数目越多，优化问题规模越大，优化收敛的速度越慢，实时性越差。

相机得到的数字图像输入计算机后，接下来进行计算机视觉的处理。

首先把标志点从图像中提取出来；

然后确定每一个标志点的二维坐标；

最后通过同一个标志点在不同图像内的二维坐标重建出三维的坐标。

2.2.2 光学运动捕捉系统的特点

光学式视觉捕捉设备之所以成为主流，主要因为相对于其他的系统有一系列的优点：

- 1) 在大部分的情况下，得到的数据的精度很高；
- 2) Marker 点的位置可以方便的更改、配置；
- 3) Marker 点的数量相对于机械式可以很多；
- 4) 通过 Marker 点可以近似的得到骨骼的结构；
- 5) 不需要有线或无线的设备；
- 6) 相对于机械式和电磁式捕捉场地可以很大；
- 7) 捕捉的频率基本上与捕捉设备没有关系，主要由相机特性和总线带宽决定

主要的缺点：

清华大学综合论文训练

- 1) 得到的数据必须经过大量后期的图像处理；
- 2) 硬件十分昂贵（专用相机、镜头）
- 3) 不能捕捉长时间的动作
- 4) 对光源和环境的要求比较高

2.3 电磁式运动捕捉设备

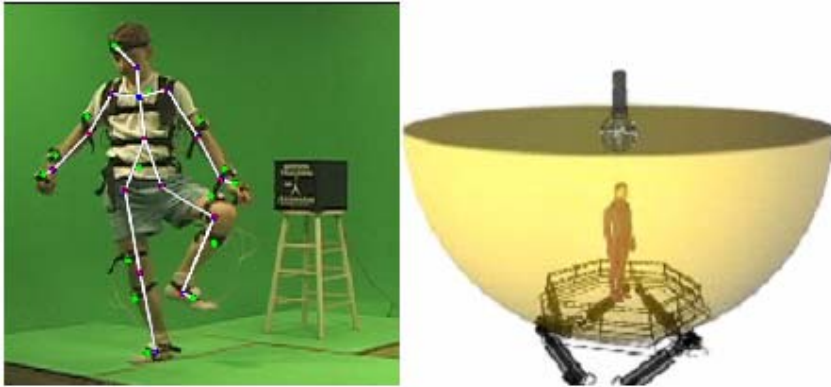


图 5 电磁式运动捕捉设备

电磁式装置的特点：

- 1) 提供实时性的数据输出
- 2) 位置和方向的数据不需要进行后期的处理
- 3) 价格便宜
- 4) 感应器不会饱和
- 5) 利用多重的设置可以捕捉多个对象的相互作用
- 6) 对金属的感应灵敏度低
- 7) 捕捉对象的运动受到线缆的制约
- 8) 采样速率较低
- 9) 捕捉区域较小

10) Marker 点的位置很难改变

2.4 机械式运动捕捉设备



图 6 机械式运动捕捉设备

机械式装置的特点：

- 1) 捕捉的范围比较大
- 2) 最便宜
- 3) 可以实时采集数据
- 4) 数据的捕捉比较容易
- 5) 感应器不会饱和
- 6) 利用多重的设置可以捕捉多个对象的相互作用
- 7) 采样速率较低
- 8) 设备笨重
- 9) 对人体的骨骼有制约
- 10) 感应器位置固定
- 11) 大部分系统无法捕捉空间位移

第3章 四元数简介

本章主要介绍关键帧提取中使用到的四元数的基本知识。更详细的介绍可以见[18, 22, 24]。

四元数是由英国天才数学家 Hamilton 发明的，它最重要的意义是在群论中提供了一个重要的例子。在计算机图形学中，它也是一种重要的工具。

四元数类似于复数，可以方便的表示空间中的旋转。

3.1 四元数的定义和性质

$q \triangleq a + \alpha, a \in R$ 称为实部, $\alpha \in R^3$ 称为虚部, 也可表示为 (a, α) 。

加法: $(a + \alpha) + (b + \beta) = (a + b) + (\alpha + \beta)$

标量乘法: $\lambda(a + \alpha) = \lambda a + \lambda \alpha, \lambda \in R$

乘法: $(a + \alpha) \circ (b + \beta) = (ab - \alpha \cdot \beta) + (a\beta + b\alpha + \alpha \times \beta)$

容易证明四元数关于上面的三种运算成为一个除环（非交换域）。

（因为四元数是加群，对乘法运算封闭，乘法满足结合律，标量乘法满足关于加法的左右分配律，故为环。又有单位元 0，显然有非零元，容易得到为除环，乘法不交换，故为非交换域）

共轭: $\bar{q} \triangleq a - \alpha$

范数: $|q|^2 \triangleq q \circ \bar{q} = \bar{q} \circ q = a^2 + |\alpha|^2$

$(\because q \circ \bar{q} = (a + \alpha) \circ (a - \alpha) = (aa + \alpha \cdot \alpha) + (a\alpha - a\alpha + \alpha \times \alpha) = a^2 + |\alpha|^2)$

容易得到:

清华大学综合论文训练

$$|q \circ q'| = |q| |q'|$$

简证:

$$\begin{aligned} \therefore |q \circ q'|^2 &= |(a + \alpha) \circ (a' + \alpha')|^2 \\ &= |(aa' - \alpha \cdot \alpha') + (a\alpha' + a'\alpha + \alpha \times \alpha')|^2 \\ &= (aa' - \alpha \cdot \alpha')^2 + (a\alpha' + a'\alpha + \alpha \times \alpha') \cdot (a\alpha' + a'\alpha + \alpha \times \alpha') \\ &= (aa')^2 - 2aa'\alpha \cdot \alpha' + (\alpha\alpha')^2 + a^2|\alpha'|^2 + 2aa'\alpha \cdot \alpha' + a^2|\alpha|^2 + |\alpha \times \alpha'|^2 \\ &= a^2a'^2 + |\alpha|^2|\alpha'|^2 \cos^2(\alpha \wedge \alpha') + a^2|\alpha'|^2 + a^2|\alpha|^2 + |\alpha|^2|\alpha'|^2 \sin^2(\alpha \wedge \alpha') \\ &= a^2a'^2 + |\alpha|^2|\alpha'|^2 + a^2|\alpha'|^2 + a^2|\alpha|^2 \\ |q|^2|q'|^2 &= (a^2 + |\alpha|^2)(a'^2 + |\alpha'|^2) = a^2a'^2 + |\alpha|^2|\alpha'|^2 + a^2|\alpha'|^2 + a^2|\alpha|^2 \quad \square \end{aligned}$$

$$q^{-1} \triangleq \frac{\overline{q}}{|q|}$$

$$\begin{aligned} \therefore \overline{q_1 \circ q_2} &= \overline{(a_1 + \alpha_1) \circ (a_2 + \alpha_2)} \\ &= \overline{(a_1a_2 - \alpha_1 \cdot \alpha_2) + (a_1\alpha_2 + a_2\alpha_1 + \alpha_1 \times \alpha_2)} \\ &= (a_1a_2 - \alpha_1 \cdot \alpha_2) - (a_1\alpha_2 + a_2\alpha_1 + \alpha_1 \times \alpha_2) \\ &= \overline{a_2 \circ a_1} \\ \therefore (q_1 \circ q_2)^{-1} &= \frac{\overline{q_1 \circ q_2}}{|q_1 \circ q_2|} = \frac{\overline{a_2 \circ a_1}}{|a_1| |a_2|} = a_2^{-1} \circ a_1^{-1} \end{aligned}$$

3.2 四元数和三维旋转的对应关系

设 $q = \cos \frac{\theta}{2} + u \sin \frac{\theta}{2}$, $u \in R^3, |u|=1$, $P \in R^3$ 。P 绕 u 旋转 θ 得到 P' 。那么有:

$$0 + P' = q \circ (0 + P) \circ q^{-1}$$

证明: (叉乘的证明中采用了 Einstein 表示)

清华大学综合论文训练

$$\begin{aligned}
 & q \circ (0 + P) \circ q^{-1} \\
 &= \left(\cos \frac{\theta}{2} + u \sin \frac{\theta}{2}\right) \circ (0 + P) \circ \left(\cos \frac{\theta}{2} - u \sin \frac{\theta}{2}\right) \\
 &= \left(-\sin \frac{\theta}{2} u \cdot P, \cos \frac{\theta}{2} P + \sin \frac{\theta}{2} u \times P\right) \circ \left(\cos \frac{\theta}{2}, -u \sin \frac{\theta}{2}\right) \\
 &= \left(-\sin \frac{\theta}{2} \cos \frac{\theta}{2} u \cdot P + \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \cdot P, \sin^2 \frac{\theta}{2} (u \cdot P) u + \cos^2 \frac{\theta}{2} P + \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P\right. \\
 &\quad \left.- \sin \frac{\theta}{2} \cos \frac{\theta}{2} P \times u - \sin^2 \frac{\theta}{2} (u \times P) \times u\right) \\
 &= \left(0, \sin^2 \frac{\theta}{2} (u \cdot P) u + \cos^2 \frac{\theta}{2} P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P - \sin^2 \frac{\theta}{2} (u \times P) \times u\right)
 \end{aligned}$$

\therefore

$$\begin{aligned}
 & [(u \times P) \times u]_i \\
 &= \varepsilon_{ijk} (u \times P)_j u_k \\
 &= \varepsilon_{ijk} \varepsilon_{jlm} u_l P_m u_k \\
 &= -\varepsilon_{ikj} \varepsilon_{jlm} u_l P_m u_k \\
 &= -(\delta_{il} \delta_{km} - \delta_{im} \delta_{kl}) u_l P_m u_k \\
 &= -u_i P_m u_m + u_k u_k P_i
 \end{aligned}$$

\therefore

$$\begin{aligned}
 & (u \times P) \times u = -(P \cdot u) u + (u \cdot u) P \\
 & q \circ (0 + P) \circ q^{-1} \\
 &= \left(0, \sin^2 \frac{\theta}{2} (u \cdot P) u + \cos^2 \frac{\theta}{2} P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P - \sin^2 \frac{\theta}{2} (u \times P) \times u\right) \\
 &= \left(0, \sin^2 \frac{\theta}{2} (u \cdot P) u + \cos^2 \frac{\theta}{2} P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P + \sin^2 \frac{\theta}{2} (P \cdot u) u - \sin^2 \frac{\theta}{2} (u \cdot u) P\right) \\
 &= \left(0, 2 \sin^2 \frac{\theta}{2} (u \cdot P) u + (\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2}) P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P\right)
 \end{aligned}$$

所以有：

$$q \circ (0 + P) \circ q^{-1} = \left(0, 2 \sin^2 \frac{\theta}{2} (u \cdot P) u + (\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2}) P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P\right)$$

从几何的角度我们可以得到同样的结果：

清华大学综合论文训练

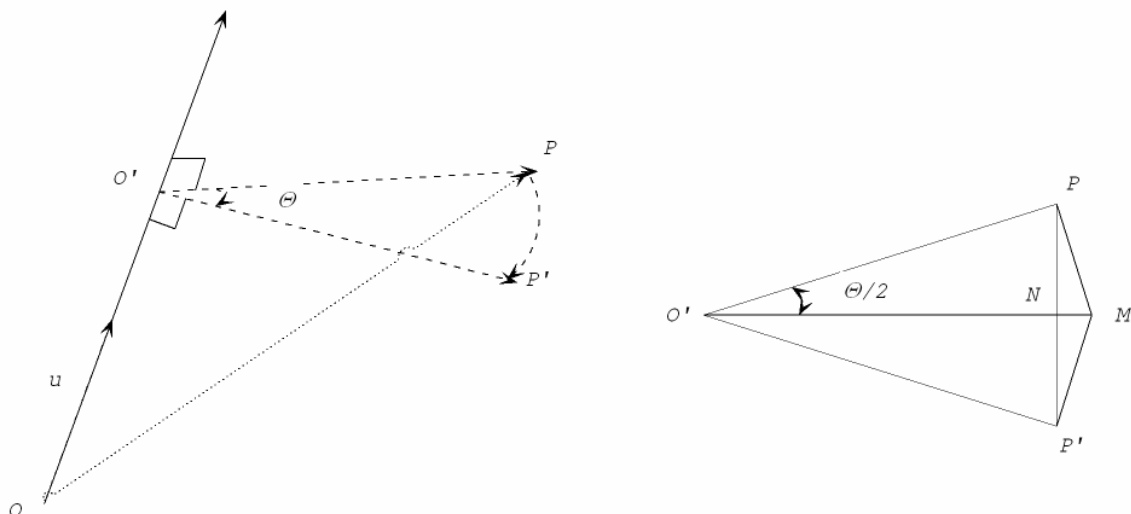


图 7 公式证明简图

$$OO' = (u \cdot P)u$$

$$O'P = OP - O'O = P - (u \cdot P)u$$

$$PM = u \times O'M \operatorname{tg} \frac{\theta}{2} = u \times (P - (u \cdot P)u) \operatorname{tg} \frac{\theta}{2} = u \times P \operatorname{tg} \frac{\theta}{2}$$

$$O'M = O'P + PM = P - (u \cdot P)u + PM =$$

$$P - (u \cdot P)u + u \times P \operatorname{tg} \frac{\theta}{2}$$

$$O'N = \cos^2 \frac{\theta}{2} O'M = \cos^2 \frac{\theta}{2} (P - (u \cdot P)u + u \times P \operatorname{tg} \frac{\theta}{2})$$

$$ON = O'N + OO' = \cos^2 \frac{\theta}{2} \left(P - (u \cdot P)u + u \times P \operatorname{tg} \frac{\theta}{2} \right) + (u \cdot P)u$$

$$= \cos^2 \frac{\theta}{2} P + \sin^2 \frac{\theta}{2} (u \cdot P)u + \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P$$

$$P' = 2N - P$$

$$= 2 \left(\cos^2 \frac{\theta}{2} P + \sin^2 \frac{\theta}{2} (u \cdot P)u + \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P \right) - P$$

$$= 2 \sin^2 \frac{\theta}{2} (u \cdot P)u + (\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2}) P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P$$

写成四元数的形式为：

$$(0, P') = (0, 2 \sin^2 \frac{\theta}{2} (u \cdot P)u + (\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2}) P + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} u \times P)$$

与 $q \circ (0 + P) \circ q^{-1}$ 的结果完全一致。

□

从上面就可以定义旋转算子 \mathbf{R} ：

清华大学综合论文训练

$$R_q(p) = q \cdot p \cdot q^{-1}$$

3.3 使用四元数表示旋转变量之间的距离

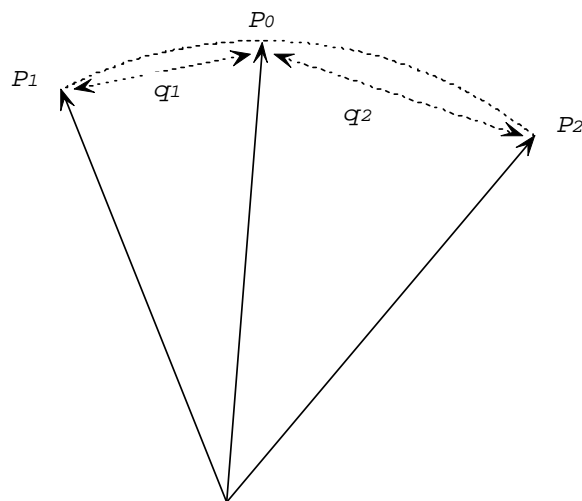


图 8 四元数之间的距离

如图, $P_0 \xrightarrow{q_1} P_1, P_0 \xrightarrow{q_2} P_2$

那么有:

$$(0, P_1) = q_1(0, P_0)q_1^{-1}$$

$$(0, P_2) = q_2(0, P_0)q_2^{-1}$$

可以证明 $P_1 \xrightarrow{q_2q_1^{-1}} P_2$

证明:

$$\begin{aligned} & q_2q_1^{-1}(0, P_1)(q_2q_1^{-1})^{-1} \\ &= (q_2q_1^{-1})q_1(0, P_0)q_1^{-1}(q_2q_1^{-1})^{-1} \\ &= q_2(q_1^{-1}q_1)(0, P_0)(q_1^{-1}q_1)q_2^{-1} \\ &= q_2(0, P_0)q_2^{-1} \\ &= (0, P_2) \end{aligned}$$

□

可见由 q_1 产生的旋转必须通过 $q_2q_1^{-1}$ 的额外旋转才能到达 q_2 , 或者说 q_1 和 q_2

清华大学综合论文训练

之间的距离为 $q_2 q_1^{-1}$ 。我们也可以使用对应的四元数的实部，即 $\text{Re}(q_2 q_1^{-1}) = \cos \theta$ 对应的角度来表示两个四元数之间的距离，即：

$$\text{distance}(q_1, q_2) = \arccos \text{Re}(q_2 q_1^{-1})$$

3.4 插值特性

在重建动画的时候，我们关心的是四元数的插值特性。

一个旋转变换映射为一个单位四元数，整个旋转群就映射到四元数空间的单位球面（从几何上是一个四维的球面）。因此四元数的插值公式对应的应该是球面插值。如果只采用矢量的线性插值，那么插值曲线不在球面上，而是在四维球的内部。这样会导致插值的错误。为了保证稳定的旋转，只能使用球面线性插值的方法。

第一个插值公式可以从群论理论中得到，使用的是 S^3 和 $SO(3)$ 的同构：

$$S\text{lerp}(q_1, q_2; u) = q_1 (q_1^{-1} q_2)^u$$

这个公式适合理论推导，对应用计算复杂性高。

第二个插值公式可以从 4D 几何理论中得到：

$$S\text{lerp}(q_1, q_2; u) = \frac{\sin(1-u)\theta}{\sin \theta} q_1 + \frac{\sin u\theta}{\sin \theta} q_2$$

这儿的 θ 满足 $\theta = \arccos \text{Re}(q_2 q_1^{-1})$ ，即 θ 为两个四元数的夹角。为了符号运算的方便，简单记做 $q_2 \cdot q_1 = \cos \text{Re}(q_2 q_1^{-1}) = \cos(q_2 \wedge q_1)$ 。可以证明上述记号可以看成是四维矢量空间中一般的点积。

证明是很简单的。因为旋转空间是封闭的，所以设 q_1 ， q_2 分别表示插值的基准四元数，两者的夹角为 θ 。设 q 为和 q_1 ， q_2 在一个大圆上的插值点，离开两个基准点的角度分别为 $u\theta$ 和 $(1-u)\theta$ 。那么 q 可以表示为 q_1 ， q_2 的线性组合，即

清华大学综合论文训练

$$q = aq_1 + bq_2$$

很容易得到 $q \cdot q_1 = \cos u\theta$ 和 $q \cdot q_2 = \cos(1-u)\theta$ 。

结合第一式，可以得到 $\begin{cases} a + b \cos \theta = \cos u\theta \\ a \cos \theta + b = \cos(1-u)\theta \end{cases}$ ，从中很容易得到：

$$b = \frac{\sin u\theta}{\sin \theta}, a = \frac{\sin(1-u)\theta}{\sin \theta}$$

但是，这种插值存在二义性。

因为根据旋转特性，有下列关系：

$$R_q(p) = q \cdot p \cdot q^{-1} = -q \cdot p \cdot (-q)^{-1} = R_{-q}(p)$$

所以 q 和 $-q$ 对应同一个旋转。这是四元数表示的奇特之处。

那么在四元数的大圆上，我们就必须判定究竟是在 q_1, q_2 还是在 $q_1, -q_2$ 之间进行插值。在没有任何信息的情况下，这两种插值都没有理由认为是错的。虽然其中一个对应的是大圆弧，一个对应的是小圆弧。但是实际的旋转情况可能是其中之一，也可能不是其中的任何一种情况。因为实际两个基准旋转之间可能相差了超过一个周角的旋转。

在本文中，我们选择小圆弧。一方面解有了唯一性，另一方面由于运动捕捉的数据率是 60f/s 甚至 120f/s，在这样密集的数据率下，关键帧率即使很低我们也可假设数据之间的关系仍然是比较大的。因为在关键帧的提取和恢复中，我们只使用球面插值从关键帧恢复原始帧序列。但是在关键帧率比较低的时候，由于相邻两帧的旋转比较剧烈，所以往往可能是大圆弧插值是正确的，很多时候甚至还要加上一圈。因此在插值恢复的时候，如果关键帧率比较低，那么很容易会出现旋转插值错误，从而导致插值的结果出现强烈的抖动。在这种情况下，四元数插值恢复的效果远远不如线性插值的效果。

为了进行小圆弧插值，可以考虑 $q_1 \cdot q_2$ ，这儿的乘法是普通的点积。如果结果为负数，那么 q_2 取反。

清华大学综合论文训练

3.5 单位四元数的指数表示

单位四元数 $q = \cos \theta + u \sin \theta$ 。根据四元数的定义，我们可以得到其类似于复数的欧拉公式的表示形式：

$$q = \exp(u\theta)$$

类似的可以定义对应的可以得到四元数的幂函数

$$q^t = (\cos \theta + u \sin \theta)^t = \exp(ut\theta) = \cos t\theta + u \sin t\theta$$

和四元数的对数函数

$$\log(q) = u\theta$$

实际上，四元数是复数的推广—超复数的一种。这就不难理解为什么它的性质和复数如此相似了。

3.6 四元数相对于欧拉角在表示旋转中的优越性

欧拉角是传统的表示旋转的数学方法。

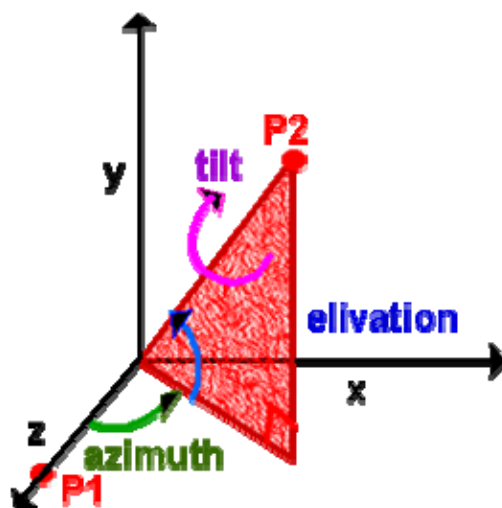


图 9 Euler 角

通过绕三个正交轴的旋转操作的组合，可以得到对应的旋转。在实际的使用中，一般是通过下面的三个矩阵的一定顺序的组合来得到最终的旋转结果：

清华大学综合论文训练

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

那么如果按从 xyz 的旋转顺序, 可以得到

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \beta \cos \gamma & \cos \beta \sin \gamma & -\sin \beta & 0 \\ \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \cos \beta & 0 \\ \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

注意这儿我们使用的是右乘矩阵 (即 $v' = vR$), 因此在形式上与有些文献可能不同。

欧拉角的表示相对于四元数的表示有如下几个缺点:

1. 存在万向节死锁 (Gimbal Lock) 的现象:

在 $R(\alpha, \beta, \gamma)$ 中令 $\beta = \frac{\pi}{2}$, 可以得到

$$R(\alpha, \frac{\pi}{2}, \gamma) = \begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin(\alpha - \gamma) & \cos(\alpha - \gamma) & 0 & 0 \\ \cos(\alpha - \gamma) & -\sin(\alpha - \gamma) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

可见退化为绕 x 轴的旋转，无法绕 z 轴旋转。即出现了旋转自由度的丧失，而且这种情况显然是无法通过改变旋转操作的顺序来完成的。

死锁情况导致欧拉角无法描述所有的旋转，也就是说对旋转操作，欧拉角是不完备的。而四元数是完备的，可以描述所有的三维旋转。

2. 插值中的问题

最简单的对欧拉角进行插值的方法是对三个欧拉角分量分别进行插值。但是这种方法默认了插值旋转和插值平移是相同的。实际上，旋转和平移有本质的不同。最关键的一点，平移矩阵满足交换率，而旋转矩阵不满足交换率。同时，欧拉角的三个分量实际上不是独立的，三者是有内在的关系的，而平移的三个分量是相互独立的。同时欧拉角还与一定的坐标系统相关联。

旋转不是 **Euclidean** 量，在正常的情况下可以使用经典的插值算法如 **Hermite** 样条、**Bezier** 样条或光顺算法如 **b** 样条，但是由于 1 中讨论的死锁性质往往会产生糟糕的效果。因为插值的结果可能恰好其中的一个分量为 $\frac{\pi}{2}$ ，接下来的操作实际上已经无法完备的表示旋转了。

但是实际使用中，正向运动学、关键帧提取、运动编辑操作如替代映射、混合操作、信号处理方法都需要由角度插值得到的旋转矩阵来进行正向运动学计算，因此 **Euler** 角的奇异性问题就必须处理。逆向运动学操作对 **Euler** 角的奇异性就更敏感了，因为基本上要使用时空约束最优化的方法来得到对应的运动轨迹，如果出现了奇异性，真正的满足约束的最优解无法取得，运动编辑的结果会出现明显的错误或明显的人造痕迹。

通常处理 **Euler** 角奇异性的方式是进行仔细的预滤波。

而这些问题，四元数几乎都完美的解决了。

3.7 结论

四元数力图确定两个空间朝向（相对于一个固定方向的两个不同的旋转）之间的唯一的简单、稳定、绕某一轴的旋转。它没有欧拉角的种种问题：万向节死锁、与坐标系统的相关性以及非常致命的插值奇异值问题。但是实际上，由于采样频率的原因，很可能我们会对不得不面对旋转欠采样的问题。尤其是在关键帧率较高的情况下，相当于旋转的采样率比较低，使用四元数插值恢复得到的旋转是错误的，对应过慢或过快的旋转。从宏观上，在序列中会出现振荡（角度差异）。而且，由于人眼对角度的差异比对长度的差异敏感，所以这种振荡非常显眼。在这种情况下，采用线性插值的效果要更好一些。

第4章 微分几何简介

古典微分几何的本质是使用一些分析量来描述参数化曲线特征。在这儿我们使用它作为一种工具来描述人体的整体运动。更详细的介绍可以参考[17, 21]。

4.1 曲线的参数方程表示

空间曲线的几何性质本质上是不依赖于坐标系的选择的。但是如果使用曲线的显函数形式

$$y = f(x_1, x_2, \dots, x_n)$$

或隐函数形式

$$F(y, x_1, x_2, \dots, x_n) = 0$$

我们对曲线的分析不得不依赖于对应的坐标系。

为了摆脱对坐标系的依赖，我们可以使用曲线的参数方程表达

$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases} \quad (a \leq t \leq b)$$

我们固然可以把参数 t 看作是时间变量，从而曲线可以看作空间质点滑过的轨迹。但一般说来，它不具有时间这个物理意义，而且它的选择不是唯一的。例如我们可以在曲线弧长坐标系中表示曲线

$$\begin{cases} x = x(s) \\ y = y(s) \\ z = z(s) \end{cases} \quad (0 \leq s \leq S)$$

其中 s 为从当前点到起始点的曲线弧长。

清华大学综合论文训练

先作如下的约定：

$$\dot{\mathbf{r}}(t) = \frac{d\mathbf{r}}{dt}, \ddot{\mathbf{r}}(t) = \frac{d^2\mathbf{r}}{dt^2}$$

$$\mathbf{r}'(s) = \frac{d\mathbf{r}}{ds}, \mathbf{r}''(s) = \frac{d^2\mathbf{r}}{ds^2}$$

4.2 Frenet 标架下的几何量：切矢，挠度和曲率

使用参数方程表示之后，我们就可以方便的得到空间曲线的几个重要的几何性质：

1) 切向量（切矢）

假设空间曲线 C , $\mathbf{r} = \mathbf{r}(t) = \{x(t), y(t), z(t)\}$ ($a \leq t \leq b$) 至少三阶连续，即 $\mathbf{r}(t) \in C^3$ 。

$$\text{我们定义切向量为 } \dot{\mathbf{r}}(t) = \lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{r}}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t}$$

$$\text{根据 } \Delta s \approx |\Delta \mathbf{r}| \rightarrow \frac{\Delta s}{\Delta t} \approx \frac{|\Delta \mathbf{r}|}{\Delta t} \rightarrow \frac{ds}{dt} \approx \frac{|d\mathbf{r}|}{dt}$$

$$\text{可以得到弧长的计算公式为 } s(t) = \int_{t_0}^t |\dot{\mathbf{r}}(t)| dt$$

用 $s(t)$ 代替 t 作为参数方程的变元，我们可以得到弧长坐标系下的参数方程：

$$\mathbf{r} = \mathbf{r}(s)$$

选取弧长作为曲线参数的一个好处就是，曲线上每一点的切向量都可以规范化为单位向量，因为 $|\frac{d\mathbf{r}}{ds}| = 1$ 。

我们把曲线在一点的单位切向量记做

$$\mathbf{T}(s) \triangleq \mathbf{r}'(s)$$

2) 曲率

清华大学综合论文训练

空间曲线在一点曲率定义为

$$k(s) = |\mathbf{r}''(s)|$$

若 $k(s)$ 不为 0，其倒数

$$R(s) = \frac{1}{k(s)} = \left| \frac{d\mathbf{T}(s)}{ds} \right|$$

可以证明

$$k(s) = \lim_{\Delta s \rightarrow 0} \left| \frac{\overline{\mathbf{r}(s + \Delta s), \mathbf{r}(s)}}{\Delta s} \right| = \lim_{\Delta s \rightarrow 0} \left| \frac{\Delta \theta}{\Delta s} \right|$$

称为曲线在该点的曲率半径。

3) 主法向量

$\mathbf{r}''(s)$ 一般不是单位向量，我们把其规范化的结果称为主法向量，即

$$\mathbf{N}(s) \triangleq \frac{\mathbf{r}''(s)}{|\mathbf{r}''(s)|}$$

4) 从法向量

在同一点，定义曲线的从法向量为

$$\mathbf{B}(s) \triangleq \mathbf{T}(s) \times \mathbf{N}(s)$$

5) Frenet 标架

上面定义的三个相互正交的单位矢量：单位切向量、主法向量、从法向量构成曲线上的动坐标系，称为 Frenet 标架，即

$$\text{Frenet 标架} \triangleq \{\mathbf{T}(s), \mathbf{N}(s), \mathbf{B}(s)\}$$

应用到运动捕捉数据上，即为下图：

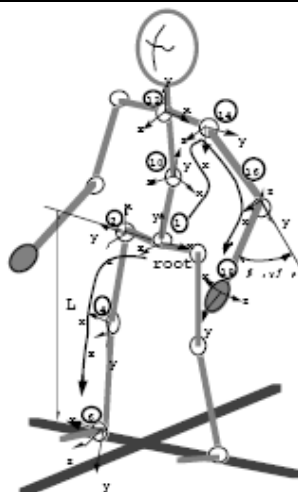


图 10 运动捕捉对象上的 Frenet 标架

6) Frenet 标架中的平面

密切平面 $\triangleq \{\mathbf{T}(s), \mathbf{N}(s)\}$;

从切平面 $\triangleq \{\mathbf{T}(s), \mathbf{B}(s)\}$

法平面 $\triangleq \{\mathbf{N}(s), \mathbf{B}(s)\}$

显然曲线在一点的特殊直线和平面的方程可以表示如下:

切线: $\rho = \mathbf{r}(s) + u\mathbf{T}(s)$

主法线: $\rho = \mathbf{r}(s) + u\mathbf{N}(s)$

从法线: $\rho = \mathbf{r}(s) + u\mathbf{B}(s)$

法平面: $\mathbf{T}(s) \cdot (\rho - \mathbf{r}(s)) = 0$

从切平面: $\mathbf{N}(s) \cdot (\rho - \mathbf{r}(s)) = 0$

密切平面: $\mathbf{B}(s) \cdot (\rho - \mathbf{r}(s)) = 0$

7) 挠度

清华大学综合论文训练

定义挠度为 $\tau(s) = -\mathbf{B}'(s) \cdot \mathbf{N}(s)$

$$\text{经计算可得 } \tau(s) = \frac{(\mathbf{r}'(s), \mathbf{r}''(s), \mathbf{r}'''(s))}{|\mathbf{r}''(s)|^2}$$

4.3 曲线几何量计算方法

结合运动捕捉系统，我们解释一下根节点曲线的切向量、曲率、挠度的意义

切向量：对应身体整体运动的速度。其大小对应身体整体运动的大小。其方向对应运动的方向；

曲率：对应身体整体的转动属性。如直线走、跑的曲率小，跳、翻等的曲率大；

挠度：对应身体整体运动的复杂性。如翻滚的复杂性远比走大。

通过这些几何量，可以得到一个运动中的控制点，即那些曲率很大、挠度也很大的点（我们不考虑切矢量的大小，其原因将在下文中说明），它们描述了整体运动中的关键点，或特征。把它们的评价值加到原有关键帧属性（仅仅考虑局部的运动）的评价中去，得到的关键帧可以令人满意的恢复原帧序列。原因很简单，这时我们对关键帧的评价同时考虑了整体运动和局部细节运动的复杂程度和代表性。这样的关键帧可以比较完整的描述运动，尤其是对于比较复杂的运动。

切向量、曲率、挠度虽然在曲线弧长坐标系内的形式很简单，但是实现的时候对应于每一个点要得到其到起始点的弧长是很麻烦的。即使是给出了曲线的方程这也不是容易做到的，更何况我们面对的是空间曲线上的离散点。因此我们使用对应于常规曲线参数方程形式的切向量、曲率、挠度形式。

可以证明，在这个坐标系下：

$$\begin{cases} \mathbf{T}(t) = \frac{|\dot{\mathbf{r}}(t)|}{\dot{\mathbf{r}}(t)} \\ \mathbf{B}(t) = \frac{\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|} \\ \mathbf{N}(t) = \mathbf{B}(t) \times \mathbf{T}(t) \end{cases}$$

对应的曲率和挠度为：

$$\tau(t) = \frac{(\dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t))}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}$$

$$k(t) = \frac{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}{|\dot{\mathbf{r}}(t)|^3}$$

因此为了计算每一个点的切向量、曲率、挠度，我们需要知道每个点的一阶、二阶、三阶导数。由于我们只有离散的数据点，所以我们需要计算对应的点的数值微分。这将在下一章描述。微分几何量的一些计算问题也将在下一章讨论。

第5章 数值微分简介

数值微分在这儿用于计算挠度和曲率，它是成熟的技术，但使用的时候针对特定的问题有值得注意的细节。下面只列出了计算需要的公式，对应的理论和更好的近似方法可以参考[15]。

5.1 数值微分基础

为了计算离散曲线上各个点的速率、挠度和曲率，首先需要计算各个点上的一阶、二阶、三阶导数。

由于只有离散的曲线点，所以可能采用的方法只有两种：

- 1) 使用样条技术进行曲线的拟合。拟合得到的高次曲线可以方便的计算各个点上的一阶、二阶和三阶导数。
- 2) 采用离散微分的方法计算各个点上的一阶、二阶、三阶差分来近似导数。

方法 1 基本上对计算速率、挠度和曲率是不太合适的。

因为如果只采用三次样条曲线，那么可以知道在两个插值点之间的点的数值、一阶导数、二阶导数、三阶导数的值分别为：

令

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j}, B = \frac{x - x_j}{x_{j+1} - x_j}, C = \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2, D = \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2$$

则

$$y = Ay_i + By_{i+1} + Cy_i'' + Dy_{i+1}''$$

$$y' = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{3A^2 - 1}{6}(x_{j+1} - x_j)y_j'' + \frac{3B^2 - 1}{6}(x_{j+1} - x_j)y_{j+1}''$$

清华大学综合论文训练

$$y'' = Ay_j'' + By_{j+1}''$$

$$y''' = \frac{y_{j+1}'' - y_j''}{x_{j+1} - x_j}$$

其中 $x \in (x_j, x_{j+1})$ 。

注意到尽管一阶、二阶导数都是连续的，但是三阶导数不连续。插值点上的三阶导数很难定义，这是它很大的一个问题。如果使用更高次的样条，一方面计算的复杂性提高，另一方面会出现过拟合的问题，算法的稳定性比较差。

同时这种方法要求曲线是欧氏曲线，但是人体的运动未必是这样的，尤其是复杂的运动，如翻筋斗、掷球等。使用传统的样条方法求解很容易出现奇异值，精度也不能达到要求。更致命的是，它的计算速度非常的慢。

最后，从方法论的角度看，由于曲线的拟合问题在本质上是一个比求曲线上离散点的各阶导数困难的问题。我们是试图通过解决一个更困难的问题来捎带解决一个简单的问题。从精度和效率上看，都不如直接解决这个简单问题来得好。Vapnik 在他的《统计学习理论》一书中对这个问题有深入的讨论。我们作了一些数值实验，也证明了这个问题。尤其是对复杂的跳跃问题，方法 1 无法避免奇异解，在方法 2 中我们完全可以避开。

方法 2 是一种直接的、有效的方法。而且由于硬件采样的性质，我们面对的是等间隔的采样点（采样间隔是帧率的倒数），所以使用这种方法更加合适。

5.2 常用数值微分公式

通过数值微分的理论技术（本质上是 Taylor 展开式的误差估计），可以得到下面一系列的结果：

1) 精度为 $O(h^2)$ 的中心差分公式：

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h}$$

$$f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2}$$

清华大学综合论文训练

$$f^{(3)}(x_0) = \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3}$$

2) 精度为 $O(h^4)$ 的中心差分公式:

$$f'(x_0) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h}$$

$$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2}$$

$$f^{(3)}(x_0) = \frac{-f_3 + 8f_2 - 13f_1 + 13f_{-1} - 8f_{-2} + f_{-3}}{8h^3}$$

利用 Lagrange 插值多项式可以得到前向后向差分公式:

精度为 $O(h^2)$ 的前向差分公式:

$$f'(x_0) = \frac{-3f_0 + 4f_1 - f_2}{2h}$$

$$f''(x_0) = \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2}$$

$$f^{(3)}(x_0) = \frac{-5f_0 + 18f_1 - 24f_2 + 14f_3 - 3f_4}{2h^3}$$

精度为 $O(h^2)$ 的后向差分公式:

$$f'(x_0) = \frac{3f_0 - 4f_{-1} + f_{-2}}{2h}$$

$$f''(x_0) = \frac{2f_0 - 5f_{-1} + 4f_{-2} - f_{-3}}{h^2}$$

$$f^{(3)}(x_0) = \frac{5f_0 - 18f_{-1} + 24f_{-2} - 14f_{-3} + 3f_{-4}}{2h^3}$$

运动捕捉数据的帧率是 120 帧/s 或 60 帧/s, 因此通过数值微分的手段可以得到较高精度的挠度、曲率和速率。而且这种方法对于运动轨迹的复杂程度要求不高, 可以较好的估计复杂运动的几何特征。

由于我们面对是曲线上的有限的采样点, 所以计算各阶导数的时候对中间的点可以采用精度 $O(h^4)$ 中心差分公式以得到较高的精度, 对于开始时的几个

清华大学综合论文训练

点和结束时的几个点可以分别选择精度为 $O(h^2)$ 的中心差分公式或前向、后向差分公式，因为我们此时无法使用中心差分公式。

5.3 数值微分计算挠度和曲率的细节问题

使用数值微分来计算挠度和曲率的时候，还要注意一个细节上的问题。

对于比较缓慢的运动，计算后得到的一阶导数非常小，由于计算机字长的限制，会被舍入到 0。以后计算该点的挠度和曲率的时候，由于公式

$$\tau(t) = \frac{(\dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t))}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}$$
$$k(t) = \frac{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}{|\dot{\mathbf{r}}(t)|^3}$$

中的分母在一阶导数为 0 的时候也为零，所以会出现数值错误。

实际上此时挠度和曲率计算式中的分子也是接近于 0，所以对于有限字长的计算机而言，这就是一个 $\frac{0}{0}$ 的问题。

应该如何解决这个问题呢？由于 $\dot{r}(t) \approx 0$ 对应较缓慢的运动，而缓慢的运动不可能进行急剧的旋转，所以它对应的曲率半径很大，即曲率应该很小。同时缓慢运动的复杂性也低，对应的挠度也应该很小。所以对于无限字长的计算机，曲率和挠率的计算结果都应该接近于 0。

因此，对这个计算奇异的问题的解决方法就很清楚了：

如果一阶导数足够大：

$$\tau(t) = \frac{(\dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t))}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}$$
$$k(t) = \frac{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}{|\dot{\mathbf{r}}(t)|^3}$$

否则

$$\tau(t) = 0$$

$$k(t) = 0$$

清华大学综合论文训练

编程实践证明，这种方法可以有效的解决上述的数值问题。

第6章 运动捕捉数据

运动捕捉设备通过接收运动捕捉对象表面的标志点发出的光学、电磁、机械信号来得到运动捕捉数据。人体运动捕捉数据近似对应人体骨架数据。

虽然运动捕捉和运动编辑的终极目标是产生在非常细微的地方也显得非常逼真的动画对象，但是由于当前运动捕捉系统精度的制约，我们只关心人体大范围的运动。实际上，更为细致的运动，如肌肉运动、面部表情的变化，都可以通过人体大范围运动的非线性映射得到，但是这已经是另一个专门的研究方向了。

6.1 运动捕捉数据的结构化

运动捕捉设备得到的数据是没有结构的原始数据，对应人体表面光学标志点的位置。要从运动捕捉数据映射到人体运动数据，必须赋予原始数据一定的结构和意义。

首先，必须把每个标志点映射到人体对应部位。这与光学标志点在人体表面的分布相关，也和运动捕捉的具体对象相关。因此在初始化的时候必须人工手动标定。

其次，人体运动捕捉数据得到的是人体关节数据，而各关节之间是存在刚性连接的。为了表示这种连接，在相联系的关节间应添加连线。这种连接需要知道运动捕捉对象的身体结构知识和光学标志点分布的知识，所以通常只能人工标定。

此步之后，运动捕捉数据的结构大致如下：

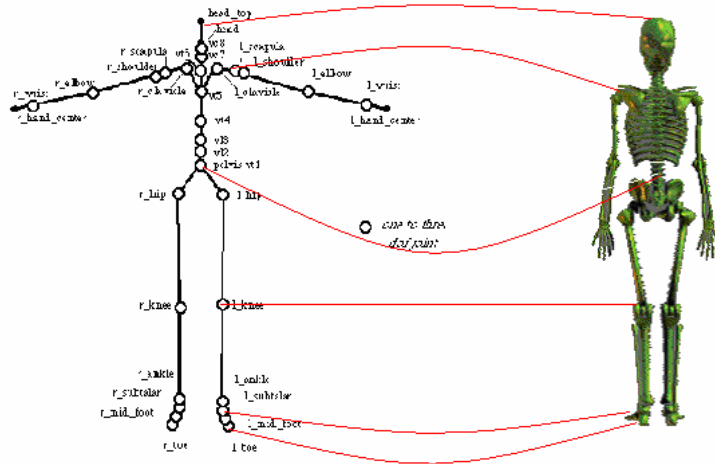


图 11 人体和运动捕捉数据的对应

再次强调，这种从数据空间向物理空间的映射很大程度上是主观的。使用不同的运动捕捉设备，标志点的位置配置不同，使用不同的运动捕捉对象，得到的结果经常是不同的。

现在要进行显示已经没有问题了。但是为了进行关键帧的提取和运动编辑的工作，我们还应该进一步结构化数据。

如果要在三维空间中表示和控制动画角色，我们需要知道或设定每个关节在各个时刻的位置和走向（这里假设是一个刚性的三维物体，相邻关节的距离不变），即它们的自由度（DOF）。注意到，为了产生合法的三维运动，我们必须加上一定的约束，如相邻关节的距离恒定、腰关节虽然有三个自由度，但是实际上旋转有一定的范围，等等。这会导致求解一个非常复杂的约束方程组。而且在每个时刻，必须反复应用和求解这个约束方程组，这对虚拟现实实时性的应用来说是无法忍受的。

对应于人体运动捕捉数据，最有效的方法是采用层次化的方法，因为人体及动物的骨架结构本质上是结构化的。从图 11 可以看到，骨架基本上是一个树形结构，树根可以选择为盆骨的位置。从运动的角度看也是合理的，因为如果把人体看作一个质点的话，为了表示它的全局运动，我们也基本上会选择盆骨的运动来代表整个人体的运动，忽略其他部分的旋转，因为它们都是细节。

建立树形结构之后，整个运动捕捉数据的描述就比较简单了，而且一些基

本的约束可以隐含在结构中，无须显式的表达。

这时只有根节点（盆骨）有完整的自由度，可以平动，也可以转动。当然，考虑到实际的人体，还可以加上一些范围约束。对于其他的节点，由于包含了刚性约束，所以相对于各自的父节点没有平动自由度，只有转动自由度。对于三维的关节，应该注意到不同的关节的转动自由度是不同的。肘关节相对于肩关节的转动自由度是 3，而踝关节相对于膝关节的自由度可以认为只有 1。同时可以加上旋转范围的约束。

这种层次化的表述方法的优点主要有：

1. 可以很方便的计算各个位置（不限于关节点）在世界坐标系下的位置和朝向。这对于运动编辑中的正运动学是必要的。

为了简单起见，以二维物体为例。如假设脚部的某一点在踝关节坐标系下的坐标为 \mathbf{v} ，那么知道了各个关节在当时相对于各自父节点的旋转角度，容易得到这一点在世界坐标下的坐标为：

$$\mathbf{v}_w = \mathbf{T}_w(x, y)\mathbf{R}_w(\theta_w)\mathbf{T}_s\mathbf{R}_s(\theta_s)\mathbf{T}_c\mathbf{R}_c(\theta_c)\mathbf{v}$$

其中 w 对应盆骨的关节，由于它是根节点，所以有完整的自由度：平移的两个自由度和旋转的自由度。 s 和 c 对应膝关节和踝关节，由于不是根节点，所以平移矩阵是常矩阵，对应和父节点的常偏移。它们都有各自的旋转自由度。

到了三维的情形，由于它自身的复杂性，我们在下面专门讨论。

2. 保证了数据结构的简单。在每个关节的控制参数的个数和关节点在父节点坐标系下的自由度成正比。

3. 简化了约束的表示形式。如果不考虑旋转的限制，那么任何控制参数的组合都可以产生合法的运动数据，因为必要的约束都已经自然的含于树形的结构中了。如果要加上旋转的限制，也仅仅是线性的约束，因为这时可以在父节点坐标系下表述，’

4. 算法实现的高效性和简单性。许多 3D 图形 API，如 OpenGL、VRML 都提供支持层次表示的机制[16]。

如 OpenGL 中的 transaction hierarchies，通过变换矩阵堆栈可以自然的实现对应的功能。

如：

```

drawRoot ();                                //draw Root, layer 0
glPushMatrix ();
    glRotatef (alpha_1, x0_1, y0_1, z0_1);
    glTranslatef (x_1, y_1, z_1);
    drawChild1 ();                            //draw child 1, layer 1
glPopMatrix ();
glPushMatrix ();
    glRotatef (alpha_2, x0_2, y0_2, z0_2);
    glTranslatef (x_2, y_2, z_2);
    drawChild2 ();                            //draw child 2, layer 1
    glPushMatrix ();
        glRotatef (alpha_3, x0_3, y0_3, z0_3);
        glTranslatef (x_3, y_3, z_3);
        drawChild_Child1 ();                 //draw child2' s child 1,
layer 2
        glPopMatrix ();
.....
.....
.....
glPopMatrix ();

```

即可以通过把父节点的状态推入堆栈，在绘制子节点相对父节点的平移或旋转，然后恢复父节点状态，再绘制另一个子节点。通过这种类似深度优先的算法，我们可以非常高效的利用显存作为堆栈。

VRML 语言中也有类似的语法。

5. 结构化的数据扩展非常容易，对程序的维护方便，从软件工程的角度具有优越性。

如果运动捕捉数据的精度提高、标志点的数目增多，那么可能更多的关节点会被考虑，那么采用层次的数据表示可以自然的加入新的数据。例如，在图 11 中，我们仅仅用一个关节点来表示手，如果我们设置更多的运动标志点，我们可以得到类似图 12 中手的模型。不费多少力就可以增加细节和表现力。



图 12 OpenGL 变换堆栈在手建模中的应用

6. 在编程中增加灵活性，这在下面的算法部分有对应的描述。

综上，我们可以得到对应的树形结构表示如下：

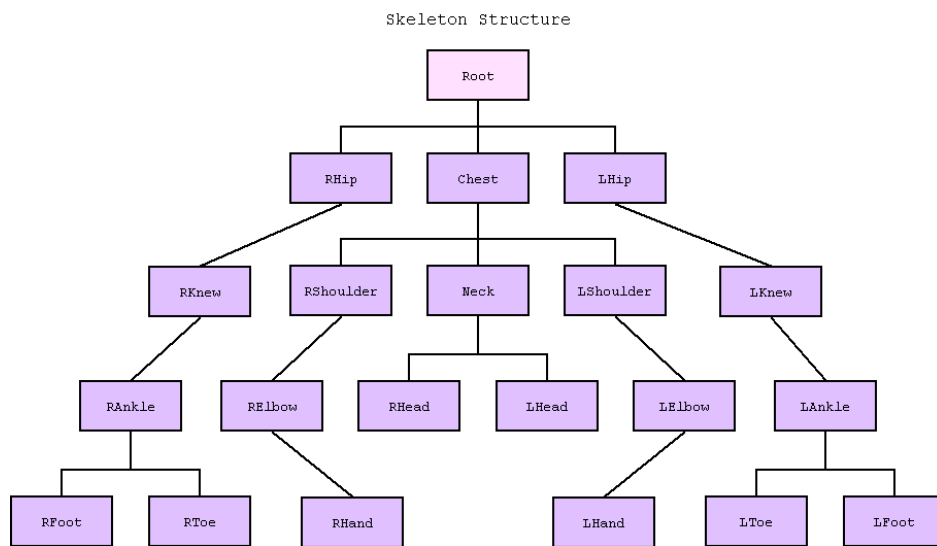


图 13 人体运动捕捉数据的树形结构

6.2 三维旋转的表示方法

三维旋转的表示方法主要有：

1. Euler 角。

已经在第三章中比较详细的介绍了它的优点和缺点，在这儿就不赘述了。

2. 矢量形式 (vector)

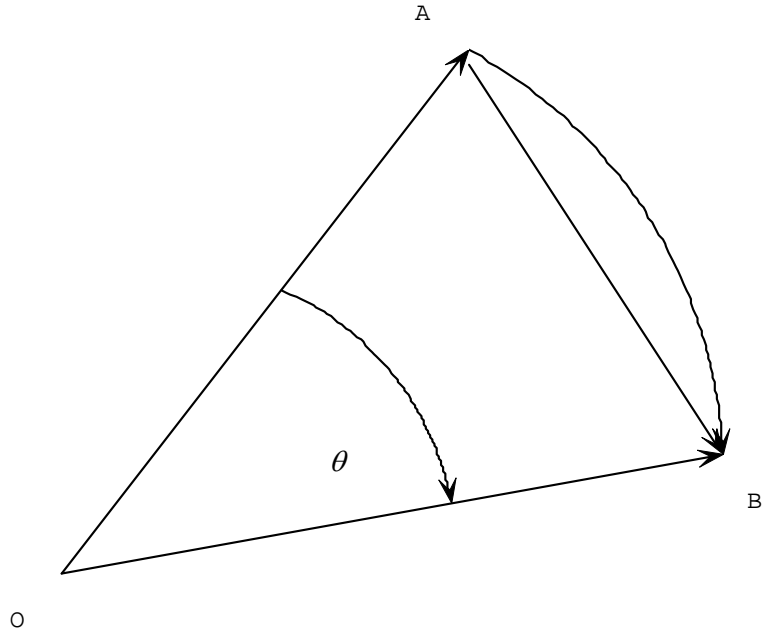


图 14 旋转的矢量表示

$\frac{|AB|}{2|OA|} = \sin \theta$ ，由于运动捕捉的帧率比较大，所以对应的 θ 比较小，我们

就可以得到

$$\frac{|AB|}{2|OA|} \doteq \theta$$

即可以使用关节点相对于父节点的矢量的变化的大小来表征旋转的大小。

这种表示的最大不足就是旋转角度无法进行插值运算，在恢复的时候有一定的困难。

3. 四元数 (quaternions)

我们主要使用的是单位四元数，它即不是标量也不是 Euclidean 量。因此我们无法使用经典的、为 Euclidean 量而设计的插值和光顺算法。适合于四元数的插值算法主要有两种：球 Bezier 曲线和闭形式的四元数曲线。前者的构造方法类似于 Euclidean Bezier 曲线的构造方法，但是它没有闭形式的解，而且不能保证二阶连续，但是实现比较容易。四元数曲线有闭形式，和 Euclidean 量的 Hermite 样条、Bezier 样条、b 样条处于平行的位置。

但是在使用四元数来表示旋转的时候必须注意，Euler 角对应的加法在这儿成了乘法，同时为了保证是单位四元数，要么进行一定的归一化，要么求对应的约束最优化问题。总之，无论进行正向运动学还是逆向运动学计算都比较复杂。

但是这是已知的唯一一种不存在角度奇异性的旋转表示。因此在本文中采取了这种方法。具体的插值算法和代数性质请参考第三章。

4. 指数映射

这种表示方法类似于四元数，但是构造的方法有所不同。它的计算较简单，本质上有奇异性。但是奇异性在采样密集的情况下可以避免。在插值恢复中我们不采用它的原因在于关键帧如果比较少的话，相互之间的差异比较大，插值时指数映射的奇异性就会显示出来。它不适合这种应用。

6.3 树形结构的优点和缺点

结合上面的讨论，树形结构的优点是很多的：

- 1.可以很方便的结构化人体数据
- 2.人体模型约束隐式的包含进来，不需要在计算时考虑约束问题
- 3.可以方便的利用硬件的加速设备
- 4.可以方便的使用旋转的四元数表示

但是树形结构也有它固有的问题，即计算的不稳定性。下面我们讨论一下这个问题。

由于子节点的坐标是通过父节点坐标来间接计算的。这虽然减少了复杂性，但是却带来了误差积累。即如果父节点坐标计算的稍有不准，那么会导致父节点的朝向发生变化。子节点的坐标是相对于父节点得到的，那么父节点朝向的微小变化会导致子节点朝向的变化。如果子节点相对于父节点的旋转半径比较大，在人体结构中即骨头的长度比较长，那么即使是微小的朝向的改变也会导致子节点位置剧烈的改变。同时越低层次的关节，如 **Foot**，**Hand** 受上层结构的误差积累的影响越大，在运动中也越不稳定。

应该注意的是，这种误差在计算中是无法避免的。因为我们反复的要从三

维绝对坐标计算四元数，在通过四元数的运算计算旋转角度，这儿的运算几乎都是非线性运算，产生的误差是比较大的。所以使用四元数插值来从关键帧序列恢复的时候，几乎不可避免的会出现底层节点的坐标很不准确的情况。

同时，这种影响对复杂高速运动的影响比对简单低速运动的影响小。因为对于简单的运动，相邻帧的差异比较小。如果有累计误差的话，很容易通过前后帧的对比看出来，同时对于简单运动关键帧率比复杂运动的关键帧率要小，这种计算误差更容易产生。而复杂运动一方面关键帧比较多，这种误差不容易产生，而且计算舍入误差很容易被实际的运动补偿掉，恢复的时候也不容易表现出来。

6.4 伪码数据结构

综上所述，我们对运动捕捉数据中一帧的描述可以采取以下的两种形式。我们没有实现 Euler 角的表示。

使用矢量的表示旋转：

```
Motion Represent
{
    Root:
    x, y, z
    Chest:
    x, y, z
    {
        Neck:
        x, y, z
        {
            RHead:
            x, y, z
            LHead:
            x, y, z
        }
        RShoulder:
        ...
        ...
    }
    ...
    ...
}
```

使用四元数/指数映射表示旋转:

```
Motion Represent
{
  Root:
  x, y, z
  Chest:
  rotQuaternion / rotExponential
  {
    Neck:
    rotQuaternion / rotExponential
    {
      RHead:
      rotQuaternion / rotExponential
      LHead:
      rotQuaternion / rotExponential
    }
    RShoulder:
    ...
    ...
  }
  ...
  ...
}
```

使用欧拉角的方法我们没有做。

第7章 运动捕捉序列的关键帧提取

7.1 关键帧提取算法

7.1.1 聚类算法整体框架

从一个运动捕捉序列中提取关键帧就是从序列中提取关键的具有代表力的元素。联想到对大量数据进行聚类的时候，聚类的中心实际上就是对数据的一个基于特征的概括。因此很自然的就会想到使用模式识别的方法来提取关键帧。如果给定了关键帧的数目，实际上就给定了聚类的数量，我们唯一要确定的，就是每一帧究竟属于哪一类。

在关键帧提取的过程中，我们知道运动捕捉数据—人体骨架关节点的三维坐标或其旋转量表示。我们也还知道对应的树形结构。但是我们对运动的结构，即对给定时间人体的骨架的位置分布

$$p(x, y, z, t) \text{ 或 } p(\theta_1, \theta_2, \theta_3, t)$$

不清楚。对于这种概率分布未知的非监督模式识别问题，很合适的一种解决方法就是动态聚类。

与一般聚类不同的是，我们现在需要对一个时序信号进行聚类。对应的，我们需要解决与特定问题相关的一些特定问题。

基于一般动态聚类的算法的捕捉数据关键帧提取

聚类算法 1 常规的动态聚类算法是在线聚类的代表算法：

```

1  begin initialize  $\eta, \theta$ 
2       $w_1 \leftarrow x$ 
3      do accepted new  $x$ 
4           $j \leftarrow \arg \min_{j'} \|x - \mu_{j'}\|$  (find nearest cluster)
5          if  $\|x - \mu_j\| < \theta$ 
6              then  $\mu_j \leftarrow \mu_j + \eta x$ 
7          else add new  $\mu \leftarrow x$ 
8               $\mu \leftarrow \mu / \|\mu\|$  (normalize weight)
9      until no more patterns
10     return  $\mu_1, \mu_2, \dots$ 
11 end

```

图 15 聚类算法 1: 常规的动态聚类算法

决定最后聚类结果的主要有序列中的元素出现的顺序和阈值 θ 。

元素出现的顺序这儿不是可控的，是运动数据本身的属性。唯一可控的元素是阈值 θ 。 θ 比较大的时候，得出的聚类个数很少但是体积很大。 θ 较小的时候，得到的聚类个数很多但是体积很小。应该注意的是，聚类是否具有实际的意义必须要经过人为的检验。如果我们没有任何其他的信息，只有数据本身，那么我们就很难确定合适的 θ 以得到比较合理的分类。

7.1.2 给定关键帧率时的调整算法

大致的，阈值和帧率、数据本身的尺度、数据的局域方差相关。帧率大，采样频率高，两帧之间的差异小，对应的阈值就应该低一些，控制关键帧的数量不要太多。数据本身的尺度对阈值也有影响，为了消除这种影响，数据应该归一化。数据的局域方差越大，说明这段时间内的运动比较复杂，应该增大阈值，提高采样率。

也可以采用自适应的阈值选择。阈值可以以一个很小的值初始化。以后计算每一帧前先动态的调整阈值。阈值的调整根据当前计算出的关键帧指标和以前的关键帧率。可能的调整算法如下：

调整算法 1

if 关键帧指标 \gg 阈值

```

then { 依然处理为关键帧， 阈值自动增加};           //可能阈值太小了

else if 关键帧指标>阈值

    {令  $p = (\text{设定关键帧数} - \text{该帧之前的关键帧数}) / (\text{总帧数} - \text{该帧之前的帧数})$ };

        产生[0, 1]之间的随机数  $r$ ,

        if  $r < p$ 

            该帧设定为关键帧;

        else

            该帧不为关键帧;

    }

else { 该帧不为关键帧; 阈值自动减小}           //可能阈值太大了

```

调整算法 2

令 $p = \frac{1}{1 + e^{-(x-x_0)}}$ ，其中 x_0 为当时的阈值

产生[0, 1]之间的随机数 r ,

```

    if  $r < p$ 

        该帧设定为关键帧;

    else

        该帧不为关键帧;

    令  $s =$  该帧处理后的总关键帧数;

    if  $s < (\text{设定关键帧数} / \text{总帧数}) * \text{当前已处理帧数}$ 

        阈值减小

    else if  $s > \text{设定关键帧数}$ 

```

阈值增大

实验的时候阈值增大和减小的幅度都选为当时值的 1/2。

这种自适应算法有些类似模拟退火算法。但试验中发现做基本上是不可行的。

实际上这两种算法的失败之处在于过分强调了贪心算法的能力。它的调整总是滞后于运动的，对于简单的物理现象和工程现象（如控制中的一些问题）这种调整可能会有用，因为那些问题基本上可以用线性的微分方程来刻画。人体运动实在很复杂，这种调整算法实际上作用不大。

当然如果不是在线的运动，要求给定关键帧率，求关键帧集合还是容易的，因为可以用 C-均值聚类的方法或其变种（如模糊 C-均值等）：

C 均值算法：

a. 选择把 N 个样本分成 C 个聚类的初始划分，计算每个聚类的均值 m_1, m_2, \dots, m_c 和各样本 y 与均值 m_i 的误差平方和对所有类相加后的结果

$$J_e = \sum_{i=1}^c \sum_{y \in \Gamma_i} \|y - m_i\|^2$$

b. 选择一个备选样本 y，设 y 在 Γ_i 中

c. 若 Γ_i 中的元素个数 N_i 为 1，那么转 b

$$d. \text{ 计算 } \rho_j = \begin{cases} \frac{N_j}{N_j + 1} \|y - m_i\|^2 & j \neq i \\ \frac{N_i}{N_i - 1} \|y - m_i\|^2 & j = i \end{cases}$$

e. 对于所有的 j，如果 $\rho_k \leq \rho_j$ ，则把 y 从 Γ_i 转移到 Γ_k 中去

f. 重新计算 m_i, m_k ，修改 J_e

g. 迭代 N 次 J_e 不变，输出关键帧

这种做法是一种离线情况下算法，对于一些实时的应用是不合适的。在这儿我们只提一下，没有做具体的实验，因为时间代价太大了。

7.1.3 关键帧属性度量—传统方法

这儿又涉及到如何评价捕捉序列中的一帧是关键帧的可能性的问题。传统的方法主要通过该帧和前面的所有提取出来的聚类的中心的差别的最小值来进行评价。在下面我们会讨论这种度量实际上无法完整的度量一帧是关键帧的可能性。

在关键帧可能性的评价中，我们需要知道两件事情：

- 1) 如何在当前的数据结构下计算两帧的差异
- 2) 如何在当前的数据结构下计算聚类的中心

如果使用 Euler 角，两帧之间的差异比较简单。使用四元数的话，差异的计算比较复杂。因此我们仅仅详细讨论使用四元数时的情况。

使用四元数的时候，我们使用前文说的树形结构表示运动，写成矢量的形式，即 $(q_1^N, q_2^N, \dots, q_n^N)$ ，其中 q_i^N 为第 N 帧的第 i 个 Marker 点在其父节点坐标系下相对于第一帧同一个 Marker 点的旋转量的四元数形式。

如前所述，两帧的关键帧不考虑根节点的位移。所以我们应该计算各个 Marker 点在两帧之间的差值，自然还要根据各个 Marker 点在树形结构的位置设置不同的权值。

从第三章的四元数的代数和性质，我们知道两个四元数之间的距离可以使用下面的式子度量：

$$distance(q_1, q_2) = \arccos \operatorname{Re}(q_2 q_1^{-1})$$

考虑到不同节点的权值，就可以使用

$$distance(F_1, F_2) = \sum_{i=1}^N \alpha_i distance(q_1^i - q_2^i)$$

来度量两帧之间的距离。

下面我们关心的是如何计算聚类中心。使用 Euler 角的时候，我们一般是使用 Euler 角 (α, β, γ) 的各个分量的平均值来近似计算聚类中的旋转量中心。

但是如果使用四元数的话，如果仅仅使用四元数的平均值的来代表聚类旋转量的中心就不合理。因为如果这样，聚类的中心 $\sum_{i=1}^N \frac{1}{N} q_i$ 并不是一个单位四元数，而旋转量的四元数表示必须是单位四元数。由于单位四元数关于线性组合不是封闭的（即组合 $\sum_{i=1}^N \alpha_i q_i$ 不是四元数，其中 $\sum_{i=1}^N \alpha_i = 1$ ）

我们一共尝试了两种方法。

第一种方法，是认为单位四元数的中心四元数到各个四元数的旋转角度是一样的。

类比一般向量集合 (v_1, v_2, \dots, v_n) 的聚类中心实际上是求 v 使得

$$\sum_{i=1}^n \|v - v_i\|$$

达到最小。

我们实际要做的就是选择四元数 $\cos \theta + u \sin \theta = \exp(u\theta)$ 使得

$$\sum_{i=1}^n \arccos(\exp(u\theta - u_i \theta_i) \cdot 1)$$

达到最小。

这个最优化问题没有解析解，但是我们可以使用优化算法来得到比较好的结果。

第二种方法是一种近似的解法，即虽然 $\sum_{i=1}^N \frac{1}{N} q_i$ 并不是一个单位四元数，

但是它的确是在 4 维空间内这些 4 维矢量的聚类中心。可以想象如果聚类比较紧密的话，它在 4 维空间的单位球面上的投影近似可以代表旋转量的中心。因

此我们可以使用

$$\frac{\sum_{i=1}^N \frac{1}{N} q_i}{|\sum_{i=1}^N \frac{1}{N} q_i|}$$

来作为近似的结果。

实验的结果表明这种近似的方法工作的非常好。其原因是捕捉数据由于采样率比较高，所以帧间的旋转量不是很大，正确聚类的旋转量聚合的很紧密。如果使用智能优化算法求解方法 1 得到的最优值，代价太大，不可取。

通过上面的算法，我们可以较好的从运动捕捉帧序列中提取关键帧。

7.1.4 聚类算法改进—考虑时序约束

上面所述的关键帧提取算法从运动序列中提取出少量的帧来表示整个的运动序列，但是这个算法是为非时序的数据而设计的，提取出来的关键帧没有保持时序关系。即很后面的帧可能聚类为很前面的类。如果运动中有周期运动的话，这种聚类的方法会过滤掉周期运动只保持一个运动的典型周期。当然在运动一个周期内，这种方法还是可以提取得到具有时序信息的关键帧的。

这种聚类特性对于建立运动捕捉数据库是非常合适的。

建立运动捕捉数据库实际上就是要对运动捕捉序列这种时序信号进行比较归类。但是因为运动捕捉序列的帧数非常多，直接进行两个序列之间比较是非常耗费时间的。如果我们从序列中提取出能够代表运动特征的帧—关键帧，那么可以通过关键帧的比较来得到序列的分类。为了进行分类，我们不需要保持运动在大范围内的时序信息。相反，我们要求周期信号坍缩为一个周期，因为周期在运动类型的比较中是没有用处的。例如走和跑的区别在一个周期内就可以区分出来。关键帧提取完毕之后，采用一些序列比较的算法，就可以达到目的。

但是如果我们的目的是为了提取关键帧用于网络上的传输，那么这种聚类的方式显然是不可取的，因为它丢失了运动中的很多信息。

实际上，只要对上述算法稍作改进就可以达到在关键帧序列中保持周期信息（对插值非常有用的时序信息）。

聚类算法 2 保持时序周期信息的动态聚类算法

```
1  begin initialize  $\eta, \theta$ 
2       $w_1 \leftarrow x$ 
3      do accepted new  $x$ 
4           $j \leftarrow \arg \min_{j'} \|x - \mu_{j'}\|$  (find nearest cluster)
5C      if  $\|x - \mu_j\| < \theta \ \&\& \ \forall j', j' \leq j$ 
6          then  $\mu_j \leftarrow \mu_j + \eta x$ 
7          else add new  $\mu \leftarrow x$ 
8           $\mu \leftarrow \mu / \|\mu\|$  (normalize weight)
9      until no more patterns
10     return  $\mu_1, \mu_2, \dots$ 
11 end
```

图 16 聚类算法 2: 保持时序周期信息的动态聚类算法

这儿我们唯一作的改动是在第 5 行。即如果一帧属于的聚类不是在时间上离它最近的，那么仍然把它作为关键帧。

但是，这样的方法还能够称做聚类算法吗？

的确，它比动态聚类得到多得多的聚类，如果序列很长，周期运动很多，聚类算法的作用只有在一个周期内部体现出来，在大范围内丧失了作用。

为了保持聚类算法的概括能力，我们可以对上述算法作一点修改：

聚类算法 3 保持时序周期信息的动态聚类算法改进

```

1  begin initialize  $\eta, \theta$ 
2       $w_1 \leftarrow x$ 
3      do accepted new  $x$ 
4           $j \leftarrow \arg \min_{j'} \|x - \mu_{j'}\|$  (find nearest cluster)
5          if  $\|x - \mu_j\| < \theta$  &&  $\forall j', j' \leq j$ 
6              then  $\mu_j \leftarrow \mu_j + \eta x$ 
N1         else if  $\|x - \mu_j\| < \theta$ 
N2             then  $\mu_j \leftarrow \mu_j + \eta x$ , memorize the cluster belong to:  $j$ 
7             else add new  $\mu \leftarrow x$ 
8              $\mu \leftarrow \mu / \|\mu\|$  (normalize weight)
9         until no more patterns
10        return  $\mu_1, \mu_2, \dots$ 
11 end

```

图 17 聚类算法 3: 保持时序周期信息的动态聚类算法改进

即我们在一帧属于的聚类不是在时间上离它最近的情况下, 依然把它归入那个聚类, 但是记下它的聚类号。在恢复的时候, 由于我们知道它属于的聚类, 就可以很容易的进行恢复。这种方法即保留了聚类的特性和性能, 又保留了周期信息, 同时压缩率可以保持和聚类算法 1 相近。当然, 在传输的时候, 还必须把该帧的根节点坐标也传输过去, 否则恢复的时候根节点坐标就错了。

7.1.5 关键帧属性度量—考虑整体几何性质

这种改进的动态聚类算法虽然可以保持时序性, 但是在如果要进行插值恢复的时候还是有问题的。

因为提取出关键帧之后, 如果要进行插值恢复, 一般都是利用前后的两个关键帧进行线性插值。但是, 线性插值的使用是有条件的。只有在两个关键帧之间近似呈线性变化的时候, 线性插值才能取得比较好的结果。

在进行聚类的时候, 我们没有考虑根节点的坐标, 因为我们关心的是人体姿态的变化而不关心人体的整体位置。从提取姿态特征的角度这是完全正确的。但是由于略去了根节点, 即人体整体运动的信息, 所以我们无从知道相邻两帧之间是否是近似线性的。这就会影响插值恢复时根节点位置的准确性。

更糟糕的是, 由于采用了树形结构, 子节点是在父节点坐标系下表示其位置的, 如果父节点的位置不准确会连带的影晌子节点的运动。这种误差的累积

效应会使得运动变形。最终得结果就是恢复的误差会非常大，这是无法忍受的。

因此一种自然的考虑是可否把根节点的平移考虑到聚类中去。

应该注意的是，我们要做的是如何保证关键帧之间的整体运动近似线性，而不是控制关键帧之间的距离。因为即使两个关键帧之间的距离很大，如果它们之间的运动是直线运动，插值的结果也是很好的。

这个问题实际上就是求空间曲线的控制点的问题。由于两点之间的曲线近似直线，其上任意一点的曲率和挠率都近似为零。复杂曲线控制点的挠率和曲率都很大，因此自然的会想到使用微分几何的数学工具来解决这个问题。

我们只需要在关键帧的度量上加上该帧根节点的挠度和曲率的度量就可以，自然要考虑一定的权值。因此对上面的关键帧属性的度量，我们可以做如下的改进定义：

- 1) 该帧和前面的所有提取出来的聚类的中心的差别的最小值
- 2) 该帧的根节点的挠率和曲率的加权

很自然的会考虑条件 1。我们考虑条件 2 的原因有两个。首先是插值恢复的时候，如果曲线比较复杂，使用简单的点和旋转的线性插值的效果不太好。其次是如果运动比较简单，那么可能即使该帧离开前面所有的聚类都很远但是依然可以通过线性插值得到很好的结果。基于以上两方面的考虑，我们在关键帧可能性的评价中引入了整体运动复杂性的度量。实验证明，这相较于简单的使用聚类算法确实有较大的优越性。这实际上是同时考虑了局部的运动细节（聚类问题）和全局整体运动（微分几何性质）的结果。我们通过这种方法比较全面的刻画了运动捕捉数据，那么关键帧的提取，即对运动序列特征的提取自然做的比较好了。

注意到我们没有使用根节点的速率，因为即使速率很大，如果以上两个评价值都比较低，说明曲线比较简单，可以使用简单的线性插值来得到较好的结果。

对应的聚类算法如下：

聚类算法 4 考虑了几何性质

```

1  begin initialize  $\eta, \theta$ 
2       $w_1 \leftarrow x$ 
3      do accepted new  $x$ 
4           $j \leftarrow \arg \min_{j'} \|x - \mu_{j'}\|$  (find nearest cluster)
C5          if  $\|x - \mu_j\| + A\kappa + B\tau < \theta$  &&  $\forall j', j' \leq j$ 
6              then  $\mu_j \leftarrow \mu_j + \eta x$ 
N1          else if  $\|x - \mu_j\| < \theta$ 
N2              then  $\mu_j \leftarrow \mu_j + \eta x$ , memorize the cluster belong to:  $j$ 
7          else add new  $\mu \leftarrow x$ 
8               $\mu \leftarrow \mu / \|\mu\|$  (normalize weight)
9      until no more patterns
10     return  $\mu_1, \mu_2, \dots$ 
11 end

```

图 18 聚类算法 4: 考虑了几何性质

这儿 κ 指该点的曲率， τ 为该点挠率的绝对值。

注意虽然挠率有符号，但是其符号一般表示螺旋线的方向，这儿不考虑，因为人体运动通常不会如此复杂。

同时，虽然还有其他可能的控制点提取的方法，如[3, 7]，但是都没有微分几何的方法来的即简单有效。

7.1.6 遗留的问题

至此，对于关键帧的提取就基本上完成了。但是，依然有一个问题：即权值的确定。

我们这儿需要确定的权值实在是非常多：

挠率和曲率的权值，树形结构中不同层的权值。而且同层节点的权值相同的假设是否合理？

这些问题的解决是非常困难的，只有通过大量的数值试验来解决。考虑到精确性和简单性的折中，我们采用人机交互的方式来试验权值。对于不同的运动序列，合适的权值可能相差很大，因此这可能是当前比较合理的解决方式。

7.2 插值恢复算法

插值算法恢复算法比较容易。

对根节点直接使用三维矢量的线性插值。

对其余节点，首先使用四元数的线性插值公式得到其相对于父节点的旋转四元数。然后使用第三章的知识得到在父节点坐标系下的坐标。加上父节点的坐标，我们就可以得到其世界坐标。

但是在实际中，我们必须考虑到数值奇异性的问题。

$$\text{Slerp}(q_1, q_2; u) = \frac{\sin(1-u)\theta}{\sin\theta} q_1 + \frac{\sin u\theta}{\sin\theta} q_2$$

($\theta = \arccos(q_1 \circ q_2)$) 在 θ 接近 0 度或 180 度的时候，分母为 0，计算出现奇异。

为了解决这个问题，我们可以利用四元数的特性和极限性质。

由于单位四元数 q 和 $-q$ 代表的是同一个旋转，所以在实际计算中，我们首先计算 $q_1 \circ q_2$ ，如果它小于 0，那么令 $q_2 = -q_2$ ，再计算 $\theta = \arccos(q_1 \circ q_2)$ 。这样 θ 只会是 0 到 90 度，不会进行多余的旋转，不会出现再 180 度附近了。

对于出现在 0 附近的 θ ，我们利用极限的性质：

$$\lim_{\theta \rightarrow 0} \frac{\sin u\theta}{\sin\theta} = u \quad (u \neq 0)$$

可以把上式转化为：

$$\text{Slerp}(q_1, q_2; u) = (1-u)q_1 + uq_2$$

这就解决了奇异性的问题。

第8章 数据文件结构

在编程实现的过程中，我深刻的感觉到了捕捉数据文件的合适结构实在是太重要了。数据格式的选择虽然看上去很简单，但是只有对问题有了比较深刻的理解以后才能选择的好。下面对三种使用过的文件结构作一评价和比较。关于对应实现的问题，可参考[2]。

8.1 TRC 文件简介

TRC 文件是 Motion Analysis 公司的一种早期的文件格式。

文件实例：

```
PathFileType 4 (X/Y/Z) C: \example.trc
DataRate CameraRate NumFrames NumMarkers Units
OrigDataRate OrigDataStartFrame OrigNumFrames
60 120 10 2 mm 60 1 10
Frame# Time LHead RHead
      X1 Y1 Z1 X2 Y2 Z2

1 0 -1089.80 1675.95 435.47 -908.83 1677.22
391.44
2 0.017 -1089.66 1675.93 434.89 -905.26 1678.51
389.11
3 0.033 -1089.24 1675.76 434.30 -902.84 1679.17
387.42
4 0.05 -1088.61 1675.55 433.77 -901.46 1679.18
386.43
5 0.067 -1088.13 1675.47 433.40 -900.61 1679.01
385.78
6 0.083 -1087.97 1675.58 433.13 -900.06 1679.07
385.14
7 0.1 -1087.97 1675.69 432.76 -899.81 1679.37
384.55
8 0.117 -1088.01 1675.66 432.19 -900.05 1679.53
384.21
9 0.133 -1088.19 1675.55 431.53 -901.01 1679.24
384.26
10 0.15 -1088.50 1675.48 430.89 -902.32 1678.65
```

可以看到文件由头部和数据部组成。头部提供了一些冗余信息：关节的名称、帧数、帧率等。数据部按帧顺序记录了每个节点的三维世界坐标，提供了捕捉时间。

从中可以看到，这种文件的最大的好处就是简单，得到的数据直接可以进行显示了。但是有一个很大的缺点：没有提供结构信息。我们不得不通过关节的名称来得到树形结构的组成。但是在不同的运动捕捉设备中，关节的名称还是有可能变化的，这时我们的程序就无法得到树形结构的信息了。

但是树形结构的信息是内蕴的，也是我们算法的基础，因此我不得不面对它。我利用 STL 的 `map` 建立了关节和关节名称的一一对应关系，然后通过这种对应关系描述树形结构。这在实际的编程中运行得很好。但是，数据必须是由特定的运动捕捉设备产生的，关节的排列顺序可以不同，但是必须仅仅是设定的几个关节。可见这种方法对文件的要求还是很严。

最后我决定改变文件，在文件头部增加下面的内容：

```
Root Global
Chest Root
RHip Root
LHip Root
RShoulder RHip
LShoulder LHip
...
...
```

即，提供了子节点名称—父节点名称的对应关系。所费不是很多，可是这就自然的得到关节的树形结构，我们不必要在名称和关节数据之间作映射了。程序实现也非常的容易了。

8.2 BHV 文件格式和 HTR 文件格式简介

BHV 是 Biovision 公司的文件格式 HTR 是 Motion Analysis 公司的文件格式。它们都在文件头中提供了结构化的信息，扩展比较容易了。BHV 和 HTR 对根节点以外的点都是使用旋转量来定义的，这一点没有 TRC 直接，显示比较麻烦，

```

-3.706973 -6.474916 -14.391472 -3.461282 -16.504230
3.973544 -3.805107 22.204674 2.533497 -28.283911
-6.862538 6.191492 4.448771 -16.292816 2.951538
-3.418231 7.634442 11.325822 5.149696 -23.069189
-18.352753 15.051558 -7.514462 8.397663 2.953842
-7.213992 2.494318 -1.543435 2.970936 -25.086460
-4.195537 -1.752307 7.093068 -1.507532 -2.633332 3.858087
0.256802 7.892136 12.803010 -28.692566 2.151862
-9.164188 8.006427 -5.641034 -12.596124 4.366460
-8.489557 4.285263 -0.621559 -8.244940 -1.784412
90.041962 8.849357 5.557910 -1.926571 -5.487280 4.119726
-4.714622 -5.790586 -15.218462 -3.167648 -15.823254
3.871795 -4.378940 22.399654 2.244878 -29.421873
-6.918557 6.131992 4.521327 -18.013180 3.059388
-3.768287 8.079588 10.124812 5.808083 -22.417845
-15.736264 18.827469 -8.070700 9.689109 2.417364
-7.600582 2.505005 -1.625679 2.430162 -27.579708
-3.852241 -1.830524 12.520144 -1.653632 -2.688550 4.545600
0.296320 8.031574 13.837914 -28.922058 2.077955
-9.176716 7.166249 -5.170825 -13.814465 4.309433

```

HTR 文件格式的例子:

```

#Comment line ignore any data following # character
#Hierarchical Translation and Rotation (.htr) file
[Header] #Header keywords are followed by a single value
FileType htr #Single word string
DataType HTRS #Translation followed by rotation and scale data
FileVersion 1 #integer
NumSegments 18 #integer
NumFrames 2 #integer
DataFrameRate 60 #integer, data frame rate in this file
EulerRotationOrder ZYX
CalibrationUnits mm
RotationUnits Degrees
GlobalAxisofGravity Y
BoneLengthAxis Y
ScaleFactor 1.000000
[SegmentNames&Hierarchy]树形结构表示!
#CHILD PARENT
Head Neck
Neck Chest
ChestHips
LeftShoulder LeftCollar

```

```

RightShoulder RightCollar
LeftElbow LeftShoulder
RightElbow RightShoulder
LeftWrist LeftElbow
RightWrist RightElbow
Hips GLOBAL
LeftHip Hips
RightHip Hips
LeftKnee LeftHip
RightKnee RightHip
LeftAnkle LeftKnee
RightAnkle RightKnee
LeftCollar Chest
RightCollar Chest
[BasePosition]
#SegmentName Tx, Ty, Tz, Rx, Ry, Rz, BoneLength
Head 0.000003 141.966248 0.000002 -37.745777 -8.179454 6.203664
80.000046
Neck 0.000000 379.566772 0.000000 32.855431 -5.194619 1.823337
141.966263
Chest0.000000 94.891693 0.000000 0.000002 12.877405 -0.000012
379.566803
..... 其余的初始位置和指向略.
#Beginning of Data. Separated by tabs
[Head]
#Fr Tx Ty Tz Rx Ry Rz SF
1 -0.000004 11.246399 -0.000001 -8.720660 -3.164685 6.637906
1.000000
2 -0.000005 11.559982 -0.000003 -8.967685 -3.106098 6.445398
0.999999
.....下面是其他节点的数据, 略.
[EndOfFile]

```

对于运动编辑和关键帧提取来说, 后两种格式比较合适, 以后将把程序改进为可以读取这两种文件。

第9章 算法实现

这儿对实现中的一些比较复杂和重要的部分进行一点讨论。对于一些比较简单或已经在前面几章详细介绍过的内容，此处不再赘述。

9.1 映射

在 TRC 文件格式中没有树形结构，但是我们的关键帧算法是基于捕捉数据的树形结构的。因此我们必须自己建立树形结构的表示。

由于文件读入时，各个点的三维数据是读入一个 `vector` 中。但是在世界坐标系和父节点坐标系的转换中，我们更愿意使用 `vector["Root"]` 而不是 `vector[6]` 来引用根节点的数据，因为后者不直观，无论是编程实现还是以后的扩展维护都极其不便。

怎样才能使用名称而不是下标来引用数据呢？

一个很好的选择就是 STL 中的 `map`。`map`，顾名思义，即在两组数据（A，B）之间建立一一对应的关系，并且可以通过 `map[A]` 来得到对应的 B。

那么我们现在只要建立 `map (string, int)` 就可以通过关节名来得到对应关节的数据在数据数组中的位置：

C++代码如下：

```
void getInfo (map<string, int> & _info, Object3D & obj3d,
Frame3DArray & fr, MocapSystemInfo & si)
{
    int _size = si.numMarkers;
    Marker3DArray _marker = fr[0].markers;
    string _name;
    for (int i = 0; i < _size; i++)
    {
        _name = obj3d.markerName (_marker[i].id);
        _info.insert (std::pair<string, int> (_name, i));
    }
}
```

9.2 三维世界坐标转换为四元数

由于世界坐标转换为四元数在恢复的时候要进行线性插值的，所以对应的旋转基点必须统一，这样插值才有意义。一个很自然的基点就是第 1 帧。

通过四元数的定义，单位四元数 $\cos \theta + u \sin \theta$ 的参数 u 对应旋转轴，参数 θ 对应绕轴 u 旋转的角度。注意这个旋转指的是沿着大圆的旋转。

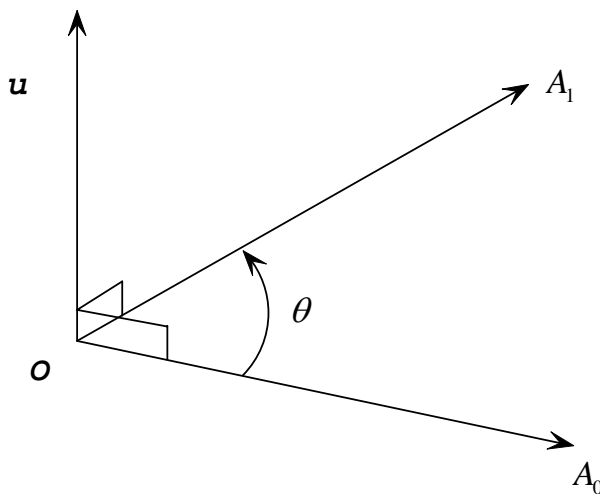


图 19 四元数旋转表示示意

因此，很容易的，根据上面的图，可以得到旋转轴为

$$u = \frac{v_0 \times v}{|v_0 \times v|}$$

旋转角度为

$$\theta = \arcsin \frac{|v_0 \times v|}{|v_0| |v|}$$

其中 v_0 和 v 为第 1 帧和当前帧的对应关节在父节点坐标系下的矢量坐标。

但是在实际编程中，我们不是转换为 $\cos \theta + u \sin \theta$ ，而是 $\cos \frac{\theta}{2} + u \sin \frac{\theta}{2}$ 。

原因是为了插值恢复的时候的方便。原因可见下文。即应有：

$$\theta = \frac{1}{2} \arcsin \frac{|v_0 \times v|}{|v_0| |v|}$$

令 P 为第 1 帧的关节点在父节点坐标系下的矢量，那么根据 $[0, P'] = q \circ [0, P] \circ q^{-1}$

我们就可以得到当前帧在父节点坐标系下的坐标矢量。但是注意到，这时的 q 应该是 $\cos \frac{\theta}{2} + u \sin \frac{\theta}{2}$ ，即对应的旋转角度是两个矢量夹角的一半。这就是为什么上文中的角度要减半。

现在似乎没有什么问题了。的确，对于复杂的运动没有什么问题。但是对于简单的数据，反而有问题！因为如果运动比较缓慢或者捕捉数据率比较高，那么有的帧和第一帧的差别很小，会有 $|v_0 \times v| = 0$ 。那么旋转轴的计算会出现奇异的情况。

如何解决这种奇异的情况呢？这时说明 $v_0 \approx v$ 即点几乎没有动，那就是没有旋转，那么我们只要随便设置一个旋转轴就可以了。

9.3 奇异值的解决思路

在这儿出现了很多的数值计算的奇异问题。这些问题的解决实际上是很简单的。

首先，检查中间数据，定位奇异值位置。出现奇异的地方一般会数据是-1。`#IND` 等代表奇异值的量，可以通过检查数据文件来完成。这是关键。

然后，找出出现奇异的原因，一般是除 0、溢出或者是反余弦的自变量值的绝对值大于 1。

最后，由于奇异值一般是出现在特殊的条件下，可以通过简单的判断来解决。这一步会非常简单。

其中有的只需从数学意义上考虑即可解决，有的还必须考虑奇异值的实际物理意义才能解决。

奇异值虽然是偶然的情况，但是对它的检查的确非常困难。通过上面的步

骤可以较快的定位改正。

第10章 实验结果

10.1 实验数据和实验设计

我们在 4 种典型数据上做测试来比较考虑了整体运动的算法和不考虑整体运动的经典算法。前一种算法在聚类的时候考虑了整体运动信息，后一种没有考虑。这是两种算法的唯一差别。

首先我们希望的实验数据应该是纯粹的，并能够完整测试算法各部分的性能。

所谓完整性就是数据的全体可以基本描述问题面对的数据空间。

我们测试的 4 种典型数据分别是：

1. **Stand**: 原地站立，仅仅头部摆动
2. **Walk**: 正常的行走
3. **Turn over**: 翻筋斗
4. **Play golf**: 打高尔夫球时的挥球动作

可以见到这 4 种运动有很简单的 **Stand**，比较复杂的 **walk** 和 **play golf** 和非常复杂的 **Turn over**。在这四种数据上进行测试可以比较全面的衡量算法的性能，对不同算法的优缺点的比较也比较客观。因为在复杂数据上效果很好的算法在简单数据上的效果未必好，反之当然也未必。

因此数据可以完整的测试算法的性能。

所谓纯粹性就是数据只测试算法某一方面的性能。比如测试滤波器低频性能的数据最好不要混杂高频的分量。因为这样得到的结果是几种性能混合表现的结果，对算法的测试就不太准确。

我们测试的四种数据实际上不满足纯粹性的要求。尤其后两种比较复杂的数据，其中混杂着大量简单的子序列。而 **Walk** 数据中也有复杂的子序列。

所以测试的数据实际上是不是很理想，但是我们现有的数据都是由其他单位提供的，所以不得不这么做。同时即使我们的运动捕捉设备完成了，捕捉数据也不能完全的纯粹。因为人体要完成复杂的运动必须从简单的运动开始。我们可能需一些对应的滤波算法。例如将复杂部分看成高频之类的。这些仅仅是想法。

我们实验的目的是关键帧提取算法的比较，所以我们的插值恢复算法必须是一样的。为了方便起见，我们都采用最简单的线性插值。因为四元数插值有稳定性的问题。实际上，我们的实验表明，虽然用四元数恢复的效果差，但下面所有实验结论依然成立。

10.2 阈值、关键帧率及恢复误差

首先，我们先看一下两种算法使用不同的阈值时得到的关键帧率及恢复误差的图像。

关键帧率=关键帧数 / 总帧数，也称为压缩率。我们这儿采用相对量是为了归一化。

恢复误差我们用两个量来衡量：1) 平均每帧的恢复误差 2) 最大恢复误差

其中第一项是各帧的恢复误差的平均值，第二项是各帧的恢复误差的最大值。采用这两个量，再加上关键帧率，基本上可以比较全面的描述算法的恢复性能。

使用使用传统算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图为：

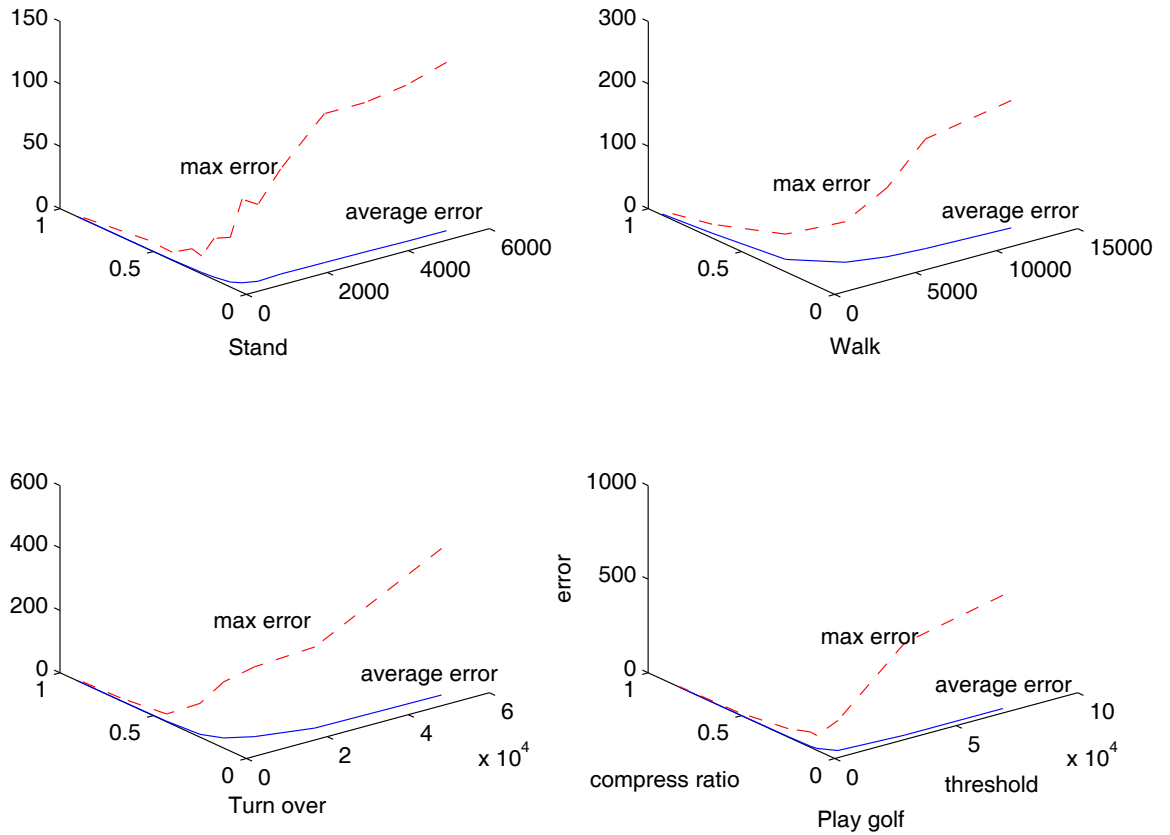


图 20 传统算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图

使用使用本文算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图为:

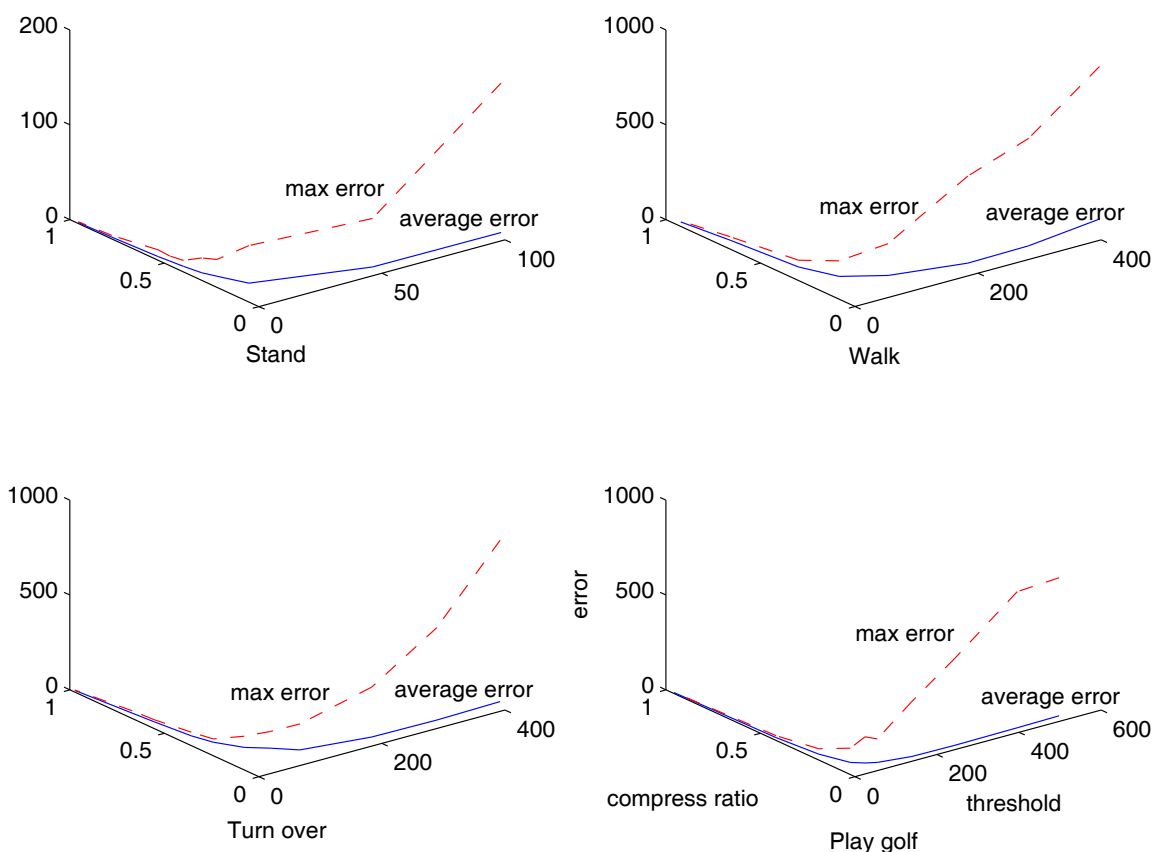


图 21 本文算法得到的 f (阈值, 关键帧率, 平均误差) 和 g (阈值, 关键帧率, 最大误差) 的曲线图

容易看到它们都满足实际的情况:

误差是随着关键帧数目的增大而减少。阈值越大得到的关键帧越小。而且不同的数据得到的曲线形状是类似的: 基本上是类指数分布。

同时, 显然的是, 在关键帧数码比较少的时候, 最大误差是远大于平均误差的。但是最大误差随着关键帧率的增大快速的趋于平均误差。

10.3 数据分析

有了阈值, 关键帧率, 平均误差三者大概的关系之后, 我们来分析一下我们的四种数据的一些特点。为了分析它们, 我们可以看一下两种算法在四种数据上得到的关键帧的恢复误差。

使用传统算法得到的平均误差在四种数据上的结果是:

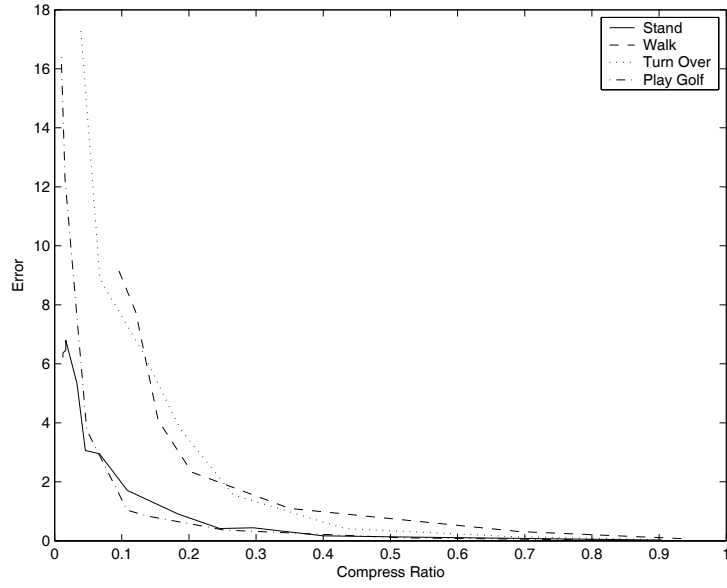


图 22 传统算法得到的平均误差在四种数据上的结果

而使用本文提出算法得到的平均误差在四种数据的结果是：

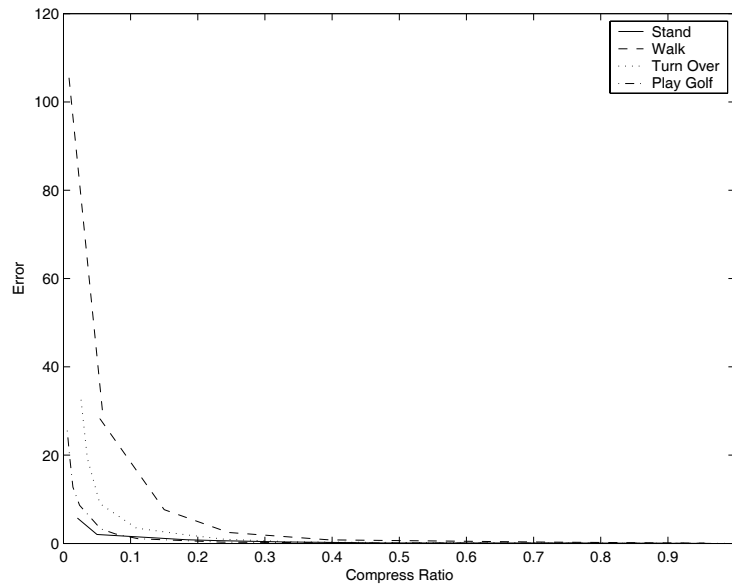


图 23 本文提出算法得到的平均误差在四种数据的结果

使用传统算法得到的最大误差在四种数据上的结果为：

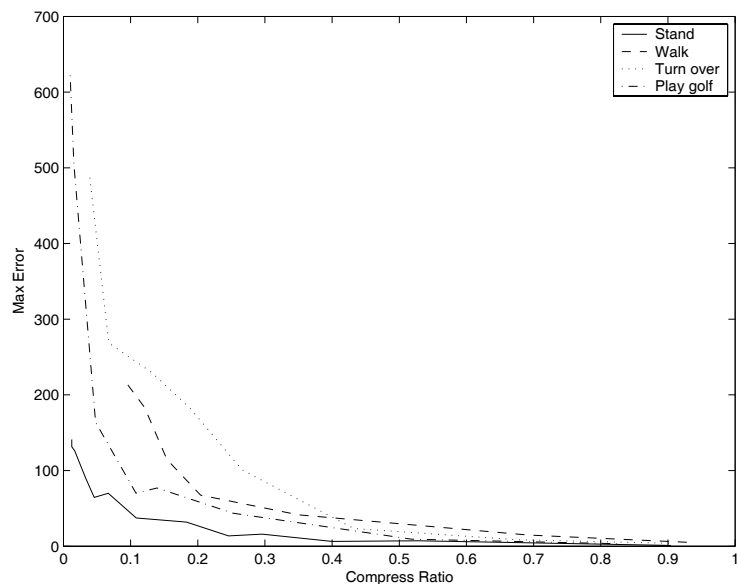


图 24 传统算法得到的最大误差在四种数据上的结果

使用本文提出的算法得到的最大误差在四种数据上的结果为：

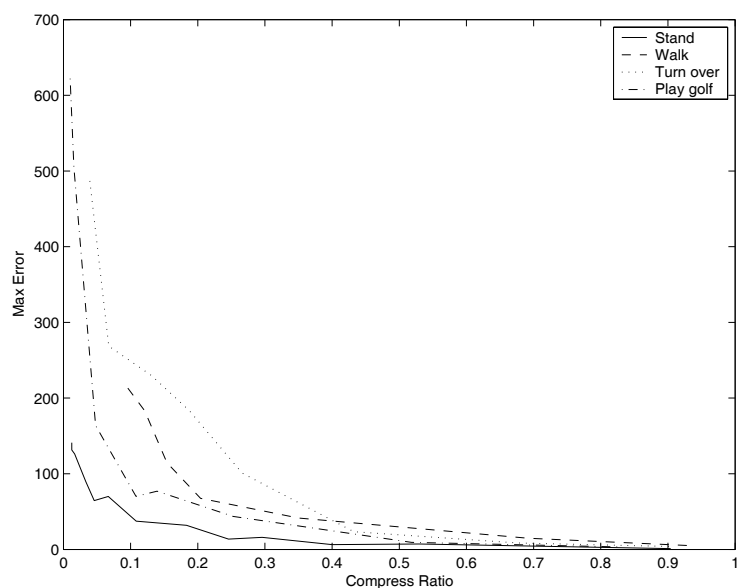


图 25 本文提出的算法得到的最大误差在四种数据上的结果

现在来对得到的结果做一点分析。

首先，四张图有一点是一致的：误差（评价误差和最大误差）都是随着关键帧数的增加而单调不增。这是和我们的预期符合的。

虽然从复杂度上是 Turn over > Play golf > walk > stand，但是两种算法得到

的平均误差都是：Walk > Turn over > play golf > stand！这似乎和我们的直觉不太符合：越复杂的数据得到的误差也应该越大啊！

但是，通过分析数据，很容易看到 Turn over 序列不是始终复杂的。Turn over 动作的过程是：原地动作→慢走→翻腾→慢走→原地动作→停止。所以除了翻腾的一段比较复杂，开始和结束的动作都比较简单。同时翻腾比较迅速，时间比较短，虽然这一段的误差比较大，但是平均之后，整体的误差反而比较小。对 play golf 动作序列的分析也会发生同样的问题。

因此，每一帧的误差都比较平均的 Walk 序列的平均误差就显得比较大了。

这个分析的正确性可以通过最大误差的结果看到：使用两种算法得到的最大误差的结果基本上都是：Turn over > Play golf > walk > stand（除了 walk 外）。

但是我们发现使用本文提出的算法得到的最大误差的结果在关键帧数比较多的时候，结果依然是 Walk > Turn over > play golf > stand。这是为什么呢？

这实际上和我们提出的算法的性质相关。

本文算法的主要改进就是同时考虑了整体的运动。这种对整体复杂性的考虑对不同运动的效果是不一样的。对复杂的运动如 Turn over，它的效果比较明显，提取的关键帧比较比传统算法合适的多，恢复误差的减少非常大。而对于整体不是很复杂的运动，如 walk，算法的效果就不明显，恢复误差的减少不明显。甚至，对于主要是细节运动的 walk，如果考虑了整体的复杂性，就必须选择合适的挠度和曲率的权值。如果选择不当，恢复误差反而会增大（这在下面可以看到）。

因此，由于使用本文的算法大幅度的改进了恢复误差，所以即使是最大误差也有 Walk > Turn over > play golf > stand。

从上面的讨论可以看到数据的不纯粹会使得实验结果比较复杂。但是经过仔细的分析可以发现结果是合理的。

10.4 算法比较

通过上面对四种数据的一些分析，我们现在基本上了解了我们算法的特点和数据的特点，接下来，开始在同一数据上进行算法比较：

以下四张图依次是两种算法在四种典型数据（Stand, Walk, Turn over, Play golf）上的比较结果：

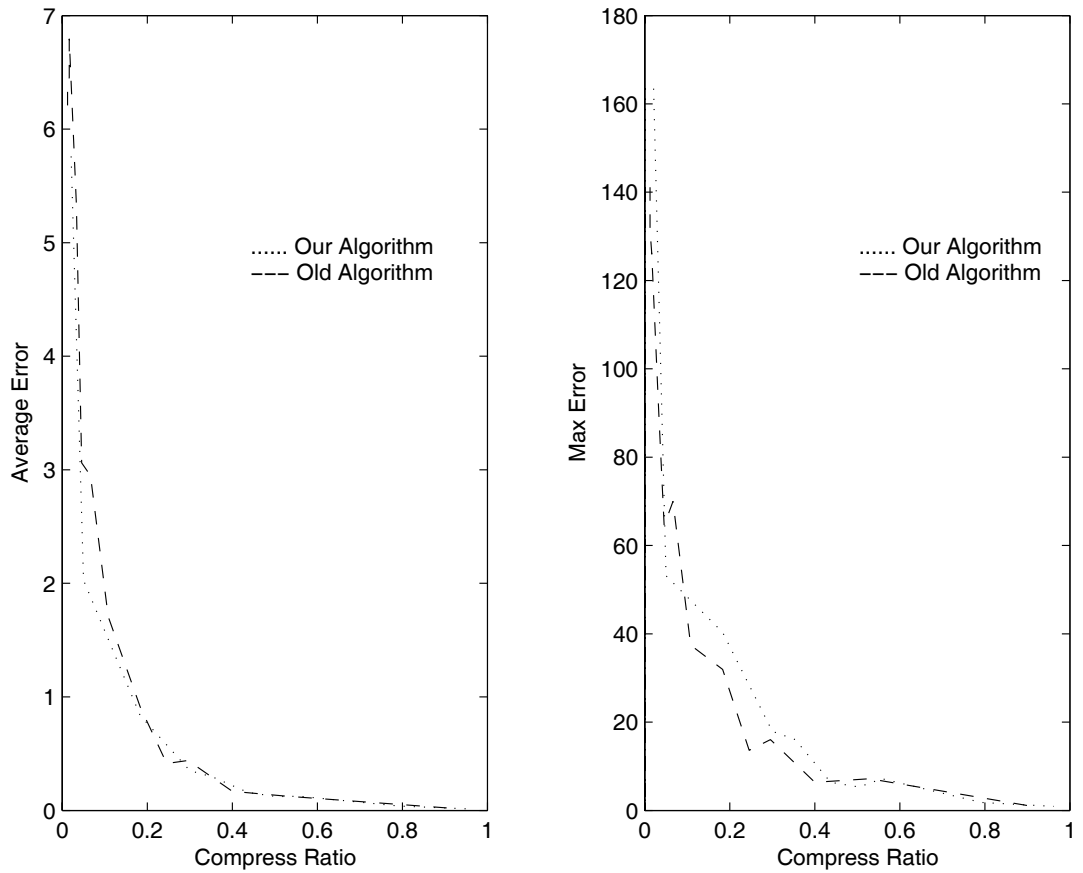


图 26 算法在典型数据 Stand 上的比较

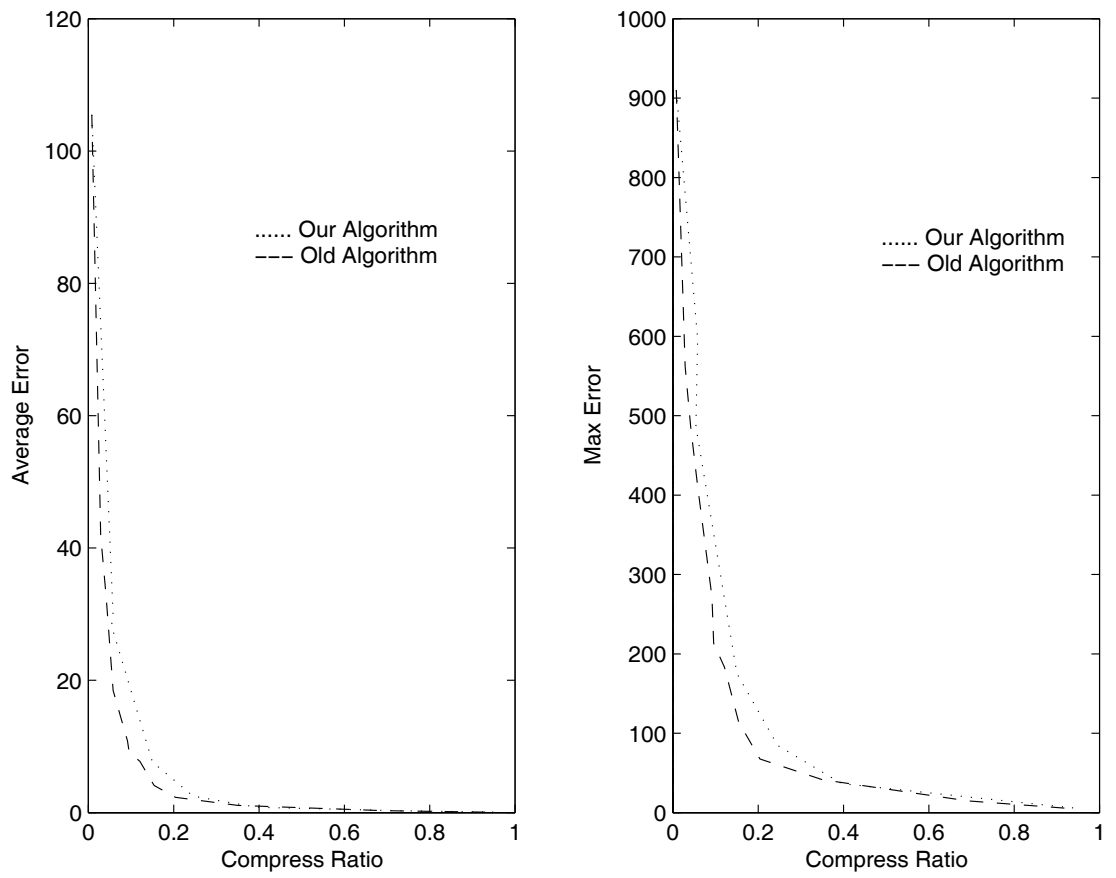


图 27 算法在典型数据 Walk 上的比较

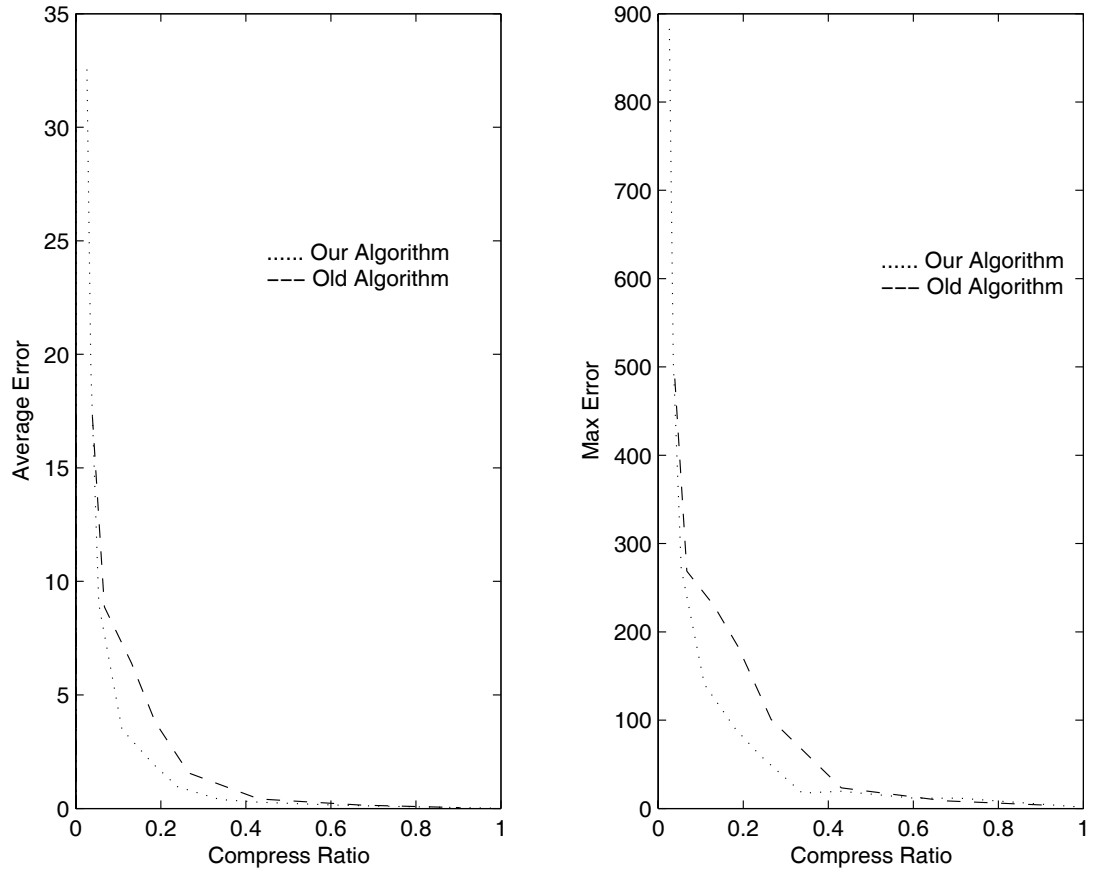


图 28 算法在典型数据 Turn Over 上的比较

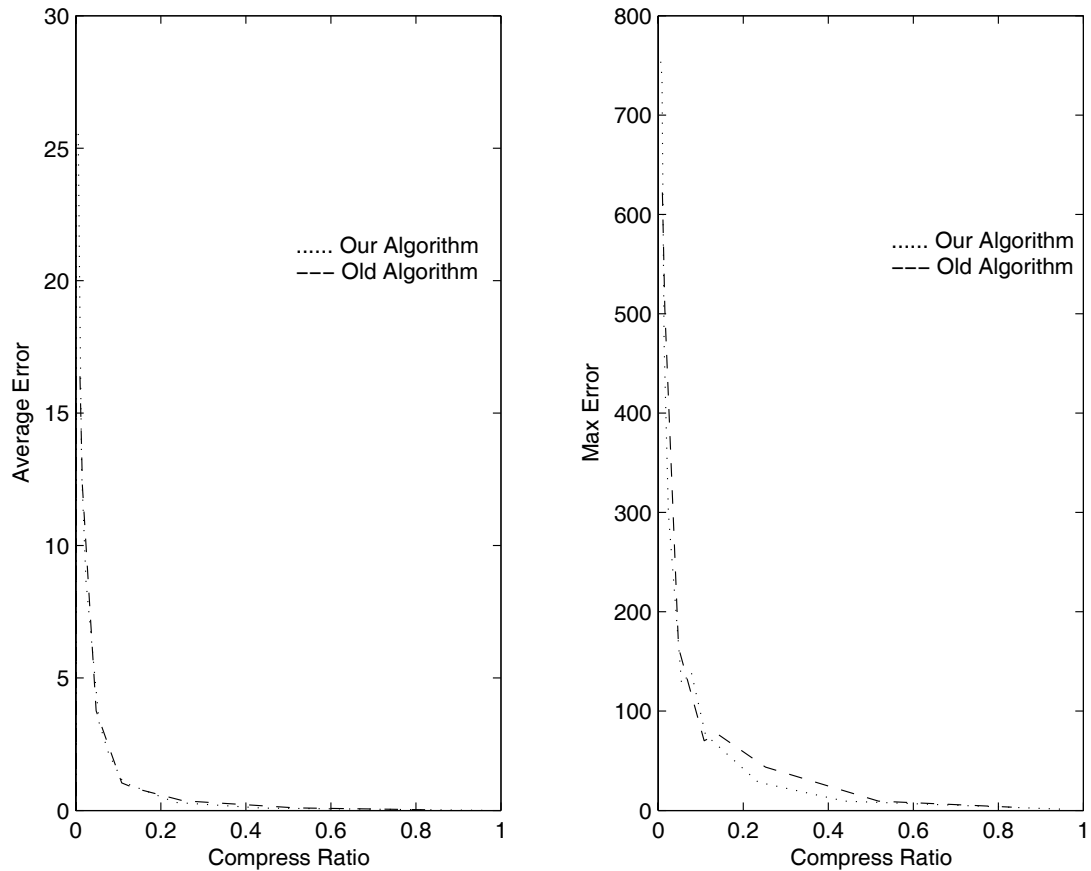


图 29 算法在典型数据 Play golf 上的比较

从以上的实验结果，我们可以很容易的看到：对于 Stand，Turn over，Play golf 三种类型的数据，本文提出的算法结果都要优于已有的算法。而且对于比较复杂的运动如 Turn over，平均误差和最大误差的改进都相当大。

但是也可以看到，在 walk 数据上的我们的算法反而性能变差了。下面我们大致分析一下原因。

因为 walk 数据在整体上比较平均，主要的是细节运动。如果考虑了整体的运动，那么由于我们对于曲线上各个点的挠率和曲率的计算都是使用的差分公式来近似的，所以得到的结果不是很准确。而实际上 walk 运动的整体曲线基本上是直线，如果曲率和挠率的误差比较大，就会不准确的判断关键帧点，从而得到的关键帧不是最优的，因此通过这些关键帧恢复的序列和原序列的差别就比较大。

同时算法的性能也没有完全的体现出来。

因为我们在提取关键帧的时候使用的微分性质的加权是固定的。而实际上比较合理的做法是在整体运动复杂的时候权值应该大，在运动细节复杂的时候应该减小权值。即这个权值应该是自适应的。但是我们只是通过经验设定了一个权值，这对算法性能的影响比较大。

10.5 结论

通过分析上面的实验结果，我们可以看到我们的算法对于复杂运动的关键帧提取可以达到很好的效果。当然对于简单的运动数据，有时也达不到最好的效果，这是算法的局限性。同时，我们的算法要达到最好的效果，必须恰当的选择整体曲线挠度和曲率的权值，而这基本上只能依靠尝试。当然，大致的数量级可以通过分析挠度。曲率和细节误差的数量级关系来确定，但是在实际的关键帧在线提取中，这也是不能预先算出的。幸运的是，我们在数值实验中发现，算法的结果对于权值不是非常灵敏。这就使得我们可以通过一定规模的数值实验来确定各种数据最佳的挠度和曲率权值。当然，这需要一定规模的捕捉数据库。

在以上的数值实验中，我们采用的权值是：挠度权值 3，曲率权值 5。实验结果说明这是可行的。

第11章 将来工作

进一步的工作包括：

1.关键帧帧数控制

即如何在指定关键帧数目的情况下得到和设定值相差不大的关键帧数目。在第7章我们有一个大概的想法，但是有点问题。以后可以进一步的研究。

2.运动编辑算法

这是将来的一个研究方向。

首先可以考虑实现已有的几种方法：信号处理、时空约束等等。然后设计对应的算法。同时可以考虑多体运动，这比单体约束运动要复杂得多，因为需要更复杂的约束计算（碰撞等）和人工智能。

3.渲染

由于我们现在只是得到了骨架，而运动捕捉实际上是可以计算机视觉得到表面信息的。因此可以结合计算机图形学的一些知识。比如把运动编辑和表面形变结合在一起。

这些内容都是比较困难的，1，2 在短期内可能会有些成果。3 实在是一个浩大的工程，在短期内可能难有突破。

参考文献

- [1] F.Sebastian Grassia.Practical Parameterization of Rotations Using the Exponential Map.The Journal of Graphics Tools, volume3.3, 1998
- [2] M.Meredith, S.Maddock.Motion Capture File Formats Explained.
- [3] Li Zhao, Wei Qi, Stan Z.Li, Shi-Qing Yang, H.J.Zhang.Key-frame Extraction and Shot Retrieval Using Nearest Feature Line(NFL) Classification. ACM Multimedia Information Retrieval 2000, Oct 2000, Los Angeles, USA.
- [4] Andrew Witkin, Michael Kass.Spacetime Constraints.Computer Graphics, Volume 22, Number 4, August 1988
- [5] 刘丰, 庄越挺, 潘云鹤, 罗忠祥.基于小波变换的运动分析及其应用.中国图像图形学报, volume 8(A), Number 1, Jan 2003
- [6] 罗忠祥, 庄越挺, 潘云鹤, 刘丰.基于时空约束的运动编辑和运动重定向.计算机辅助设计与图形学学报, Volume 14, Number 12, Dec 2002
- [7] Ik Soo Lim, Denial Thalmann.Key-Posture Extraction Out of Human Motion Data by Curve Simplification. Proc. EMBC2001, 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society
- [8] F.Sebastian Grassia.Motion Editing:Mathematical Foundations. In course: Motion Editing: Principles, Practice, and Promise, SIGGRAPH 2000, New Orleans, Louisiana, USA, 2000
- [9] Michael Gleicher.Motion editing with spacetime constraints. Proceedings of the 1997 symposium on Interactive 3D graphics, p.139-ff., April 27-30, 1997, Providence, Rhode Island, United States

-
- [10] M.Gleicher, P. Litwinowicz. Constraint-based motion adaptation. *J. Visual. Comput. Animat.* 9 (2) (1998) 65-94
- [11] Jehee Lee, Sung Yong hin. A hierarchical approach to interactive motion editing for human-like figures. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, p.39-48, July 1999
- [12] Michael Gleicher. Retargetting motion to new characters. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, p.33-42, July 1998
- [13] Armin Bruderlin, Lance Williams. Motion signal processing. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, p.97-104, September 1995
- [14] Feng Liu, Yueting Zhuang, Zhongxiang Luo, Yunhe Pan. A Hybrid Motion Data Manipulation: Wavelet Based Motion Processing and Spacetime Rectification. *Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, p.743-750, December 16-18, 2002
- [15] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in C++, The Art of Scientific Computing, Second Edition.* The Press of the University of Cambridge, Cambridge, England. 2002. 108~126.
- [16] Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner 著, 吴滨, 段海波, 薛凤武 译. *OpenGL 编程权威指南.* 北京: 中国电力出版社, 2001. 1~244.
- [17] 马力. *简明微分几何.* 北京: 清华大学出版社, 2004. 2~66.
- [18] Saeed B. Niku 著, 孙富春, 朱纪洪, 刘国栋等译, 孙增圻审校. *机器人学导论—分析、系统及应用.* 北京: 电子工业出版社, 2004. 1~104.
- [19] Ken Shoemake. Animation Rotation with Quaternion Curves. *Siggraph 85, Volume 19, Number 3*, 1985.
- [20] Feng Liu, Yueting Zhuang, Fei Wu, Yunhe Pan. 3D Motion retrieval with motion index tree. *Computer Vision and Image Understanding* 92, 2003. 265-284.

-
- [21] 苏步青, 刘鼎元.初等微分几何.上海:上海科学技术出版社, 1985.1~71.
- [22] David Eberly.Quaternion Algebra and Calculus.Magic Software, Inc.September 27, 2002
- [23] C.Barron , I.A. Kakadiaris.A convex penalty method for optical human motion tracking.IWVS'03, November 7, 2003, Berkeley, California, USA
- [24] 鲍虎军, 金小刚, 彭群生. 计算机动画的算法基础.杭州:浙江大学出版社, 2000.196~222

致 谢

本文完成于清华大学自动化系控制理论与技术研究所和中国科学院自动化研究所模式识别国家重点实验室.

感谢我的指导教师徐文立教授,感谢他的严格督促和在毕业设计期间提供的帮助.感谢我的导师杨青教授和潘春洪教授,感谢他们提供的指点和良好的锻炼机会,以及在学习生活上提供的帮助.杨老师严谨的科研态度和敬业精神是我学习工作上的榜样.

感谢实验室的师兄和朋友们,他们给了我无私的帮助.

感谢所有过去对我传道授业解惑的师长和朋友.

谨以本文献给所有关心我的人.

基于要素的动态人物运动的参数化表示

Yuehi Abe C.Karen Liu Zoran Popovic

University of Washington

摘要

本文提出了一种系统，可以用于高速运动捕捉数据的快速编辑。这个系统的核心是一种优化算法。这种算法可以在转化捕捉数据使得其满足高层用户的约束的前提下保证运动的线性和角度要素在物理上是正确的。与过去的大多数运动编辑的方法不同，我们的算法不需要特定的姿势或者模型的化简，用户只需要对输入的运动指定高层的变化。为了保证输入运动的动力学特性，我们引入了一种符合运动捕捉数据线性和角度要素的、基于样条的参数化表示。由于我们的算法给出了一个很好的优化初始条件，所以算法可以很快的收敛。因此用户可以通过单个的输入运动高效的产生大量的实际运动。然后，算法可以通过简单的插值生成运动的动力学空间，有效的参数化实际的运动空间。我们将显示这个框架可以用于产生一个有效的接口，用于快速的生成动画，也可以用于实时的驱动角色的运动。

类别和主题描述符(根据ACM CCS): I3.7[计算机图形学]: 三维图形学和动画。

1. 介绍

尽管近年已经取得了很大的进步，但是为现实人类的运动的合成制作有效的工具依然是计算机动画领域一个开放的问题。对人物的快速运动，如跑、跳以及其他在特写镜头和电子游戏中经常出现的体育和竞技活动，这个问题尤其重要。合成这样的运动是非常有挑战性的，因为在这些运动中，任何物理的不精确都是尤其的惹人注意的。

时空优化与控制器的合成两种方法都被提出用于动态人物运动的直接合成。尽管这些方法确实符合物理学定律，但是它们似乎过度光滑了，而且有时显得机器人化。更重要的是，这些方法没有提供交互控制，在产生动画序列前经常需要大量的线下时间。另外，它也很难达到为了进一步控制而要求的退化。

与这些直接合成相比较，基于运动捕捉数据改写的方法可以得到更真实的运动，尤其是在捕捉数据样本的领域内。它们的速度可以进行交互，因为它们使用的是插值技术。不幸的是，这种方法需要大量的样本。如果动画师希望交互式的控制一个动画的一个特定的参数，比如一个特技动作的落地位置，对大运动集的需求是特别显著的：插值技术需要一个已经存在的大的运动序列集，这些运动的唯一不同在于落地点的位置。收集这样的数据集不仅是累人的而且需要捕捉到的运动在除此以外的所有方面都是相似的(例如：其他的落地点，初始位置和最终位置，所有的模式)—这是一个实际的演员很难重复的方面。实际上，在电子游戏的制作领域，产生这样的参数化的运动是数据采集领域最具挑战性的问题 [Buc]。此外，动画师经常希望产生一些违反实际物理规律的非现实的运动，这是运动捕捉无能为力的地方。

我们采用的获取相似运动的方法是对单个运动序列进行几次修改来合成一个维持物理约束的运动族。这种方式创造的运动可以满足动画师的特定要求，而且与初始图像的背离最小。作为理想的情况，我们希望可以利用最少的源运动数据，可能只是一个捕捉的运动，来创造大范围的其他运动。最近提出了许多动态运动的更改算法[PW99, ZH99, TSK02, SP04, SHP04]，本文介绍的工作也属于这个范围。本文介绍了基于要素的运动编辑技术。与已知的其他方法不同，我们提出的框架对大范围运动变化师鲁棒的。例如，我们可以更改向前跳跃的运动，得到包括向后跳或者360度旋转跳跃在内的各种不同方向的跳跃。

使用我们的运动编辑框架,我们将显示如何基于动画师对于交互控制的需要得到一系列的运动。因为我们的运动族对空间的采样间隔比较的,满足一定的约束,或者与初始的运动序列背离最少,我们可以使用实时的插值技术来获得对合成的运动空间的实时访问。我们描述了一系列实时动画工具,它们可以使用这些合成运动族来构造。例如交互式的约束调整(例如,改变落脚点的位置),也可以逆向控制实例,例如决定可以触及在空间一定位置的排球的鞋钉的位置。此外,我们将描述如何使用合成采样/插值方法来从单个运动捕捉的跳跃得到跳跃运动的实时控制器。

2. 相关工作

最近计算机动画的研究集中在通过已知数据的再映射得到新场景下的实例的技术。本文我们主要参考了基于物理和插值的运动编辑方法。

2.1. 基于物理的运动编辑

由Witkin和Kass介绍的最优轨迹方法[WK88]为在客观的评价函数下搜索最合适的运动时强加运动学的约束提供了一种有力的框架。然而,把基于物理的优化扩展到完全的人形上却遇到了极大的挑战,主要的原因是由于动态约束的非线性,而且对优化的起始点非常敏感。由于可以利用动态捕捉数据,所以对初始点的敏感性在某种程度上可以得到缓解。Popovic和Witkin在1999年发展了在保持物理特性的情况下转化运动捕捉数据的第一种方法。他们发现可以通过简化人体模型来得到结果。最近,基于时空优化的运动捕捉数据的编辑已经成为产生现实人物动画的一种流行的策略[RGBC96, SP04, SHP04]。这些方法通过对优化空间增加运动学约束来保证在改变运动捕捉数据的同时物理上的特性大致不变[如: RGBC96],或通过PCA降维的方法来解决[SHP04]。最近的研究显示,如果对关节的角度、力矩、Lagrange乘子给予适当的缩放和估计的话,动力学约束的简化不一定是必须的[SP04]。我们的工作使用了相似的时空优化框架。与其他方法不同的是,我们对复杂物体模型的要素约束的公式表达要简单的多,没有对肌肉力进行显式的求解,这一点和[LP02]很相似。因为我们没有计算关节的内部力矩,缩放比例和收敛问题在我们的优化框架中并不是一个很大的问题。

我们的基于物理的编辑方法的基础是由Liu和Popovic引入的要素约束。在那份工作中,要素约束被用于从简单的、并不包含足以产生整个运动的信息的动

画中合成高度动态的运动。结果是，引入了过渡姿势用于进一步的约束优化空间。与完全的动力学约束相比，要素约束有两个主要的优点。首先，相比于动力学约束被简化为全局的要素模式，我们现在只需要在一个小的多的未知集合上求解，而且约束集合的性质也更好。这使得我们可以更快地求解。同时，我们的经验是，这些约束不会因为许多全局的极小点而使结果很差，这使得我们可以找到离开初始点很远的解。要素约束的第二个优点是它们除了物理上的正确以外还更多的编码了自然的运动。例如，在自然的运动中，腱和韧带等被动的要素在弹道运动中存储和释放能量。为了用纯粹的动力学系统来建模，我们必须包含一个复杂的肌肉模型。要素约束通过特定的要素模式有效的记录了自然的力矩效应和能量的存储和释放的总体效应。在要素约束中嵌入的额外的信息保证了更改后的运动不仅仅是物理上正确的，而且它自然的要求要素的转化模式和自然中观察到的是一致的。与过去介绍要素约束的文章相比，我们的方法直接在运动捕捉数据上应用要素约束。我们的方法不需要在空中和地面之间的状态中增加过渡点的约束。更进一步的，我们引入了要素方式的一种新的基于样条的表示，这可以被用来内在的增加产生的运动和输入的运动之间的相似性。

除了明确的叙述了基于物理的优化，动态滤波器是对小尺度的运动进行编辑的另一个有效的方法。基于逐帧的框架极大的减少了计算时间，为用户提供了交互式的编辑接口[TSK02, SKG03]。

不幸的是，逐帧的方法意味着动画师可以更改约束的空间位置，但是无法改变约束的时间位置。Tak等应用Kalman滤波器估计了对于给定的约束，当前帧对象的最优姿势。这个估计的结果然后使用最小均方的方法来保证物理上合适的运动[TSK02]。Shin等人，通过在跳跃过程中校正人体的各个要素以及使用地面约束的平衡，估计了对原始的运动捕捉数据的改变[SKG03]。一般的，与全局的运动例如增强平衡相比，这些方法与本地的变化相联系。我们的方法能够控制运动的全局变化，例如把一个向前跳的过程转化为360度向后跳的过程。另一个动态滤波的分支是使用动态跟踪的方法[ZH99, PR01]。这些方法把运动捕捉数据和动态仿真结合到一起，在提供和环境交互的能力的同时保持数据的拟人细节。这些方法产生的运动不会与输入数据产生很大的偏离，但是依赖于拥有和使用者希望的相类似的捕捉运动的存在。

2.2. 基于插值的运动编辑

对关节角度的直接插值通常不能保持初始数据的物理真实性。然而，许多方

法显示运动的小改变可以很容易的由关节角度的线性插值来得到[BW95, WP95, WH97]。把插值和运动学约束结合在一起, Gleicher把原始运动变化为新的对象的同时保持了环境约束, 比如地板和脚的接触点[Gle98]。还有一种更加高级的插值, 使用的是放射基函数来把运动序列和多种逆运动学目标[RSC01]或不兼容的模式[RCB98]结合在一起。不幸的是, 这些方法的数据获取或预处理是一个很大的挑战, 因为运动序列需要仔细的加工这样它们才能包含相同的内容但是具有不同的样式。我们的方法只需要单个的运动捕捉序列作为种子。这个种子用来产生一组运动来得到参数化的运动学空间。

Lee和Shin提出了多级B样条表示法, 它可以通过直接的控制将已知的运动转化为适合特定的约束[LS99]。使用B样条表示法, 运动编辑可以被限制到用户特定的频带, 提供了一种更有效的优化框架。我们的工作对使用B样条的表示法来约束最优值的搜索的方法做了一定的改进。我们使用和原始运动相适应的B样条表示法对参数曲线进行建模, 这样优化的搜索空间被限制到和原始运动由相似的运动学行为的解上。

3. 总的看法

我们的算法是基于在转化运动捕捉数据满足高层用户的约束的同时保持物理真实性的优化算法。作为输入, 系统使用单个运动捕捉序列和用户定义的更改。我们分三个部分来描述我们的算法: 运动预适应、优化和插值(见图1)。

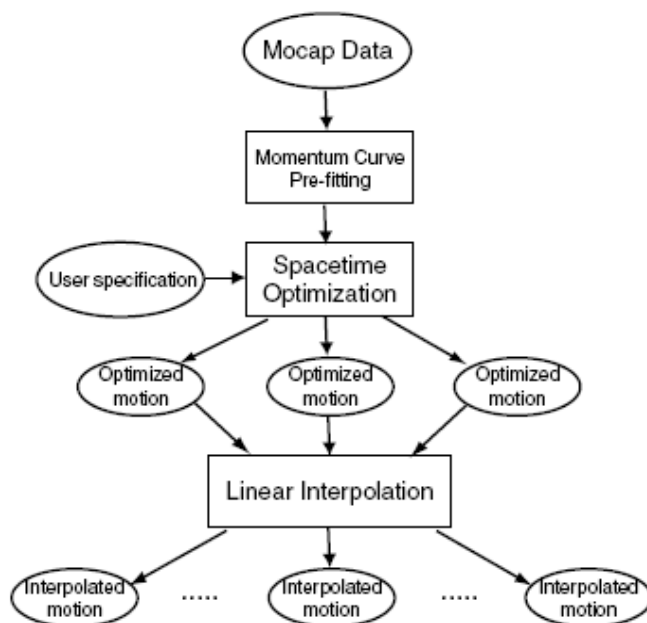


Figure 1: System overview

预适应最优化用于建模参数曲线的系数集合，这样它们受到约束，必须和初始曲面的形状是一致的。系统然后为新的运动求解一个时空优化表达式，使得高层的约束和用户定义的特性都得到满足。使用一组这样优化过的参数化一定的动态空间的运动，我们可以使用简单的线性插值在动态空间中实时的产生任意的新运动。

4. 使用要素约束的运动编辑

我们的算法改写了基于要素的约束使得它适合运动编辑的任务[LP02]。运动编辑必须在保持动画师附加的约束的同时求解保持原始数据的逆问题，而不是恢复丢失的数据。任何形式的关键帧都没有必要，因为运动已经是在一个很好的初始状态开始的。这个系统使用的任何潜在的物理模型都必须足够灵活以精确的描述运动的初始状态，同时又足够严格以在整个编辑过程中维持原始运动的外貌。

4.1. 运动预适应

我们的算法的核心是一个整体的角度和线性参数曲线的集合。这些曲线约束了被编辑的运动使得它满足物理的现实性，而不必模范昂贵的运动学特性如关节的力矩和接触力。参数曲线使用一个预解的、与输入的运动尽可能接近的参数集

合参数化。这种方法的优点是双重的。首先，一个参数系数的好的初始化状态导致了最优化过程的快速收敛。第二，控制参数形状的参数可以在编辑的过程中固定，有效的实现了在参数空间中的相似运动搜索的有偏的搜索。

在运动使用光学系统捕捉之后，再经过处理以适应人物的骨架结构，我们使用在[LP02]中描述的约束检测技术来把运动分为与地面接触和在空中两个阶段。因为动画师可能不时的希望产生不受地球引力约束的、物理上不可能的跳跃，同时因为每个输入运动序列的采样频率是不同的，我们需要决定两个动画帧之间的时间间隔。重力和时间是直接相关的，因为我们可以等价的选择合适的引力常量使得运动在给定的单位时间间隔中显得是真实的。在自由落体的过程中，线性参数仅仅受重力的影响，角参数是常量。通过观察遵循相似的轨迹的模型的质量中心(COM)，我们可以通过下面的等式计算出重力常量 g :

$$\begin{aligned} \mathbf{p}(t) &= 1/2\mathbf{g}t^2 + \mathbf{v}_0t + \mathbf{C}_0 \\ \mathbf{p}(t_n) &= \mathbf{C}_n \\ \mathbf{p}(t_{n/2}) &= \mathbf{C}_{n/2} \end{aligned}$$

这儿的 $t_{0..n}$ 是自由落体运动的时间步数。 $\mathbf{C}_{0..n}$ 是对应的物体质量中心的值， \mathbf{v}_0 是对应的COM的初始速度。

当物体受外力作用的时候，参数曲线不再由简单的线性方程的集合表示。我们这时使用三阶的非单位B样条来描述参数曲线，因为它们基于参数的灵活方便的节点。在我们的样条表示中，第一个和最后一个节点的自由度是4，保证最后点的插值[FvDFH92]。

一个运动的决定性的特征就是它的形状和它的参数曲线的数量(见图2)。在我们的样条表示的形式中，控制点决定了曲线的量级，点的分布也会影像形状。我们注意到这种表述方式可以比过去使用的硬线方式捕捉到参数方式的更大的可变性[LP02]。这在处理现实世界中可能的大量不同的策略运动捕捉数据时显得特别重要。为了找到一个控制点的集合， $\{\mathbf{c}_i | i \in 1..k\}$ 和点的集合 $\{\mathbf{u}_i | i \in 1..k+4\}$ ，它与输入运动的参数模式的结合比较紧密，我们为每一个参数样条 $\mathbf{S}(t, \mathbf{c}_{0..k}, \mathbf{u}_{0..k+4})$ 求解下面的优化问题。

$$\min_{\mathbf{S}} \sum_{i=0}^n (\mathbf{m}_i - \mathbf{S}(t_i))^2 \quad \text{subject to} \quad \begin{cases} \mathbf{S}(0) = \mathbf{m}_0 \\ \mathbf{S}(n) = \mathbf{m}_n \\ \dot{\mathbf{S}}(0) = \mathbf{v}_0 \\ \dot{\mathbf{S}}(n) = \mathbf{v}_n \end{cases}$$

$$\mathbf{u}_i - \mathbf{u}_{i-1} < \epsilon, \text{ for } i \in 1..k+4$$

这儿的 \mathbf{m}_i 是输入运动在时间步 i 的参数， $\mathbf{v}_i = \mathbf{g}M$ ，这儿的 \mathbf{g} 是相邻的飞行阶段的重力加速度常量， M 是物体的体质量。换句话说，我们对地面阶段的参数曲线进行最小方差回归，同时在过渡到飞行阶段的时候保持 C^1 的连通性。

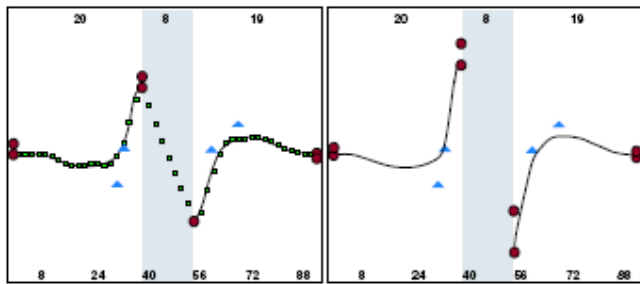


Figure 2: Linear momentum of a jumping motion in vertical direction. The gray area indicates the flight stage. Left: The control points $\{\mathbf{c}_i | i \in 1..k\}$, visible as red circles, determine the magnitude of the curve. The spacing of the knots $\{\mathbf{u}_i | i \in 1..k+4\}$, visible as blue triangles, influence the shape. Pre-fitting phase solves for a set of control points and knots that closely match the momentum pattern of the input motion (shown as green squares). Right: During the spacetime optimization, \mathbf{u}_i is held fixed while \mathbf{c}_i is part of free variables. In this example, the optimized control points \mathbf{c}_i result in a more energetic jumping motion.

对上面描述的问题有一些例外。当没有相邻的飞行阶段的时候，我们把对应于 \mathbf{v}_i 的约束从问题的状态中移去。同时，在预适应矢量线性参数曲线的时候，对应于 \mathbf{v}_0 的约束完全移去，因为从自由落体阶段到地面阶段的过渡是完全由冲量决定的，这在垂直分量上不是 C^1 连续的。

4. 2. 运动编辑和优化

本节我们系统中运动编辑的过程。和[LP02]中的一样，我们建立的运动模型是一个有约束的动态优化的过程。在一般的情况下，我们优化的目标是产生的运动应该尽可能的光滑，和原始运动尽可能的相似。对解的约束保证了人体不会不

自然的弯曲，如人的脚不会穿过地面，人体的整体运动曲线和预设置的样条曲线符合。被优化的自由度包括在 \mathbf{QUG} 中，这儿 \mathbf{Q} 是描述运动的随时间变化的关节角度集合， \mathbf{G} 是控制运动曲线的控制点集合。在优化的初始状态 \mathbf{Q} 是通过原始运动和用户特定的平移和朝向之间的一个线性插值，这是对目标运动的很好猜测。 \mathbf{G} 包括动量系数的预适应。除了在 [LP02] 中使用的约束和目标函数，我们引入了一个相似的目标函数和一个伪平衡的目标函数，这些将在下一节阐述。

4.2.1. 相似性目标函数

相似性目标函数是为了使得优化后的运动和初始运动尽可能的相似。我们建立的目标函数的公式是 DOF 的原始运动矢量 \mathbf{Q}_{init} 和解运动的矢量之间的平方距离。每一个 DOF 通过它的自然界来衡量。我们希望最小化的能量函数即为：

$$E_s(\mathbf{Q}) = (\mathbf{Q}_{init} - \mathbf{Q})^2$$

4.2.2. 伪平衡目标函数

人类有逃避极端倾斜运动的偏好，因为在现实中会导致滑倒在地。但是我们的模型没有考虑这点。解运动可能会出现人体在没有支撑的情况下不自然倾斜的情况。为了使优化的解脱离这些不稳定的区域，我们包含了一个伪平衡的目标函数。我们使用的目标函数试图最小化模型在第一个时间段 t_0 和最后一个时间段 t_f 之间的 COM，即 $\mathbf{C}(t)$ 的二次距离，即运动的初始和最终着地的两步。对于内部的着地的各步，我们转而优化中间帧的 COM，即 $\mathbf{C}(t_m)$ 和输入的线性插值帧之间的 COM，即 $\mathbf{C}_{orig}(t_m)$ 在同一步时两者之间的距离。换句话说，我们引入了附加的目标函数项，对于初始和最终的两帧是 $E_b(\mathbf{Q}) = (\mathbf{C}(t_0) - \mathbf{C}(t_f))^2$ ，对于内部的着地的几步，是 $E_b(\mathbf{Q}) = (\mathbf{C}_{orig}(t_m) - \mathbf{C}(t_m))^2$ 。我们发现这些目标函数合适的权值对于不同的运动相差并不相同。实际上只要这个权值相对于目标函数的其他部分是合适的，那么这个值几乎就是万能的。

4.2.3. 时空优化

总结一下，现在我们系统中的未知量 \mathbf{Q} 和 \mathbf{G} ，是 DOF 和运动样条控制点的特征。注意样条的节点为了保持和原始运动的相似性而忽略了。优化包含两方面的约束：环境约束 \mathbf{K}_e ，例如脚的位置应该在地面上；运动约束 \mathbf{K}_m 。下面的时空表达式得到了在满足所有优化的情况下最小化目标函数的 \mathbf{Q} 和 \mathbf{G} ：

$$\min_{\mathbf{Q}, \mathbf{G}} E_s(\mathbf{Q}) + E_b(\mathbf{Q}) \quad \text{subject to} \quad \begin{cases} \mathbf{K}_e(\mathbf{Q}) & = 0 \\ \mathbf{K}_m(\mathbf{Q}, \mathbf{G}) & = 0 \end{cases}$$

4.2.4. 用户接口

我们的系统提供了几种高层的运动特化工具这样动画师就不需要在编辑的时候考虑约束优化的问题。首先，运动自动的分割成可选的飞行状态和地面状态。用户也可手工的调整分割来保证准确。接下来，用户使用鼠标来调整地面的几步来进行平移和改变朝向。系统把这些特化看作原始的着地状态的一种偏移。换句话说，给定原始的位移 \mathbf{qT} ，和原始的朝向 θ ，用户分别指定偏置 $\Delta\mathbf{qT}$ 和 $\theta + \Delta\theta$ 。

为了得到飞行状态的一个好的假设解，系统通过相邻两个着地帧的偏置的线性插值来进行。得到的结果是最终结果的一个粗略的估计，但是提供了时空优化的一个很好的初始状态。动画师可以通过交互式的规定轨道的可见形状来得到飞行状态轨道的高度。这在创建不符合重力定律的非现实运动时尤其有用。一旦用户对编辑的结果感到满意了，优化过程大约要化1到5分钟。同时也可以以批处理的方式产生几种运动。

4.3. 产生运动的动态空间

本节我们阐述通过单个的运动捕捉序列来产生一系列连续的物理上很像真实的运动的技术。这种技术通过一个样本运动的集合的DOF的简单的加权平均来实时的产生输出的运动。一族运动可以先通过系统的改变一个或多个着地的步的位置和朝向来得到输入，然后进行类似的优化。

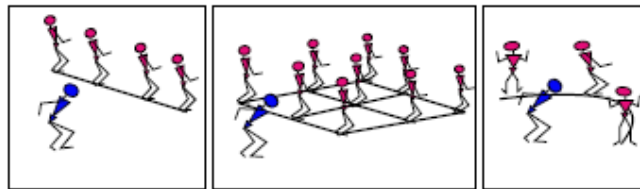


Figure 3: Left: Line motion family that varies the translation of a ground stage along a line. Middle: Grid motion family that varies the translation of the ground stage along a 2 dimensional grid. Right: Circle motion family that varies both the translation and orientation of the ground stage along a semi-circle such that the orientation of the character is always aligned with the normal vector on the arc.

4.3.1. 运动族

我们为三个最常使用的运动族(见图3)提供了接口。第一种是着地的运动在一条线上变换,第二种情况是着地的运动在一个二维的网格上运动。第三种情况是着地的运动在一个半圆上运动,这样人体的朝向在椭圆的法向上连续的分布。样本空间的大小和密度在必要的时候可以进行调整。其他类型的运动族可以很容易的加入。

一旦一个运动族产生了,我们可以很容易的通过扭曲最近的 2^n 个点来产生任意的中间运动,这儿 n 是参数空间的维数。我们选择简单的线性弯曲有这样几个原因。首先也是最重要的是算法必须快速,可以适合于飞速产生输出运动的应用。因为运动族是在线产生的,它们可以任意密的产生来增加插值的精度。其次因为运动族的各个成员是通过相同的优化步骤产生的,仅仅是在特定维上有所不同(如着地位置,高度,朝向等等),经常出现的情况是它们已经弯曲的很好以至完全不需要非常密的采样。在我们的结果节中,我们最多需要九个样本就可以采样运动的动态空间了。

4.3.2.滑步

尽管滑步是大多数运动完全技术最令人讨厌的人造痕迹,我们发现这种影响对线和网格运动族的影响微乎其微。但是当愈多农的整体朝向和位移同时插值的时候,即在圆形运动中出现的情况,很小数量的滑步会比较严重。一个简单的修补的方法就是使用一个每帧IK算子来微调身体的下部来保证每条腿上的位置约束。求解身体下部的IK问题不但可以在保持运动的整体外观的情况下保证脚牢固的在地面上,而且它的影响非常小,可以保证实时收敛。

4.3.3.逆控制

到目前为止我们已经显示了如何通过样本的插值来产生动态运动空间。这儿我们讨论一种更直观的控制这些动画的方法。在许多应用中最重要的是空是在什么时间什么位置物体和环境中的物体接触。考虑一个头顶足球人的运动,这里需要人的头部在适当的时间始终和球相接触。从单个的输入运动开始我们可以通过产生一个格点运动族来产生任意的顶球者。在任意时刻的输出序列的节点配置被定义为矢量 $q(x, y, t)$,其中位置为 (x, y) ,时刻为 t 。如果我们用函数 $h(q)$ 来表示人的头部,那么原来试图找一个运动满足人的头部在 tc 时刻满足球的位置 pc 的问题就转化为了寻找 (x, y) 满足方程 $pc=h(q(x, y, tc))$ 。这就转化为最小化能力函数 $E(x, y)=(pc-h(q(x, y, tc)))^2$,这可以通过简单的梯度下降的方法来解决。梯度通过有限差分的方法来得到。一个值得注意的地方是 q 实际上是一个分段函

数, 表现为4个最近点的双线性插值。当一个样本运动另一个大小为4的代替的时候, q 不再是C1连续的, 导致梯度下降法出现收敛的问题。一个简单的解决方案是把线性弯曲函数 $f(x)=x$ 和 $g(x)=1-x$ 用光滑的输入/输出函数如 $f(x) = \sin^2(x)$ 和 $g(x) = \cos^2(x)$ 来解决。这样就通过过渡保持了C1连续。

4.4.交互式控制

我们的运动产生算法的一个优点就是它可以实时的提供很宽范围的物理上看上去是真实的动画。为了示范这种方法的全部好处, 我们创建了一个视频游戏的接口, 用户通过一个多方向控制板来控制跳跃物体的轨迹(见图6)。我们从一个运动捕捉数据来产生两个连续的跳跃。这个运动的一个有趣的方面就是人体必须在第一跳的运动中显示出一定的远见, 这样才能在中间的着地时刻产生正确的作用力来为第二次跳跃产生适当的动量。时空方法是编辑这样运动的理想方法, 因为它内在建模了运动的不同时刻之间的内在独立性。我们方法从时空方法中继承了同样的关键好处, 但是允许我们实时的产生运动。

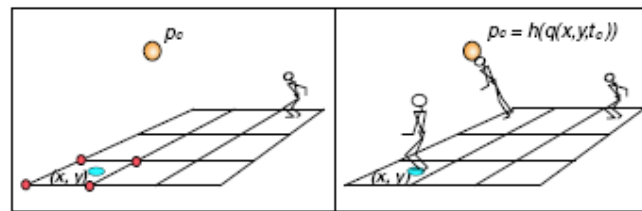


Figure 4: In the example of the soccer header motion, the user specifies the contact point of the head and the soccer ball p_c at timestep t_c . Inverse control mechanism is able to immediately determine the four nearest neighbors among the sampled motions as well as their weights that interpolate the desired motion. An efficient gradient descent method solves for the landing position (x, y) by optimizing $E(x, y) = (p_c - h(q(x, y, t_c)))^2$.

在显示中我们希望控制第一跳和第二跳的水平平移矢量, 各自为 $d1$ 和 $d2$ 。首先我们通过在一个 3×3 的网格上变化着地点来产生运动族。整个的运动族包含由9个起始点和9个终止点排列组合产生的81个最优点。这对于两跳之间着地时刻的完整采样是必要的。我们然后就可以通过最近邻的最优解的线性插值来产生整个空间。在这种情况下, 我们的样本空间相对于矢量 $d1$ 和 $d2$ 的维数是4, 形成了一共16个最近邻的运动。因此我们可以把输出运动表达成矢量函数 $q(d1, d2)$

的形式，这儿 d_1 和 d_2 在样本空间的范围中。为了使得我们的显示更有趣一些，我们把跳跃运动链接起来，这样就形成了连续的循环跳跃。这可以通过将第一个运动的第二个空中阶段 $q_a(d_{a1}, d_{a2})$ 变形成第二个运动的第一个空中阶段 $q_b(d_{b1}, d_{b2})$ 。为了使得变形起效果，我们简单的要求 $d_{a2} = d_{b1}$ 。换句话说我们要求变形的运动在长度和方向上都是一致的。

最后的结果就是一个交互式的跳跃仿真，这里用户控制人体跳跃的方向，然后就可以看到物理上似真的运动。根据前面讨论的远见，人体必须总是对接下来的两个跳跃的方向有先验的知识。这就导致如果在运动即将产生的时候用户设定了运动的方向的话会产生一定的延迟。但是这仅仅是轨迹运动的确定性特征。

5. 结果

我们显示中的运动序列是以120f/s的速度捕捉的，使用的是光学运动捕捉系统。人体由18个刚性链接和43个自由度构成。SO(3)旋转使用指数映射的方式表达。模型的质量分布近似是按[DL96]中的方法获得的。我们使用SNOPT[GSM96]一种非线性约束优化软件包来解决时空优化和参数曲线的预匹配的问题。在给出的视频片断中的大多数编辑在1分钟内完成。对每个运动的优化在2GHz的Pentium4机器上使用2~4分钟(见表1)。

Motion	Sequences	Frames	Time
Forward jumps	1	46	2 min
Two-step hop	1	49	3.5 min
360 degree spin	1	79	3.5 min
Volleyball slam	9	44	17 min
Interactive controller	81	56	4.5 h

Table 1: Computation time for optimizations

5.1. 运动编辑

我们的系统提供了UI工具来帮助用户快速的对已知的运动进行特定的修改。在单脚跳的运动中，动画师交互式的产生每一步的位置，高度和朝向。人体必须跨越一个长距离，到达更高的高度，在更改构的单脚跳运动中假设了新的方向，这样她必须降低质心，更向右倾斜，轻微的自转以为起跳做准备。尽管有这

些改变，结果运动依然和原始运动在姿态上是一样的。为了显示我们的运动可以在原始运动上进行强烈的修改，我们编辑了一个运动，先是360度的旋转紧接着是180度的反方向旋转(见图5)。

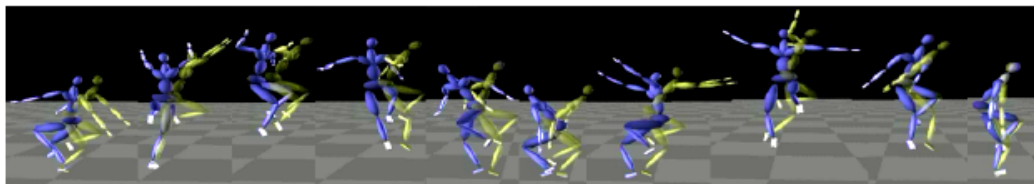


Figure 5: A forward hopping motion (shown in yellow) is modified to make a 360 degree spin in the clockwise direction followed by a 180 degree spin in the opposite direction (shown in blue).

5.2. 实时插值

为了显示实时的运动插值，我们更改了一个有两个连续的跳跃的运动。我们让用户在一个均匀的网格上控制着地点和起跳点，来产生一组参数化的运动。因为插值可以实时的显示，我们可以通过在参数空间内的任意的起跳和着地位置来交互式的产生运动。另一个例子显示了一个顶球的运动，这个人好像丢了球。一开始我们通过增加跳跃的高度来修正运动，以使在接触点碰到球。然后，我们使用我们的运动编辑算法来产生一个运动族，它是在运动的着地位置空间上参数化的。通过原始运动的插值，我们可以实时产生任意的中间运动，使得人体在样本空间的任意位置都可以接触到球。

5.3. 逆控制

一个在任意的位置约束下编辑运动捕捉数据的更直观的方法是使用我们的实时逆控制原理。在打排球的运动中，用户交互式的指定人体在中间飞行过程中的手的位置，我们的系统立即通过4个最邻近的样本的线性插值来满足手的约束。样本运动在地板上的亮度显示了和每个样本相关的权值。我们使用9个样本运动，它们都是同一个输入序列的编辑结果。演示程序显示了通过排球的轨迹来引导人体运动实时产生的不同的打击运动(见图6)。

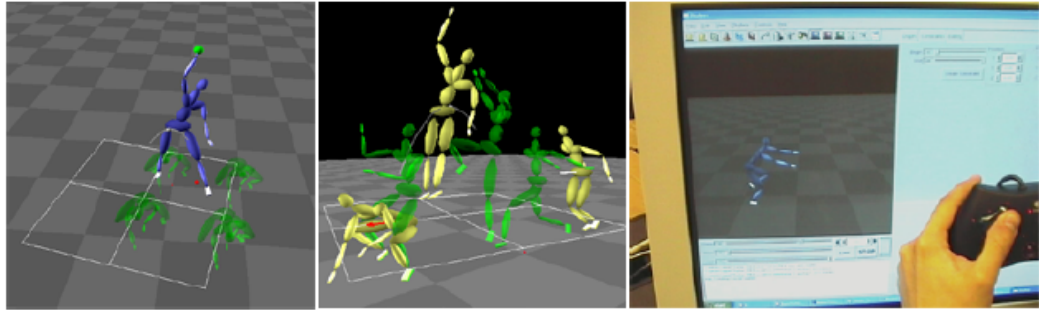


Figure 6: *Left: For a volleyball slam motion, the user interactively specifies the position of the character's hand in mid-flight. The system then determines the correct linear interpolation of the sampled motions to meet the positional constraint on the hand. Middle: The volleyball motion in profile. Right: The user interactively controls the direction the character jumps with a multi-directional control pad.*

5.4.非真实运动

我们的系统也可以一类非真实的运动，以允许人体显示超人力量和违反物理定律。考虑这样一个例子，我们希望编辑一个跳跃的运动来达到一个比原有的运动在相同的时刻更高的中点。我们首先看到这在不改变重力常数的情况下在物理上是不可能的，因为它决定了人体从一定的高度下落着地时的最大速度。在我们的系统中可以很方便的在一个或更多的着地时刻更改引力常数。人体依然需要在起跳的时候获得必要动量来达到设定的高度，而且在着地的时候也产生相同的动量。这需要一种超人的肌肉强度，但因为我们不直接对肌肉力量进行建模，我们对它们的强度也没有限制，我们的系统可以很容易的操纵这些想象的环境。从动画师的角度看，编辑非真实感的动画实际上和编辑其他所有的运动是一样的。为了增加飞行时刻的高度，动画师只需要简单的操纵飞行阶段的运动轨迹使之和需要的高度一致即可，然后指定系统是否需要改变重力常量或者飞行阶段的总时间。如果动画师选择保持重力常量不变，系统增加了飞行阶段每一帧的时间度量，然后和正常情况一样进行编辑。在一个例子中，我们把一个向前的跳跃编辑为一个向后的两米远的跳跃(见图7)。

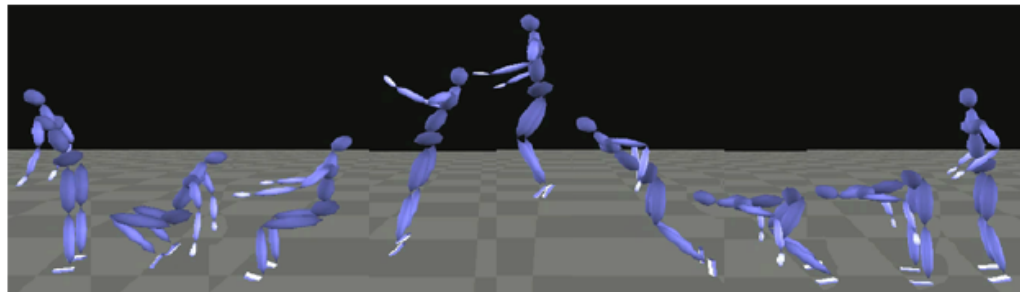


Figure 7: The timeline of this animation goes from left to right. To demonstrate a motion that is impossible to achieve in the real world, the animator altered a forward jump to a 2-meter-long backward jump.

6. 结论

这个工作建立在基于物理的运动综合和基于插值的运动编辑方法的基础之上。本文中我们提出的建议是，使用基于物理的调整来创建样本运动以供数据插值之用可能是这两种方法的一个很好的结合点。一旦数据集建立了，这个范例使得动画师可以交互式的编辑动态运动。

这个工作的基本贡献是提出了一种新的调整轨迹人体运动的方法。和过去的基于运动学的调整算法相比，我们的框架可以产生和原有运动有很大不同的运动。我们的方法不需要模型的简化或运动空间的简化。因为我们并不求解每个关节角度的一般力，我们的方法也比其他的基于物理的变换方法要快的多。这种速度允许我们在合理的时间内创建大量的运动。

一旦参数化的运动样本创建好了，我们描述了交互式的框架，使得动画师可以面对真实的运动空间。我们也显示了这个框架同时也可以用于逆控制的调整。最后，我们显示了如何从一个运动捕捉序列来构造实时的人体运动的数据驱动控制器。

自然的，我们的框架并不能处理所有的现实人体运动。它专门是应用于有飞行阶段的高速运动物体的。我们推测基于要素的方法不能很好的适合能量较低的运动比如走动。而且需要的采样数目和空间的维数是指数相关的，因此当前的框架的一大阻碍就是必须进行大数据集的离线计算。有很多方法可以通过利用我们实际上是在解一系列相似的问题来简化计算。一个更智能的采样策略对于把我们的方法扩展到多维的运动空间是很重要的。因为我们的模型没有对真实的肌肉和接触处的摩擦力进行建模，对一些极端的情况不能产生真实的运动。增加诸如平衡等启发式方法可以在很大程度上降低这些问题的重要性。

7.致谢

尤其感谢Mira Doncheva对创建视频提供的帮助。我们也感谢Keith Grochow对处理运动捕捉数据提供的帮助。此项工作由UW动画研究实验室，NSF基金CCR-0092970，ITR 基金IIS-0113007，EIA-0121326，NSF REU基金，Alfred P. Sloan奖学金，Electronic Arts 公司，Sony和Microsoft Research资助。

综合论文训练任务书

姓名 潘佳 学号 2001011783 班号 自14

系别 自动化 同组姓名 指导教师 徐文立

一、 课题名称：

三维运动捕捉数据的关键帧提取与恢复

二、 论文主要内容及进度安排：

开始—3月中旬:前期资料调研,确定研究方向,学习对应的计算机视觉和射影几何基础.

3月中旬—4月中旬:微分几何学基础,熟悉平台.数据格式的读写,可以显示提供的三维视觉捕捉数据.

4月中旬—5月中旬:编程实现关键帧的提取算法、利用关键帧的插值算法,误差的比较.

5月中旬—6月初:优化代码,并最终完成论文.

三、 论文主要要求（如主要指标）:

- 1.数据可以高效准确的读取
- 2.可以比较有效的提取关键帧,关键帧应该有一定的代表性
- 3.算法的效果应该超过已有有算法
- 4.关键帧应该可以有效的插值恢复
- 5.算法的速度限制:处理1000 帧的时间应该在1~2分钟内

四、 主要参考文献:

- 1 Feng Liu,Yueting Zhuang,Zhongxiang Luo,Yunhe Pan.A Hybrid Motion Data Manipulation: Wavelet Based Motion Processing and Spacetime Rectification.Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing, p.743-750, December 16-18, 2002
- 2 David Eberly.Quaternion Algebra and Calculus.Magic Software,Inc.September 27,2002
- 3 William H.Press,Saul A.TeuKolsky,William T.Vettering,Brian P.Flannery.Numerical Recipes in C++,The Art of Scientific Computing,Second Edition.The Press of the University of Cambridge,Cambridge,England.2002.108~126.
- 4 刘丰,庄越挺,潘云鹤,罗忠祥.基于小波变换的运动分析及其应用.中国图像图形学报,volume 8(A),Number 1,Jan 2003
- 5 罗忠祥,庄越挺,潘云鹤,刘丰.基于时空约束的运动编辑和运动重定向.计算机辅助设计与图形学学报,Volume 14,Number 12,Dec 2002

教学负责人签字 李强

05 年 6 月 29 日

指导教师评语:

论文针对三维运动捕捉数据的特点,利用整体运动曲线的挠度与曲率信息,使用了改进的时序动态聚类算法,实现了关键点提取,并进行了插值恢复。由于引入了曲线的几何性质,该算法的整体效果超过了文献中已有算法,论文工作有创新性。

指导教师(签字)

徐文立

评阅教师评语:

论文针对三维运动捕捉数据的特点,利用整体运动曲线的挠度与曲率信息,实现了关键点提取,并进行了插值恢复。论文工作具有创新性,提出的算法在整体效果上取得了很好的效果,达到了学士论文的要求。

评阅教师(签字)

李小明

答辩小组意见:

该生答辩中叙述清楚,
回答问题正确。

组长(签字)

石宗英

成绩:

92

教学负责人(签字)

李小明

05年6月29日