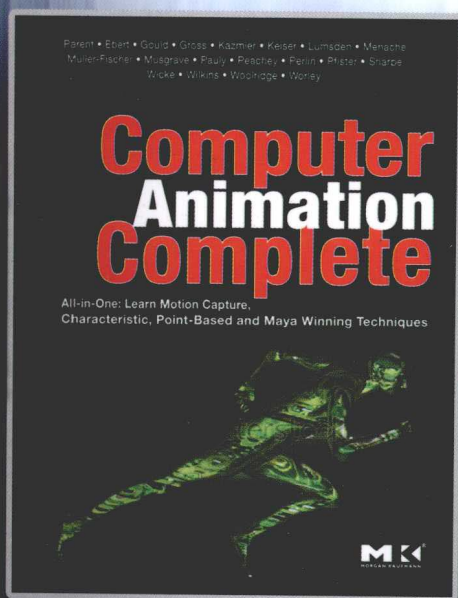


计算机动画设计指南

——运动捕捉、角色特征、点图及Maya Winning技术

Rick Parent 等著
王锐 王冠群 冷林霞 译

All-in-One: Learn Motion Capture, Characteristic, Point-Based, and Maya Winning Techniques



清华大学出版社

计算机动画设计指南

——运动捕捉、角色特征、点图及 Maya Winning 技术

Rick Parent 等著

王锐 王冠群 冷林霞 译

清华大学出版社

北 京



内 容 简 介

本书详细阐述了与计算机动画相关的高效解决方案及相应的数据结构和算法, 主要包括计算机动画概述、技术背景、运动捕捉入门、运动数据、面部特征动画、实体空间的动画、流体和气体的动画化、动画生物学、基于点的动画、Maya 内部原理解析、MEL 动画、MEL 的刚体动力学范例、MEL 的角色装配范例、建立你的角色以及 MEL 命令基础等内容。此外, 本书还提供了相应的算法、代码以及伪代码, 以帮助读者进一步理解相关方案的实现过程。

本书适合作为高等院校计算机及相关专业的教材和教学参考书, 也可作为相关开发人员的自学教材和参考手册。

Computer Animation Complete

Rick Parent, David S. Ebert, David A. D. Gould, Markus Gross, Chris Kazmier, Richard Keiser, Charles J. Lumsden, Alberto Menache, Matthias Müller-Fischer, F. Kenton Musgrave, Mark Pauly, Darwyn Peachey, Ken Perlin, Hanspeter Pfister, Jason Sharpe, Martin Wicke, Mark R. Wilkins, Nicholas Woolridge, Steven Worley

ISBN: 9780123750785

Copyright © 2010 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by Elsevier (Singapore) Pte Ltd. Press and Tsinghua University Press.

Copyright © 2013 by Elsevier (Singapore) Pte Ltd. and Tsinghua University Press. All rights reserved.

Published in China by Tsinghua University Press under special arrangement with Elsevier (Singapore) Pte Ltd.. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中国大陆地区 (不包括香港、澳门特别行政区以及台湾地区) 出版与发行。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

本书封面贴有 Elsevier 防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

计算机动画设计指南——运动捕捉、角色特征、点图及 Maya Winning 技术 / (美) 帕伦特 (Parent, R.) 等著; 王锐等译. —北京: 清华大学出版社, 2013.4

书名原文: Computer animation complete All-in-One: learn motion capture, characteristic, point-based, and Maya Winning techniques

ISBN 978-7-302-31239-0

I. ①计… II. ①帕… ②王… III. ①动画制作软件 IV. ①TP391.41

中国版本图书馆 CIP 数据核字 (2013) 第 002293 号

责任编辑: 赵洛育

封面设计: 刘超

版式设计: 文森时代

责任校对: 赵丽杰

责任印制: 何芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京世知印务有限公司

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 203mm×260mm 印 张: 27 字 数: 716 千字

版 次: 2013 年 4 月第 1 版 印 次: 2013 年 4 月第 1 次印刷

印 数: 1~4000

定 价: 89.00 元

产品编号: 043580-01

译 者 序

这是一本“大而全”的计算机动画方面的权威著作，其内容可谓无所不包。本书从动画的起源和历史变迁着手，系统而不失详尽地介绍了运动捕捉技术、角色和面部动画技术、刚体动画、流体动画、动画相关的生物学知识，以及基于顶点的动画。本书的另一大亮点就是充分运用 Maya 这一计算机动画制作软件，向读者展示了快速制作各种复杂的刚体与粒子动画、基于物理的动画，以及角色动画的技巧。其主要的制作方法并不是通过传统的用户界面调用，而是直接使用 MEL 脚本语言来执行动画的构建。有了这些更为清晰和有价值的参考内容，相信可以帮助相关从业人员快速而系统地学习 Maya 动画制作的深层次技巧，而非简单地照猫画虎。

作为一本理应偏重于具体技术实践的图书，本书重视理论的讲解、数学模型的建立，以及相关过程的推导；而另一部分章节则包含了大量与实践密切相关的源代码或伪代码内容。因此，无论是动画的具体实施人员，还是从事相关理论研究的人士，都应将这本书彻底读懂吃透。事实上，它所包含的大量的数学、生物学和物理学的理论知识，让笔者在翻译过程中也遇到了极大的挑战。

总体来说，本书是一本全面、系统、详尽地讲解计算机动画技术以及相关知名软件产品的著作，相关的作者也足有 18 人之多。他们多年来在第一线积累下来的宝贵经验，理应被更多的计算机动画行业从业者和爱好者所了解，以及传承。

本书的翻译工作由王锐主持，其中第 1、2、4、6、8、9 章由王锐负责，第 3、5 章由冷林霞负责，第 10~15 章由王冠群、迟亮负责，第 7 章由刘桢楠负责，全书由王锐统筹并定稿。此外，冷琴、陈节、刘剑也参与了本书的部分翻译工作，在此向他们表示感谢。

由于译者水平有限，而本书的内容又十分繁杂，涉及知识面很广，书中难免有疏漏和错误之处，敬请读者指出和谅解。

王 锐

目 录

第 1 章 计算机动画概述	1
1.1 感知.....	1
1.2 动画的变迁.....	3
1.2.1 早期设备.....	3
1.2.2 早期的传统动画.....	5
1.2.3 迪斯尼 (Disney).....	6
1.2.4 其他贡献者.....	7
1.2.5 其他动画媒介.....	8
1.2.6 动画的原理.....	8
1.2.7 电影制作的原理.....	9
1.3 动画的生产.....	12
1.4 计算机动画的生产.....	16
1.4.1 计算机动画的生产任务.....	17
1.4.2 数字编辑.....	19
1.4.3 数字视频.....	21
1.4.4 数字音频.....	22
1.5 计算机动画简史.....	23
1.5.1 早期的活跃者 (1980 年以前).....	23
1.5.2 中间的几年 (1980 年左右).....	25
1.5.3 动画时代 (20 世纪 80 年代中期至今).....	27
1.6 本章总结.....	30
参考文献.....	30
第 2 章 技术背景	33
2.1 空间与变换.....	33
2.1.1 显示流水线.....	34
2.1.2 齐次坐标和变换矩阵.....	38
2.1.3 变换的合成: 级联变换矩阵.....	38
2.1.4 基本变换.....	39
2.1.5 任意方位的重新表达.....	41
2.1.6 从矩阵中分解变换.....	44

2.1.7	显示流水线中的变换操作	45
2.1.8	误差估计	46
2.2	方位的表达	49
2.2.1	固定角度 (Fixed-Angle) 表示法	50
2.2.2	欧拉角度 (Euler Angle) 表示法	52
2.2.3	角度与轴 (Angle and Axis) 表示法	53
2.2.4	四元数 (Quaternion) 表示法	54
2.2.5	指数映射 (Exponential Map) 表示法	56
2.3	本章总结	56
	参考文献	56
第 3 章	运动捕捉入门	57
3.1	运动捕捉与表演动画	57
3.2	表演动画在娱乐业的发展历史	57
3.2.1	转描 (Rotoscope)	57
3.2.2	Brilliance	58
3.2.3	Pacific Data Images	60
3.2.4	deGraf/Wahrman	63
3.2.5	Kleiser-Walczak 公司	63
3.2.6	Homer and Associates	64
3.3	动作捕捉的类型	66
3.3.1	光学动作捕捉系统	66
3.3.2	电磁追踪器	69
3.3.3	机电设备	72
3.3.4	数字电枢	73
3.3.5	其他动作捕捉系统	75
3.4	动作捕捉在各行各业中的应用	77
3.4.1	医学界	77
3.4.2	体育界	78
3.4.3	娱乐界	78
3.4.4	法律界	79
第 4 章	运动数据	81
4.1	运动数据类型与格式	81
4.1.1	Acclaim 文件格式	82
4.1.2	.asf 文件	83
4.1.3	.amc 文件	85

4.1.4	.bva 文件格式	87
4.1.5	.bvh 文件格式	88
4.1.6	.trc 文件格式	91
4.1.7	.htr 文件格式	94
4.2	编写运动捕捉解析工具	102
第 5 章	面部特征动画	132
5.1	人脸	132
5.1.1	面部结构解析	132
5.1.2	面部表情编码系统 (FACS)	132
5.2	面部模型	134
5.2.1	构建连续的表面模型	135
5.2.2	纹理	139
5.3	制作面部动画	139
5.3.1	参数化模型	139
5.3.2	融合变形	140
5.3.3	肌肉模型	140
5.3.4	表情	143
5.3.5	总结	144
5.4	口型同步动画制作	144
5.4.1	发音器	144
5.4.2	音素	145
5.4.3	协同发音	146
5.4.4	韵律学	146
5.5	本章总结	146
	参考文献	146
第 6 章	实体空间的动画	149
6.1	动画路径	150
6.2	实体纹理 (solid texture) 的动画化	150
6.2.1	大理石生成	151
6.2.2	大理石运动	153
6.2.3	实体纹理透明 (solid-textured transparency) 的动画	154
6.3	气体体积的动画	155
6.4	三维晶格体	162
6.4.1	访问晶格体数据	163
6.4.2	功能流程形式的晶格体	163

6.4.3	功能流程函数	164
6.4.4	功能的集合	167
6.5	超纹理 (hypertexture) 的动画化	170
6.6	粒子系统: 另一种过程动画技术	171
6.7	本章总结	174
	参考文献	174
第 7 章	流体和气体的动画化	176
7.1	特殊的流体类型	176
7.1.1	水的模型	176
7.1.2	云的模型 (作者: David Ebert)	185
7.1.3	火的模型	192
7.2	计算流体力学	194
7.2.1	建模流体的一般方法	195
7.2.2	计算流体力学方程	196
7.3	本章总结	200
	参考文献	200
第 8 章	动画生物学	203
8.1	概述	203
8.2	动画和电影的感知	203
8.2.1	视觉的简述	203
8.2.2	运动和动画的视觉	205
8.3	动画师的工作流程	206
8.4	工作流程三段论	207
8.4.1	流程阶段 1: 预生产	208
8.4.2	流程阶段 2: 生产	213
8.4.3	流程阶段 3: 生产后处理	220
8.4.4	放在一起考虑	220
8.5	动画	221
8.6	Maya	221
8.6.1	过程式动画与关键帧动画	221
8.6.2	关键帧与内存	222
8.6.3	Animation 菜单集	222
8.6.4	设置关键帧	223
8.6.5	自动关键帧	224
8.6.6	图示动画	224

8.6.7	删除关键帧	226
8.6.8	时间单位	226
8.6.9	回放设置	226
8.7	教程 08.01: 关键帧动画	227
8.7.1	准备工作	227
8.7.2	设置关键帧	228
8.7.3	播放、拖动和停止动画	229
8.7.4	编辑动画曲线	229
8.7.5	Graph Editor	230
8.7.6	Graph Editor 的 Graph View	230
8.7.7	Graph Editor 工具栏	231
8.7.8	移动关键帧项	232
8.8	Hypergraph 和 Attribute Editor 中的动画节点	232
8.9	教程 08.02: 简单的过程式动画	233
8.9.1	动画表达式概述	233
8.9.2	创建动画表达式	233
8.9.3	动画表达式节点	235
8.10	本章总结	235
	参考文献	236
第 9 章	基于点的动画	237
9.1	导言	237
9.2	无网格的有限元	237
9.2.1	概述	237
9.2.2	连续弹力	238
9.2.3	无网格的离散化	241
9.2.4	移动最小二乘法	242
9.2.5	更新应变与应力	243
9.2.6	通过应变能量计算受力	244
9.2.7	弹性物体的动画化	245
9.2.8	塑料	247
9.2.9	被动的表面点集 (surfel) 对流	248
9.2.10	总结	248
9.3	碎裂材质的动画	248
9.3.1	概述	249
9.3.2	历史背景	250

9.3.3	不连续的建模	250
9.3.4	表面模型	252
9.3.5	裂缝的初始化和增殖	253
9.3.6	拓扑控制	253
9.3.7	体积采样	255
9.3.8	破碎的控制	257
9.3.9	模拟流水线	257
9.3.10	结论	258
9.4	流体模拟	258
9.4.1	概述	258
9.4.2	模拟方法	258
9.4.3	平滑粒子的流体动力学	260
9.4.4	表面表达	264
9.4.5	使用采样点进行表面跟踪	265
9.4.6	总结	268
	参考文献	268
第 10 章	Maya 内部原理解析	272
10.1	为什么要剖析 Maya 内部原理	272
10.2	从属图、属性、属性连接	272
10.3	范例 1: 使用 Hypergraph 浏览从属图	277
10.4	变换层次与父子关系	280
10.5	检查层次结构	281
10.6	变换节点 (Transform) 和形状节点 (Shape)	282
10.7	范例 2: 了解变换节点和形状节点、实例化, 以及历史节点	282
10.8	MEL 和 Maya 用户界面	285
10.9	Maya 场景的后台处理备忘录	285
第 11 章	MEL 动画	286
11.1	动画	286
11.1.1	时间	286
11.1.2	实时回放	288
11.1.3	动画曲线	289
11.1.4	骨骼系统	305
11.1.5	运动路径 (motion path)	317
第 12 章	MEL 的刚体动力学范例	320
12.1	范例 1: 粒子碰撞	320

12.1.1	创建场景	320
12.1.2	与粒子碰撞	323
12.1.3	对碰撞进行控制	325
12.1.4	geoConnector 中的其他控制手段	325
12.1.5	在 MEL 中完成场景	326
12.2	范例 2: 碰撞事件	327
12.2.1	概述	328
12.2.2	添加发射器和粒子	328
12.2.3	动力学关系	330
12.2.4	限制粒子数目, 添加重力	331
12.2.5	添加更多的碰撞	332
12.2.6	事件	332
12.2.7	篮子的表达式	334
12.2.8	编辑设置来修复问题	336
12.2.9	速度	337
12.3	范例 3: 刚体动力学的物体间碰撞	337
12.3.1	编写复制和定位的脚本	339
12.3.2	组装字符串变量时的常见错误	340
12.3.3	创建碰撞盒	341
12.3.4	反转碰撞盒的法线	342
12.3.5	主动和被动的刚体	343
12.3.6	将每个网格平面都变成被动碰撞对象	346
12.3.7	打开碰撞数据选项	347
12.3.8	改变网格碰撞时的颜色	348
12.4	范例 4: 刚体动力学与粒子	351
12.4.1	使用 MEL 创建瞄准窗口	352
12.4.2	向工具栏添加新的窗口控件	353
12.4.3	将平面转换为多边形, 并且分解为多片	354
12.4.4	向分片添加动力学和表达式	356
12.4.5	创建完整的 MEL 脚本	358
12.4.6	加载场景并运行脚本	360
12.4.7	向 panelBreakup 传递一个浮点数	361
第 13 章	MEL 的角色装配范例	362
13.1	范例 1: 角色控制	362
13.1.1	场景载入	363

13.1.2	场景概述	363
13.1.3	mrBlah 控制的概述	364
13.1.4	锁定属性	366
13.1.5	手臂控制	367
13.1.6	建立 mrBlah 的脊柱控制	368
13.1.7	选中足部时产生摇摆的效果	371
13.2	范例 2: 创建角色的用户界面	373
13.2.1	加载保存的 mrBlah 场景	374
13.2.2	设计用户控件	374
13.2.3	创建 mrBlahControls.mel	375
13.2.4	创建滑块来控制 spinCtrl 的属性	378
13.2.5	为窗口空间创建布局	379
13.2.6	测试窗口	380
13.2.7	向用户显示有限的信息	381
13.2.8	给窗口创建一个脚本节点	381
13.2.9	创建工具栏图标来打开窗口	382
13.2.10	mrBlahControls.mel 的完整代码	382
第 14 章	建立你的角色	386
14.1	设置角色的旋转数据	386
14.2	设置角色的平移数据	387
14.3	提示与技巧	396
14.3.1	改变旋转的顺序	396
14.3.2	旋转的分配	398
14.3.3	使用三次参数曲线	402
14.3.4	插值	403
14.3.5	关键帧简化	405
14.3.6	捕捉数据的重适应	406
第 15 章	MEL 命令基础	407
15.1	不写脚本也能使用 MEL 吗	407
15.2	命令行和命令反馈行	407
15.3	Shell	408
15.4	脚本编辑器	408
15.5	脚本编辑器 VS Shell	409
15.6	脚本编辑器的信息作为 MEL 代码	410
15.7	把一个 MEL 脚本制作成工具栏按钮	411

15.8	保存一个 MEL 脚本.....	411
15.9	状态消息区的危险诱惑.....	412
15.10	whatIs 命令.....	413
15.11	MEL 命令的基本结构.....	413
15.12	在网络上哪里可以找到关于 Maya 和 MEL 的信息.....	414
15.13	如何使用在网上找到的 MEL 脚本.....	414
15.14	备忘录.....	415

第 1 章 计算机动画概述

对于许多人来说，计算机动画基本上等价于那些大银幕上的故事，如《星球大战》、《玩具总动员》、《泰坦尼克号》等。但是并不是所有的，或者说事实上只有一部分计算机动画是在好莱坞这样的地方完成的。对于星期六早晨播放的那些卡通片而言（美国电视频道通常在星期六早晨播放预定的动画节目），它们很有可能完全是使用计算机来生成的。计算机游戏同样受益于各种最先进的计算机图形学技术，并且已经成为计算机动画技术研发的主要驱动力。实时的、性能驱动的计算机动画已经在 SIGGRAPH^①上提出，并应用在《芝麻街》（*Sesame Street*，一个风靡美国的儿童电视节目）上。台式计算机的动画制作已经可以维持在一个合理的开销水准上，而基于网络的计算机动画设计也有各种规程可循。使用数字模拟技术来完成飞行员、SWAT 反恐小组的训练项目，以及核反应堆的模拟也已经不是什么新闻了。这种种应用之间最显著的差别就在于它们的成本、画面要求质量，以及可用的交互行为的种类和数量。在这本书里，我们不会列举某一个应用领域的优势与不足，而是向读者介绍所有这些案例当中，计算机动画的算法实现和相关技术的使用。

这里所说的计算机动画，指的是各种基于计算机实现的运算，其目的是产生多幅图像以构成对运动的感知。本书的重点在于三维图形数据的处理算法和相关技术。总的来说，任何可以被改变的数据量都可以被动画化。某个对象的位置和方向显然都是动画的组成元素，而以下这些内容同样可以构成某种形式的动画：物体的形状、着色参数、纹理坐标、光源的参数，以及摄像机的参数等。

对于计算机动画的技术而言，通常有 3 种运动控制的实现方式。

- 艺术动画（artistic animation），设计者需要全力负责运动的制作过程。艺术动画的构建基础是插值技术（interpolation）。
- 数据驱动动画（data-driven animation），此时我们需要将各种即时录制的运动过程数字化并且映射为图形对象。
- 过程式动画（procedural animation），也就是建立一种计算模型以用于运动的控制。通常来说，这种动画的表现形式是对某种物理事件或行为的模拟，并且需要设置其初始条件。

在走进计算机动画的世界之前，我们很有必要先了解一下它的传统、历史，以及一些相关的概念。本章的其余部分将讨论运动的感知、动画技术的变革、动画的生产过程，以及计算机动画业界中一些值得留意的作品。总之，本章将着重介绍计算机动画的基础概念及相关知识。

1.1 感 知

一幅图像可以迅速地传达大量的信息，这是由于人类的视觉系统是一个精密的信息处理设备。依

^① SIGGRAPH 是计算机协会（Association for Computing Machinery, ACM）针对计算机图形学组建的一个特别兴趣团体（Special Interest Group）。计算机协会（ACM）是计算机行业科学家的一个主要的专业性组织。

此类推，运动的多幅图像就有可能在短时间内传递更多的信息。当然，为了生存的需要，人类的视觉系统同样在多变的世界中不断地进化，它同样可以察觉并解译那些运动的信息。

一个被普遍认同的事实是，如果一系列图像迅速地依次显示，那么观察者会将其辨认为一幅运动的图像，这种情况的发生需要适当的观察条件和播放速度。此时由于人眼和大脑的组成结构，我们有可能将一个静态图像的序列感知为一个连续的图像。对此，一种通常的看法是这种体验源于人体的视觉暂留，即眼睛可以在图像刺激被移除的一瞬间保留其视觉印记。但是值得争议的是，这样的印记（又被称做独立静物的正视觉后像（positive afterimages）），是否真的能填补连续变化的图像之间的感知空隙。对此，Peter Roget 于 1824 年提出了自己的理论，即光照的痕迹可以保留在视网膜之上^[1]。但是视觉的暂留并不等同于运动的感知。一个白色光源的旋转如果能够达到足够的速度，那么我们将感受到一个静止不动的白色环。虽然这种现象可以归因于视觉的暂留，但是其结果却是静止的。而电影院标牌上的一组霓虹灯连续开关，则可以产生一个循环运动的发光体的效果。在这里我们感知到了运动，但是却不包含任何视觉的暂留，因为我们并没有输出独立的图像。近年来，越来越多的人开始质疑（生理学上的）视觉暂留机制的产生及其结果，而运动的感知也开始被归因于一种（心理学上的）拟动现象（phi phenomenon），即外在的运动最终指向了一种名为 β -运动（beta motion）的行为^{[2][3][4]}。

无论其内部机理是怎样的，我们的结论依然不变，即对于影片和视频而言，如果一个图像序列可以以一种足够快的速度被依次显示的话，我们的眼睛将被愚弄并将它们解析为连续的影像信息。如果我们的器官无法建立起对这种连续影像的感知，那么我们会看到闪烁（flicker）的画面。这个时候，我们的眼睛和大脑事实上把动画识别为了快速的静止图像序列。依据外部条件的不同，如室内光照和观察的距离，我们能够把独立的图像序列感知为连续影像所需的播放速率也不相同。这个速率被称为临界闪烁频率（critical flicker frequency）^[5]。

我们已经知道，运动感知的最低限度就是建立对连续影像的感知，那么人眼的感知能力是否也有最高限度呢？答案是肯定的。人眼的感觉器官会频繁地对环境中的光照进行采样，而运动感知的极限，在一定程度上，也就取决于这些传感器官的反应时间，以及其他机械运动（例如眨眼和追踪观察）的限制。如果一个物体的运动对于观察者而言非常迅速，那么眼睛的感受器官将无法向大脑发出足够快的响应信号，也就无法轮廓分明地、独立地分辨物体的细节，从而产生模糊的运动效果^[6]。在一个静止图像的序列中，这种运动模糊（motion blur）是由于物体运动的速度和场景采样的时间间隔的共同作用而产生的。对于静止的摄像机来说，如果快门速度相对于物体本身的速度而言足够快的话，那么就算是快速运动的物体也不会产生模糊。在计算机图形学中，如果我们总是可以及时和精确地对场景进行采样的话，就不会出现运动模糊的情形。如果要产生运动模糊，我们需要以大于默认时间间隔的方式对场景进行采样，或者通过手动控制来模拟这种采样的效果^{[7][8]}。如果我们没有实现运动模糊的效果，那么快速运动的物体的图像序列将变得脱节，就好像在一些现场演出中使用频闪灯一样。这种效果通常称做频闪（strobing）。在手绘实现的动画里，我们通常需要对快速运动的物体在运动方向上进行拉伸，这样就可以在这个物体的连续各帧图像上产生交叠^[9]，从而减轻频闪的问题。

正如上面的讨论所提及的，我们事实上需要关注两种类型的速率。一种是播放或者刷新的速率，即浏览动画时每秒钟显示的图像数目；另一种是采样或者更新速率，即每秒钟出现的不同图像的数目。播放速率与动画的闪烁与否相关，而采样速率决定了运动的平滑程度。举例来说，NTSC 制式（national television standards committee）的电视信号要求显示图像的速率为大约 30fps^①，但是它是一种交错信

^① 更准确地说，美国广播电视系统的制式是由国家电视标准委员会（NTSC）颁布的，其指定的帧速率为 29.97fps^[48]。

号^①，因此在通常的观察条件下，整个场的播放实质上要达到 60fps 才能避免闪烁的问题^[10]。在有些电视节目中（例如一些周六早晨播放的卡通片），每秒钟可能只需要 6 幅不同的图像，而每幅图像需要重复显示 5 次。通常来说，与口型同步的动画是隔帧绘制的（每两帧绘制一次），因为如果我们每帧都对其进行绘制的话，其显示可能会变得过于频繁。在电影院播放电影的速率通常是 24fps（美国标准），但是为了减轻闪烁的问题，电影的每一帧实际上要被显示两次（double-shuttered），这样有效刷新速率将维持在 48fps。另一方面，由于 NTSC 电视信号是交错的，我们可以在 1 秒内每 60 帧采样 1 次场景，从而产生平滑的运动效果（但是事实上所有帧的播放速率是 30fps）^[5]。计算机的显示设备通常采用逐行扫描的形式（非交错式），因此其刷新速率一般大于 70fps^[10]。

1.2 动画的变迁

在人们最宽泛的理解中，动画^②意味着“赋予生命”，包括那些实地表演的木偶戏（例如我们在《芝麻街》节目中所看到的）、使用各种机电设备操控的运动偶人（动画人偶，animatronics）。人们一直在尝试各种赋予这些事物生命的方法。这一演变过程中集合了神话、欺骗、娱乐、科学以及医学的知识。许多有关动画的历史故事都采用了一种将生命拟人化的形式：从皮格马利翁（Pygmalion，希腊神话人物）到普罗米修斯（Prometheus，希腊神话人物），从瓦格纳的霍尔蒙克斯（Wagner's Homunculus，源自歌德的《浮士德》）到玛丽·雪莱的弗兰肯斯坦博士（Dr. Frankenstein）。有些人还试图创造可以模拟某些人类行为的机械设备，例如 Jacque Vaucanson 在 1730 年左右创造的机械吹笛手、鼓手和大便鸭（defecating duck），Wolfgang von Kempelen 在 1769 年设计的象棋手，以及 Pierre Jaquet-Droz 在 1774 年设计的自动书写机，更不用说如今日趋流行的运用机械电子技术开发的各种人型机器人。早期（18 世纪到 19 世纪）的机械装置设计主要来源于各种类型的科学辩论，而非真正取自人体的机械学原理（例如，Julien Offray de La Mettrie 在 1747 年写就的 *L'homme Machine* 一书，英译为 *Man a Machine*，一直广受争议）。而拟人化的机械装置的设计原动力，则是来自一群天才的魔术师、钟表匠、哲学家、科学家、艺术家、解剖学家、手套制造者以及外科医生（参见 Gaby Wood 的著作^[11]，这是一部有关机械化生活的有趣的调查资料）。但是在本书中，我们的关注点则是各种使用独立静止图像序列来创建运动图像效果的设备，因为这些设备与传统的手绘动画之间有着更为紧密的联系。

1.2.1 早期设备

19 世纪的人们对于视觉暂留，以及我们将一系列静止物解析为运动图像的能力进行了广泛的研究^[12]，这之后人们才发明了电影摄像机。对于这一现象的认识和随后的研究成果促使了一系列小玩具的发明^{[13][14]}。其中最简单的一种早期玩具可能就是留影盘了，它是一种在两边都绘制了图像的扁平圆盘，通过圆盘边缘的两根弦将它们各自反向连接起来（见图 1-1）。通过旋转弦线，我们可以快速地

^① 交错显示技术将每一帧分为两个场，每个场包含了奇数或者偶数行扫描线。奇数场或偶数场会交替显示在相应的设备之上^[5]。

^② 我们也可以在一些文献中找到动画的更严格的定义，即使用一个静止物的序列来构建可视的运动图像。诸如动画人偶或者皮影戏一类的技术形式，并不能归入这一严格定义的动画范畴。

反复翻转圆盘。如果翻转的速度足够快，两侧的图像就好像被叠加在一起一样。一个经典的留影盘会在一侧绘制一只鸟的图案，在另一侧绘制一个鸟笼；那么旋转的圆盘看起来就仿佛把鸟放入了鸟笼之中。另一种与之原理相同的技术是翻书效果，首先在一摞纸上面依次绘制一些独立的图案，当这些纸张快速地翻动时，我们就可以看到一个运动的场景了。

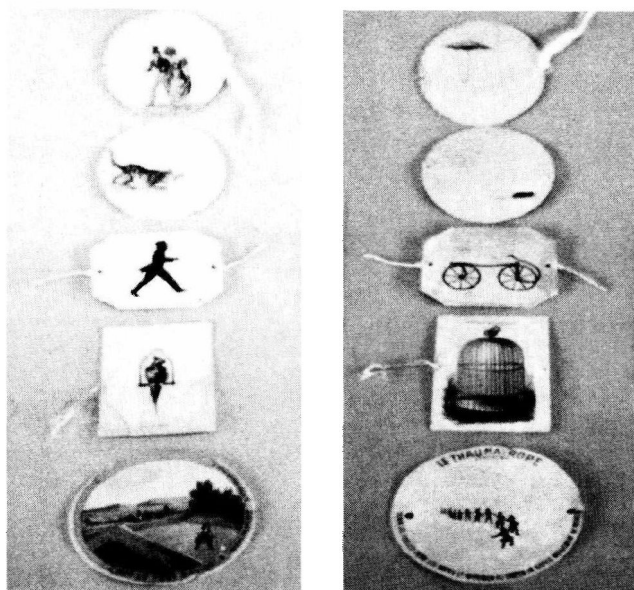


图 1-1 留影盘 (Thaumatrope)

在那些久负盛名的早期动画设备中，西洋镜（也称为 wheel of life）是一个不得不提的名字。在其对称轴上放置了一个短粗的圆筒；在圆筒的内壁上环绕了一系列图样，每一幅图与邻近的图都稍有不同；在圆筒的每两个相邻图样之间都有一个垂直的长切割槽。这样当圆筒本身绕轴旋转时，我们的眼睛就通过这些槽来观察对面壁面上的图样（见图 1-2）。当圆筒绕着对称轴旋转时，这些槽就会依次掠过我们的眼睛，在眼中形成一个图像的序列，从而形成了运动的效果。

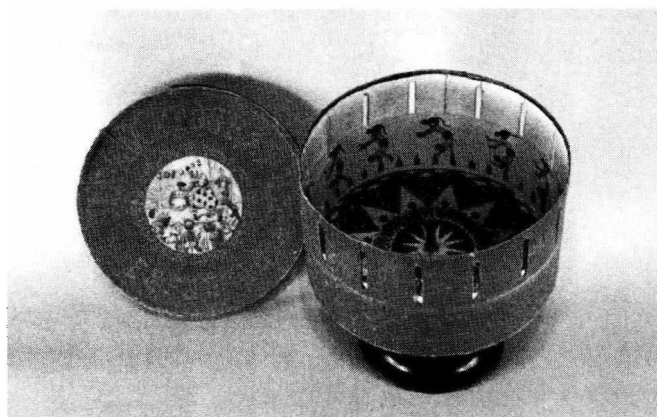


图 1-2 西洋镜 (Zoetrope)