

一种基于质心的卡通角色自动匹配方法

张韶华¹, 张宝印², 崔建军³

(1. 解放军信息工程大学 测绘学院 遥感信息工程系, 郑州 450052; 2. 西安测绘研究所 三室, 西安 710054; 3. 长安大学 地质工程与测绘学院, 西安 710054)

摘要: 引入了角色质心的概念, 并将其作为匹配条件应用到卡通角色对应关系的自动检索中, 有效解决了卡通角色在图形对象数量不一致时的匹配对应问题, 实现了卡通角色的自动匹配。该方法已经在实际动画制作过程中进行了应用, 应用结果表明它能有效解决动画造型之间相互对应的问题。

关键词: 动画; 卡通角色; 图形序列; 质心; 自动匹配; 关键帧内插

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2012)05-1744-03

doi:10.3969/j.issn.1001-3695.2012.05.038

Auto-matching method of cartoon role based on centroid

ZHANG Shao-hua¹, ZHANG Bao-yin², CUI Jian-jun³

(1. Dept. of Remote Sensing Information Engineering, College of Surveying & Mapping, PLA University of Information Engineering, Zhengzhou 450052, China; 2. Mimuro, Xi'an Surveying & Mapping Research Institute, Xi'an 710054, China; 3. College of Geology Engineering & Geomatics, Chang'an University, Xi'an 710054, China)

Abstract: This paper introduced the concept of the role's centroid, and applied it to automatic retrieval of the corresponding relationship cartoon roles as matching condition, effectively solved matching problem between cartoon roles which had different number of graphic objects, realized the automatic matching process of cartoon roles. The method had been applied in the actual animation process. The application results show that it can effectively solve the corresponding problem between cartoon roles.

Key words: animation; cartoon role; graphic sequence; centroid; automatic matching; key frame interpolation

动画产业经过近些年的快速发展,采用计算机进行全数字动画制作的效果已经初见端倪,特别是随着“无纸动画”理念在动画行业的不断深入和贯彻,计算机辅助下的数字化二维动画制作方法最终已经逐渐成为二维动画制作的发展潮流。在这一潮流中,如何利用计算机技术实现动画帧的自动内插是促进二维动画制作发展的关键所在,而其核心又在于如何确定不同帧之间卡通角色的匹配对应关系。目前国外流行的 Flash、Toon Boom Studio 等动画制作软件采用的做法都是动画师在绘制原画时就严格地按照前一幅原画的图形内容和顺序进行绘制^[1],也就是通过原画师人为地控制原画之间图形的对应关系,没有利用计算机技术来实现自动化的匹配对应,从而也很难做到原画帧的自动内插,这就导致采用上述软件进行动画制作时有质无量或有量无质。因此,能否快速、智能地建立原画图形之间的相互对应关系,已经成为数字化制作卡通动画的关键问题之一。

1 算法描述

“动画”一词,包含着两个极为重要的元素——“动”和“画”。“动”是动画区别于其他绘画艺术的重要特征。因此,制作动画需要遵循的首先就是一系列与运动有关的原理^[2]。实际上,“动”的主体是角色,即卡通动画中的角色不会脱离现

实的关系变化,也就是说组成卡通角色的图形之间是具有一定相互关系的,而且这个关系一般不会随着角色动作的变化而变化,即使发生了变化,这个变化也是有规律可循的^[3,4]。因此,就可以充分利用卡通序列的内部信息(角色各多边形之间相互位置信息),根据角色各多边形的质心位置变化具有连续性这一规律,在相邻帧内寻找各角色各多边形之间的匹配关系。总体思路如下:a)计算每个多边形的质心;b)根据多边形的面积求其“质量”;c)把角色看成由各个有质量的多边形组成的整体的质心位置,计算出该角色的公共质心位置;d)把各个多边形质心与公共质心的关系作为匹配条件完成各多边形之间的配准。

1.1 多边形的质心

在同一角色内有若干个多边形,可用 $S = \{C_{t_i}, i = 0, 1, \dots, N-1\}$ 来表示。设 N_s 为该角色上多边形的个数,通过对每一个多边形 C_{t_i} 分别求出其质心 $C_i = (x_i, y_i, z_i)$,可得到质心点的集合 $C(z) = \{C_i, i = 0, 1, \dots, N_s - 1\}$ ^[5]。质心的计算方法如下:

用函数 $f(x, y)$ 表示多边形,其在多边形内时取值为 1,其余情况下均为 0。则函数 $f(x, y)$ 的矩集定义为

$$M_{jk} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^j y^k f(x, y) dx dy$$

其中: j 和 k 为非负整数。它的一阶矩和零阶矩分别为

$$M_{10} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy$$

收稿日期: 2011-09-19; 修回日期: 2011-10-26

作者简介: 张韶华(1976-),男,浙江东阳人,工程师,博士研究生,主要研究方向为数字摄影测量与遥感应用、图形图像处理(cainiao_god@126.com);张宝印(1965-),男,研究员,博士,主要研究方向为图形图像处理、卡通动画制作;崔建军(1974-),男,副教授,博士,主要研究方向为图形图像处理。

$$M_{01} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y^k f(x, y) dx dy$$

$$M_{00} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy$$

其中, M_{00} 是多边形的面积, 则这个多边形的质心坐标为

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}} \quad (1)$$

1.2 多边形的质量

把角色内每一个多边形都当做一个物体, 当各多边形代表的物体质量密度相同时, 各多边形的质量与其所围的面积成正比(在具有先验知识的情况下, 也可根据实际情况对不同多边形赋予不同的权值, 以体现密度差异)。所以其质量 Q_i 可表示为

$$Q_i = \Delta_i A_i \quad (2)$$

其中: Δ_i 为多边形 C_{l_i} 的密度, 设为 1; A_i 为多边形的面积。

对于多边形面积的计算, 本文采用了目前较常用的弧段面积算法, 即认为多边形由一条或多条弧段构成, 用 $y = f(x)$ 表示多边形的每条弧段, $[a, b]$ 为弧段区间, 则弧段 $y = f(x)$ 、 $x = a$ 、 $x = b$ 与 x 轴所构成的封闭区域如图 1 所示。

设前进方向的右侧为区域的方向, 则该区域的面积为

$$A_{arc} = \int_a^b f(x) dx \quad (3)$$

其中 $f(x) dx$ 就是直角坐标下的面积元素, 它表示高为 $f(x)$, 底为 dx 的矩形面积。

通常, 图形的弧段由离散化的坐标串 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 表示, 因此可利用 x 将弧段区域分成一些窄曲边梯形, 则利用梯形面积计算方法, 可将式(3)转换为

$$A_{arc} = \int_a^b f(x) dx = \frac{1}{2} [(y_1 + y_2)\Delta x + (y_2 + y_3)\Delta x + \dots + (y_{n-1} + y_n)\Delta x] \quad (4)$$

其中, $\Delta x = x_{i+1} - x_i$, 再对式(4)进行离散化, 得到

$$A_{arc} = \frac{1}{2} \sum_{i=1}^{i=n} (y_{i+1} + y_i) (x_{i+1} + x_i)$$

设多边形由 m 条弧段构成, 则多边形面积为

$$A_{polygon} = A_{arc1} + A_{arc2} + \dots + A_{arcm} \quad (5)$$

1.3 角色的公共质心

在利用式(5)求出角色内各多边形的面积后, 即可通过式(2)计算出各个多边形的“质量”, 而多边形的质心由式(1)求得。当角色内所有多边形的质心和质量都得到后, 即可以算出该角色的公共质心位置 $C_g(z)$, 计算公式如下:

$$c_g(z) = \left[\frac{\sum_{i=0}^{N_s-1} m_i x_i}{\sum_{i=0}^{N_s-1} m_i}, \frac{\sum_{i=0}^{N_s-1} m_i y_i}{\sum_{i=0}^{N_s-1} m_i}, z \right] \quad (6)$$

多边形与公共质心的关系如图 2 所示。其中, 黑色点代表多边形的质心, 白色点即为公共质心点。

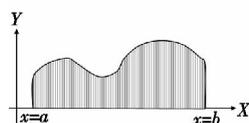


图1 弧段面积计算

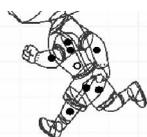


图2 角色公共质心

由于角色公共质心的位置直接影响匹配效果, 因此除了从角度上对公共质心进行判断和计算外, 还要在距离上对公共质心进行进一步的修正, 即计算角色各多边形的质心与角色当前公共质心的距离^[6,7]。此时需要对式(6)进行距离加权重新计

算公共质心, 距离越大, 权值越小, 这样就减少了距离公共质心较远的多边形发生较大变化时上下两帧的公共质心产生较大偏移的问题。

2 卡通角色匹配

对待匹配的前一帧角色内各多边形 $C_{l_i}^u$, 将其质心的 x, y 坐标 (x_i^u, y_i^u) 转换为相对于公共质心 C_g^u 的极坐标 (ρ_i^u, φ_i^u) ; 同样对待匹配的后一帧中角色内的各多边形 $C_{l_j}^l$, 也将其质心的 x, y 坐标 (x_j^l, y_j^l) 转换为相对于公共质心 C_g^l 的极坐标 (ρ_j^l, φ_j^l) 。这样如果对于前一帧角色内每个多边形质心的极坐标 (ρ_i^u, φ_i^u) , 后一帧的角色内也存在一个相对应的多边形 $C_{l_j}^l$, 其质心的极坐标 (ρ_j^l, φ_j^l) 和面积都非常接近, 那么这一对多边形就被认为是相匹配的多边形。实际上就是要满足以下三个条件: a) 造型质心的极坐标半径值最接近; b) 造型质心的极坐标幅角最接近; c) 造型多边形的面积最接近。

这是因为幅角相差很大的多边形间不会存在匹配关系(如图 3(a) 中多边形 A 与图 3(b) 中的多边形 C' 不相匹配), 所以在计算过程中只需在相邻角色和幅角在一定阈值范围内的多边形之间寻找匹配关系。设在后一帧图形内的多边形中, 与前一帧图形内多边形 $C_{l_i}^u$ 质心极坐标幅角相差在阈值 θ 以内的多边形集合是 $PM_i = \{C_{l_{i1}}^l, C_{l_{i2}}^l, \dots, C_{l_{in}}^l\}$ 。如图 3 所示, 对前一帧中的多边形 A , 此集合 PM_i 即对应于图 3(b) 中扇形区域内的多边形集合。

此时, 设计一个度量函数 $f(i, j)$, 以衡量多边形 $C_{l_i}^u$ 与 PM_i 中各卡通角色匹配的可能性, 公式如下:

$$f(i, j) = |\rho_i^u - \rho_j^l| + \alpha |\varphi_i^u - \varphi_j^l| + \beta |\gamma_i^u - \gamma_j^l|; j = i_1, i_2, \dots, i_N$$

其中, α 和 β 分别是调节这三项比例的调节因子, 具体值可以通过实验来确定。上式中的三项分别对应于第二种方法中的三个条件, 其中第一项用于衡量极坐标半径值相差大小; 第二项用于衡量极坐标幅角相差大小; 第三项用于衡量多边形大小相似程度。这样在 PM_i 中, 使 $f(i, j)$ 值最小的一个多边形 $C_{l_{is}}^l$, 即为 $C_{l_i}^u$ 在后一帧角色内的匹配多边形 $C_{l_{m(i)}}^l$ 。

由于角色公共质心在相邻帧会有一定的移动, 因此距公共质心较近的多边形在相邻帧上幅角的变化会比较大, 此时前面所叙述的只对幅角相距较小的多边形才进行匹配的方法难以达到好的匹配效果。其解决的办法是设定一阈值 Ω , 然后对多边形质心与公共质心距离大于 Ω 的多边形, 使用上述方法处理; 反之, 对于多边形质心与公共质心距离小于 Ω 的多边形, 则需要比较它与后一帧角色内所有多边形的度量函数 $f(i, j)$ 的值, 以找到最优匹配多边形。

3 实验结果和分析

本文采用该方法对两幅动作较大的原画进行了匹配和内插实验。一般情况下每幅原画中只有一个造型, 因此匹配的主要目的是确定组成造型的各个多边形之间的对应关系, 也就是根据前一帧原画中的多边形自动检索后一帧原画中对应的多边形。多边形对应关系建立以后, 就可以利用多边形之间的内插关系对其进行内插, 从而得到中间帧。但实践证明, 简单的线性插值在对动画造型进行内插时, 往往会造成动画造型的缩减和变形, 而单纯采用圆弧内插又往往造成复杂多边形内插时

的扭曲变形或改变相互之间的连接关系。因此,为了避免造型实体的缩减同时结合实际的动作原理,本文以线性插值和圆弧法相结合的算法来进行中间帧插值计算,同时为了避免造型的扭曲变形,则在圆弧法插值之前,采用了骨架绑定方法,将角色造型绑定到一条或多条骨架线上,然后对其进行内插计算,最后根据骨架线的内插结果反算角色造型。

实验数据截取自正在制作的动画片《如意兔之开心农场》,原画序列如图 4 所示。



图3 相邻帧质心匹配

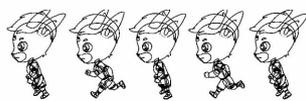


图4 实验数据的原画帧序列

对图 4 中第一帧和第二帧进行匹配, θ 设为 Π , 并内插了一帧中间帧,内插帧如图 5 所示。

图 6(a) 和(b) 分别代表原画帧序列中第一帧和第二帧去掉头部后的图形(头部不参与匹配和内插)。



图5 自动匹配之后进行内插的效果



图6 匹配计算的对应关系

其中,序号 1 代表的是身体多边形;序号 2 代表的是左小臂多边形;序号 3 代表的是右小臂多边形;序号 4 代表的是右大臂多边形;序号 5 代表的是左大腿多边形;序号 6 代表的是左小腿多边形;序号 7 代表的是右小腿多边形。表 1 中列出了图 6(a) 中加黑的多边形(造型的小臂)与(b) 中多边形进行匹配计算过程中的度量函数值,坐标原点位于屏幕左下角。

表 1 匹配过程中部分多边形的质量、质心、度量值

序号	质量	质心	$f(i, j)$
1	7 358. 24	(1436. 0, 367. 1)	1 787. 39
2	1 465. 75	(1486. 2, 371. 4)	56. 57
3	1 435. 33	(1401. 5, 354. 7)	21. 35
4	1 124. 67	(1428. 6, 384. 2)	101. 94
5	4 537. 13	(1475. 9, 307. 3)	941. 34
6	985. 69	(1436. 1, 258. 8)	181. 99
7	4 497. 68	(1483. 6, 300. 8)	957. 74

/pixel

从实验结果中可以看到,度量函数能较好地反映出多边形之间的对应关系,内插结果也都能较好地符合人物运动的基本常理,说明本文的匹配方法能够较好地实现关键帧中角色的图形配对,并能对经过配对的图形对象内插出良好的动画效果。

该方法已应用于自主开发的如意通二维动画自动生成系统,并用于实际的动画制作。通过实践笔者发现,应用该方法对两幅原画中的多边形进行匹配内插,匹配成功率可达到 90% 以上,但对于躺下、转头等透视变化及多边形数量不一致情况下的造型多边形匹配效果不是很理想,对从有到无或从无到有的多边形仍需手工进行处理。实践证明,大多数情况下,该方法能够帮助年轻的动画制作人员快速确定造型多边形之间的对应关系,避免了复杂的人工对应过程,能够有效提高动画制作的效率。

4 结束语

本文针对目前计算机辅助动画制作中存在的 key 问题,深入研究了图形实体的数学表示方法和动画角色的变化规律,提出了基于角色质心的卡通序列匹配方法,充分利用角色中各个层次及多边形之间的相互位置信息来判别两帧图形之间的对应关系,并进行了实验。实验结果表明该方法具有良好的全局性,且计算简单、迅速,能有效解决自动化建立常规动画角色动作的原画图形之间相互对应关系的问题,进而提高计算机辅助动画制作的效率,具有很高的实用价值。

参考文献:

- [1] 齐朝晖,郑飞. Flash 动画原理与实践[M]. 北京:电子工业出版社,2008.
- [2] 叶风华,叶正华. 动画制作原理[M]. 长沙:湖南师范大学出版社,2008.
- [3] 吴刚,杨敬安,李道伦. 基于目标不变量的识别方法研究[J]. 计算机科学,2000,27(12):81-85.
- [4] JAGADISH H V. A retrieval technique for similar shapes[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York:ACM Press,1991:208-217.
- [5] 杨洪. 图论常用算法选编[M]. 北京:中国铁道出版社,1998.
- [6] 张宝印. 基于矢量的时序图形图像处理技术研究与实践[M]. 西安:西安地图出版社,2008.
- [7] 张宝印. 时序图形相邻两帧配准问题的研究[J]. 测绘学院学报,2001,18(4):21-24.