/Theory/In/Practice

数据可视化之美

Beautiful Visualization

通过专家的眼光洞察数据

O'REILLY®
机械工业出版社
China Machine Press
HZ BOOKS
华章科技

Julie Steele & Noah Iliinsky 编
祝洪凯 李妹芳 译

Data Analysis

数据可视化之美

可视化是数据描述的图形表示，旨在一目了然地揭示数据中的复杂信息。可视化的典型示例是纽约地铁图和人脑图。成功的可视化的美丽之处既在于其艺术设计，也在于其通过对细节的优雅展示，能够有效地产生对数据的洞察和新的理解。

在本书中，20多位可视化专家，包括艺术家、设计师、评论家、科学家、分析师、统计学家等，展示了他们如何在各自的学科领域内开展项目。他们共同展示了可视化所能实现的功能以及如何使用它来改变世界。

阅读本书，您将：

- 通过简单的可视化实践探索讲故事的重要性。
- 了解颜色如何传达我们尚未充分意识到而大脑已经识别出的信息。
- 发现我们购买的书籍和我们的交际圈如何揭示内心的自我。
- 通过对民航交通的可视化探索识别航空旅行的混乱的一种方法。
- 揭秘研究人员如何调查未知问题，包括从最初的草图到发表的论文。

本书的作者包括：

Nick Bilton
Michael E. Driscoll
Danyel Fisher
Jessica Hagy
Todd Holloway
Noah Iliinsky
Eddie Jabbour
Valdean Klump
Aaron Koblin
Robert Kosara
Valdis Krebs
JoAnn Kuchera-Morin等
Adam Perer
Anders Persson
Maximilian Schich
Matthias Shapiro
Julie Steele
Moritz Stefaner
Jer Thorp
Fernanda Viégas
Martin Wattenberg
Michael Young

本书所有作者的版税将捐赠给“人道建筑组织”（Architecture for Humanity）。

客服热线：(010) 88378991，88361066
购书热线：(010) 68326294，88379649，68995259
投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com
华章网站：*http://www.hzbook.com*
网上购书：*www.china-pub.com*

O'Reilly Media, Inc.授权机械工业出版社出版

此简体中文版仅限于在中华人民共和国境内（但不允许在中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China (excluding Hong Kong, Macao and Taiwan)

O'REILLY®
oreilly.com.cn

ISBN 978-7-111-33796-6

9 787111 337966

定价：89.00元

数据可视化之美

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权机械工业出版社出版

机械工业出版社

图书在版编目（CIP）数据

数据可视化之美/（美）斯蒂尔（Steele, J.）等编；祝洪凯，李妹芳译. －北京：机械工业出版社，2011.6

（O'Reilly精品图书系列）

书名原文：Beautiful Visualization

ISBN 978-7-111-33796-6

I. 数… II. ①斯… ②祝… ③李… III. 可视化软件 IV. TP31

中国版本图书馆CIP数据核字（2011）第045478号

北京市版权局著作权合同登记

图字：01-2010-4822号

书　　名/　数据可视化之美

书　　号/　ISBN 978-7-111-33796-6

责任编辑/　秦健

封面设计/　Karen Montegomery，张健

出版发行/　机械工业出版社

地　　址/　北京市西城区百万庄大街22号（邮政编码100037）

印　　刷/　北京京师印务有限公司

开　　本/　178毫米×233毫米　16开本　28.5印张（含5.25印张彩插）

版　　次/　2011年6月第1版　2011年6月第1次印刷

定　　价/　89.00元（册）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991; 88361066

购书热线：(010) 68326294; 88379649; 68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

目录

前言

Toby Segaran和Jeff Hammerbacher的《数据之美》探索了从数据收集到数据存储、组织和分析等与数据相关的方方面面。很自然地，编著本书的想法正是基于此书。在编著《数据之美》一书的过程中，我们就很清晰地认识到可视化——把信息作为艺术品展现给人们——是一个值得我们另行审视且非常有深度和广度的话题。成功的可视化，如果做得漂亮，虽表面简单却富含深意，可以让观察者一眼就能洞察事实并产生新的理解。我们希望帮助新手在可视化这个不断发展的领域中了解专家们为实现这一目标所采用的方法和决策过程。

饶有趣味的是，在收集潜在的撰稿人列表时，我们发现“美丽”一词可以有非常多的诠释方式。Andy Oram和Greg Wilson的《Beautiful Code》（该书中文版《代码之美》已由机械工业出版社于2009年1月出版，ISBN：978-7-111-25133-0）一书奠定了该“之美”系列，它把“美丽”定义为解决某些问题的一种简单优雅的方式。但是，可视化——作为信息和艺术的融合——自然地结合了问题求解和艺术这两个方面，允许我们同时通过理性和传统的感官方式来感受美丽。

我们希望你会和我们一样喜欢本书所展现的丰富多彩的背景知识、项目和方法。虽然各章涉及的背景、项目和方法不同，但它们确实为那些善于思考和观察的人们提供了一些主题。整本书围绕着寻找数据的思想展开讨论，包括讲故事、色彩使用、数据中的粒度级别和用户探索。抓住这些线索，看看它们可以给你的工作带来什么启发。

本书的版税将捐赠给“人道建筑组织”（Architecture for Humanity，*http://www.architectureforhumanity.org*）。该组织致力于通过为最需要的地方提供设计、建造和开发服务，以使得世界变得更加美好。我们希望你会思考自己的设计过程如何改变世界。

本书的组织方式

以下是本书的概览：

第1章“论美”。Noah Iliinsky 给出了在可视化情境下，美所蕴涵的意义，为什么值得追求，以及如何追求。

第2章“曾经的堆叠时间序列：讲述故事在信息可视化中的重要性”。Matthias Shapiro阐述了讲故事对于可视化的重要性，引导读者一起创建一个自己可以实现的、简单的可视化项目。

第3章“Wordle”。Jonathan Feinberg介绍了他所发明的流行的可视化文本的内部工作方式，探讨了其在这个过程中从技术和审美角度上所做的选择。

第4章“色彩：数据可视化的‘灰姑娘’”。Michael Driscoll阐述了如何有效地使用颜色来表达我们尚未意识到而大脑却可以识别的其他维度的数据。

第5章“信息映射：重新设计纽约地铁图”。Eddie Jabbour以探索简陋的地铁图作为基本的可视化工具来理解复杂的系统。

第6章“飞行模式：深入探索”。Aaron Koblin和Valdean Klump对美国和加拿大的民航交通进行可视化，揭示了一种“疯狂”的空中旅行方法。

第7章“你的选择揭示你是谁：社会模式的挖掘和可视化”。Valdis Krebs深入探索行为数据，证明了通过我们购买的书和交往的人能够更深入地揭示自我。

第8章“美国参议院社交图（1991～2009）的可视化”。Andrew Odewahn通过“定量”的证据来评价美国参议院关于投票联盟的“定性”的故事。

第9章“鸟瞰图：搜索和发现”。Todd Holloway通过已经应用于YELLOWPAGES.COM网站和Netflix颁奖中的近似图形化技术来探索搜索和发现的动态特征。

第10章“从社交网络可视化的混杂之中寻找美丽的感悟”。Adam Perer通过结合可视化和统计的交互技术，以帮助读者深入探索混杂的社交网络可视化。

第11章“美丽的历史：对维基百科可视化”。Martin Wattenberg和Fernanda Viégas从最初的设计草图到发表的科学论文，通过可视化带领读者走向未知领域的探索。

第12章“把表转换成树：把并行集发展成意义深远的项目”。Robert Kosara重点描述了数据的可视化展现和基础的数据结构或数据库设计之间的关系。

第13章“‘X byY’的设计：奥地利电子艺术节档案的信息美学探索”。Moritz Stefaner描述了努力寻找的一种信息展现方式，这种方式不仅有用且信息充实，而且是感性的、令人回味的。

第14章“矩阵探秘”。Maximilian Schich揭秘了资料数据库中由于管理员的本地操作和数据源的异构性产生的一些非直观的结构特征。

第15章“1994年：基于《纽约时报》上的文章搜索API的数据探索”。Jer Thorp引领读者使用API对《纽约时报》资料库的数据进行探索和可视化。

第16章“《纽约时报》的一天”。Michael Young和Nick Bilton描述了《纽约时报》研发组是如何使用Python和Map/Reduce来处理美国以及全世界的Web站点和手机网站的流量数据。

第17章“深入揭秘复杂系统”。Lance Putnam、Graham Wakefield、Haru Ji、Basak Alper、Dennis Adderton和JoAnn Kuchera-Morin教授描述了AlloSphere项目通过尖端高科技可视化和可听化技术实现的非凡的科学探索。

第18章“解剖可视化：真正的黄金标准”。Anders Persson描述了使用新的成像技术来收集和分析人类和动物尸体数据。

第19章“动画可视化：机遇和缺点”。Danyel Fisher尝试提出设计动画可视化的一种框架。

第20章“带索引的可视化”。Jessica Hagy提出了对可视化这头“大象”的各个方面的洞察，因此可以对全局有更透彻的理解。

本书使用的体例

本书遵循以下字体体例：

斜体（*Italic*）

表示新的术语、URL、Email地址、文件名和文件扩展名。

等宽字体（`Constant width`）

用于程序清单以及段落中的程序单元如变量或函数名称、数据库、数据类型、环境变量、声明和关键字。

等宽粗体字（**Constant width bold**）

显示命令或者其他应该由用户逐字输入的文本。

等宽斜体字（*Constant width italic*）

表示必须根据用户提供的值或者由上下文决定的值进行替代的文本。

使用本书的样例代码

本书是为了帮助你完成工作。通常来说，你可以在你的程序和文档中使用本书的代码。除非你使用了本书的大量代码，否则你无需联系我们以获取许可。例如，写一个程序用到本书的几段代码不需要获得许可；销售和分发O'Reilly丛书的例子代码光盘需要获得许可；引用本书的样例代码来解决一个问题不需要获得许可；结合本书的大量代码到你的产品文档中需要获得许可。

我们不要求你（引用本书时）给出出处，但是如果你这么做，我们对此表示感谢。出处通常包含标题、作者、出版社和 ISBN。例如："*Beautiful Visualization*, edited by Julie Steele 和 Noah Iliinsky. Copyright 2010 O'Reilly Media, Inc., 978-1-449-37986-5."

如果你觉得你对本书样例代码的使用超出了这里给出的许可范围，请和我们联系：*permissions@oreilly.com*。

联系方式

请把对本书的评论和问题发给出版社：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly的每一本书都有专属网站，你可以在那找到关于本书的相关信息，包括勘误列表、示例代码以及其他的信息。本书的网站地址是：

http://www.oreilly.com/catalog/9781449379865/

对于本书的评论和技术性的问题，请发送电子邮件到：

bookquestions@oreilly.com

关于本书的更多信息、会议、资料中心和网站，请访问以下网站：

http://www.oreilly.com

http://www.oreilly.com.cn

致谢

首先，我们要感谢各位作者投入这么多的时间和精力来分享他们的智慧。他们共同的愿景和经历给我们留下了深刻的印象，并且激发我们在工作中的创作灵感。

Julie：感谢家人Barbara、 Pete和 Matt，感谢他们一直以来的支持，感谢他们激发了我对世界的好奇心。感谢Martin，感谢他的陪伴和永远跳动着的思维，他给我带来了很多灵感。

Noah：感谢在过去这些年来帮助我探索的每一位人，尤其是我的老师、同事和家人，他们总是给我提出很好的问题，帮助我更好地思考。

第1章

论美

Noah Iliinsky

本章探讨了在可视化情境下，“美”所蕴涵的意义，它为什么值得追求，以及如何追求。我们将首先探讨美的组成部分，审视一些正例和反例，然后再重点说明实现可视化之美的关键步骤[注1]。

何为美

当我们认为一个可视效果很美时，其中有什么涵义呢？它是“美”这个字在传统意义上的一种审美判断吗？可能是。但是，当我们在这种场景下讨论可视化时，可以认为“美”包含4个关键因素，而审美判断仅仅是其中的一个。一个称得上“美”的可视效果，它不但必须美观，而且也必须新颖、充实和高效。

新颖

一个可视效果要想真正做到“美”，它必然不仅仅是作为信息渠道，还必须具备某些新颖性：一种崭新的视角观察数据，或者一种风格可以激发读者的激情从而达到新的理解高度。众所周知的可视化展现方式（如散点图）可能易于理解且有效，但是在绝大多数

注1：在本章中，可视化（visualization）和可视效果（visual）两个词是等价的，表示所有结构化的信息表现方式，包括图形、图表、示意图、地图、故事情节图以及不是很正式的结构化插图。

情况下，它们无法使我们感觉充满惊奇和乐趣。通常情况下，让人赏心悦目的设计并非是为了新颖而设计，而是为了更加有效而设计；新颖性只是为了有效地展示对世界的一些新的洞察所衍生的一个副产品。

充实

对于任何可视化而言，不论美丽与否，其成功的关键是提供了获取信息的途径，人们可以借以增长知识。不能达到这个目的的可视化是失败的。信息传递能力是判断整体成功与否的最重要的因素，因此它是可视化设计的主要驱动力。

在创造一个有效的视觉效果中，需要考虑几十个因素，如场景、感知和认知等。虽然其中很多因素都超出了本书的讨论范围，我们将重点考虑两个特殊因素：想要表达的信息和应用场景。除了关注数据本身，同时还关注这两个因素，将会在使数据可视化更有效、成功和美丽的道路上走得更远；我们将在稍后部分对这两个因素进行更为深入地探讨。

高效

美丽的可视化具备一个清晰的目标、传递一种信息或者提供一个特别的角度来表达信息。访问这些信息必须尽可能地直截了当，而不需要牺牲任何必要的相关复杂性。

可视化不允许包含太多和主题无关的内容或信息。在页面上放太多的信息可能会（也可能不会）给读者传递更多的信息。然而，展现的信息越多，往往意味着读者需要花费更长的时间来查找需要的那部分信息。不相关的数据如同噪音，如果无益，则很可能有害。

美感

图形化构建——包括坐标轴、布局、形状、色彩、线条和排版——是实现可视化之美的“必要”因素而不是“充分”因素。合理地利用这些因素来引导用户、传播信息、揭示关系、突出结论以及提高视觉魅力是必要的。

图形方面的设计必须主要服务于表现信息这个目标。在图形处理中，任何无助于表现信息的微小方面都可能成为表现信息的潜在障碍：这些方面可能会降低效率，妨碍可视化的成功。在图形设计部，通常是展现的数据越少，表示的信息越丰富。同样道理，展现的数据如果无益，则很可能有害。

通常，新颖的视觉处理方式是创新性的解决方案。然而，如果一个独特的设计是为了与众不同，而且其新颖性与使数据更易于访问并没有必然联系，那么几乎可以确定该可视化结果是更难以使用的。在最坏情况下，新颖的设计只不过是自负的产物，或者是希望

创造一些视觉上令人印象深刻的欲望的产物，完全没有考虑到目标受众、使用方式或功能。这种设计对任何人都没有使用价值。

学习经典

大量平庸的信息可视化完全基于标准格式。基本的可视化展现方式，如条形图、折线图、散点图、饼图、组织流程图，以及其他一些格式是可以很容易通过各种软件生成的。这些格式无处不在，并且提供了便捷、常规的开始使用方式。可视化创造者和消费者都可以很好地理解这些格式的理论意义和使用方式。基于这些原因，这些方法是常见可视化问题的良好且强大的解决方案。然而，使用这些方法的最佳方式局限于一些特定的数据类型，而且其标准性和普遍性意味着它们基本无法达到新颖性。

“赢得”声誉和财富的美丽的可视化则不同于上述传统的可视化。它们不必源于创造者和消费者所熟悉的惯例（虽然它们可能会充分利用一些熟悉的视觉因素和处理方法），而且它们通常与期望的数据格式有一定偏差。这些图像通常不会受限于传统的可视化协议：它们会根据非传统的数据类型进行灵活地变动，这足以使人惊喜和兴奋。

最重要的是，美丽的可视化可以反映出所描述数据的品质，显式地揭示源数据中内在和隐式的属性和关系。读者了解了这些属性和关系之后，可以因此而获取新的知识、洞察力和乐趣。为了说明这一点，我们一起来欣赏两个闻名于世的美丽的可视化，观察它们是如何充分利用其源数据结构的。

元素周期表

我们探讨的第一个例子是门捷列夫（Mendeleev）的元素周期表，它是可视化的一个杰作，一张表中囊括了至少4种、通常9种或者更多类型的数据编码（见图1-1）。元素的属性呈周期性变化，将所有元素排列成一张表格，以表格的行和列表示属性的变动周期。这是关键点，因此我再重述一遍：元素周期表的天才之处在于通过元素的编排组织揭示了元素之间的相互关系以及周期性变化的物理属性。表的结构直接取决于其所表示的数据。在这张表上，元素的属性一目了然，因此，借助这张表就可以快速地认识和理解给定元素的属性特征。除此之外，根据元素周期表上的空白，能够精确地预测尚未发现的元素。

毋庸置疑，元素周期表信息丰富，其高效性也是可以证明的，而且为在此之前一直没有良好的可视化解决方案的问题提供了一种全新的视角。基于以上种种原因，元素周期表被视为复杂数据可视化早期的一个杰作。

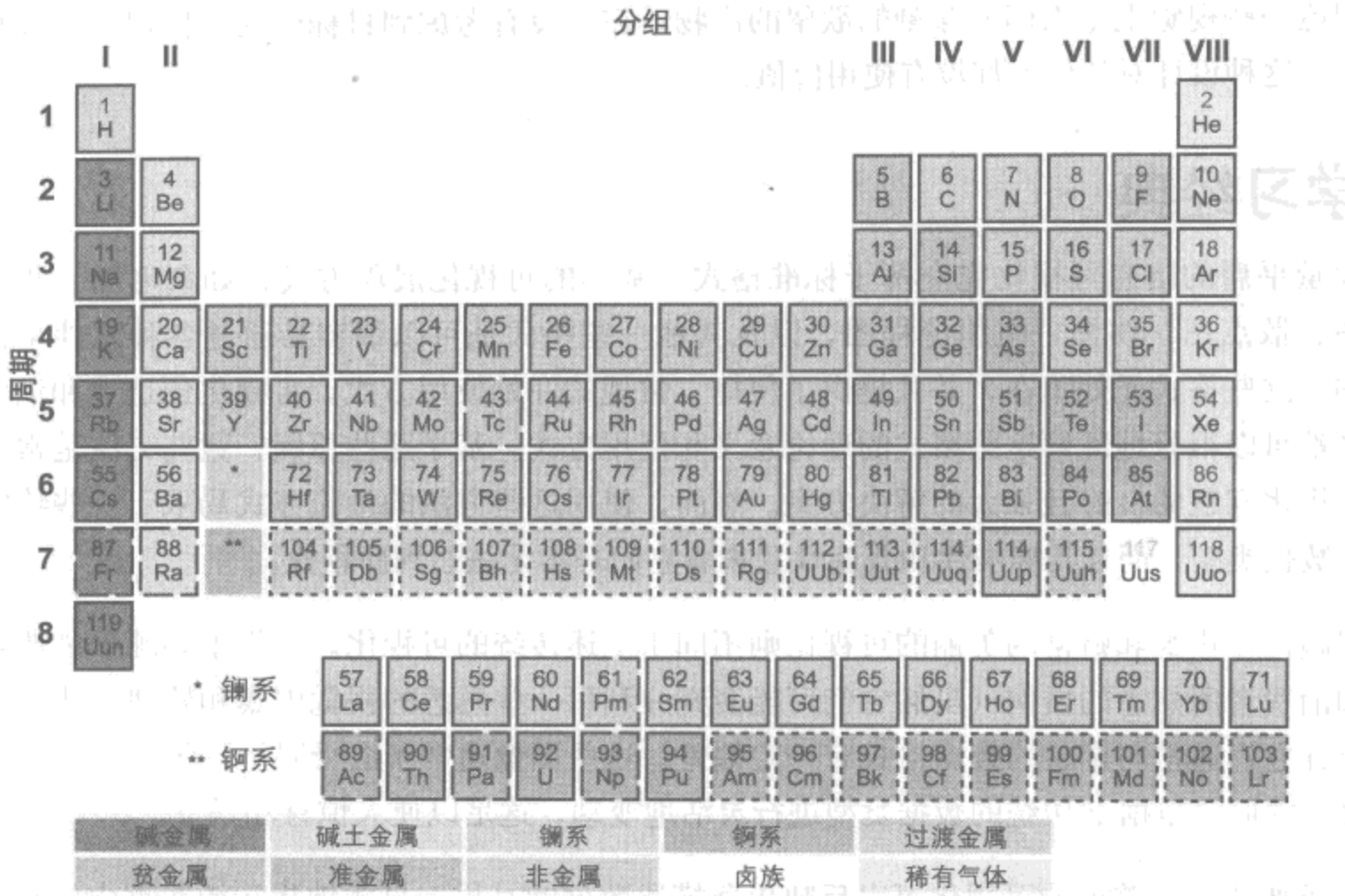


图1-1：门捷列夫的元素周期表（见彩图1）

值得指出的是，为达到理想的效能和取得非凡的成绩，元素周期表所需的图形化处理绝对是最少的；实际上，最早的版本是纯文本的，可以通过打印机打印出来。强大的图形设计处理并非是可视化美丽的必要条件。

伦敦地铁图

第二个美丽的可视化经典是Harry Beck的伦敦地铁图（见图1-2）。该地铁图（Tube map）受到可视化的惯例和标准的影响，但是却没有被这些制图者的陈规所束缚。Beck的背景是画电路草图：他习惯于绘制45°和90°的电路布局图，并把这种习惯带到了地铁图的绘制过程中。

这种风格把地图从准确的代表地理位置特征的束缚中解放出来了，引领了更简单地反映地铁旅行实际情况的抽象视觉风格：在地铁系统中，人们最关心的是自己和系统中其他位置的逻辑关系。精确地显示了地理特征的其他地图可以帮助你找出在地面上可以做什么，但是当你在地铁里时，你能够到达的地面位置仅仅是那些地铁站。

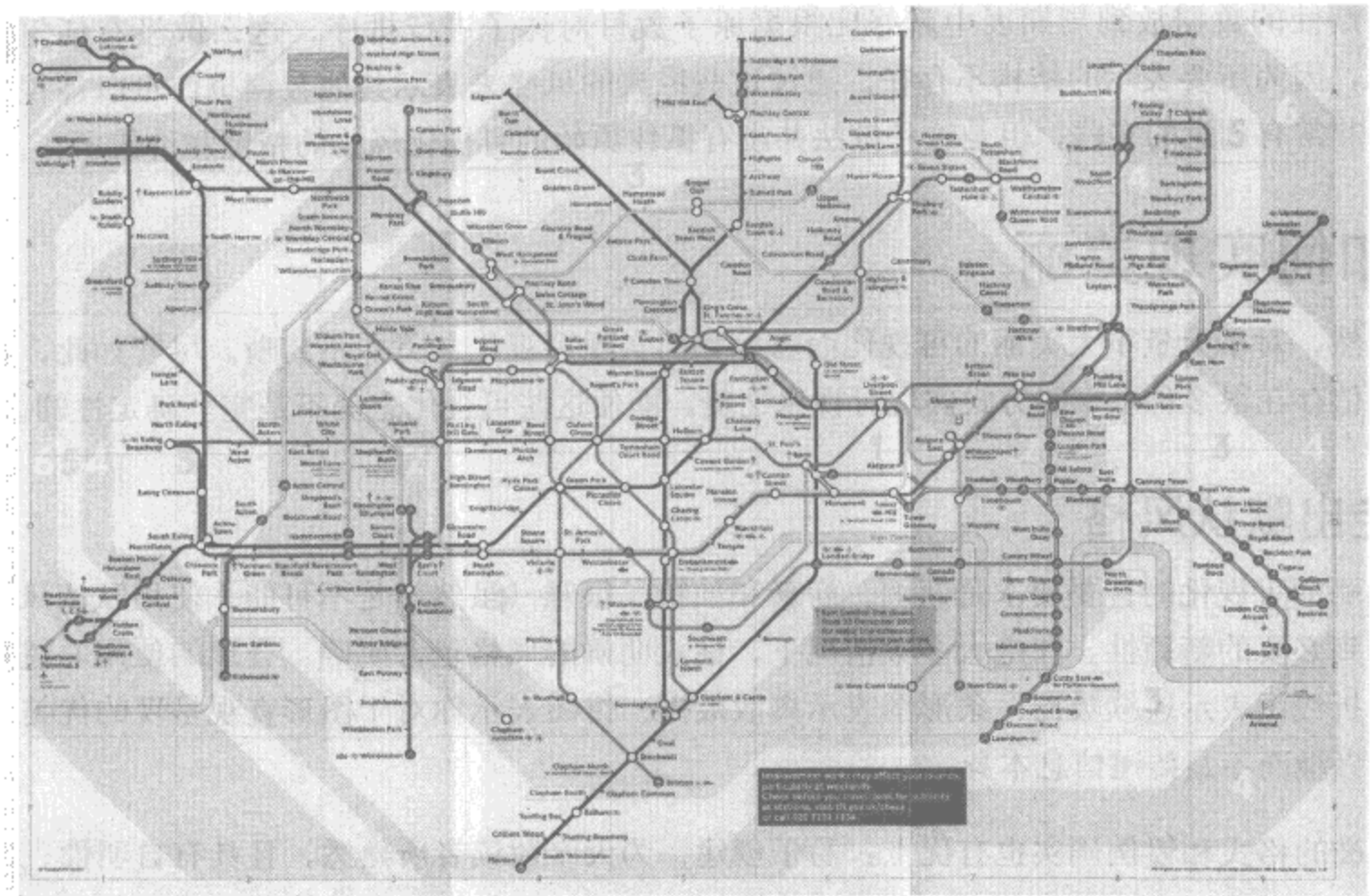


图1-2：伦敦地铁图：2007年伦敦地铁图。伦敦交通博物馆收藏（已授权使用，见彩图2）

伦敦地铁图突出显示了最相关的信息，剔除了很多不相关的信息，使得相关的数据可以更容易被访问到。它独特鲜明的图形风格已经成为标志。它是一个公认的杰作，一个无可争议的美丽的可视化。

其他地铁图和周期表仅仅是弱仿制品

由于元素周期表和伦敦地铁图的成功，其他数据的表现方式往往会模仿它们的风格。几乎你能想象的所有东西都有周期表：食品、饮料、动物、爱好，更为可悲的是甚至包含了可视化方法[注2]。所有这些都没有抓住可视化的精髓。类似地，地铁图的风格也用于表示不同风格的电影[注3]、技术公司之间的关系[注4]、公司并购时间表[注5]，以及其他城市的地铁系统。

这些例子中，关于伦敦地铁图的风格的唯一合理的使用方式是采用该风格来表示其他城市的地铁图（很多城市，如东京、莫斯科等，在这方面都做得非常好）。对该风格的其他使用方式都没有理解其产品的特别之处：产品和源数据的真正关系和表现形式。把非

注2：见*http://www.visual-literacy.org/periodic_table/periodic_table.html*。

注3：见*http://blog.vodkaster.com/2009/06/25/the-top-250-best-movies-of-all-time-map/*。

注4：见*http://informationarchitects.jp/wtm4/*。

注5：见*http://www.meettheboss.com/google-acquisitions-and-investments.html*。

周期性的数据放到周期表中就好比根据原子数目对袜子进行排序，这么做没有任何意义，因为所要表示的结构不存在。通过这些经典的风格来表示其他数据也许可以被视为是非常有创意的实践，但是这种做法并没有抓住原始的可视化风格的精髓和价值所在。

如何实现美丽

显然，对于大量不太美丽的可视化而言，如何实现可视化之美并不清晰。尽管如此，我相信存在很多种可靠的方式实现可视化之美，即便这些可视化之美不是完全确定性的。

走出默认风格

美丽的可视化的首要要求是新颖、崭新和独特。很难（虽然不是不可能）使用默认风格达到必要的新颖性。在绝大多数情况下，格式明确的风格包含明确、合理的使用习惯：用折线图表示连续数据、条形图表示离散数据、饼图表示你对于图形直观呈现的优美更感兴趣而不是传递信息本身。

标准的格式和惯例确实也有优点：易于创建、为大多数读者所熟悉，且具有自明性。绝大多数时候，应该遵从并充分利用这些惯例。然而，通常情况下，使用实用型的格式难以实现新颖性；默认方式很有用，但是存在其局限性。抛弃默认格式并采用更好、更强大的解决方案必须是为了传递信息而非多样化。

在不适宜的情况下使用默认的表现方式，可能也存在陷阱。我遇到的一个例子是一个制造公司的Web站点，在该站点中，它以零售商为第一列并按照其名字的字母序排列，以零售商们所在的城市和州为第二列。这个系统对于设计它的人来说当然很有意义，但是该设计并没有考虑到该列表会如何使用。如果我已经知道了我所在区域的零售商，按照字母序排列将很有用。

不幸的是，我知道自己的位置，但是不知道零售商的名称。在这种情况下，根据最易获取的信息——位置排序的列表比默认的以零售商名字的字母序排列的列表将会更有意义。

使可视化信息更充实

正如我之前所提到的，成功的可视化必须是信息充实且实用的。为了确保可视化的实用性，有两个方面需要考虑：预期的信息和使用场合。考察并整合这些方面的认识通常是一个迭代的过程，随着设计的演进，会涉及在这些因素间的来回变化。此外还应该考虑惯例，以支持设计的可达性（谨慎使用某些惯例，可以帮助用户对数据做出一些假定，比如关于美国政治上使用红色和蓝色来表现视觉效果）。

预期的信息

首先要考虑的问题是你想要传达什么知识，想要回答什么问题，或者想要讲述什么故事。这个阶段完全是抽象地规划可视化功能，在这个阶段开始考虑特定的格式或者实现细节还有些为时过早。这是一个关键步骤，而且很值得投入时间。

一旦确定了可视化要传递的信息或者要达到的目标，接下来需要思考的是如何使用可视化。读者和他们的需求、行话和偏好必须纳入考虑之中。在这个阶段，明确用户需要完成的任务或者明确他们需要从可视化中获取的知识将很有帮助。用户的专业知识刚开始可能不能很好地被理解，但是这是在设计过程中需要牢记的关键因素。

如果你最终不能以读者和他们的需求的方式准确地阐明你的目标，你就没有目标，也无法来衡量你到底成功与否。上文所举的两个案例的目标可能可以如下陈述："我们的目标是，提供一张伦敦地铁系统的视图，使得乘客可以轻松地选定乘车路线"；或者是"我的目标是，以一种可以很清晰地显示元素的物理特征并且可以据此对它们的行为作出预测的方式来显示元素。"

一旦对自己的信息以及受众的需求和目标有了清晰的理解，就可以开始考虑你的数据。对可视化目标的理解将允许你有效地选择需要包含哪些方面的数据，判断哪些方面的数据是没用的、甚至更糟的是会分散你的注意力。

使用场景。意识到以下两种设计目的在可视化上的区别也是很重要的：一是旨在揭示设计师所知道的；二是为了帮助未知事物的研究（虽然设计师可能提前猜想到其结果）。前者是演示工具，后者是探索工具。这两种设计方式都可能采取标准的或者非传统的方式，而且都可以从过程和处理中受益。然而，明确区分清楚到底属于哪种可视化设计类型是非常重要的，因为这一点会影响后续的所有设计选择。

旨在揭示已知事物的可视化是无处不在的。只要一方向另一方传达信息的方式不仅仅是文本，就存在这种可视化。我们遇到的绝大多数的图形和图表是为了传达特殊的见解、消息或者潜在底层数据中的清晰知识：团队如何分工、绩效如何划分、公司如何组织、给定的输入如何影响最终结果，以及不同产品如何比较等。数据可能还会揭示其他的知识或者见解，但是如果它们对于当前的目标不重要，该设计就不需要考虑展示这些消息或者趋势的方式。因此，定义良好的目标有助于设计这些可视化的过程。

旨在促进探索的可视化通常存在于更专业的、面向研究的科学、商业和其他领域之中。在这些情况下，其目标通常是为了验证假设，回答具体问题或者发现任何趋势、行为或者值得注意的关系。如果对于数据可能揭示的规律不清楚，设计这些可视化会变得更具有挑战性。在答案不确定的情况下，设计一些不同的可视化可能是有用的。

周期表是这些目的的有趣的混合体，因为它是用于对已知和未知的信息进行可视化。该周期表的结构是通过那时已知的元素的属性定义的，因此在该情况下，它对现有已知的知识提供了参考，正如今天所使用的。然而，该结构导致了周期表中产生一些空白，这些空白后来用于预测未发现的元素的存在和行为。在后一种模式下，表格是研究和发现的工具。

使可视化变得高效

在确保可视化富含信息量之后，下一步是要确保它是高效的。当为了高效而设计时，值得考虑的最重要的方面是：可视化的每一部分内容都将使用户花费更长的时间来找到在该可视化中的任何元素。页面上的噪音数据和视觉噪音越少，读者找到他们需要寻找的东西就越简单。如果你所明确的目标无法证明某些内容存在的必要性，试着去掉这些内容。

视觉上突出重要的因素

当你已经确定了必要的内容，考虑其中的某些部分（某种特定的关系或者数据点）是否特别相关或者有用。这些内容在视觉上可以通过几种方式突出显示。它可以更大、更粗、更亮、更详细，或者通过圆圈、箭头或标签来标识。 另一方面，不太相关的内容可以通过较柔和的色彩弱化显示，线条更细或者缺乏细节信息。例如，在伦敦地铁图中的各个区域，在视觉上没有被突出显示：虽然它们确实存在，但是其相关程度显然弱于那些地铁线路和站点。

注意，强调相关性的策略通常适用于数据展现，而不是数据研究：设计师通过改变突出的重点，有意地改变传递的信息。此外，突出未知数据的不同方面或者子集是发现可能淹没在噪音数据中的关系的有效方式。

使用轴线表达含义并展示自由信息

在减少可视化噪音数据和文本数量的同时仍能保留足够的信息的一个非常棒的方法是定义轴线，然后使用这些轴线来指导可视化中其他模块的位置。定义轴线的优雅之处在于可以通过轴线对可视化中的每个节点赋值，而且不需要涉及额外的标注操作。举个例子，周期表是由定义清晰的行（周期）和列（分组）组成的。可以通过查看一个元素占有的周期和所属的分组来了解关于该元素的很多信息。因此，信息不需要显式地展现在元素的表格单元中。轴线还可以用于定位数据集中的某个部分或者某个成员，比如查找特定周期的元素、查找南方的一些州或者查找已知位于伦敦的西北地区的一个地铁站。

定义良好的轴线对于定性数据和定量数据都有效。在定性环境中，轴线可以定义（无序的或者杂乱的）领域或分组。而定量的轴线可以提供信息，支持相关值的查找。

相关部分的切分

减少可视化混乱，使得信息更易于理解的最后一种方式是，把大数据集划分成多个相似或者相关的子集并分别可视化。如果可以获取的信息可以独立使用，这种方式效果不错；而如果需要和其他数据集中一起使用，则收益会很小。其中的风险在于当把所有数据集中一起显示时，可能会发现看起来不相关的数据集中存在的相关的、尚未察觉的关联关系，这种关系在这种显示方式下才会变得很明显。

慎重使用惯例

当已经充分考察预期的信息、应用场景和数据对你的特定情景的影响时，在可视化中应用一些标准的展现方式和惯例是值得的。有意识地、恰当地运用惯例将会加速学习，便于读者记忆。在使用了惯例的情况下，只要和前述的几项因素没有冲突，采用惯例会非常强大且实用。本文所举的两个例子使用了默认的、传统的表现方式来表示元素符号、地铁线色彩和指南针方向。这些因素绝大部分看起来太自然了，不值得一提或注意，而实际情况也正是如此。它们很容易被理解，而且可以精确地表达消息，用户可以轻易迅速地理解以这种方式表达的信息，而且几乎不需要用户或者设计师做出任何额外的努力。这正是默认方式和惯例发挥作用的理想方式。

充分利用美感

一旦满足了充实和高效的需求，终于可以考虑可视化设计的美感了。审美元素可以是纯粹装饰性的，或者是增加可视化成果被接纳的机会的又一个因素。在某些情况下，可视化处理方式可以对信息进行冗余编码，因此一个给定的值或分类可能使用位置和颜色来描述，可能使用文字标签和形状的大小来描述，或者使用其他的属性对来描述。与单一编码相比，冗余编码可以帮助读者更快、更容易地区分感知和了解更多信息。

可以选择一些其他方式以帮助理解：熟悉的色彩板、图标、布局，以及和参考文档或者期望的使用场景相关的全局风格。熟悉的外观和感觉可以使读者更轻松或者舒适地接受展现处理的信息。（但是，要注意避免仅仅为了风格本身而使用熟悉的风格，避免像那些拙劣地模仿周期表和地铁图的设计师们陷入同样的陷阱。）

有时，设计师可能想要做出某些选择以干扰一些或者所有的可视化使用方式。这可能是通过弱化显示其他信息，以此为代价来突出某些特定的消息，为了以艺术性的表达方式、为了使可视化适应于某个有限的空间，或者只是为了使可视化更让人赏心悦目或者感兴趣。只要它们是在对全局效用的影响已经了解之后的有意为之，这些都是合理的选择。

付诸实践

我们一起来看另一个成功的、数据驱动的可视化例子，该例子把这些可视化原则付诸应用：《纽约时报》的2008年总统竞选地图[注6]。图1-3是美国的标准地图，每个州都以颜色编码来表示在该州竞选获胜的候选人（红色表示共和党候选人在该州竞选获胜，蓝色表示民主党候选人获胜）。该图看起来像是一个利用了默认框架的非常合理的可视化：一张国家地理图。然而，实际情况是这样的：准确的地理描述，最好情况下这些信息充其量也只是无关紧要的，而最坏情况下它们可能会产生很多误导。

图1-3：地理上准确的美国竞选投票结果图（见彩图3）

新泽西州（呈花生形状的州，在宾夕法尼亚州的东部和纽约州的南部，面积太小以致无法标注出来）的面积是略多于8700平方英里。Idaho、Montana、Wyoming、North Dakota 和 South Dakota这5个州的所有区域的面积总共超过47.6万平方英里，大约是新泽西州的面积的55倍，如图1-4所示。如果我们对于每个州的准确的地理、形状、大小和位置感兴趣，这将真的是一个很不错的地图。然而，在总统竞选这样的背景下，我们关心的是基于每个州的选票计数的影响。实际上，以上5个州的选票加起来总共只有16张，仅仅比新泽西州的15张选票多出一张而已。因此，地理上准确的地图实际上对于反映选举方面的影响是非常不准确的。

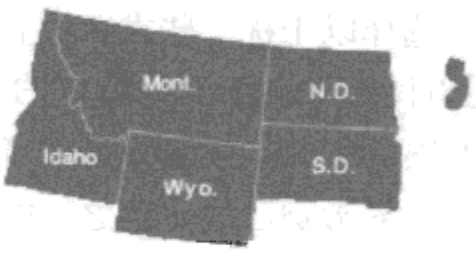


图1-4：5个州和新泽西州的相对面积大小（见彩图4）

注6：　数据来源：*http://elections.nytimes.com/2008/president/whos-ahead/key-states/map.html*。

一个州的面积和它对选举产生的影响力没有太大关系；在这种情况下，需要一种完全不同的可视化来准确地表示相关的数据，满足可视化需求。为此，《纽约时报》还生成了另一个地图视图（见图1-5），在该地图中，每个州是由相当于选票数的很多方块组成。和州的大小相比，这种选举上相应的视图已经失去了地理准确性，而考虑到州的大小，则几乎失去了所有的地理准确性。

然而，美国各个州的相对位置基本上还保留着，它允许读者找到他们感兴趣的特定的州并探测区域趋势。这里牺牲地理位置的好处是当显示每个党派赢得的选票和每个州的相对影响时，该可视化是非常准确的。举个例子，当我们查看新的地图，把新泽西州和前面提到的5个州的大小做比较，可以准确地描述15到16个竞选联盟，如图1-6所示。

图1-5：按相应比例加权的美国选票的结果图（见彩图5）

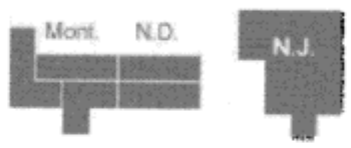


图1-6：5个州和新泽西州的相对选票影响

你可能已经注意到这里做出了另一个权衡：因为读者无法看清每个方块的边界，他们无法简单地在我们比较的每个领域都计数15到16个方块。此外，因为要尽可能地维持每个州的形状，图1-6所示的红色和蓝色聚集的分区形状区别显著，使得难以一眼比较它们的相对区域面积。因此，这是充分利用惯例（在这个例子中是各个州的形状）达到必要的平衡以及高效地、直白地表现数据的一个很好的例子。

该可视化的成功之处在于设计师愿意摆脱标准的、默认的地图，从而创建一个主要基于相关的源数据的可视化表示。其结果是一个高度定制的图像，该图像对于预期目标更精

确和有用，即使不能很好地适应于典型的地图任务，如导航。（在那种情况下，它类似于地铁图，为非常特殊格式的信息查找进行了优化，其代价是牺牲了通用的地理上的准确性。）

结束语

虽然本章只是简要介绍了设计成功的可视化的一些策略和考虑，但是它为成功的可视化奠定了坚实的基础。实现可视化之美的核心在于专注于使可视化有用、相关和高效，并且使用默认方式和有意的艺术解决方案。这些建议将帮助我们确保最终产品是新颖、充实和美丽的。

第2章

曾经的堆叠时间序列

—— 讲述故事在信息可视化中的重要性

Matthias Shapiro

信息可视化这门艺术在某种程度上似一头“怪兽”。很少有学科需要其从业人员具备如此多的技能。最佳可视化的创建者不仅需要具备一些天赋，而且还要能够快速地在不同技能之间切换。此外，在完成可视化的最后阶段，创建者可能会发现前期舍弃的某些信息对于充分理解作品是至关重要的，也可能发现前期的某个计算结果是不精确的。

Ben Fry在他的优秀著作《Visualizing Data》（O'Reilly出版社）中指出创建信息可视化包括以下7个阶段：获取、解析、过滤、挖掘、展现、提炼和交互。每个阶段都需要具备特定水平的技术或艺术才能，而信息可视化需要兼具多项才能。在数据获取和解析阶段，信息可视化艺术家可能已经开始思考应该如何和它交互。而在对展现信息进行提炼的过程中，他也可能会回想起，在过滤阶段的某个处理步骤过滤掉的某些数据实际上是相关的。最佳可视化往往是由知识面宽广、多才多艺的个人独立构想和完成，或者是通过一个能够紧密协作的小团队合力完成。在这种小型、灵活的环境下，各种才能可以相互影响促进，进而创造出令人震撼的图像或交互产品，它所描述概念的方式比起一串数字让人感觉更贴切自然。

创建好的信息可视化需要具备很多才能，虽然这已经被人们广为认可，但是仍然存在一项技能在更正式的场合下往往会被人们忽略——可能因为几乎每一个可视化创造者都潜意识中做到了这一点，也可能因为它是整个可视化过程如此自然而然的一个部分以至于看起来似乎不值一提。这种技能就是讲述故事的能力。

故事拥有非凡的魔力，可以让我们集中注意力，帮助我们理解为什么所展现的数据对我们生活的某些方面是重要的或相关的。只有在特定的场景下，数据才是有意义的，而将数据作为故事的一部分是让数据产生持久效应的最佳方式。最有效的信息可视化会成为读者（或者用户）心中的故事或叙事的中心情节。

不是每一个信息可视化都需要讲述一个故事。有些可视化看上去就很美，其本身就是优雅的艺术作品。然而，绝大部分可视化都有一个目标，需要把数据置于某种故事情节中以有意义的方式进行展示。

问题 + 可视化数据 + 场景 = 故事

绝大多数可视化故事会以某类问题作为开场，引导读者进入某个主题或者场景中，在该主题或场景中，数据所蕴含的意义最为丰富。这种引导方式可以是显式的，也可以是隐式的，但是其场景必须清晰明确。作为开场白的问题包含了该故事的前提和引言，引领读者到达数据能够控制整个故事线索的关键点上。

故事的多个关键部分会作为一些组成环节嵌入到可视化的特定场景中。我们经常发现可视化场景是作为信息图片或者可视化的介绍文本的一部分。可视化场景提供了解答下述问题的信息：

- 我们正在看的是什么数据？
- 这份数据存在于什么时间段内？
- 哪些显著的事件或者变化影响了这些数据？

请看图2-1所示的可视化。假设用户没有相应的背景知识，当他看到该图时，我们确定他会理解这份数据是按照时间轴映射的，而该时间轴与某次选举有关。除此之外，几乎没有任何有价值的场景信息可以引导用户去弄清该可视化的含义。

如果更进一步，假设用户对该可视化作品上展现的一些较为有名的名字比较熟悉，我们就可以假定他将了解到该可视化作品展示的是2008年美国总统选举前两年的总统候选人的一些衡量指标。

只有当用户点击了右上角的问号标记，才会显示完整的场景说明，那时该用户才会知道这个可视化作品映射的是每位总统候选人某一周在《纽约时报》上被提及的次数。一旦了解了这个信息，用户就可以明白该可视化粗略地反映了由《纽约时报》撰稿人决定的新闻对这些总统候选人的关注度。

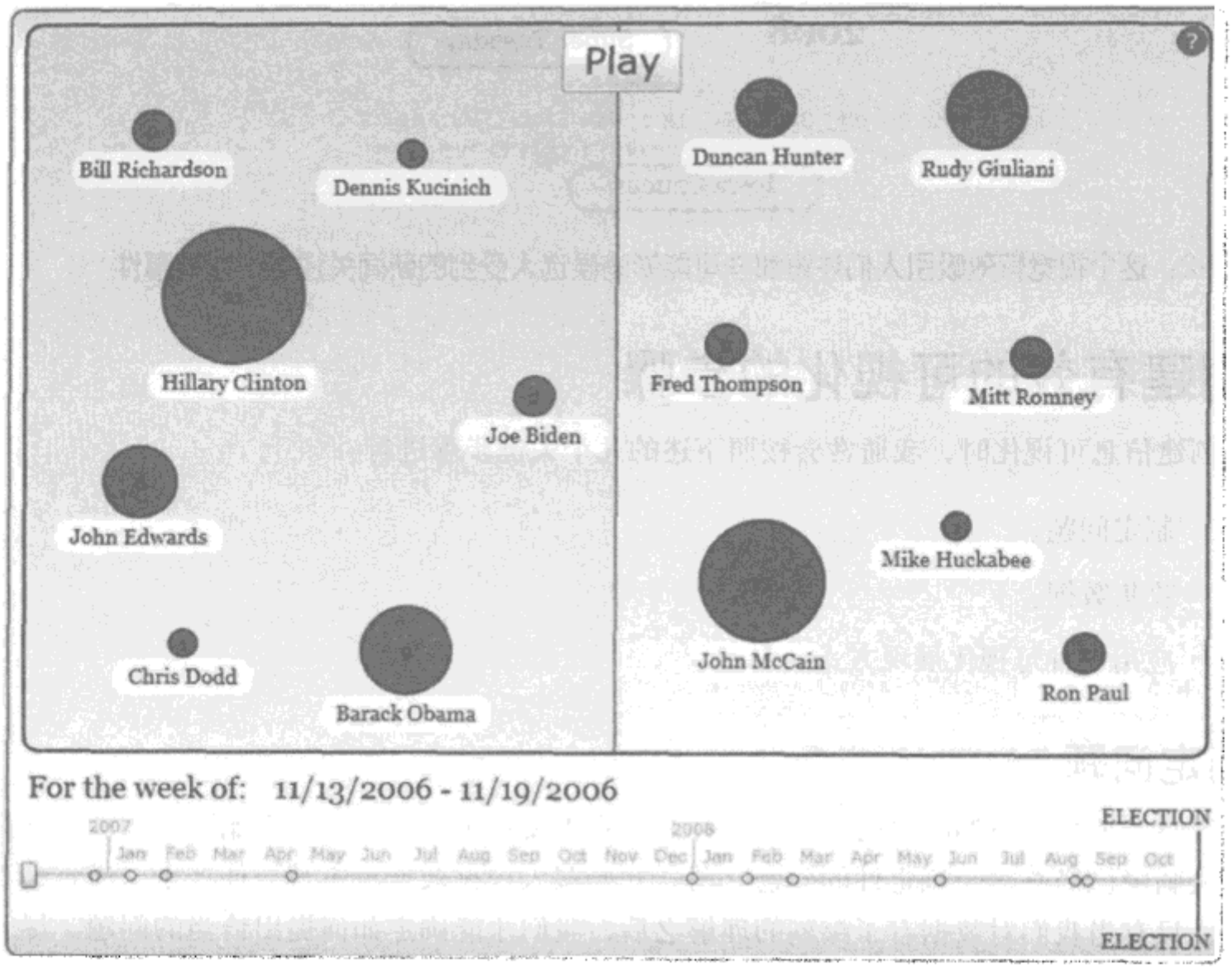


图2-1：设计工具Silverlight[注1]生成的可视化（见彩图6）

回到我们之前列出的那些问题，我们现在已经知道正在看什么数据以及其时间范围。该可视化是交互式的：如果用户点击最上方的Play（播放）按钮，它会沿着时间轴顺次弹出一些点，显示可能以某种方式对数据产生了影响的重要事件（见图2-2）。

除了这些线索，用户还可以把自己所知的总统竞选知识作为该数据的额外的场景信息。他可能回想起民主党内竞选时发生在希拉里·克林顿（Hillary Clinton）与巴拉克·奥巴马(Barack Obama)间的激烈角逐，这一点在现实中的反映就是从2008年4月到5月，他们俩都保持了很高的新闻关注度，而约翰·麦凯恩(John McCain)因为早在3月初即已经确保了在共和党内竞选的胜利地位，因而在那段时间的新闻关注度上落后于他们俩。

当提出一个问题“在2008年总统竞选过程中，《纽约时报》提及各个候选人的频度有多高？”之后，就开始引发一个故事。该可视化为这个故事提供了吸引人心的可视化部分，帮助用户在一分钟内重温这一历时两年的总统竞选大戏。

注1：参见*http://tr.im/I2Gb*。

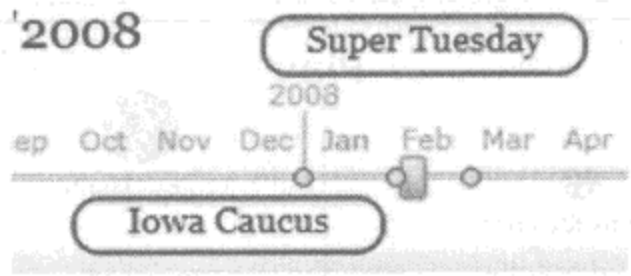


图2-2：这个视觉特效吸引人们注意那些可能影响候选人受到的新闻关注度的重要事件

创建有效的可视化的步骤

在创建信息可视化时，我通常会按照下述的几个关键步骤进行：

1. 制定问题。
2. 收集数据。
3. 应用一种可视化展现方式。

制定问题

提出驱动所要讲述的故事的问题，这并非一定需要在可视化之旅的开始阶段完成。在你的大脑中已经有一个确定性的问题之前，开始深入挖掘数据也不是一件坏事。通常情况下，只有当我们对数据有了深刻的理解之后，我们才能确定如何提出恰当的问题。尽管如此，在收集和过滤必要的数据时，提出一个问题（或者至少大脑中思索一个或者几个问题）会大有裨益。

随着收集到更多的数据，你可能考虑从某个主题切入，专注于数据搜索和问题提炼。举个例子，假设我们想表达这样一个观点：执行美国人口普查是一项庞大的任务。对于启动数据搜索而言，这是个不错的主题，因为其涵盖面足够宽广，所以存在很多数据能够提供场景，支持这个观点。我们可以找到相关的数据，并创建基于下述几点的一个可视化：

- 收集到的调查问卷的数量。
- 使用过的铅笔的数量。
- 人口普查工作人员的行程英里数。

我最喜欢的与美国人口普查相关的数据是美国联邦雇员数。统计数据显示了某个人口普查年的3月到7月间，联邦雇员数从20万飙升到30万。而当人口普查结束后，雇员数又会回落。

我们最终所选用的具体问题对最终的可视化展现有很大影响。举个例子，我们可能会问：“一次人口普查所需的全部信息需要多少纸张来记录？”，然后展示调查一座小城

市所需的一摞纸张；或者我们可能这样问：“对这个国家的所有人点一次名，需要花费多少人力？”，然后用一些图像来展示在人口普查期间联邦雇员数的上升。这些问题都和美国人口普查范畴下最原始的话题相关，但是由于选用了不同的数据集，生成的可视化作品也完全不同。

当为创建信息可视化而提出问题时，我们应该尽可能地关注以数据为中心的问题。那些以“在哪里”(where)、“什么时间”(when)、“有多少”(how much)或者“有多频繁”(how often)开头的问题通常都是不错的开始：它们使我们专注于在特定的参数集合内查找数据，因此更有可能找到适用于可视化的数据。

对于以“为什么”(why)开头的问题，需要格外小心。它意味着你开始从对数据的较为正式的描述转入数据分析。

收集数据

准确地找到所需的数据是一个非常困难的任务。通常，最好从已经可用的数据着手并尽量找到一种方式来描绘它，而不是尝试自己去收集数据。也就是说，最好从一个数据集出发（正如之前所提到的），从数据中找到一些模式之后再构建问题。如果你是为了一个既定的目标创建一个数据可视化，而不是出于兴趣或者纯粹的好奇心，那么很有可能你已经有了一个可用的数据集。尽管如此，仍然存在一些数据集，它们可能可以在工作的某些方面激发你的灵感或者提供某些信息。

有很多不错的地方提供了可以访问的数据。其中一个最大、最丰富的资源库是Data.gov网站（*http://www.data.gov*）。这个站点上存放了庞大的数据集合，它涵盖了大量领域，既包括鸟类的迁徙，也包括专利目录，还包括国债收益统计和联邦预算数据。其他优秀的数据源还包括：

- 美国人口普查局（*http://www.census.gov*）的网站上提供了种类广泛的人口统计和地理信息数据。
- 美国劳动统计局（*http://www.bls.gov*）提供了美国就业方面的广泛数据（点击“Databases and Tables”标签，然后向下滚动页面到历史新闻发布表单（Historical News Release Tables）处，可以找到最简单的数据访问入口）。
- 《纽约时报》的API（*http://developer.nytimes.com*）提供了对海量数据集易于访问的API接口，包括国会投票、畅销书列表、文章检索、影评、纽约市的房地产开盘和销售信息等。

一旦获取到了原始数据，就需要考虑数据的解析、组织、分组或者修改，以便可以从中识别出模式或者抽取出想要描绘的特定信息。这个过程通常就是众所周知的“数据再加

工”（data munging）过程，而且通常是即时地“玩弄”数据直到感兴趣的模式出现。如果感觉这个过程听起来有些含糊或者不够具体，不要担心，在下一小节中我们将以实践指南的方式完整地介绍一个数据再加工的例子。

应用一种可视化展现方式

既然我们获取到了数据，接下来需要做的就是确定应该如何描述它。这意味着需要决定采用何种可视化展现方式来描述数据才能帮助读者更好地理解。

一种可视化展现方式就是某种可视化维度，不同的数据以不同的维度展示。举个例子，一个*XY*坐标图就是一种简单的可视化展现方式，它把*x*,*y*数据点映射到一个二维平面中。当对足够多的数据点进行映射后，即使原始数据本身没有可以立即识别的模式，可能还是会产生显而易见的可视化模式。

让我们一起查看一些最常用的可视化展现方式。

尺寸

尺寸可能是最常用的可视化展现方式，而且是理所当然的。当辨别两个对象时，我们可以通过尺寸来快速地区分它们。此外，使用尺寸可以加快理解两组不熟悉的数字之间的区别。听说或知道美沙酮（一种镇静剂——译者注）是英国最致命的毒品是一回事，而看到如图2-3所示的因吸食美沙酮而致死的人数与吸食其他毒品而致死人数的信息则完全是另一回事儿。

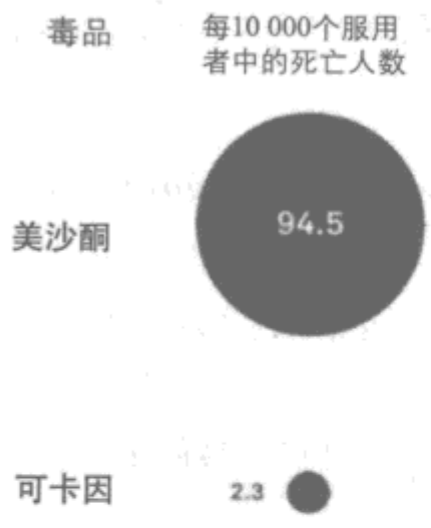


图2-3：来源于David McCandless对“世界上最致命的毒品”的信息可视化

虽然尺寸是一种非常实用且直观的展现方式，但它也经常被滥用。很多结构不良的图形只是起到了误导和混淆视听的作用，这往往是因为其作者虽然想要对一些数据进行可视化，但是却仅仅只知道一种可以展示它们的可视化方式。

色彩

色彩是展现大数据集的一种优秀方式。我们可以通过色彩识别出很多层次和色调，可以以很高的分辨率来查看区别。这一点使得色彩成为展现宏观趋势的必然选择，这种用法我们经常会在气象图中看到。由于这个原因，色彩通常被用于标识大数据集中存在的模式和异常。

图2-4是与股票相关的历时3个月的一组数据缩放图。

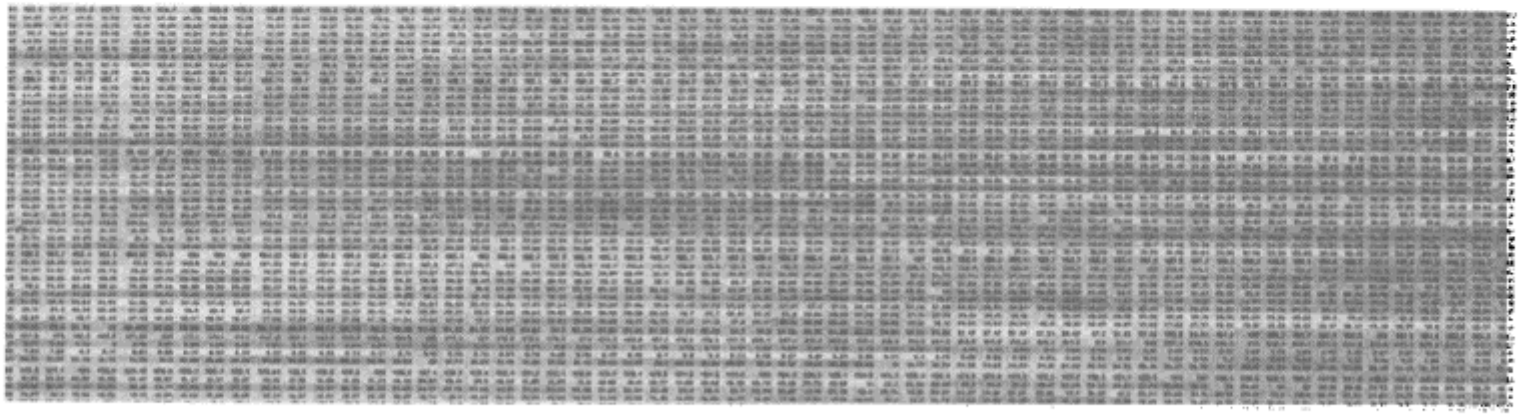


图2-4：Motley Fool CAPS[译注1]网站上在几个月内关注度最高的30只股票，使用红绿色阶对其进行了可视化（见彩图7）

虽然该可视化因为类型太小以致无法阅读，但我们却可以很容易识别出正增长或者负增长的行。我们可以很轻松地对数据中的趋势做出全面的评估。

对于规模较小的数据集或者相互之间区分度不大的数据，色彩的作用就不明显。如果数据中没有鲜明的色阶变化，即使是训练有素的人，也难以识别出其中重要的区别。

例如，假设我们有个范围1～100的数据集，以及一个色彩板，其颜色变换从红色（表示1）到黄色（50）到绿色（100）。在这样的色彩板中，对于图2-5中所示的只有10个百分点之差的两个数据[译注2]，正如你所观察到的，其区分度很小，而且可能对于很多读者都难以分辨。

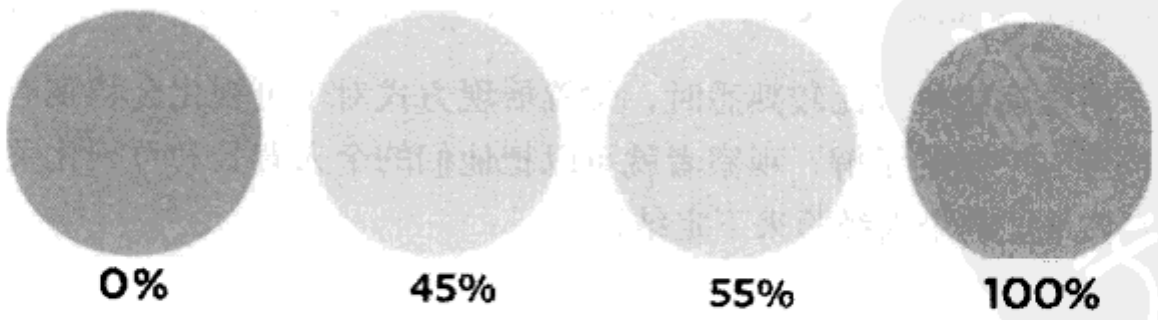


图2-5：在色彩可视化中，色彩图像在45%和55%范围之间的区别的展现（见彩图8）

译注1：Motley Fool CAPS是一个理财咨询网站，其主页是*http://caps.fool.com/*。

译注2：指的是图2-5中位于中间的45%和55%的两个数据。

如果你正在创建可视化，确保读者能够区分出在45%和55%的数据点是很重要的。为此你可能需要改变一些颜色需要发生变换的点，或者完全不拘泥于色彩展现，不采用色彩作为主要的展现方式。

还应该增加文字说明以帮助色盲的读者，因为几乎每10个人当中就有一个是色盲。如果你希望你的可视化能够覆盖尽可能多的读者，你可能会考虑使用黑白色阶，而不是红绿色阶。关于设计和色盲方面的更多信息，请访问We Are Colorblind（我们是色盲）网站（*http://wearecolorblind.com*），该Web站点专门为色盲人士而设计。

位置

基于位置的展现方式就是把数据和某些类型的地图关联起来，或者把它和一个真实或虚拟地方相关的可视化元素进行关联。日常生活中基于位置的可视化的一个例子是，为了方便选择座位而提供给顾客的关于飞机或剧院的一个简单的轮廓。

在图2-6中，我们观察到在美国Florida州的地图上显示的从1996年到2008年的各个郡的犯罪比率。

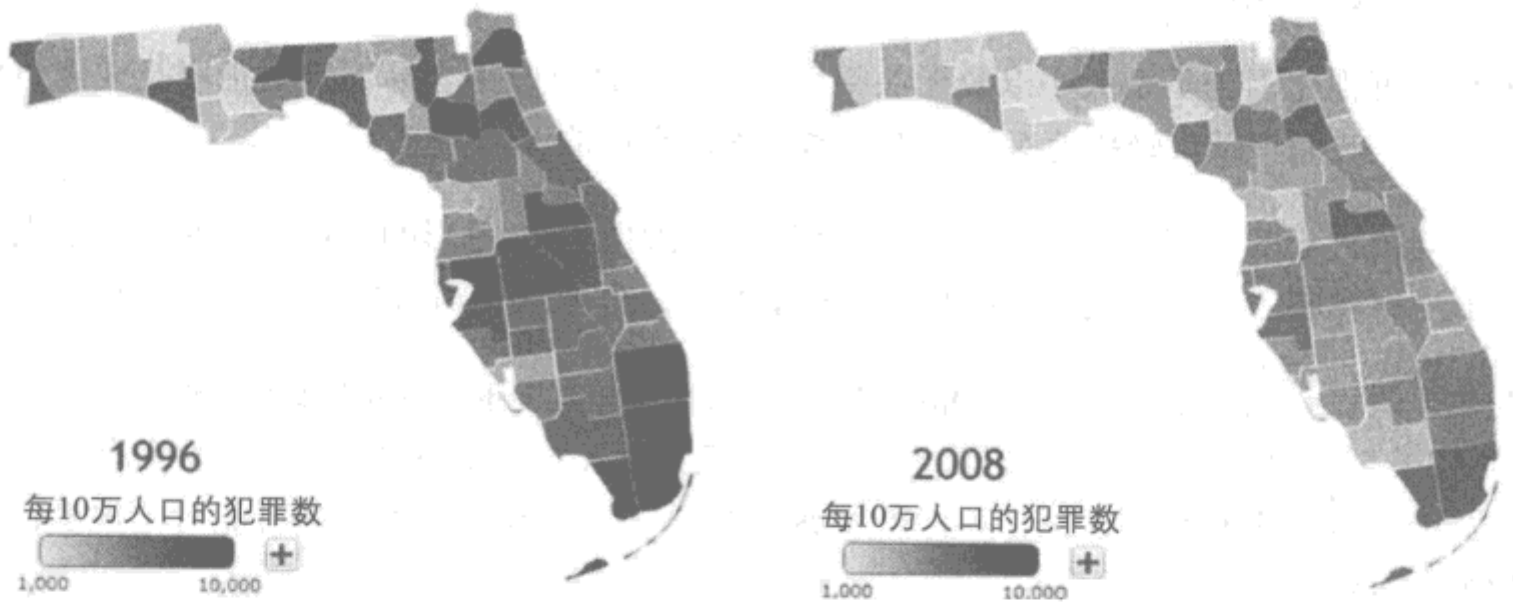


图2-6：Florida州各个郡的地图，通过不同的颜色深度来表示每个郡的犯罪比率（见彩图9）

当观察者对于所描述的位置比较熟悉时，位置展现方式对于可视化会特别有价值。只要对所展现的位置有一定的了解，观察者就可以把他们的个人背景和可视化关联起来，并且可以基于对该地区的个人经验来下定结论。

网络

网络展现方式显示了数据点之间的二元连接，在查看这些数据点之间的关系时很有帮助。在线网络可视化如雨后春笋，它们使得人们可以看到他们在Facebook上的朋友或者在微博Twitter上的关注者的地图[译注3]。

图2-7显示了我的Facebook朋友以及他们当中彼此互为朋友关系的人数的网络可视化。

通过该网络映射，我们可以一目了然地看出我所拥有（或被拥有）的不同的社交网络。此外，各个组的密度和它们的社交亲密关系的对应非常吻合。

对于网络可视化，需要记住的一点是，如果这些可视化不是精心构建的，那么成千上万的数据点可能会变成视觉凌乱的连接，它们对于我们增强了解这些连接的涵义是没有帮助的。

图2-7：我的Facebook朋友关系的网络可视化的关系渲染图

译注3： Facebook是当前美国最火的社交网站，Twitter是当前美国最火的微博。

时间

随时间变化的数据（股票报价、选举结果等）通常是根据时间轴进行描绘。然而，最近几年，具备动画功能的软件使我们能够以不同的方式来描绘这些数据。像《纽约时报》的动画“Twitter Chatter During the Super Bowl”[注2]（见图2-8）把一段较长的时间进行压缩，从而使得我们可以在加速环境中观察到数据的变化。

点击动画左上角的Play（播放）按钮启动动画，在全国范围内，和美国橄榄球超级杯大赛（Super Bowl）相关的tweet（微博）消息中使用最频繁的单词，在比赛过程中会随着其使用频率的增长或减少而被展示出来。

该可视化为用户提供了一系列有用的随时间变化的脉络线索，显示了在那时发生的主要事件。通过这种方式，作者提供了宝贵的背景信息，使用户无须特意记住比赛是如何结束的。相反，他们可以专注于全国范围内的tweet消息中所用到的单词，当有重要事件驱动数据时，让应用给他们发出报警。

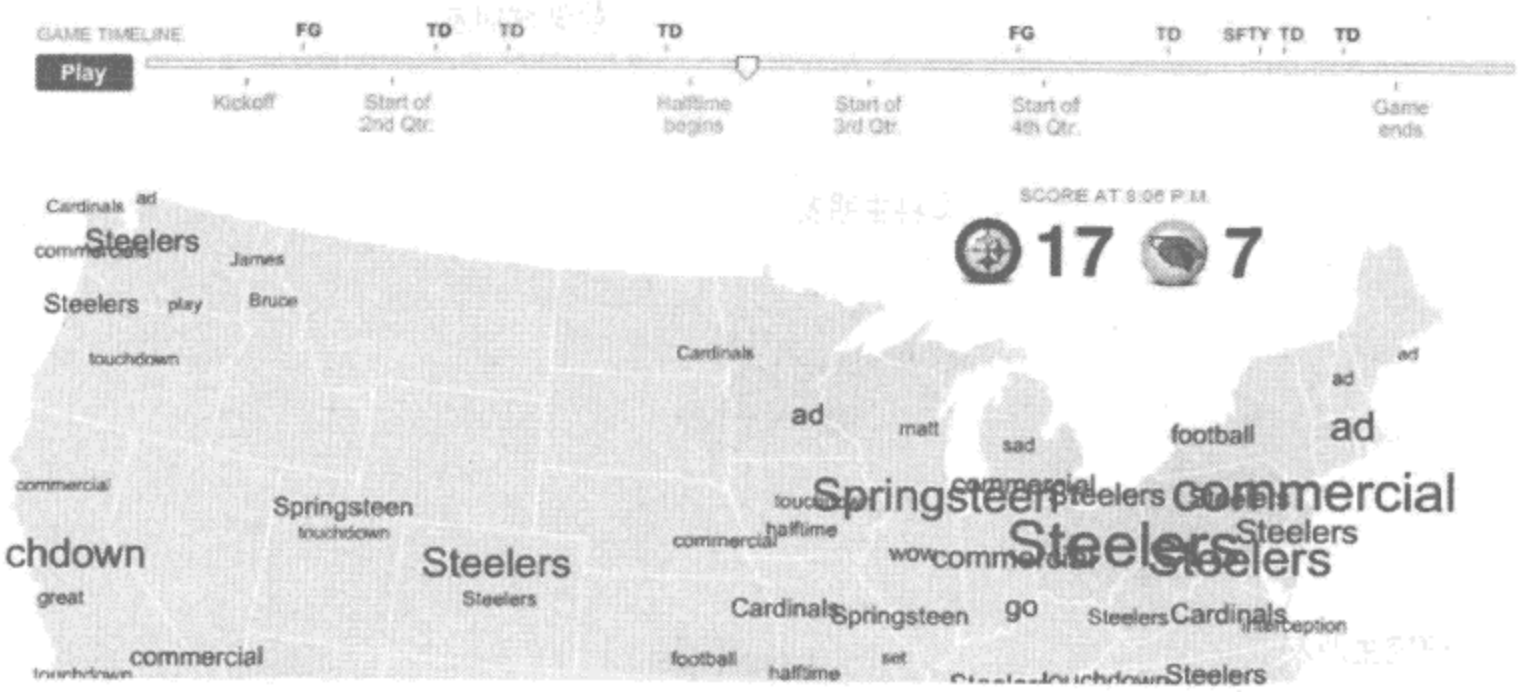


图2-8：《纽约时报》对和2009年美国橄榄球超级杯大赛相关的tweet消息中使用最频繁的单词的可视化

应用多种可视化展现方式

很多优秀的信息可视化使用多种视觉展现方式来全面展现数据。在一个在线应用NameVoyager（*http://www.babynamewizard.com/voyager*）中，用户可以输入一个名称的前几个字母，然后查看历史上有多少人以该字母为开头给他们的孩子命名（见图2-9）。

注2：参见*http://www.nytimes.com/interactive/2009/02/02/sports/20090202_superbowl_twitter.html*。

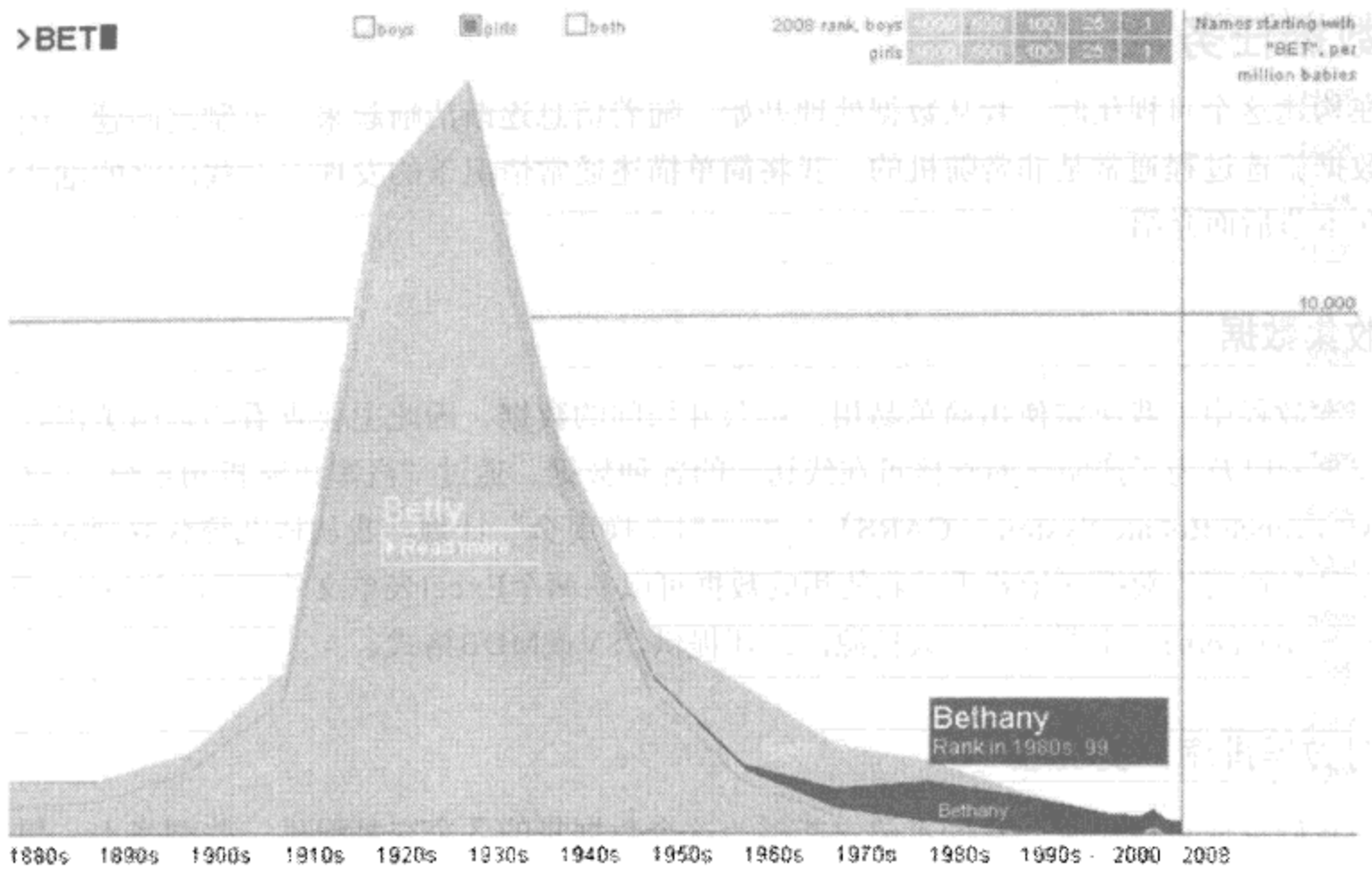


图2-9：NameVoyager的孩子名字探索图显示的逐年的名字频率（见彩图10）

该图使用两个维度进行可视化。第一个维度是时间：通过时间轴，对以输入的字母为开头的名字的使用频率进行展现。第二个维度是大小：图像上的阴影区域表示在某些年份以某些名字命名的孩子的个数。

这种特定类型的图形被称为堆叠时间序列，它是一种非常标准的可视化方式，将多种可视化方法以相互结合却又各自独立的方式应用于信息的多维可视化。

可视化创建实践

到目前为止，我们已经讨论了通常情况下信息可视化的一些基础知识，现在一起来完成一个可视化的构建。我们将创建一个静态可视化，通常称为信息图表（infographic）。

为了完成这个可视化实例，我们需要以下工具：

- Microsoft Excel（或者Gdoc）
- Adobe Photoshop（或者GIMP，一个免费的图像处理程序）

为了尽可能地重现该过程，我将以实际事件发生的顺序来描述这个过程，而不是以之前的“问题—数据—展现”的方式来描述。

数据任务

在构建这个可视化时，我从数据处理开始，随着信息逐渐清晰起来，再制定问题。因为数据筛选过程通常是非常随机的，我将简单描述通常情况下的发现。这些内容的细节将在本节后面介绍。

收集数据

在本教程中，我决定使用简单易用、可公开访问的数据，因此主要查看的是由美国政府收集的以及为了透明化而直接可在线访问的各种数据。通过“汽车津贴折扣系统”（Car Allowance Rebate System，CARS），即“旧车换现金”计划，我从该系统获取到的汽车交易和购买的数据开始着手。我使用的数据可以从两个Excel表单文件（*http://www.cars.gov/carsreport*）中获取。该数据源网站还提供CSV或MDB格式。

对数据排序：发现版

当我们完成可视化时，我们希望它能够为这个数据集的各个交易提供一些洞察力。想象一下，有个人开着一辆破旧的车，自思自忖着很快就能够摆脱这辆又老又破的汽车，换成一辆崭新的汽车。

她正在开着一辆什么样的车呢？她是否期望寻找一辆相似的但是更新、更高效的车呢（“以旧换新”）？或者她是否希望把汽车换成一些完全不同的车（更像是“两厢的SUV”这一类的车）？

我们要查看的数据包含超过65万的个人故事，每个故事都需要动机、驱动、时间和付出。我们无法从数据中梳理出每个人的故事，但是我们的可视化可以有助于揭示这些人所做出选择的背后所蕴涵的故事。我们的目标是找到一种方式来讲述一个故事，使得该故事对于用户/观察者有趣而又新鲜。

以下是我为了发现故事对数据进行排序和过滤的一些处理步骤。

当下载完数据集，我开始查看回购的数据，试着通过很多种不同方式对它们进行分组。对汽车型号进行分组刚开始看起来很有意思，但是这个过程很乏味，因为汽车是通过发动机和变速器进行分组，因此相同型号的汽车可能存在一些不同的登记方式。

然而，在通过汽车型号查看汽车的过程中，我发现某些汽车型号有非常高的回购量，我对此感到很震惊。我开始好奇人们是否更期望购买某种型号的汽车，因此我开始根据汽车型号对车辆进行分类。

警告：当创建可视化时，提出类似“人们是否更热衷于回购某种型号的汽车？”这样

的问题是很危险的。数据会告诉我们很多东西，但是数据很少会给我们提供和人类动机一样复杂的良好信息。描绘数据本身是一回事，而解释数据涵义却又是另一回事。如果因为福特汽车比其他型号的汽车有更高的回购量，就在可视化中表明人们更渴望摆脱福特车可能是错误的。这种表述将忽略很多重要的变量，包括如市场份额、销售的汽车类型、福特这一型号在汽车销售中的地位、汽车的使用年份等。限制可视化的一个好的经验法则是：只从数据本身查看问题，允许用户或者观察者自己下结论。

介绍完以上这些，对可视化内在的问题提出质疑将是发现规律的有效驱动，因此不要怯于在早期提出这些问题——而是要避免在最后的可视化中回答这些问题。

我开始通过汽车型号对它们进行排序，对回购的汽车交易额进行汇总，我觉得比较不同型号（本田、丰田、通用、福特和克莱斯勒）的回购量和新车购买量是很有意思的。随着开始收集这些数据，我们逐渐发现汽车型号太多了，以致难以清晰地描绘 很多不同的数据点。因此，我开始通过“母公司”对汽车型号进行分组，即把同一汽车公司制造的不同型号的汽车放在一组中。举个例子，雷克萨斯是丰田公司生产的一个汽车型号，因此我把雷克萨斯型号和回购的丰田型号的汽车统一以丰田公司作为分组，把这两个不同型号的汽车组合在一起。

最后，我认为最有利的信息描述方式是把所有型号以生产国家进行分组，把同一国家不同型号的汽车组合在一起。这种方式的好处是可以把汇总的数据点的总数减少到十几个，并把信息以不是非常明显的方式组合在一起。通过这种方式，我们能够以崭新的方式来查看数据。

对数据排序：技术版

既然我们已经理清了对数据排序的思路，现在我们一起开始文件的处理过程。

如果下载了Excel文件，在打开这些文件后，你可以发现这些数据首先是根据汽车行业进行分类的（卡车第一，轿车第二），然后对汽车型号按字母序排序（Acura、Audi、BMW等）。为了根据汽车的生产国家对数据进行排序，最简单的方式是通过汽车型号对数据进行分类，然后，我们将决定哪一种型号汽车和其“母公司”所在的国家一致。

为了对Excel表单中的数据进行排序，只需要在newvehicles文件中选择New_Vehicle_Make这一列，或者在*trade-in-vehicles*文件中选择Trade_in_make列，然后选择“Sort & Filter→Sort A to Z”。如果Excel文件弹出对话框，问是否要扩大选择范围，则接受该选项。

你可以通过以下方式把特定型号的汽车的购买和回购的数量汇总起来：输入“=SUM(”，然后使用鼠标选择Count列中特定型号的所有的单元格。作为第一次尝试，

把所有的Acura这一型号的汽车的购买数量加起来，结果应该是991辆汽车。把所有型号的汽车购买数量进行汇总，把结果值放到另一个页面中，这样可以帮助你更方便地查看数据。

如果你有这个爱好，这是尝试探索数据的最佳时机。试着弄清楚哪一款汽车销量最高，或者哪一年的回购量最大。即使是对于和当前的数据集一样小的数据集，也可以提出几十个有意思的问题。可能灵机一动，你就想到其中某个问题，并激发你创造新的、有吸引力的可视化。至少，这是去实践查看数据的一个非常好的机遇。

对这种数据进行排序存在很多种方式。可能写个脚本或程序来处理该CSV文件、并且把数据放到更易于查看的汇总文件中是更高效的（而且更让人印象深刻）。在这个例子中使用Excel是为了帮助不熟悉编程的人们参与数据处理和可视化创建。

制定问题

到了现在这个阶段之后，我们应该对自己要做什么才能为可视化制定充实的问题有了坚实的认识。我们的问题是：“在‘旧车换现金’项目中，汽车的购买和其生产厂家所在的国家的比例分布是什么？”

基于该问题场景，我们可以选择构建很多相关的信息来相应地搭建可视化，记住我们的目标受众可能并不会马上对该主题感兴趣。以下几项有助于为数据增添场景信息：

- 该项目花费了2 850 162 500美元，提供677 081辆汽车的采购资金。
- 对于购买的每辆汽车，有一辆是回购并报废的。
- 该项目始于2009年7月1日，终于2009年8月24日。
- 回购的汽车每加仑油耗的行驶里程必须少于18英里(MPG)。
- 购买的汽车每加仑油耗的行驶里程必须大于22英里。

为了可视化，我们最感兴趣的是汽车购买和汽车报废之间的关联关系。这在人们想要摆脱的各种汽车以及他们想要购买的新的汽车之间产生了一个有趣的平衡现象（也即某种程度的戏剧化）。当我们把数据和可视化放在一起，我们需要记住这种平衡特征，并相应地调整可视化。

把问题弄清楚之后，我们已经有了坚实的基础，我们可以通过分组和排序来进一步对数据进行处理了。

对数据分组

这个步骤需要做一些调研。为了按生产国家对不同型号的汽车进行分组，我们必须查明

哪个汽车型号对应哪个公司。在公司信息和汽车型号信息这两个文件中包含50多种汽车型号，因此需要花一些时间进行调研。对于这项任务，Wikipedia是很好的助手，因为它可以快速地为各种不同型号的汽车提供其所属的公司（举个例子，在这个数据集中，克莱斯勒汽车公司拥有6种汽车型号）以及这些型号的汽车的总部所在的国家。

为了节省您的时间，我提供了一个包含这些数据信息的有用的表（见表2-1）。

表2-1：通过型号、所属公司和所在的国家进行分组的汽车

型号	所属公司	国家	型号	所属公司	国家
Jaguar	Tata	England	Hyundai	Hyundai	South Korea
Land Rover	Tata	England	Kia	Hyundai	South Korea
BMW	BMW	Germany	Volvo	Volvo	Sweden
MINI	BMW	Germany	Saab		Sweden
Mercedes-Benz	Daimler	Germany	American motor	Chrysler	U.S.
smart	Daimler	Germany	Chrysler	Chrysler	U.S.
Audi	Volkswagen	Germany	Dodge	Chrysler	U.S.
Porsche	Volkswagen	Germany	Eagle	Chrysler	U.S.
Volkswagen	Volkswagen	Germany	Jeep	Chrysler	U.S.
Acura	Honda	Japan	Plymouth	Chrysler	U.S.
Honda	Honda	Japan	Ford	Ford	U.S.
Isuzu	Isuzu	Japan	Lincoln	Ford	U.S.
Mazda	Mazda	Japan	Mercury	Ford	U.S.
Mitsubishi	Mitsubishi	Japan	Merkur	Ford	U.S.
Infiniti	Nissan	Japan	Buick	GM	U.S.
Nissan	Nissan	Japan	Cadillac	GM	U.S.
Subaru	Subaru	Japan	Chevrolet	GM	U.S.
Suzuki	Suzuki	Japan	GMC	GM	U.S.
Lexus	Toyota	Japan	Hummer	GM	U.S.
Scion	Toyota	Japan	Oldsmobile	GM	U.S.
Toyota	Toyota	Japan	Pontiac	GM	U.S.
			Saturn	GM	U.S.

然而，需要记住的是，这种通过型号对汽车进行分组的方式对数据提出了一些问题，我们在继续下一步探讨之前需要回答这些问题。举个例子，Jaguar[译注4]是一个典型的总部

译注4：Jaguar即捷豹，是一款很名贵的汽车。

设在英国的英国公司，但它却为印度公司Tata汽车公司所有。那么，我们应该把Jaguar划分为英国汽车还是印度汽车呢？

处理这类问题的“正确”的方法主要是由个人喜好决定。重要的是在可视化展现中，对此类问题的决定应该保持一致性，并且向读者传达这样的信息：你以某种方式做出了决定。通常情况下，在可视化中给一个脚注进行说明就足够了。

应用可视化展现方式

在这个阶段，我们应该以自己期望的方式获取所有数据：回购或新购买的汽车，通过国家进行分组。现在应该开始选择数据的可视化展现方式。

在该可视化中，我们将展现两个维度的信息。第一个维度是按照国家进行分组的汽车的数量，第二维是购买和回购的汽车之间的区别。购买的汽车和“以旧换新”的汽车之间是“独一无二”的，因此在信息上不存在任何交叠，这将简化展现方式。为了区分购买的和回购的汽车，我们可以使用一种简单的方法来表示：用红色表示“回购”、绿色表示“购买”。

由于我们要处理的数据包含的数据点很少，但是其变化却很多，通过尺寸来表示这种信息是最有意义的。这种展现方式将以直观、有力的方式引起人们对这种变化范围的关注。最简单的实现方式将是使用不同大小的圆圈或者条形图来表示回购和购买汽车的数量。

关于面积和圆圈的注意点

如果我们使用圆圈来表示数据，必须记住的是我们将需要改变圆圈面积，而不是该圆圈的半径或直径。如果我们选择了购买的美国汽车的数量（575 073），并且半径用50个像素来表示，我们将使用以下Excel公式来计算其他每个圆圈的大小：

```
SQRT((US_Baseline_Radius^2 * Target_Vehicles)/US_Vehicles)
```

我指出这一点是因为这种计算方式可能是在一般情况下，用圆圈或者面积对信息进行可视化时最常犯的错误之一；正确的关系如图2-10所示。通过线性增大半径或直径的长度来增大圆圈时，圆圈面积的增加或减少将是呈指数级变化的，如图2-11所示。

至此，我们讲清楚了以上几个问题，但是实际上我们不会使用圆圈。不要着急，我这么做是有充足理由的。

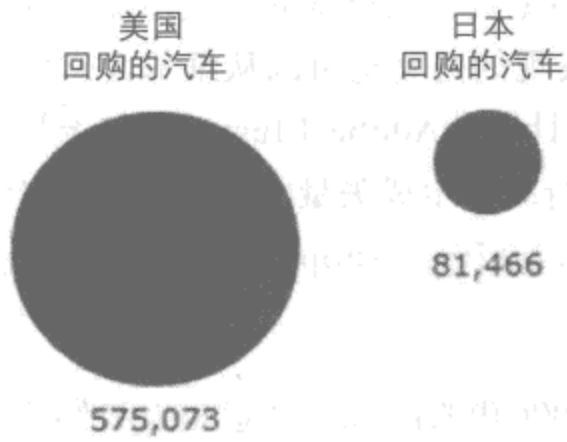


图2-10：正确的方式（增大面积）

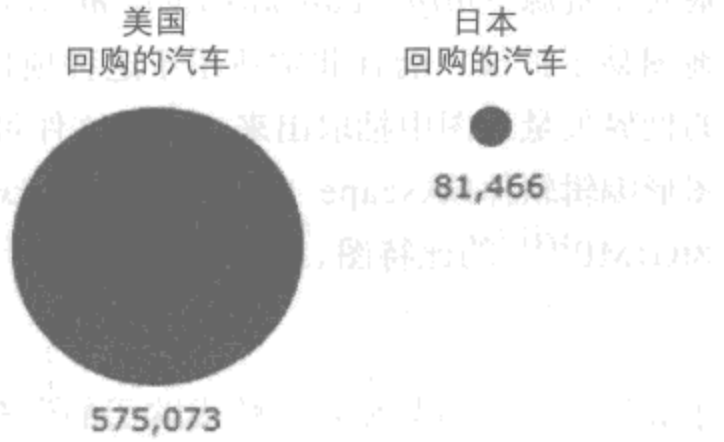


图2-11：错误的方式（增大半径）

通过国家地图展现数据

既然我们的信息可视化是以国家为中心，我们将使用各个国家的形状地图来展示可视化，并相应地调整这些地图。这种展现方式可以给我们的可视化增加一些有价值的附加信息。

首先，使用国家地图将使该可视化项目为读者带来视觉上的亲密感。如果读者的祖国在列表上，他就可以马上找到他的祖国，并且会倾注其注意力。同样地，我们可以拉近读者和其祖国或者他们所熟悉的任何其他国家间的情感。这样的情感拉近使得读者更有可能记住或者推荐该可视化产品。

其次，使用国家形状地图而不是圆圈使得该可视化可以通过很多不同的尺寸大小来传递信息。即使可视化中只有拇指般大小的图形，还是可以轻易识别出国家形状，使得用户可以知道该可视化是和不同国家之间有关系。而一组拇指般大小的圆圈看起来就仅仅是一组圆圈而已。

再次，如果我们只使用圆圈或者条形图，我们将需要依赖文本来表达可视化中的国家的名字。这一点不一定是坏事，但是会增加读者对可视化的理解所需要的时间，因为读者需要阅读文本才能理解可视化。这种方式将可能削弱可视化带来的直接影响效果。

最后，读者习惯在世界地图中看到的这些不同的国家，它们的相对大小比例总是相同。如果在可视化中不考虑读者所熟悉的这些形状，而展现为韩国比德国大或者美国比日本小，该可视化将会和读者的预期差别很大。它会被读者认为“扭曲”了真实的事实。

确定了应该使用国家形状而不是圆圈的方式来展现可视化之后，我们需要在列表中找到国家的可视化展现。最可靠的方式是搜索以“.svg” 为后缀的文件中的国家名字。SVG表示可缩放矢量图形（Scalable Vector Graphics），是由W3C协会倡议的开放标准的矢量图形[译注5]。它是一种流行的矢量图像标准，尤其适用于免费的图像和地图，很多矢量控制应用程序都支持它。

译注5： 可访问其主页获取更多信息，*http://www.w3.org/Graphics/SVG/*。

维基共享资源（*http://commons.wikimedia.org*）包含很多免费、高质量的矢量地图。这些地图易于扩展，而且非常适用于这种项目。有些难以发现的国家也可以从维基共享资源的世界矢量地图中抽取出来。这些文件可以通过图形设计软件Adobe Illustrator或者矢量图形编辑软件Inkscape（*http://www.inkscape.org*）进行编辑生成矢量文件[译注6]，或者作为GIMP[译注7]的比特图。对于Illustrator，其矢量对象可以在Photoshop中直接拷贝和粘贴。

为了简化，我们将只显示回购或购买的汽车的数量超过1000辆的国家。这意味着我们的可视化需要美国、日本、加拿大、德国、瑞典和英国的地图。

一旦我们有了这些国家的图像，我们就为可视化的最后一步（即调整图像大小）做好了准备。

构建可视化

在将图像加载到图像处理程序之后，我们需要调整它们的大小，以便能够合理地表示回购和购买的汽车的比例。

处理该问题的方法是采纳最大的数据块（在这种情况下，即美国制造的汽车的回购数量：575 073辆），然后把它的大小调整到适合于一个信息图形的画布大小。这种锚形状（anchor shape）是非常实用的，可以确保没有一种图形元素会因为尺寸太大而影响可视化显示上给人的优雅的感觉。把这种数据作为锚，我们可以对所有其他数据元素相应地调整大小。

一旦确定了锚形状的大小，我们需要计算其中包含多少像素。Photoshop和GIMP软件处理图像时存在技巧，使我们可以很容易地计算在特定层选定的像素的个数。这两款软件都有一个菜单窗口名为“直方图”（Histogram），它显示了当前选定的像素的个数。使用该工具，我们可以确定锚的像素个数，通过以下公式，可以计算其他形状需要包含多少像素：

Target_Size = Target_Number * Anchor_ Size / Anchor_Number

译注6：Adobe Illustrator是Adobe公司推出的图形设计软件，可以通过公司的网站产品介绍*http://www.adobe.com/cn/products/illustrator/*了解更多；Inkscape是一款开源的矢量图形编辑软件，使用W3C标准的SVG文件格式。

译注7：GIMP：GNU图像处理程序（GNU Image Manipulation Program），是一款位图图形编辑软件。可以访问其网站*http://www.gimp.org/*了解更多。

举个例子，日本汽车的回购数量是81 466辆。如果我们调整美国地图大小为25 000个像素，那么计算日本地图大小的等式如下：

Japan_Size = 81 466 * 25 000 / 575 073 = 3 542 像素

通常使用Excel来计算，因为这样可以很容易地保存、检查和复制。

利用直方图的技巧，我们可以对目标国家的不规则图形重新调整大小，直到它们包含适合相应数据点可视化的像素数量。

为了适应于展现可视化的媒体（对本书而言是一个页面），我决定通过一条垂直轴对这些国家进行排列。这种方法为色彩元素增添了对称性，增强了数据中的绿色/红色、新买的/以旧换新的二分区别。

现在，我们已经完成了可视化需要的核心工作。在介绍性宣传单上提供一些背景信息，增加关于Jaguar和Land Rovers的起源国家的标注[译注8]，得到如图2-12所示的结果。

该可视化满足了我们的标准。在它的最上方给出了故事的介绍信息，以鲜明的布局展示方式吸引了读者的注意力，而且可以立即被理解。我们通过颜色编码表示“购买的/回收的”汽车之间的二分区别，通过物理上的对称性增强了该展现效果（如果我们希望那些色盲人员也能够理解该信息图，对称性是很重要的）。该可视化说明了我们期望给读者一个真正激动人心的故事。

结束语

该教程谈到的只是创建有效可视化的技巧的一小部分。如果在以下领域具备更深层次的基础，如色彩理论、印刷术、计算数据挖掘和编程，以及关于数据主题的一些背景知识，那么在创建吸引人心的可视化中都将提供很有价值的帮助。

虽然不同领域都为可视化创建过程提供了一些不同的信息，但它们都属于一个统一的整体，因为每个可视化都是某个故事的一部分。即使显示一个公司的盈利数据的最简单的条形图也是从一个更大范围（可能是管理风格上的变化）、更令人难忘、更有价值的信息中获取到的。正是这些不同的场景以及和它们相关的故事，赋予了可视化长期持久的影响和力量。

译注8：Jaguar和Land Rovers这两款汽车都是属于Tata公司的，该公司总部在英国，但是属于印度的公司。

WINNERS & CLUNKERS

Between July 1 and August 24, 2009, the federal goverment provided 677,081 rebates to individuals who traded in an older, inefficient vehicle for a new fuel efficient one.

This is a visual of the countries from which vehicles were "clunked" and the countires that built the cars for which they were traded.

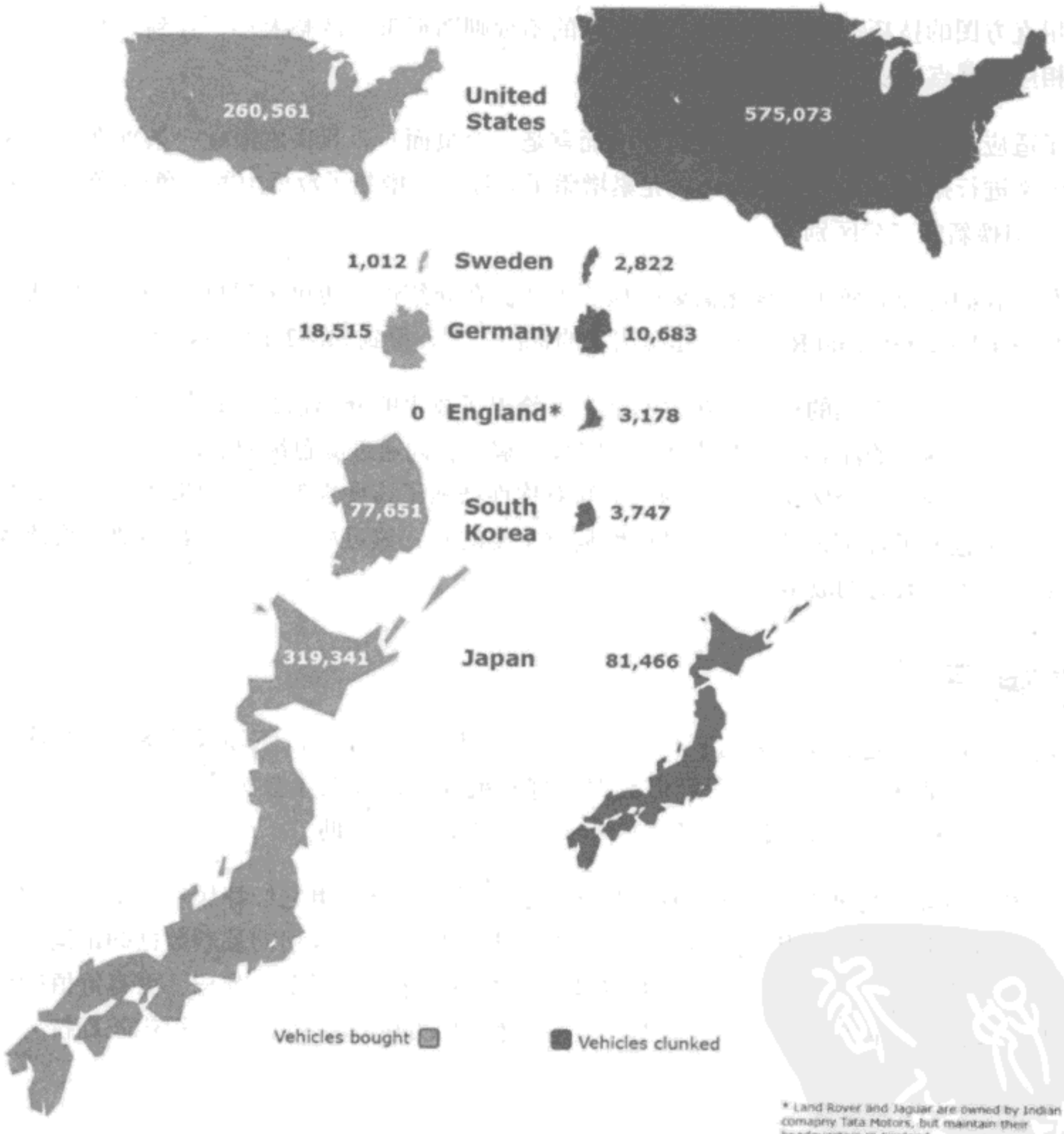


图2-12：最终生成的可视化（见彩图11）

第3章

Wordle

Jonathan Feinberg

图3-1：本章的一个Wordle图例（见彩图12）

到目前为止，即使是从未听过“信息可视化”的人对于绚丽多彩的单词拼贴“Wordle”[译注1]也都很熟悉，Wordle被认为是“文本分析处理的‘入门仙丹’”（the gateway drug to

译注1：　Wordle是一种工具，能够根据提供的文本，生成单词拼贴图形（即单词云，word cloud）。

textual analysis）[注1]。正如很多这样的“仙丹”一样，虽然Wordle起源于像del.icio.us和Flickr这样的站点对这种功能实用的标签云的推广，但它的诞生却仅仅是出于好玩。

Wordle的起源

在2004年，我的同事Bernard Kerr和我一起制作了一个社会标签应用，Bernard 把它命名为“dogear”[译注2]（Millen、Feinberg和Kerr，2006）。任何一个应用，只要允许用户对内容添加标签，就必定会提供一个“标签云”（tag cloud），它是由可点击的关键字组成的一个模糊的矩形集合。因此，当我们设计dogear应用时，我们确定对每个页面都添加了醒目的“标签云”标识（见图3-2）。

图3-2：在dogear应用中显示的作者的标签

我之前从未发现过标签云在可视化上有什么特别有趣或者让人赏心悦目之处。没有足够的证据表明标签云对于导航或者其他交互任务会确实很有用[注2]。但是，当Matt Jones[注3]在他的博客上把del.icio.us网站的标签以美丽、排版上生动活泼的图像发布出来时，我感到非常激动。我认为一个计算机程序一定会创造出类似的效果。至少，我希望最后可以是通过某种方式——类似Jones的云标签——把点“i”放到点“g”的下方，这一点超出了标签云当时力所能及的范围。

注1：参考*http://www.profhacker.com/2009/10/21/wordles-or-the-gateway-drug-to-textual-analysis/*。

译注2：dogear是“书页折角”的意思。它是IBM的一个协作式用户体验项目，可以访问*http://domino.watson.ibm.com/cambridge/research.nsf/0/1c181ee5fbcf59fb852570fc0052ad75*了解更多。

注2：参考*http://doi.acm.org/10.1145/1240624.1240775*。

注3：参考*http://magicalnihilism.com/2004/07/04/my-delicious-tags-july-2004/*。

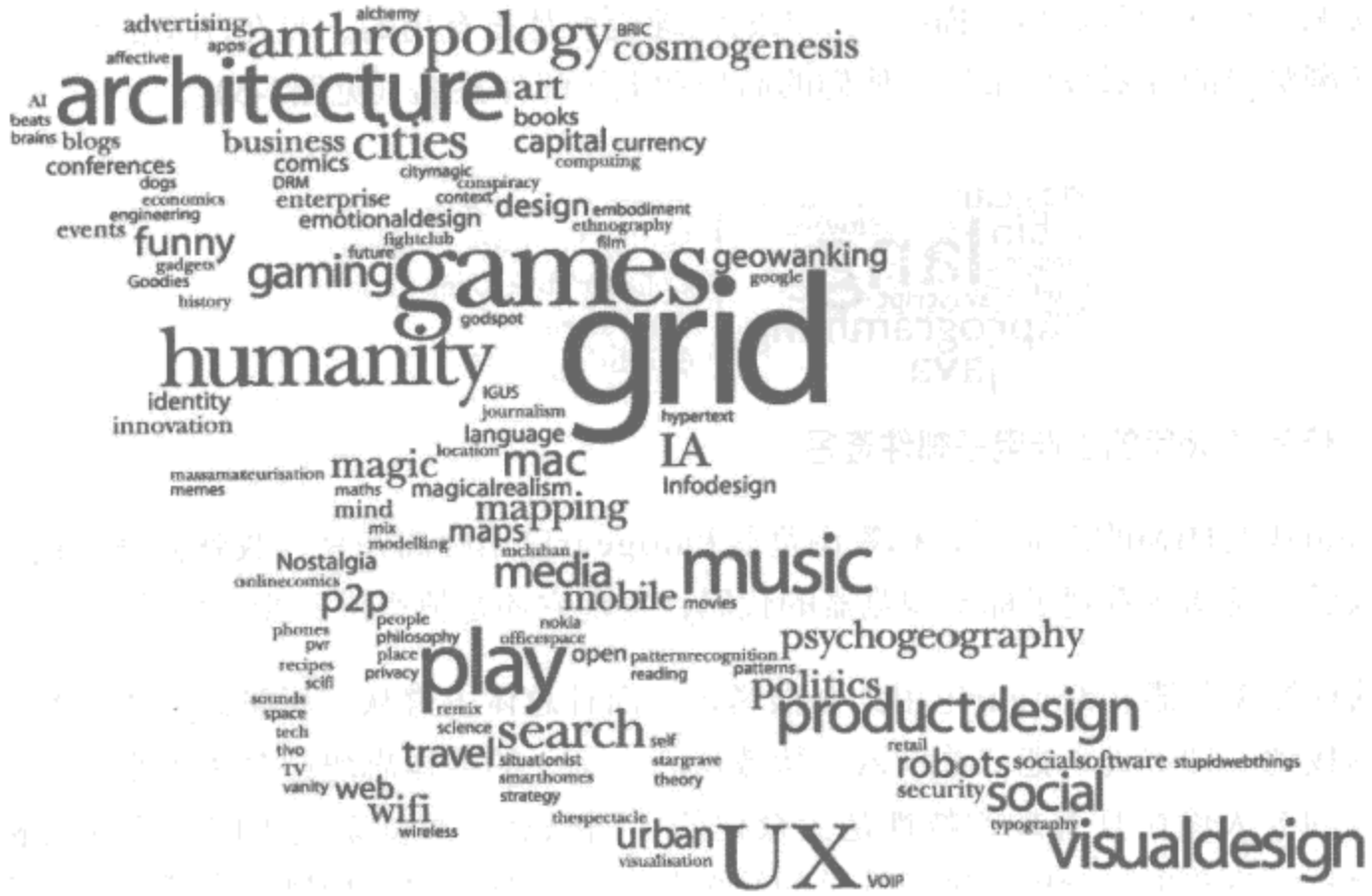


图3-3：Matt Jones做的排版上可识别的标签云

我花了一周左右的时间编写代码，实现了所谓的“标签浏览器”（见图3-4），它是一个Java应用小程序。这个小程序使得用户可以通过dogear应用，点击和当前内容相关的标签来浏览页面。

图3-4：Dogear标签浏览器[注4]

注4：参考*http://www.flickr.com/photos/koranteng/526642309/in/set-72157600300569893*。

显而易见，标签浏览器用于描绘一个人的兴趣爱好是很有用的。我在IBM的很多同事使用标签浏览器的屏幕截图来显示他们的简历和电子邮件签名（见图3-5）。

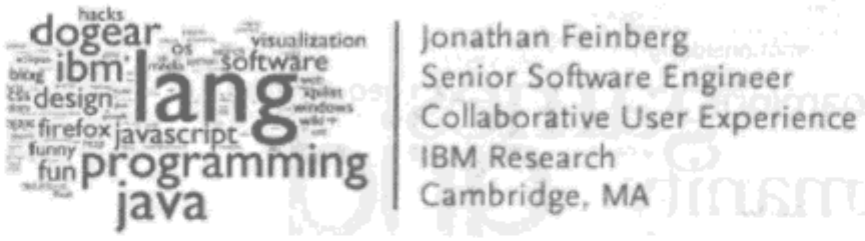


图3-5：作者2006年的工作电子邮件签名

当dogear成为IBM的产品[注5]，标签浏览器和dogear应用脱离开来，我就完全忘记了它。几年以后，我偶然看到了标签浏览器的代码，发现它还是值得进一步开发的。

原始的标签浏览器和dogear应用结合很紧密，而且总体上遵从“标签云”的思想。我希望能够找到一种方式，把“单词云”特效从“标签”的整体思想中分离出来，因为单词云具有的令人愉快且有趣的特性是大众化的，而只有了解复杂技术的人们才对标签熟悉。因此，我产生了对单词进行计数的简单想法。一旦决定了构建一个系统来查看文本而不是标签，对单词执行其他任何处理而不仅仅只是在页面上展现看起来都是多余的。我决定还是以快乐作为设计的目标，秉承Charles Eames的讲话精神：“谁会认为快乐是不重要的？”反过来，这个决定使得最后确定应该保留哪些特征、舍弃哪些特征以及如何设计交互界面变得简单容易（见图3-6）。

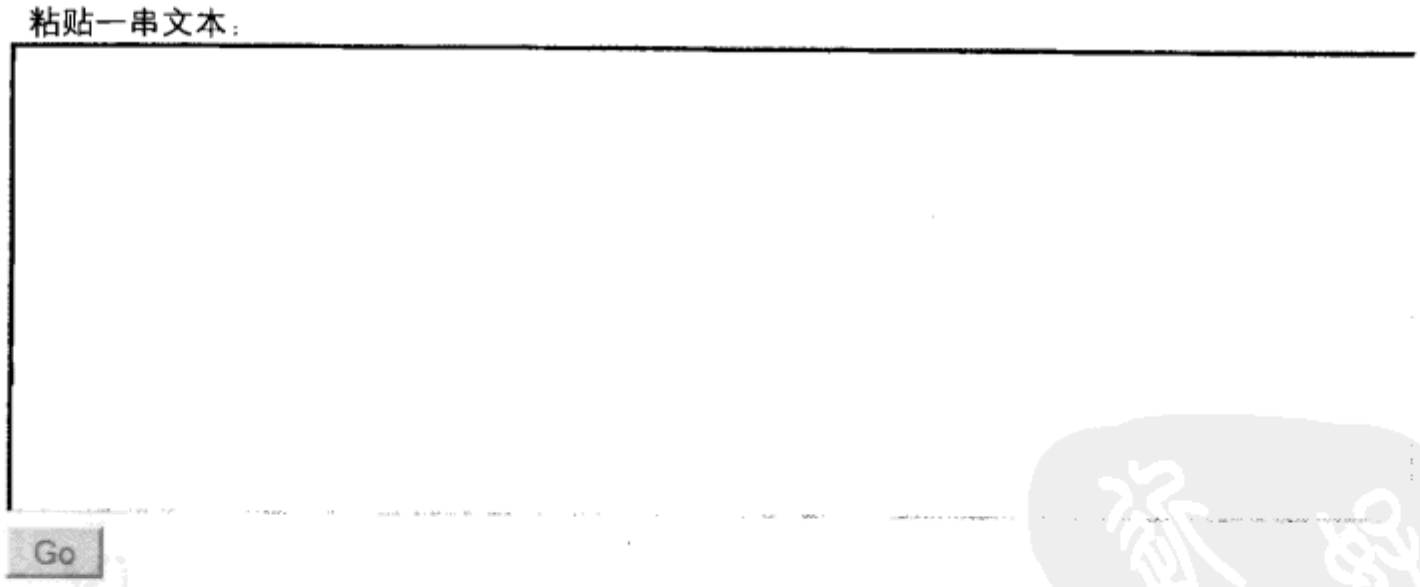


图3-6：Wordle的文本分析用户界面

因为Wordle（正如该词听起来的那样）的目的是为了使人愉快，我需要考虑字体和调色板的表现力（见图3-7）。

注5：参考*http://www-01.ibm.com/software/lotus/products/connections/bookmarks.html*。

图3-7：Wordle提供了各种不同的调色板、字体和布局（见彩图13）

我相信自己为了简化Wordle以及强调商业乐趣上所付诸的努力，已经达到了事半功倍的效果。人们对Wordle的使用方式是我从未预料到的，其使用人数也远远超出了我的期望。Wordle的成功一部分归功于Web应用设计本身，由于它的“一次粘贴/一次点击”给人们带来的瞬间的满足感。虽然Wordle可视化设计本身为其普遍性带来积极影响，可是在我们详细探讨什么是Wordle以及它是如何工作之前，有必要分析一下什么不是Wordle。

解剖标签云

典型的标签云应用是以“嵌入型”的环绕方式组织的[注6]。如果某行的字体大小比其他行大，字体小的周围的空白处将更大，这看起来会很不协调。例如图3-8，“everett hey”的上方有很大的空白，因为该行的字体大小是由其相邻词“everett everett”决定的。

图3-8：“迷失”在空白中[注7]（见彩图14）

减轻这种由于强烈的对比造成空白参差不齐的一种方式是把不同字体大小的单词放入几个不同的区块中，如del.icio.us所做的那样。在图3-9中，“programming”这个标签被用了55次，而“scripting”只被用了1次，但是使用更频繁的字体大小仅大出50%。还应该注意使用字体权重（粗细）来增强不同权重的字体之间的对比度。

注6：如果你想深入研究标签云设计，请查看该网址*http://www.smashingmagazine.com/2007/11/07/tag-clouds-gallery-examples-and-good-practices/*，它包含非常有见地的评论。

注7：参见*http://manyeyes.alphaworks.ibm.com/manyeyes/page/Tag_Cloud.html*。

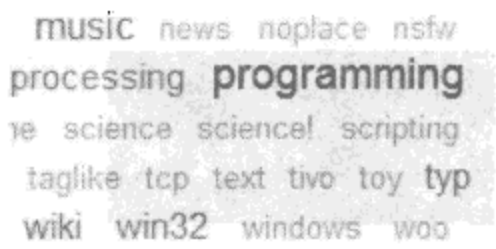


图3-9：借助字体权重来增加区分度

实际上，del.icio.us站点主要是通过计算对数的方式来缩放单词的权重。当源数据遵从幂率分布时，使用对数或者平方根的方式对字体权重进行缩放是合理的，如标签那样[注8]。

在Wordle的真实、有用的设计和充满幻想的世界中，存在一些其他更具有实验研究性的接口。比如WP-Cumulus[注9]的博客插件，提供了旋转的、三维的标签画面（见图3-10）。

图3-10：WP-Cumulus: 几乎无法点击的“标签云”

把可视化和导航结合起来在设计“单词云”上提出了一些约束。但是一旦我们能够从“实用性”中解放出来——一旦我们不再需要提供导航功能——我们就可以拥有更大的发挥空间。

填充二维空间

有很多计算机科学博士因为逐步改进所谓的“装箱问题（bin-packing problems）”[注10]而被授予博士学位。幸运的是，一种简单的方法有一个很不错的名字：随机贪婪算法。该算法是随机的（randomized），你可以随意把单词拖放到屏幕中某个期望的位置附近，

注8：参考 *http://www.citeulike.org/user/andreacapocci/article/1326856*。

注9：参考*http://wordpress.org/extend/plugins/wp-cumulus/*。

注10：参考*http://en.wikipedia.org/wiki/Bin_packing_problem*。

而如果该词和其他词存在交叠，就重新再试一次，直到它不和任何其他词交叠为止。随机贪婪算法的“贪婪（greedy）”之处在于字体大的单词更容易被选中。

Wordle的特定字符依赖于一些限制条件。首先，给定一组包含关联（有意义的）权重的单词列表。我们不能多次显示一个单词，而且不希望显示超出了单词的字体大小而扭曲了单词的形状。不过，如果我们取消这些约束条件，可能会产生很多其他美丽有趣的效果。

例如，你可以使用贪婪算法来填充几乎任何一个区域（不只是一个矩形），只要你有一组单词作为“调色板”，从该调色板中你可以任意次数地选择任意字体的任意单词（见图3-11）。

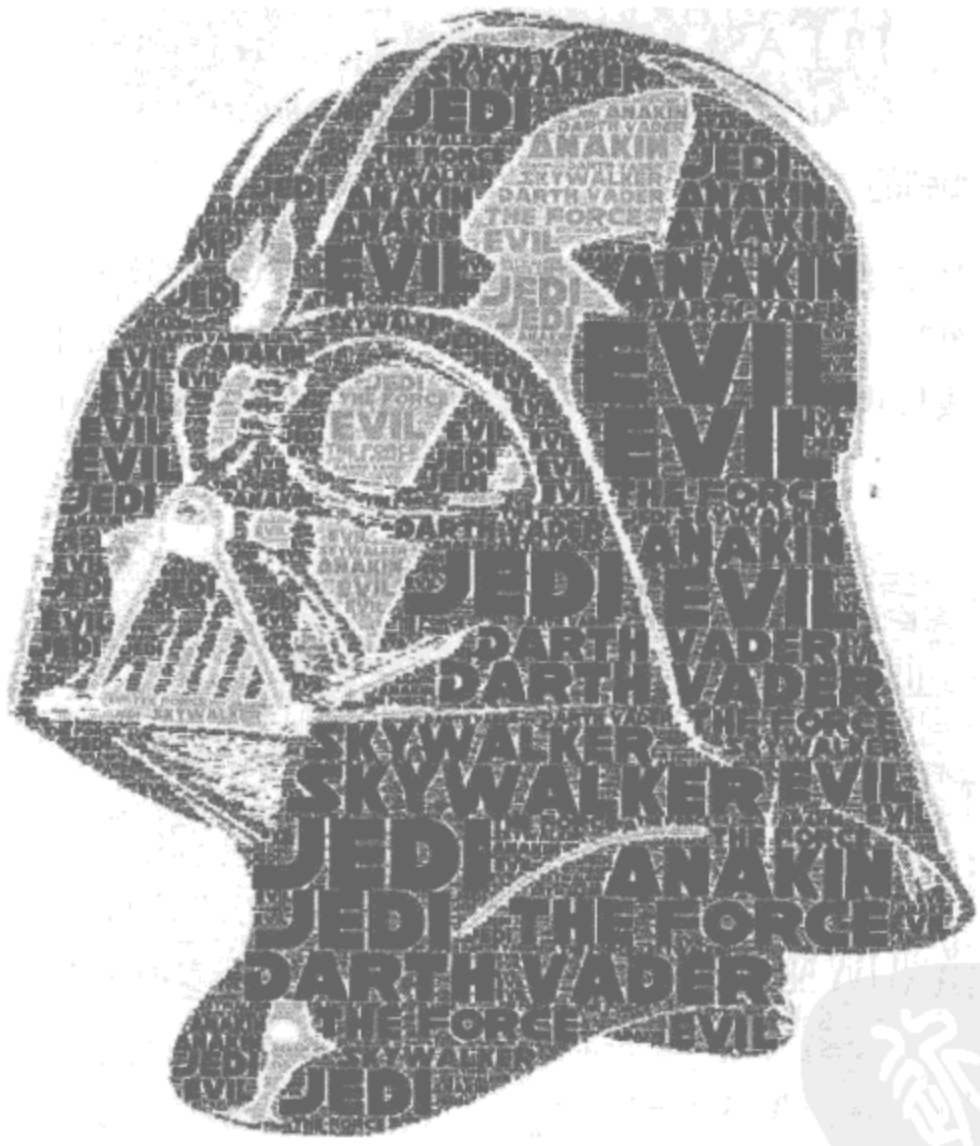


图3-11 不要低估随机贪婪算法的“力量”（见彩图15）

考虑Jared Tarbell的细腻的“情感分形（Emotion Fractal）”[注11]（见图3-12），它递归地把空间细分成更小的随机矩形，用字体更小的单词来填充空间。这种效果依赖于大量的、随机选择的、任意权值的候选单词。

注11： 参考*http://levitated.net/daily/levEmotionFractal.html*。

图3-12：Jared Tarbell的“情感分形”

如果你不介意按照需要拉长或者压缩字体，还可以产生其他的效果。例如，图3-13显示了典雅的“树形图”（treemap）[注12]的变体，它使用文本，而不是矩形来填充空间。每个单词填充的区域与其出现的频率成一定比例；每个矩形区域包含了在原文文本中相互强关联的单词。

图3-13：奥巴马演讲的单词树形图（见彩图16）

注12：参考*http://www.cs.umd.edu/hcil/treemap-history/*。

必须指出的是，早在Processing图形处理软件（Processing sketches）[译注3]和Flash应用小程序产生之前，人们就开始探索在大众媒体和艺术作品上的排版创作（见图3-14）；我们长时间一直在探索文字的格式和字体之间的分界（见图3-15）。探索这些算法的目标是使这些例子中所蕴涵的智慧和优雅能够给文本数据的展现带来良好的效果。

鉴于以上关于Wordle所涉及的技术和艺术背景的简短介绍，我们现在可以更详尽深入地查看Wordle中蕴涵的技术和美学。

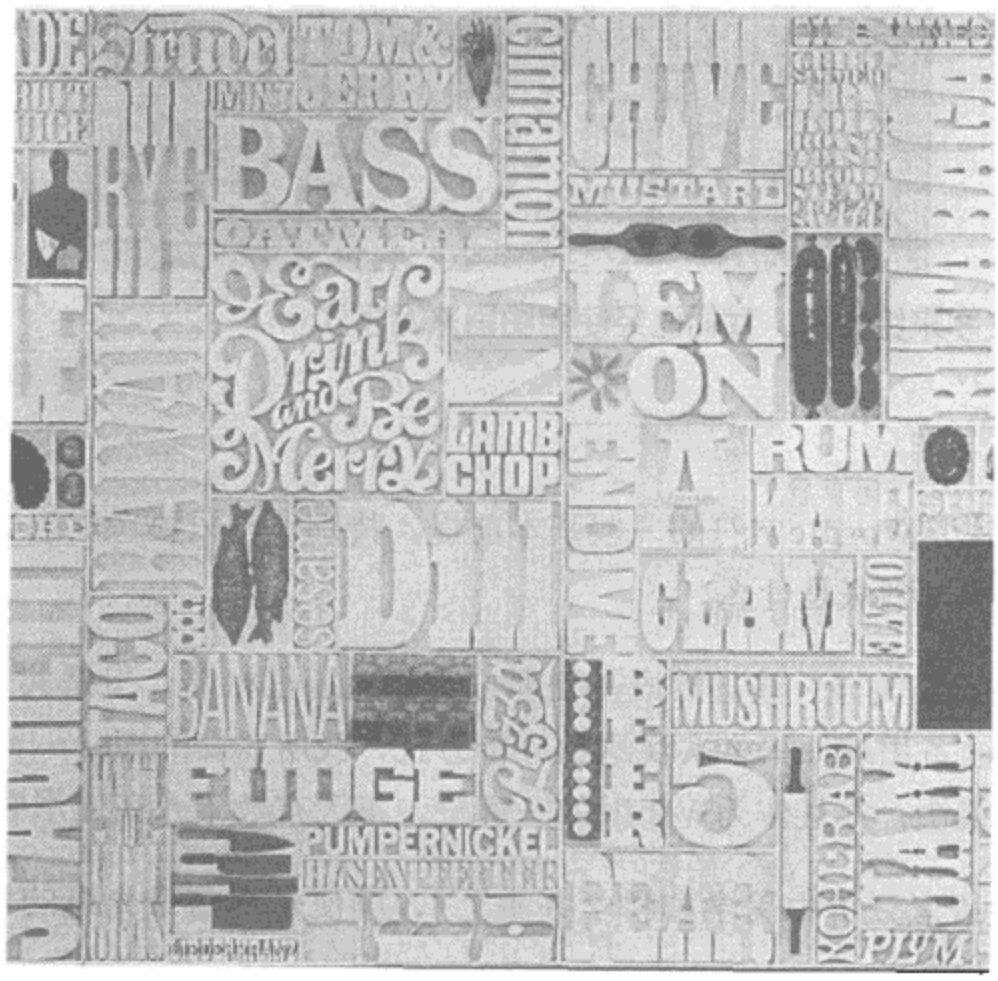


图3-14：Herb Lubalin和Lou Dorfsman的印刷排版组合（由设计研究中心提供，见彩图17）

Wordle如何工作

Wordle是通过Java应用小程序实现的，因此这里提供的一些技术细节是以Java特有的一些语言特性描述的。这里所描述的都可以通过其他语言、使用其他库或者框架来实现，不过Java对Unicode文本处理和二维图形（通过Java2D API）的支持，使得用Java实现Wordle变得简单容易。

译注3：Processing是一款优秀的开源编程语言，人们可以用它创建二维、三维的图形、动画和一些交互应用等。你可以访问其主页*http://processing.org/*来了解更多。

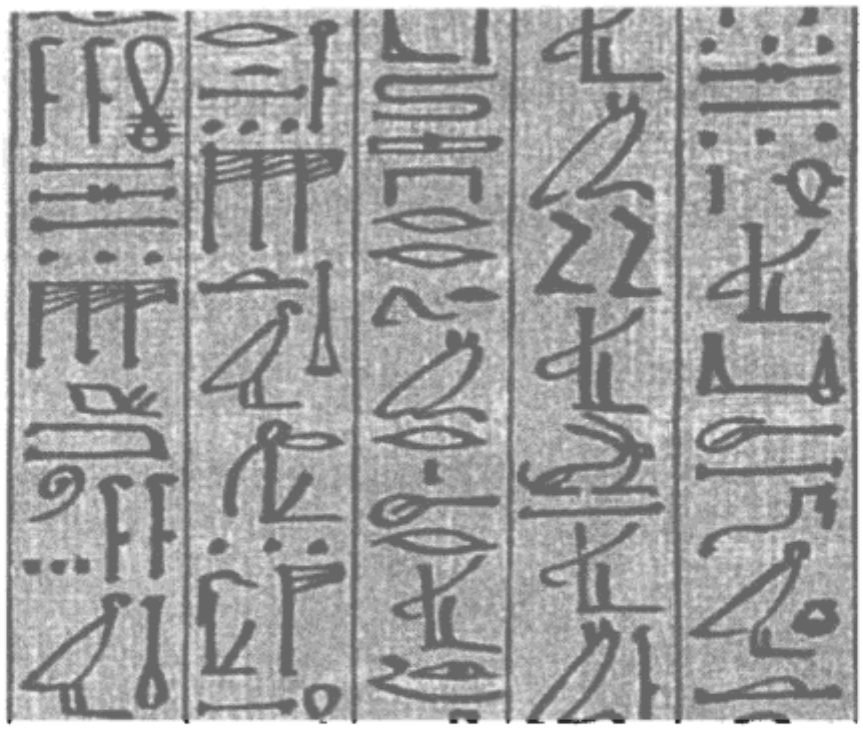


图3-15：在使用文字来绘图前，我们已经使用图画来写字了

文本分析

我们先回顾一下决定Wordle字符的一些基本前提。特别地，只要考虑到Wordle，我们需要首先确定“文本”到底是什么。

虽然这种文本分析方式与一些自然语言处理方式相比还很粗糙，但是其实现起来却也相当繁琐。如果你准备用Java语言实现这种分析方式，我所开发的库cue.language[注13]会很有帮助。它很小很快，并且作为Wordle的一部分，每天有数千人在使用它。

请记住，在自然语言分析中美学和科学并重[注14]，即使是当前最先进的计算工具，其中也需要用到判断和审美。

查找单词

Wordle使用单词进行绘图，每个单词有一定权值，单词的大小由这些权值决定。Wordle如何判定一个“单词”？Wordle构建了一个正则表达式，它能够识别很多不同字体类型的单词，然后通过递归方式，把该正则表达式应用于给定的文本，生成一组结果单词列表，如例3-1所示。

注13：参考*http://github.com/vcl/cue.language*。

注14：如果你想了解自然语言理解这门艺术，请查看本书的姊妹篇《数据之美》中 Peter Norvig 写的关于自然语言处理这一章。

例3-1：如何识别“单词”

```
private static final String LETTER = "[@+\\p{javaLetter}\\p{javaDigit}]";
private static final String JOINER = "[-.:/''\\p{M}\\u2032\\u00A0\\u200C\\u200D~]";
/*
A word is:
   one or more "letters" followed by
   zero or more sections of
       one or more "joiners" followed by one or more "letters"
/*
private static final Pattern WORD =
   Pattern.compile(LETTER + "+(" + JOINER + "+" + LETTER + "+)*");
```

在Wordle中，字符是Java的Character类所能够识别的以下任何一个字符，具体包括：“字母”、“数字”、“@”号和“+”号。连接符包括以下几方面：Unicode的M类，它描述了很多非空格标识和组合标识；URL中经常出现的其他标点符号（因为Wordle的用户期望把URL也作为字符串）；省略号以及一些其他非正式地表示省略号的字符（如单引号字符（'），U+2032）。Wordle支持把波浪符（~）作为单词连接符，但是在输出时把波浪符替换成一个空格，因此用户可以非常容易地“把这些单词连接在一起”，而不需要知道真正地把各个字符连接在一起的“魔术连接符”。

确定字体类型

抽取了一组单词之后（不论这里“单词”的涵义是什么），我们需要知道如何把这些单词展现给观众。我们首先要知道需要展示的字符有哪些，从而可以选定一种字体，能够支持这些字符。

Wordle的字体集是按照每种字体可以支持的字体类型（script）的方式来组织的，一种字体类型（语系）即你所能想到的一个字母：一个符号（字型）集合，可以以一种或多种语言来可视化表示字符序列。一个给定的字体类型，在Unicode中是组织成一个或多个分块。因此，Wordle的任务是通过给定文本中所表现的分块，确定用户可能想要使用哪一种字体。

Java提供了静态方法`UnicodeBlock.of(int codePoint)`来确定给定的代码点属于哪个分块。Wordle获取文本中最常见的单词，并检查每个单词中的首字符。在常见情况下，首字符是属于Latin分块，我们进一步查看该单词的其他字符，看是否包含任何Latin-1编码符（我们认为这种方式可以排除某些字体类型）或者任何Latin扩展分块（这种方式可以排除更多的字体类型）。最后，Wordle会选中最常见的分块作为最终分块。

为了保证响应速度和限制网络资源的使用，Wordle在设计上一次只允许使用一种字体。特征功能更全的单词云可能会为不同的单词选择不同的字体；这种方式可以为展现提供另一种视觉维度，如不同的源文本。

在撰写本章时，Wordle已经支持拉丁语（Latin）、西里尔文（Cyrillic）、梵文（Devanagari）、希伯来文（Hebrew）、阿拉伯文（Arabic）和希腊文（Greek）。Wordle本身有意不支持CJKV字体类型，包括中文、日文、韩文和越南文。因为CJKV字体数据非常大，需要花费用户很长时间下载（因而其带宽成本很高）。此外，确定表意符号的字体边界需要非常复杂的机器学习算法和大量的运行时数据结构，Wordle无法提供这些。

Unicode内核

由于Wordle只能处理Unicode文本，为了理解后面的一些术语和符号，你首先需要了解以下内容。

Unicode[注15]标准提供了一套通用的编码字符集和一些在计算机中表示这些字符的规范（即字节序）。

字符是一个抽象的概念，是表示书面语言的原子单位。它和“字母”不是一个概念——比如一些Unicode字符（重音符号、元音变音符号、零宽连接符）只有和其他字符组合时才是有意义的。每个字符都有一个名字（如希腊大写字母ALPHA）以及很多属性，比如：是否是数字、是否是大写字母、表示方式是否是从右到左、是否是变音符等。

一个字符集或者字符指令系统则是另一种抽象：它是字符的无序集合。一个给定的字符或者属于、或者不属于一个给定的字符集。Unicode的目标是提供一种通用的字符集合——包含当前正在使用中的以及历史上曾经使用过的每一种书面语言的每一个字符——其标准也在不断地修改以使得它能够更接近该目标。

一个编码的字符集会为每个字符唯一指定一个整数作为这个字符的码点。一旦为字符分配了特定的码点，就可以通过数字来代表这些字符了。通常情况下，码点的描述是由一个大写的U、一个加号字符“+”以及一个十六进制数字组成。例如，本章之前提到的单引号字符的码点是U+2032。

编码的字符是按照它们所属的语系来组织的，而语系内部会进一步将各种强相关的字符组合在一起进而划分为多个分块。举个例子，拉丁文语系（很多欧洲语言都是属于该系）被划分成基础拉丁文（包含足够表示拉丁文和英文的字符）、Latin-1补码（包括一些特殊符号和一些控制符号的结合）、拉丁扩展A、拉丁扩展B等。

当需要真正地把文字显示在屏幕上时，计算机程序解释字符序列并使用一种字体来生成符合上下文所需要的顺序和位置的字形。

注15：参见*http://unicode.org*。

猜测语言并删除停用词

文本中包含很多“the”、“it”和“to”既不有趣也不会令人惊奇。为了避免Wordle变得让人感觉无聊，需要删除在每一种可识别的语言中包含的这些停用词。对于给定的文本，想要知道应该删除哪些停用词，我们首先需要猜测该文本是什么语言。

识别字体类型和识别语言不同，因为很多语言可能使用相同的字体（例如法语和意大利语，都是使用拉丁语字体）。

Wordle采用了一种直截了当的方式来猜测文本所属的语言：它从文本中选择50个最常见的单词，计算这些单词在每种语言的停用词列表中出现的次数。哪个停用词列表的计数值最高，就认为该文本的语言即为该停用词列表所属的语言。

如何创建一个停用词列表？如同之前所述的关于一个“单词”的定义，这种问题属于主观上的判断问题，而不是科学。通常情况下，首先对一个大语料库的所有单词进行计数，选择出现频率最高的单词。然而，你可能会发现某些高频词对输出结果起到良好的效果，而其他低频词看起来只是给结果增加噪音干扰，因此可能需要稍微调整一下停用词列表。

Wordle的很多停用词列表来自于用户的收集，他们希望Wordle能够更好地支持他们的语言。Wordle的Web站点对这些用户表达了谢意。

默认情况下，Wordle在下一步处理之前，会从单词列表中删除包含的选定语言的停用词，但Wordle用户也可以通过设置菜单复选框，来修改默认值的设置。

给单词分配权重

Wordle采用直截了当的方式为每个单词赋数值权重。其采用的公式是权重=单词计数。

布局

一旦你对文本进行了分析，结果就是一个单词列表，每个单词都有一个基于其在文本中的频率计算得到的数值权重。Wordle会对这些权值以任意尺度进行范化，这样就决定了影响结果图片的不同常数的尺度（如本章后面所述的层次边界框的最小尺寸）。你现在可以把文字转变成图形对象并把这些对象放到空间的某个位置。

把加权单词转换成图形

对于每个单词，Wordle构建了一种字体，其点大小和该单词缩放的权值相等，然后使用字体来生成Java2D图形（见例3-2）。

例3-2：如何把字符串转换成图形

```
private static final FontRenderContext FRC
    = new FontRenderContext(null, true, true);

public Shape generate(final Font font, final double weight, final String word,
                      final double orientation) {
    final Font sizedFont = font.deriveFont((float) weight);
    final char[] chars = word.toCharArray();
    final int direction = Bidi.requiresBidi(chars, 0, chars.length) ?
        Font.LAYOUT_RIGHT_TO_LEFT : Font.LAYOUT_LEFT_TO_RIGHT;
    final GlyphVector gv =
        sizedFont.layoutGlyphVector(FRC, chars, 0, chars.length, direction);
    Shape result = gv.getOutline();
    if (orientation != 0.0){
        result = AffineTransform.getRotateInstance(orientation)
                        .createTransformedShape(result);
    }
    return result;
}
```

展现区域

Wordle通过以下几种方式来估算最终的单词云所能覆盖的所有区域：检查每个单词的边界框，对区域面积求和，调整字体小的单词和边界大的单词的面积使得它们显示上更紧凑。结果区域和目标区域成一定比例（目标区域是根据Wordle的应用小程序在运算时的布局的长宽等维度的数值计算得到的）。

用于调整“展现区域”的常量和Wordle的布局所在的区域，是通过“久经考验”的传统方式实现，即应用不同的数值进行尝试，直到整体看起来“不错”且运行“良好”。正如实际情况所示，展现区域的精确的面积大小是至关重要的，因为区域边界在布局中是作为约束条件。如果你的展现区域面积太小，在该区域放置单词就会很慢，绝大多数单词会“出局”，留下一个圆圈（因为一旦一个单词不能被放置在展现区域中，Wordle系统将放宽约束条件，结果是所有单词都会被随机分布在一些初始位置）。如果展现区域面积太大，结果将是杂乱的一团（因为任何不存在交叠的位置都是可以接受的）。

需要特别注意的一点是，对于异常长的单词，它的某个维度的取值可能比计算其所基于的区域的宽度和高度的值都要大。你必须保证你的展现区域面积足够大，至少可以包含最长的单词。

请记住，展现区域只是一个抽象的空间，一个和像素、尺寸或者任何衡量尺度不相关的坐标系统。在这个抽象空间中，你可以对单词形状进行布局，并执行交叠检查。当需要真正地将像素放到屏幕上时，你还可以对屏幕单元进行缩放。

放置

为放置单词创建完展现区域后，需要把单词放置到该区域中。全局的放置策略是采用随机贪婪算法，其中单词是一次一个地放置在展现区域中。一旦放置了某个单词，它的位置就不会再改变。

Wordle为用户提供了不同的放置策略选项。这些策略可以通过确定每个单词“想去何方”的方式，来影响Wordle最终生成的图像的形状和结构。在Wordle网站上，可以有两种选择，即中心线和字母中心线。这两种策略都把单词放在展现区域的水平中心线附近（并不是严格地在该中心线上，而是随机分布分散在中心线附近）。字母策略以字母顺序对单词进行排序，然后沿着*X*坐标轴把单词分散放置在展现区域里。

通过使用更智能的放置策略可以产生有趣的效果。例如，给定聚类数据——包含哪些单词通常互相使用的信息，这种放置策略可以确保每个出现在该聚类中最后一个单词附近的单词都会被放置到该展现区域中（见图3-16）。

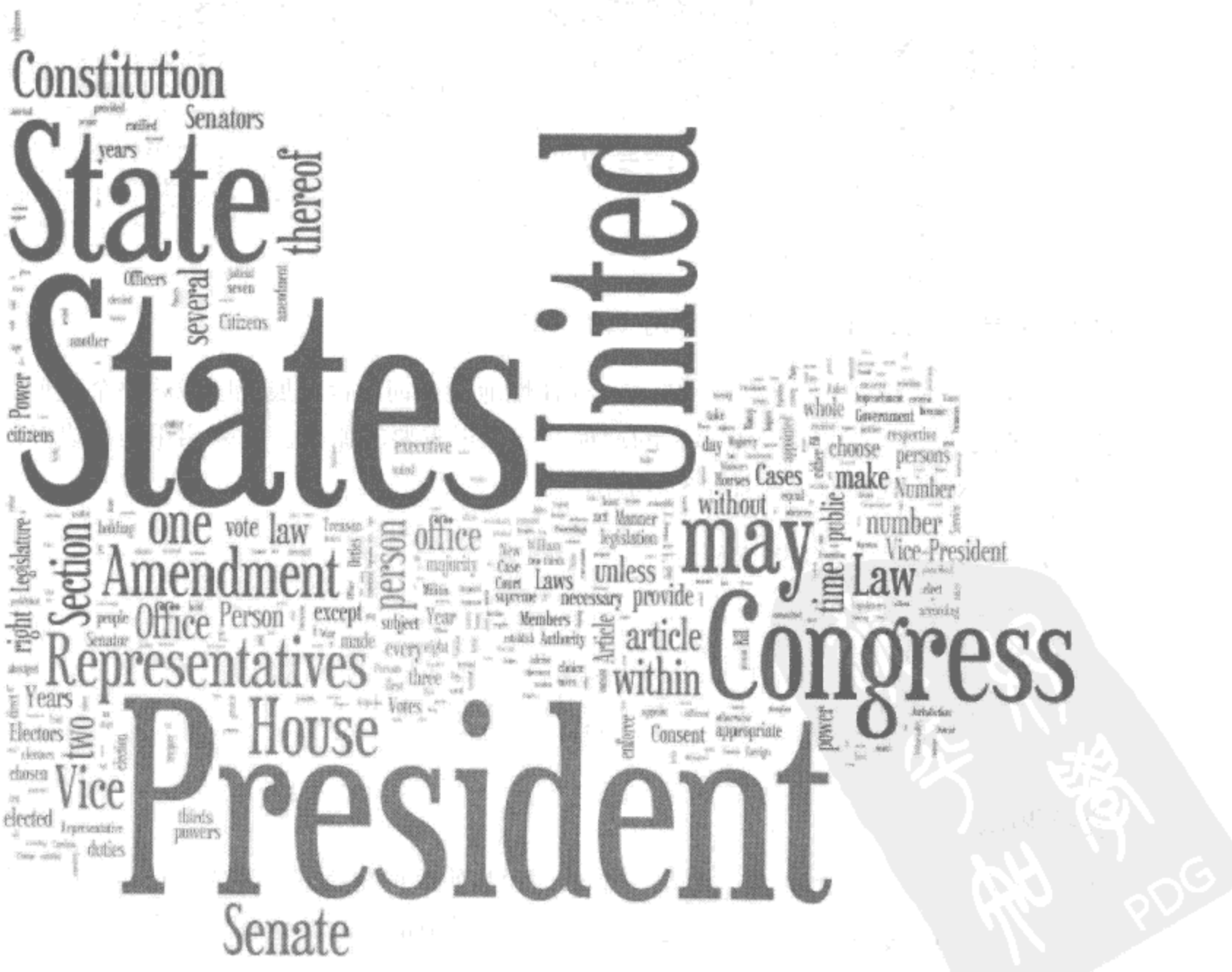


图3-16：聚类放置策略结果（见彩图18）

单词的形状是通过相应的权值降序排序。如例3-3所示的布局，其结果如图3-17所示。

例3-3：最后终于揭秘的Wordle算法

```
For each word w in sorted words:
    placementStrategy.place(w)
    while w intersects any previously placed words:
       move w a little bit along a spiral path
```

图3-17：单词“Denmark”所走过的路（见彩图19）

为了使事情变得更加复杂一点，Wordle选择性地使单词包含在展现区域的整个矩形边界内，这也是为什么要猜测整件事情的规模的一个很重要的原因。如果启用矩形边界约束条件，交叠处理规则看起来如例3-4所示。

例3-4：展现区域中的约束性单词

```
while w intersects any previously placed words:
   do {
      move w a little bit along a spiral path
   } while any part of w is outside the playing field and
           the spiral radius is still smallish
```

交叠测试

例3-4所示的伪代码清楚地表明如果一个单词和其他单词交叠就应该改变该单词的位置，但是它并没有指出如何确定单词是否交叠。基于插值的交叠测试代价很高，而简单地选

择几对值进行比较又是完全不可行的。以下是Wordle所采取的方法，它保证处理速度能够足够快：

层次边界框

第一步是减少测试两个单词是否交叠的成本。检测疏漏的一个简单的办法是比较两个单词的边界框是否交叠，但是却经常出现两个单词并没有交叠而其边界框却存在交叠的情况。Wordle充分利用了矩阵比较代价低的特性，它递归地把单词的边界框分成更小的矩形，生成一棵由矩形框生成的树，其叶子节点包含单词的形状分块（见图3-18）。虽然构建这样的层次边界框成本很高，但该成本在布局中得到了很大的降低。为了测试冲突，程序递归地处理相互重叠的矩形框，当存在两个叶子节点重叠或者当所有可能存在重叠的分支都被排除时程序就结束。通过处理最小尺寸的叶子矩形和对叶子矩形框进行稍微“膨胀”扩大，在布局上，单词边框之间就可以得到“免费”、让人舒心的边距。

图3-18：层次边界框

宽阶段冲突检测

在选择一对单词测试它们是否存在交叠时，最简单的方法是对当前的候选单词和所有已经置位的单词进行测试。这种冲突检测方法需要比较的次数为N^2，当你有200个左右的单词需要测试时，冲突检测速度就会过于缓慢。因此，Wordle采取了一些额外的措施来尽量避免冲突测试。

缓存

对以上冲突检测方法的一个简单有效的改进是基于如下的观察：如果单词A和单词B交叠，如果稍微调整A的位置，很有可能A还会和B交叠。因此，Wordle把和一个候选单词最经常交叠的单词缓存起来，首先测试这些经常交叠的单词。

空间索引

为了进一步减少冲突检测次数，Wordle采用了计算几何学中的“区域四叉树”（region quadtree）算法，它递归地把二维空间（在Wordle中，即展现区域）划分成4个矩形区域。在区域四叉树算法中，四叉树作为空间索引树，能够高效地把单词列表和其他候选项进行比较。一旦在展现区域中放置了某个单词，

Wordle就会搜索包含该单词的最小的四叉树节点。然后，当放置下一个单词时，就可以通过查询该四叉树，在交叠测试中排除很多已经置位的单词。

高效的冲突检测是一个很大的研究方向，在Christer Ericson的书《Real-Time Collision Detection》（2005年）对其中一些研究成果做了很好的综述。那些对类似于Wordle中所用到的图形算法感兴趣的人，我很推荐这本书；我自己对四叉树的实现也是基于该书对这种算法的论述。

Wordle是优秀的信息可视化吗

如果你认为Wordle是严格意义上的信息可视化工具，它在设计的某些方面存在误导或者干扰用户的潜在可能，有必要指出并加以批判。以下是我认为Wordle存在的不足之处。

单词大小调整太初级

Wordle在计算其字体大小时，并没有考虑单词的长度，或者它所绘制的文字的字形。其结果是，给定使用次数相同的两个单词，包含的字母越多的单词在屏幕上会占用更多的空间，这可能会给读者带来这样的印象：单词越长，其出现频度越高。

此外，据我所知，在关于单词大小和感知上的相对权重的关系上没有任何研究。更糟糕的是，常见的策略是把单词的权值根据其平方根进行缩放（为了体现单词拥有区域而不仅仅是长度这一事实），这只会使Wordle显得很无聊。

颜色毫无意义

在你的电脑屏幕的中央提供了最宝贵的几个维度，令人吃惊的是，Wordle在颜色使用上非常“散漫”。在Wordle中，颜色是毫无意义的；它仅仅是用来提高单词边界的对比度和增加一些美感。

颜色可以用于对各个维度进行编码，如聚集（意味着这些单词通常是一起使用的）或者统计学意义（如图3-19中的总统就职演说的单词云）。Wordle还可以使用颜色在同一空间表示两种或者更多不同的文本。

值得一提的是，Wordle并没有为色盲的读者做出什么特意安排，虽然人们总是可以通过应用小程序的色彩菜单栏创建一个定制的调色板。

图3-19：这个演讲使用了很多“Government”（政府）这个词，但是在其他演讲中该词用得更多；“pleasing”（愉快）只被用了几次，但是它在语料库中是一个不寻常的单词；“people”（人民）被用了很多，其频率在该演讲中非同寻常（见彩图20）

字体是使人充满遐想的

Wordle的很多字体都更倾向于美学和表现力，而不是可读性。这么做的原因一部分是因为Wordle的Web站点设计——如果缺乏形式多样的字体，画面将是单调的。最重要的是，Wordle中的字体必须看起来很优美，这意味着它不一定很适合于正文文本。

对于易读性至关重要的应用，Wordle提供了Ray Larabie的Expressway字体[译注4]，该字体被美国运输部作为标准字母。

字数计数不够具体

Wordle对《New Testament》[译注5]中的每一卷的页面中出现的“Lord”（上帝）这个单词的次数进行了求和，但是它没有提供任何关于各个章节的区别的信息。仅仅简单地对单词计数并不能对相似的文本做出有意义的比较。比如有一个博客帖子，突出该帖子和该博客的其他帖子的不同之处，或者说明它和其他博客的在同一主题上的区别，甚至是说明该帖子和新闻报道文章的用语的不同，这些方面的说明可能是最具有启迪性的。

存在很多统计学方法，可以应用于一篇“样本”文章，来基于一些“范文”的正文来抽取“样本”中的特定字符，尤其注意一些单词的使用在统计上是更重要的。除了单词出现频度，还可以对单词权重进行更细致深入地探析，然后应用Wordle布局算法来展示结果。

译注4：Ray Larabie是加拿大的一名字体设计学家。他创作提供了很多免费字体，Expressway是其中一种。

译注5：《新约全书》，共27卷。

在分析每个总统就职演说[注16]时，我都探索了这个想法，把每个演说都和当时最接近的5个演讲、10个最接近的演讲以及所有其他的就职演说进行了比较。这种分析的优点是可以揭示一些不可预见的单词。举个例子，图3-20是哈里·杜鲁门在1948年的就职演说的可视化。左侧是该就职演说中使用的单词的Wordle形式的展现，右侧是他那个时代的其他总统所使用次数更多的单词的展现。该可视化展现说明了杜鲁门的演说强调的是对外政策。

图3-20：哈里·杜鲁门在1948年的总统就职演说：和他同时代的其他总统就职演说相比，杜鲁门的演说当中很明显缺乏那些红色标注的单词（见彩图21）

注16：　参考*http://researchweb.watson.ibm.com/visual/inaugurals/*。

如何真正使用Wordle

Wordle不是为可视化专家、文本分析专家甚至是有经验的计算机用户而设计的。我试着把Wordle做得尽可能像个工具。

在撰写本章时，人们在Wordle画廊中已经创造、保存了超过140万的“单词云”。这些单词云被用于：总结和修饰商务演示和博士论文，插图说明新的文章和电视新闻报道，提炼和抽象受害者个人痛苦的回忆。Wrodle还发现形形色色的充满热情的教师社区，他们使用Wordle来展示拼写列表、总结话题以及促使不识字的青年参与到享受文本的乐趣中。

如表3-1的调查结果表明（Viégas、Wattenberg 和 Feinberg，2009），使用Wordle激发了人们的创造力，人们会觉得他们正在做创造性的事情。

表3-1：人们创造Wordle时的感受

	赞成%	中立%	不赞成%
激发我的创造力	88	9	4
我感到一种情绪反应	66	22	12
从文本中我学到了一些新的知识	63	24	13
它证实了我对文本的理解	57	33	10
它勾起了我的回忆	50	35	15
Wordle使我感到困惑	5	9	86

因此，通过对可视化效能应用传统的学术评估——“从文本中我学到了一些新的知识”——至少可以认为Wordle是比较成功的。但是Wordle真正闪光之处在于其交际作品的创作。使用Wordle的人们感觉他们似乎创造了一些东西，它成功地表示一些有意义的事物，并准确地反映或增强了源文本。这种意义看起来主要是直观的，因为很多人并没有意识到单词大小和单词频度是相关的（相反地，猜测该大小表示“情感重视”甚至是“单词意义”）。

Wordle的特性缘于文本的特性。只是简单地把一个单词放到屏幕上，其字体要么对单词本身的涵义进行补充，要么对其进行反衬，可以马上使读者产生共鸣（实际上，在公共画廊上保存了成千上万的单词）。当你把两个或者更多的单词并排展示时，一个有文化的人就会自然而然地去理解该序列化单词。Wordle对单词的随机组合给人们创造了喜悦、惊喜、某种程度的认可，以及如诗般激发了人们的洞察力。

为传统的信息可视化使用Wordle

Wordle的信息可视化分析用途当然可以为专业用户所用，更不用说Wordle所具备的特定的情感和交际特性。为了满足那些使用Wordle给“加权文本”创建可视化的用户，其权重不一定是基于单词的出现频度，Wordle的Web站点提供了“高级”用户界面，用户可以输入包含任意（可选）色彩的加权单词或短语的表格数据。

Wordle的更高级的使用方式可能是通过“单词云生成器”控制台应用程序，可以通过IBM的alphaWorks Web站点进行查看[注17]。

ManyEyes协作式数据可视化网站还把Wordle作为文本可视化选项，其他的还有创新型的Phrase Net和Word Tree可视化（以及更传统的标签云）[注18]。

结束语

人们通常希望保存和分享他们创作的Wordle；他们利用Wordle进行沟通。美丽的可视化在揭示事物的本质时，也给人们提供了乐趣。

致谢

我想要感谢IBM CUE的Martin Wattenberg和Irene Greif使我参与到本书的写作中。非常感谢Ben Fry、Katherine McVety、Fernanda Viégas和Martin Wattenberg，他们都非常认真地阅读了本章，并给出很多改进意见。关于那些帮助我们创建和改进Wordle的人员信息，请参阅*http://www.wordle.net/credits*。

参考文献

1. Ericson, Christer. 2005. *Real-Time Collision Detection*. San Francisco, CA: Morgan Kaufmann.

2. Millen, D. R., J. Feinberg, and B. Kerr. 2006. “Dogear: Social bookmarking in the enterprise.” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22–27, 2006). http://doi.acm.org/10.1145/1124772.1124792.

3. Viégas, Fernanda B., Martin Wattenberg, and Jonathan Feinberg. 2009. “Participatory visualization with Wordle.” IEEE Transactions on Visualization and Computer Graphics 15, no. 6 (Nov/Dec 2009): 1137–1144. doi:10.1109/TVCG.2009.171.

注17：参考*http://www.alphaworks.ibm.com/tech/wordcloud*。

注18：参考*http://manyeyes.alphaworks.ibm.com/manyeyes/page/Visualization_Options.html*。

第4章

色彩：数据可视化的“灰姑娘”

Michael Driscoll

避免带来灾难成为给信息增添色彩时的首要原则：
最重要的是，不要造成伤害。
——Edward Tufte，
《Envisioning Information》（Graphics出版社）

色彩是数据可视化中滥用和忽视最严重的工具之一：当我们做出不好的色彩选择时，我们滥用了它，而当我们依赖于功能很弱的软件默认值设置时，我们忽视了它。虽然历史上工程师和最终用户都没有用好色彩这个工具，然而如果能够善用它，它将是一个无与伦比的可视化工具。

绝大多数人在穿着亮红色的Underoos[译注1]出门前会三思而后行。要是我们在为资讯图像选择色彩时也能如此慎重就好了！其区别在于我们当中很少有人设计自己的衣服，而我们都需要修饰自己的资讯图像，使得色彩能够符合我们的目的（至少直到好的色彩板（如ColorBrewer）变得普遍起来）。

在思索如何实现Dataspora实验室的PitchFX观察仪的色彩时，我提出了一个基本的目标取向问题：为什么在数据图像中使用色彩？我们随后将探讨该问题。

为什么在数据图像中使用色彩

对于一个简单的数据集，单一色彩是足够的（甚至是更好的）。例如，图4-1显示了大联盟[译注2]棒球员Oscar Villarreal在2008年的287次投掷的散点图。只需要描述二维数据——

译注1：Underoos是一种内衣品牌，由Fruit of the Loom公司制造，其特征是花哨，充满性感和幻想。

译注2：美国职业棒球联赛中档次最高的一级。

“好球带（strike zone）”[译注3]的x轴和y轴坐标——黑白两色就足够了。实际上，这种散点图是数据集的无损表示（假定没有数据点完全重叠），也是其最佳的选择。

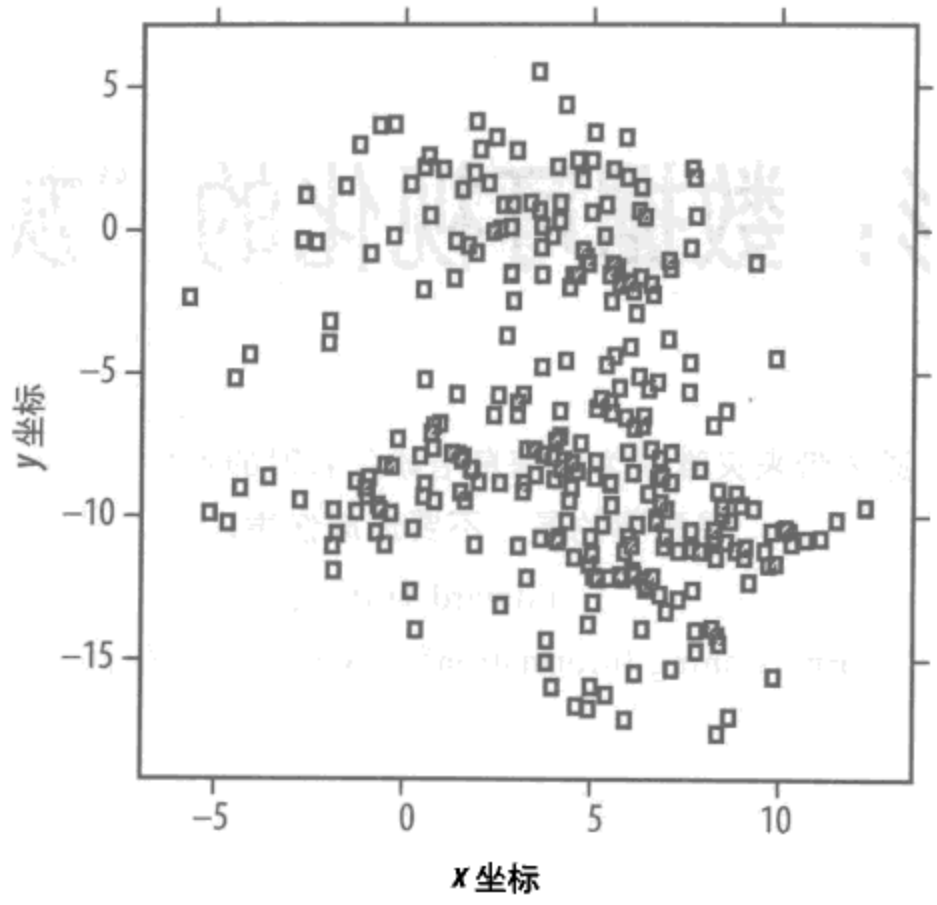


图4-1：使用x/y坐标平面图表示的投掷位置

但是如果我们希望了解更多，该怎么做？举个例子，不同的投掷（曲线球、快球）最后的落地点在哪里？它们的速度如何？可视化占用了两个维度，但是其所描述的现实世界的范畴却要宽泛得多。

数据可视化的典型挑战是把高维度的数据投影到低维度的画布上。通常来说，我们永远都不要把二者颠倒（对数据可视化生成比已有更多的维度）。

回到我们之前讨论的有关的棒球投掷的例子，如果想要对它增加一维数据——投掷类型——到汇总图中，我们可以通过以下几种方式来实现：

1. 绘图符号。可以改变我们所使用的图形（圆、矩形等）。
2. 小的多重图形。我们可以在空间上增加一些额外维度，创建一系列小的图形。
3. 色彩。我们可以对数据进行着色，在一个色彩空间内对额外的维度进行编码。

译注3：“好球带”指的是以棒球击球员之肩部上缘与球裤上缘之中间平行线作为上限，以膝盖下缘作为下限，通过本垒板上方的空间。

在可视化中你应该采用哪一种技术取决于数据的本质和展现的画布媒介。我将通过例子来描述这3种方法。

使用多种绘图符号

在图4-2中， 我通过使用不同的绘图符号，在绘图中增加了投掷类型的属性维度。

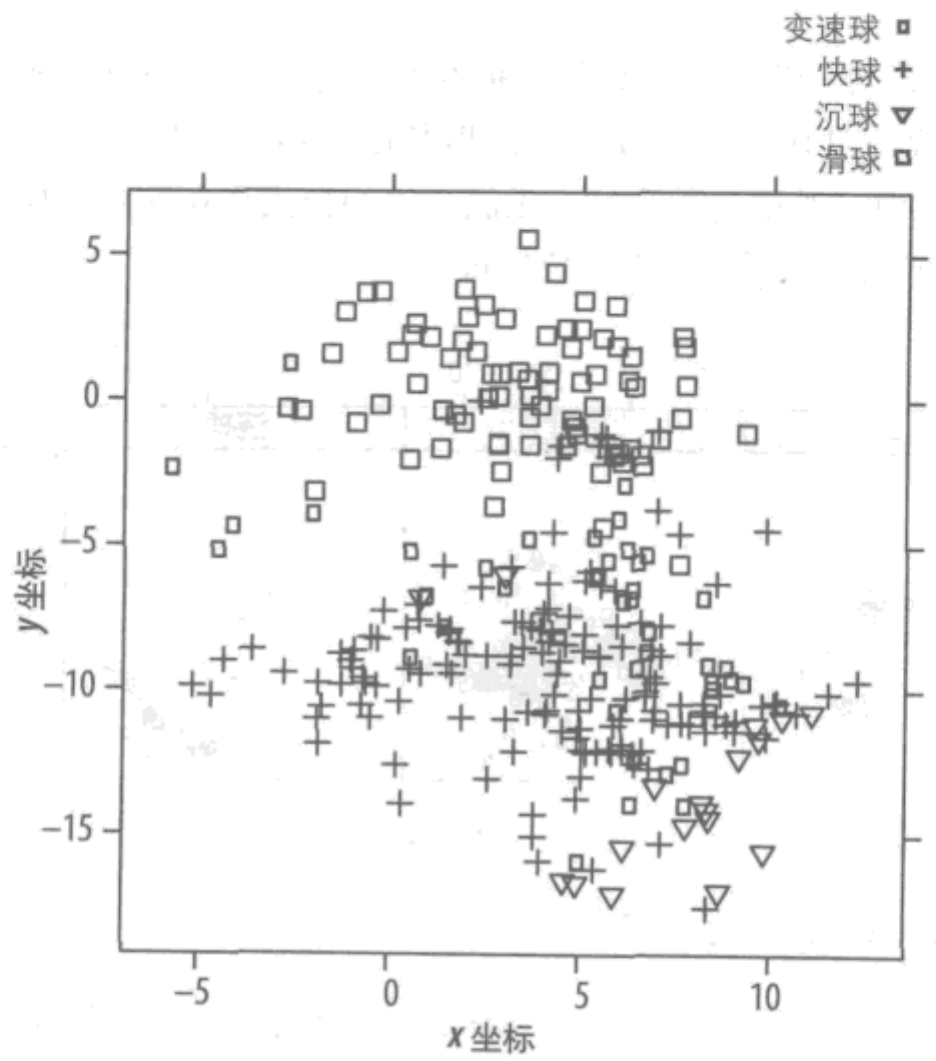


图4-2：绘图符号表示的位置和投掷类型

我认为该可视化是可耻的失败。有两个原因令我们对这类图形感到头痛：一是区别显著的图形需要分散我们额外的注意力（和学术上所谓的像色彩那样“前意识注意的（preattentively processed）”[译注4]暗示不同），二是即使我们对符号进行视觉解码，我们必须把这些符号和它们的语义属性进行匹配（诚然，通过切尔诺夫脸谱图（Chernoff faces）[译注5]或者其他符号标识，可以减少工作量，因为它们的属性映射是不证自明的）。

译注4： preattentive processing指的是在信息可视化中自动从整个可视化区域中识别出基本的特征。可以参考*http://www.infovis-wiki.net/index.php/Preattentive_processing*了解更多。

译注5： 切尔诺夫脸谱图是一种多元统计学表示方式，它以多元数据显示人脸，通过形状、大小、位置、方向各个变量来表示眼睛、耳朵、嘴巴、鼻子等。其思想是人们可以很容易识别人脸的微小变化。请参考*http://en.wikipedia.org/wiki/Chernoff_face*了解更多信息。

在画布上使用小的多重图形

虽然Edward Tufte已经做了很多工作来促进小的多重图形在信息图形中的应用，在分块化的画布中增加额外的维度是一款很优秀的方式。这种技术已经被应用于方方面面，从伽利略的“太阳黑子说明图”到William Cleveland的“网格图”。随着Scott McCloud因创建了令人惊喜的卡通漫画而变得人所皆知，连环画能够讲述故事，而这一能力是单一、整体的画布所缺乏的。

如图4-3所示，Oscar扔出的4种类型的投掷在水平方向上的分组。通过减少图像尺寸，我们降低了在位置信息显示上的分辨率。但是由此换来的是，在第一张图像中无法识别、在第二张图像中（通过多种符号）无法分辨的模式现在这张图像中开始变得清晰了（Oscar投掷的快球位置很低，而滑球位置很高）。

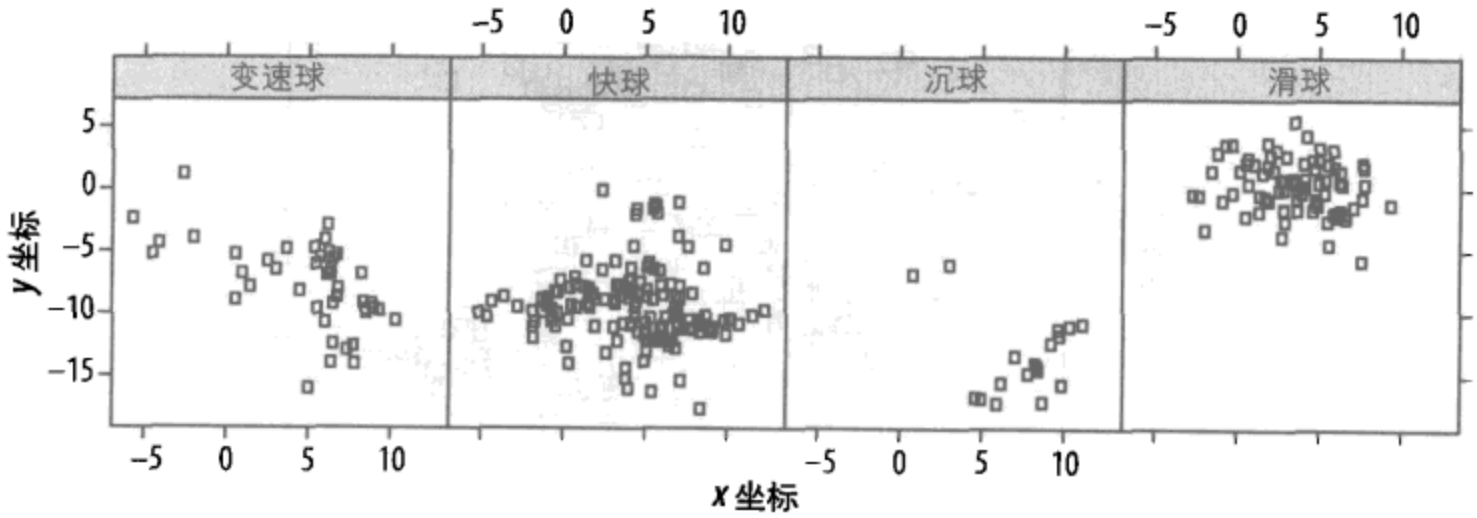


图4-3：通过切面显示的位置和投掷类型

在印刷媒介上，在空间上切分多重图片效果尤其显著，这种方式可以显示一个屏幕上每平方英寸所显示的点数的10倍。额外的图形还可以通过列和行的方式进行排列，作为散点图矩阵显示（请参阅统计工具R的splom函数[注1]）。

给数据增添色彩

在图4-4中，我使用了颜色对投掷数据的第四维进行编码：投掷的速度。我选择的色彩板是在Lab色彩空间[注2]中沿着一个维度变化的（可以把它想象成“红－蓝”维度），且同时能够维持恒定的亮度。

注1：关于统计工具R，你可以访问*http://www.r-project.org/*了解更多。

注2：参考*http://en.wikipedia.org/wiki/CIELUV_color_space*。

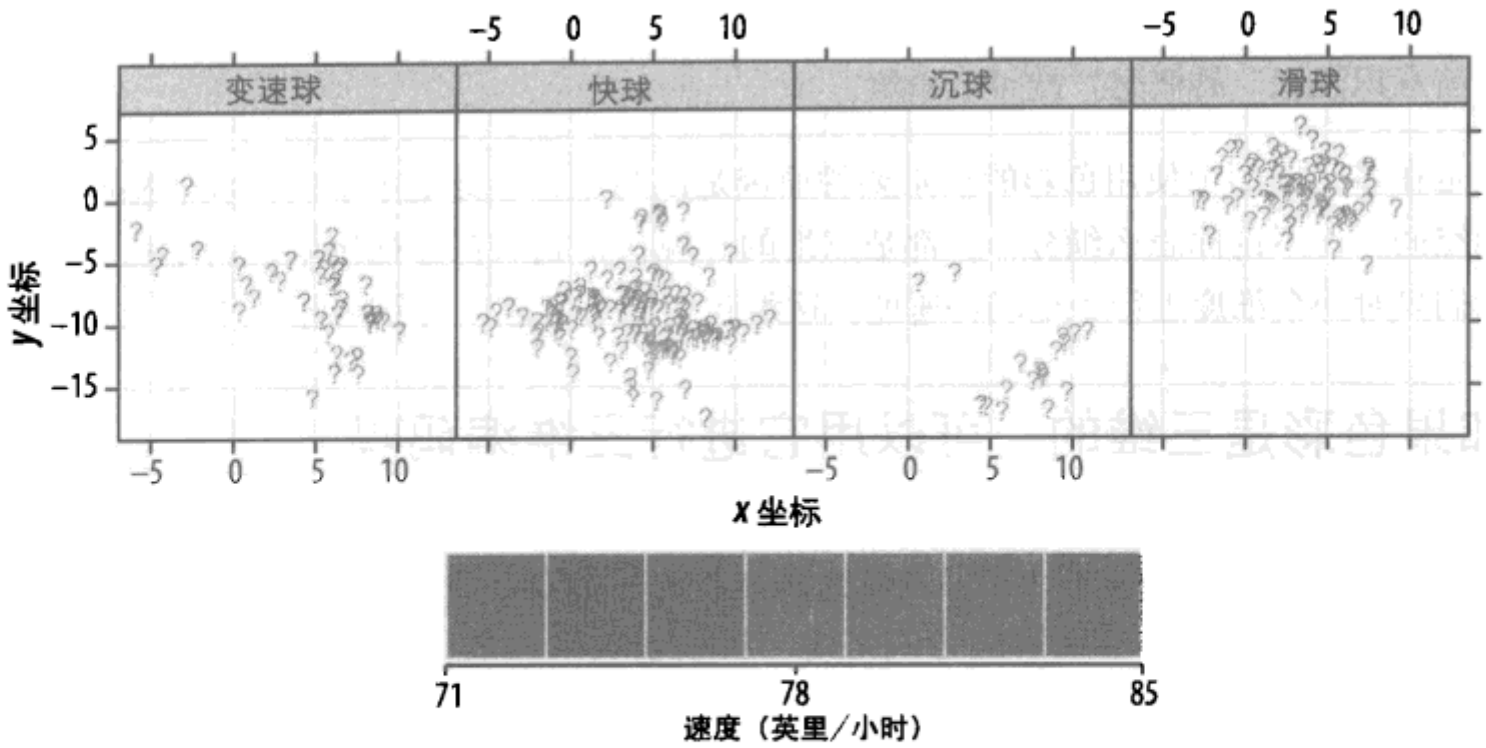


图4-4：位置和投掷类型，投掷速度是通过一维的色彩板来显示的（见彩图22）

一方面，维持恒定亮度有好处，因为亮度（luminosity）（类似于明亮度（brightness））决定了一种色彩给人们所带来的视觉影响。亮丽的色彩会突出显示，暗淡的色彩会显得模糊。采用亮度随色彩变换的色彩板会把人工选择的部分数据点的色彩作为艺术品展示。

另一方面，亮度和色调不同，亮度具有色调所不具备的内在次序特征，这一特征使得它适合于描述定量（而不是绝对）维度的数据。

因为在本章后面我将使用亮度对数据的另一维度进行编码，所以我决定在这里采用色调对速度进行编码；它足以达成我们的目标。我只选择7种色阶，因此（以有损方式）降低了对速度的采样频率。把色调板划分成过多的色阶会使我们难以辨别不同的色调。

在该版本的编码显示中，不同于所有先前绘图所用的空心圆圈，我还选择使用实心圆圈作为画图符。这种编码显示通过色彩改进了对每次投掷的速度的直观可视化：面积小的色彩块看起来不明显。然而，这种选择把投掷速度的可视化展现和一系列更小的图片组合在一起，其结果是存在更多的数据点重叠。为此，我们进一步降低了一些位置信息的分辨率（我们很快将试着恢复其中一些信息）。

为什么要使用颜色

和绝大多数的印刷媒介相比，电脑的显示空间更小，但是其能够显示的色阶范围更广。因此，丰富的色彩是电脑在显示上的很大优势。

对于多维数据，色彩可以表达单位空间内额外的维度，而且可以即时达到这种效果。

颜色差异可以在200毫秒内被检测到，甚至在你注意到它的变化之前（即我之前提到的“前意识注意”的概念）就可以检测。

但是在多元图形中使用色彩的最重要的原因是因为色彩本身是多维的。我们感官上的色彩空间——不论你怎么细分——都是三维的。现在，我们在可视化中引入了色彩，但是我们只对一个维度进行了编码：速度。这给我们带来了另外一个问题。

如果色彩是三维的，可以用它进行三维编码吗

理论上，答案是肯定的——Colin Ware（2000年）曾经使用红色、蓝色和绿色作为三维坐标轴研究了这个问题。（我们将很快看到其他有用的色谱划分方式。）然而，该研究在实践上却很困难。最终解决方式是请一批观察员来评估“红色”、“蓝色”和“绿色”的点的数量并进行展示，但是这种方式很不直观。

另一个复杂的因素是有某种色盲（也称为双色盲（dichromacy），一种不同于正常的三原色盲(trichromacy)的色盲类型）的人数占的比例不低。它可以有效地把对色彩的感知减少到两个维度。

最后，事实上我们对所有维度的色彩的感知不是等同的：有的对黄色感知力比较弱，而有的对蓝色感知力比较弱。我们认为紧密相关的“红”和“绿”接收器是通过复制单一长波的接收器而产生的（据载，这种方法对于检测水果是否成熟很有用）。

因为色盲人口在整个人口中的比例很高，而且对三维色彩进行编码挑战很大，我相信如果使用色彩对数据编码，数据的维度最好不要超过两个。

亮度作为恢复局部密度的方法

作为对投掷数据可视化的最后一次迭代，如图4-5所示，我将介绍使用亮度对局部数据点的密度进行编码的方法。运用这种方法，我们可以通过增加绘图符号的大小的方式来恢复一些损失的数据。

这里，我们有效地运用了二维色彩板，其中蓝色和红色沿着一条坐标轴来表示速度变化，亮度沿着另一条坐标轴来表示局部密度的变化。正如“方法”一节中所详细描述的，这些绘图是通过使用统计工具R的“色彩空间包”（color space package）来创建的，该包提供了在任何一个主色彩空间（RGB、HSV、Lab）中指定颜色的功能。因为Lab色彩空间的颜色变化和亮度无关，我选择该色彩空间来创建这个特定的二维色彩板。

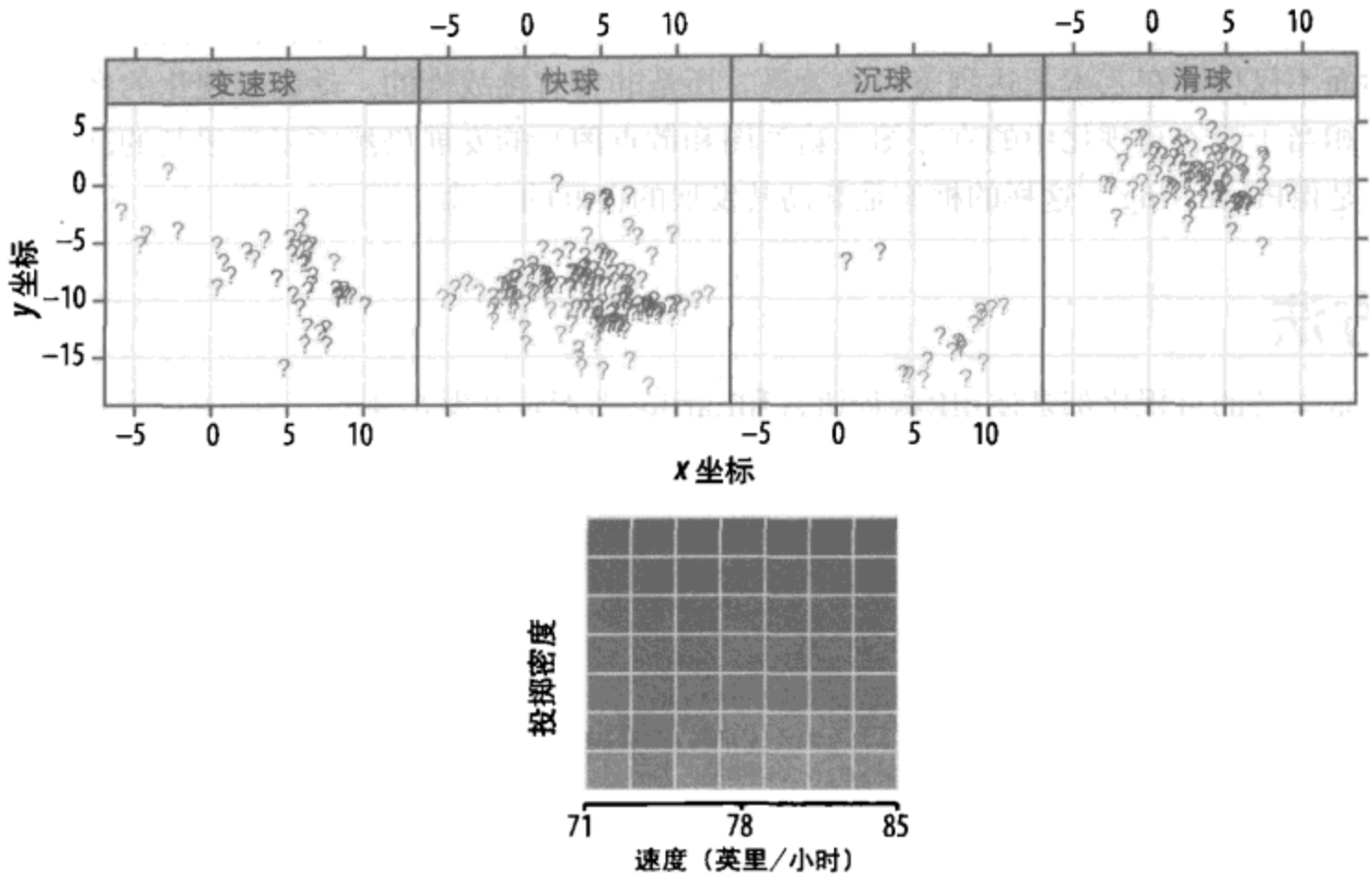


图4-5：位置和投掷类型，通过二维色彩板表示投掷速度和局部密度（见彩图23）

关于亮度的最后一点是在数据可视化中观测多种不同颜色涉及编程学上的“重载”。也就是说，我们依赖于认知函数，这些函数为了某个应用而开发（如展示lions），而实际中可以用于其他应用中（如展示lines）[译注6]。

我们可以对颜色以任何方式进行重载，但是只要可能，我们还是应该选择自然的展现方式。用亮度表示投掷密度给人感觉很自然，因为在投掷绘图中颜色越深意味着投掷越远。类似地，当通过色彩空间进行抽样时，我们不妨选择自然界中真实的颜色来表示。自然界中存在了人们已经通过肉眼凝视了几百万年的“色彩板”，它远远早在出现RGB色彩空间之前就已经存在了。

展望未来：关于动画

本章讨论的重点是使用一般的静态图形，尤其是使用色彩作为多元数据可视化方法。我有意地忽略了数据中另一个非常强大的维度：时间。时间可以把图像变成动画，从而把几个数量级的信息量都纳入可视化中（一个震撼人心的例子是Aaron Koblin对美国和加拿大的飞行模式所做的可视化，在第6章中将会深入探讨）。但是把这些信息植入随

译注6： lions和lines只差一个字母，可以利用“重载”，使用同一色彩空间进行展示。

时间变化的数据结构之中需要付出很多努力，而且让数据以动画这种富信息化的方式展示而不仅仅是在艺术上达到美观的效果，还是非常有挑战性的。动画可视化的经典方式（相当于静态可视化中的直方图、箱型图和散点图）的发展仍然需要有很长的路要走，但是像Processing[注3]这样的框架是帮助其发展的良好的开端。

方法

本章谈及的可视化都是使用R编程语言和Lattice图形包开发实现的。通过R语言构建二维色彩板的代码如下所示：

```
## colorPalette.R
## builds an (m x n) 2D palette
## by mixing 2 hues (col1, col2)
## and across two luminosities (lum1,lum2)
## returns a matrix of the hex RGB values
makePalette <- function(col1,col2,lum1,lum2,m,n,...) {
   C <- matrix(data=NA,ncol=m,nrow=n)
   alpha <- seq(0,1,length.out=m)
   ## for each luminosity level (rows)
   lum <- seq(lum1,lum2,length.out = n)
   for (i in 1:n) {
       c1 <- LAB(lum[i], coords(col1)[2], coords(col1)[3])
       c2 <- LAB(lum[i], coords(col2)[2], coords(col2)[3])
       ## for each mixture level (columns)
       for (j in 1:m) {
           c <- mixcolor(alpha[j],c1,c2)
           hexc <- hex(c,fixup=TRUE)
           C[i,j] <- hexc
       }
   }
   return(C)
}

## plot a vector or matrix of RGB colors
plotPalette <- function(C,...) {
   if (!is.matrix(C)) {
       n <- 1
       C <- t(matrix(data=C))
   } else {
       n <- dim(C)[1]
   }
   plot(0, 0, type="n", xlim = c(0, 1), ylim = c(0, n), axes = FALSE,
           mar=c(0,0,0,0),...)

   ## helper function for plotting rectangles
   plotRectangle <- function(col, ybot=0, ytop=1, border = "light gray") {
       n <- length(col)
       rect(0:(n-1)/n, ybot, 1:n/n, ytop, col=col, border=border, mar=c(0,0,0,0))
```

注3：　参见*http://processing.org*。

```
    }

    for (i in 1:n) {
        plotRectangle(C[i,], ybot=i-1, ytop=i)
    }
}

## Let's put it all together.
## We make two colors in the LAB space, and then plot a 2D palette
## going from 60 to 25 luminosity values.
library(colorspace)
lightRed <- LAB(50,48,48)
lightBlue <- LAB(50,-48,-48)
C <- makePalette(col1=lightBlue, col2=lightRed, lum1=60, lum2=25, m=7, n=7)
plotPalette(C, xlab='speed', ylab='density'
```

结束语

正如本章给出的例子所展示的，色彩（如果可以慎重、负责地使用）在对高维度数据进行可视化时可以作为一个非常宝贵的工具被使用。其最终产品——对2008年赛季的所有数据的五维投掷图——可以通过由PitchFX Django驱动的Web工具，在Dataspora实验室进行深入探索（*http://labs.dataspora.com/gameday/*）。

参考文献和补充阅读

1. Few, Stephen. 2006. *Information Dashboard Design*, Chapter 4. Sebastopol, CA: O' Reilly Media.
2. Ihaka, Ross. Lectures 12–14 on Information Visualization. Department of Statistics, University of Auckland. *http://www.stat.auckland.ac.nz/~ihaka/120/lectures.html*.
3. Sarkar, Deepayan. 2008. *Lattice: Multivariate Data Visualization with R*. New York: Springer-Verlag.
4. Tufte, Edward. 2001. *Envisioning Information*, Chapter 4. Cheshire, CT: Graphics Press.
5. Ware, Colin. 2000. *Information Visualization*, Chapter 4. San Francisco, CA: Morgan Kaufmann.

第5章

信息映射：重新设计纽约地铁图

Eddie Jabbour（*Julie Steele* 执笔）

地图是已有的最基本的数据可视化中的一种，我们已经有几千年的地图制作历史。然而，我们并没有把地图作为理解复杂系统的一种工具并加以完善——拥有26条地铁线、468个站点并覆盖了5个市区的纽约地铁系统，毋庸置疑是相当复杂的。KickMap是我为了设计一种更为有效的地铁图所做的探索的成果，其最终的目标是增加乘坐地铁的人次。

需要更好的工具

我出生在纽约的皇后区（Queens），在布鲁克林区（Brooklyn）长大。我看到的第一张地铁图是我父亲的，时间大约在1960年。它给我留下了深刻的印象，因为它当时吓到了我。通过该地铁图，我看到的是一个灰色的纽约，红色、绿色和黑色线条纵横交错，看起来像一个网格，如图5-1所示，而且地图上面还有数以百计的站点名字[注1]。它让我想起了一张自己无法理解的复杂无比的电路图；它看上去带着一股“成年人的肃穆”，甚至有点恐怖。我希望自己永远都不要和它打交道。

注1：我现在知道该地图是Salomon设计的地图的早期版本。多年以后，当我为创作KickMap调研时，我应该感谢这张地图体现的设计之美。

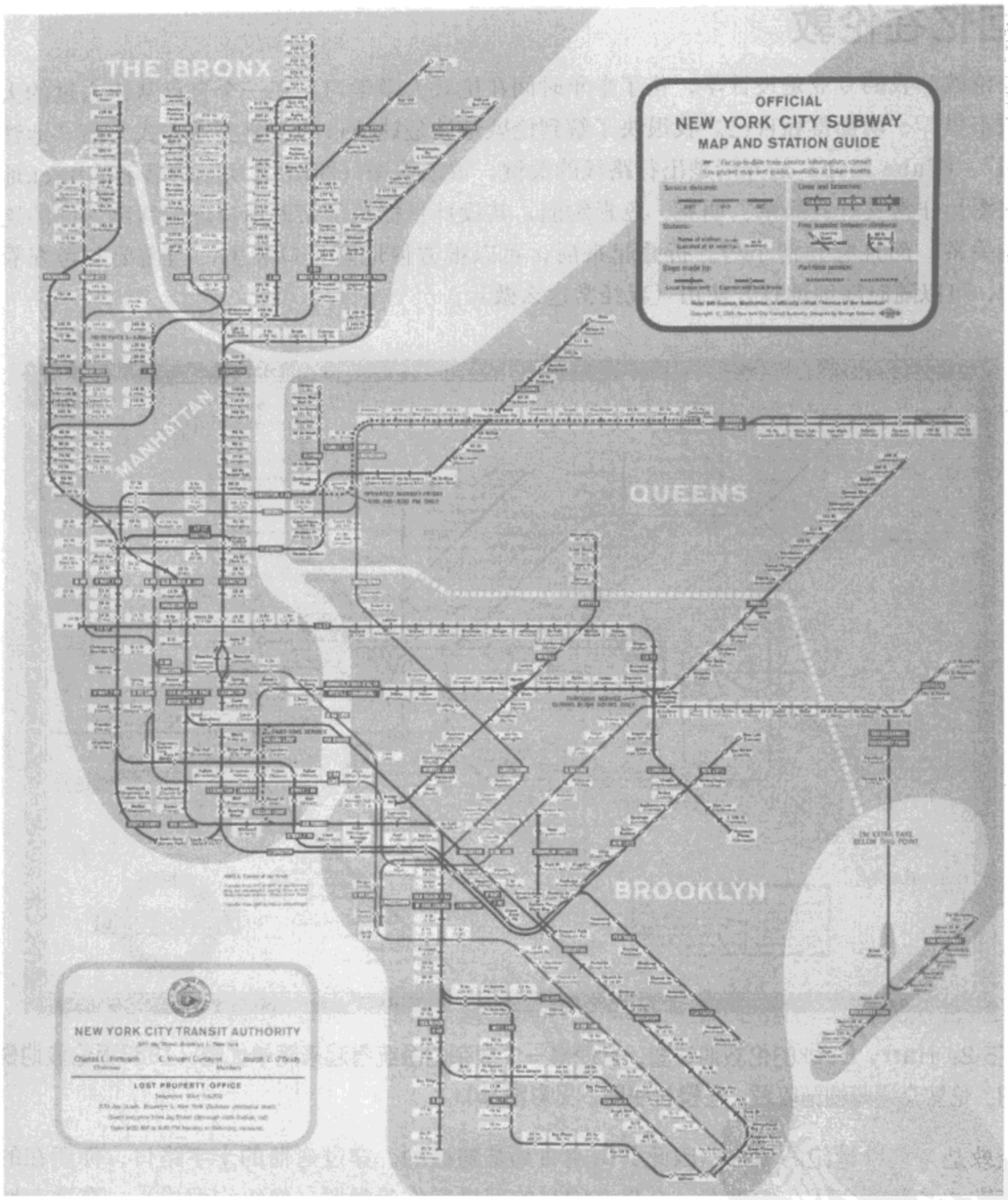


图5-1：由George Salomon设计的1958版的纽约地铁图（1958版纽约地铁图。MTA纽约城市运输图。已授权使用，见彩图24）

回忆在伦敦

在伦敦，我的专业是设计学，花了半年时间在伦敦大学学习。在一个自己从未去过的大城市里，一切都得靠自己。我很快了解到伦敦地铁是该城市的主要交通方式，而“地铁图”（Tube map）是弄清楚出行路线的关键。该地铁图（即图5-2所示的著名的Beck地铁图）非常友好：简单、明亮、色彩绚丽，其设计目标在于帮助用户理解线路之间的连接关系，而且它非常小巧。折叠起来后，可以很方便地塞到口袋里，当需要参考查看时，可以随时随地地打开使用（我经常这么做）。

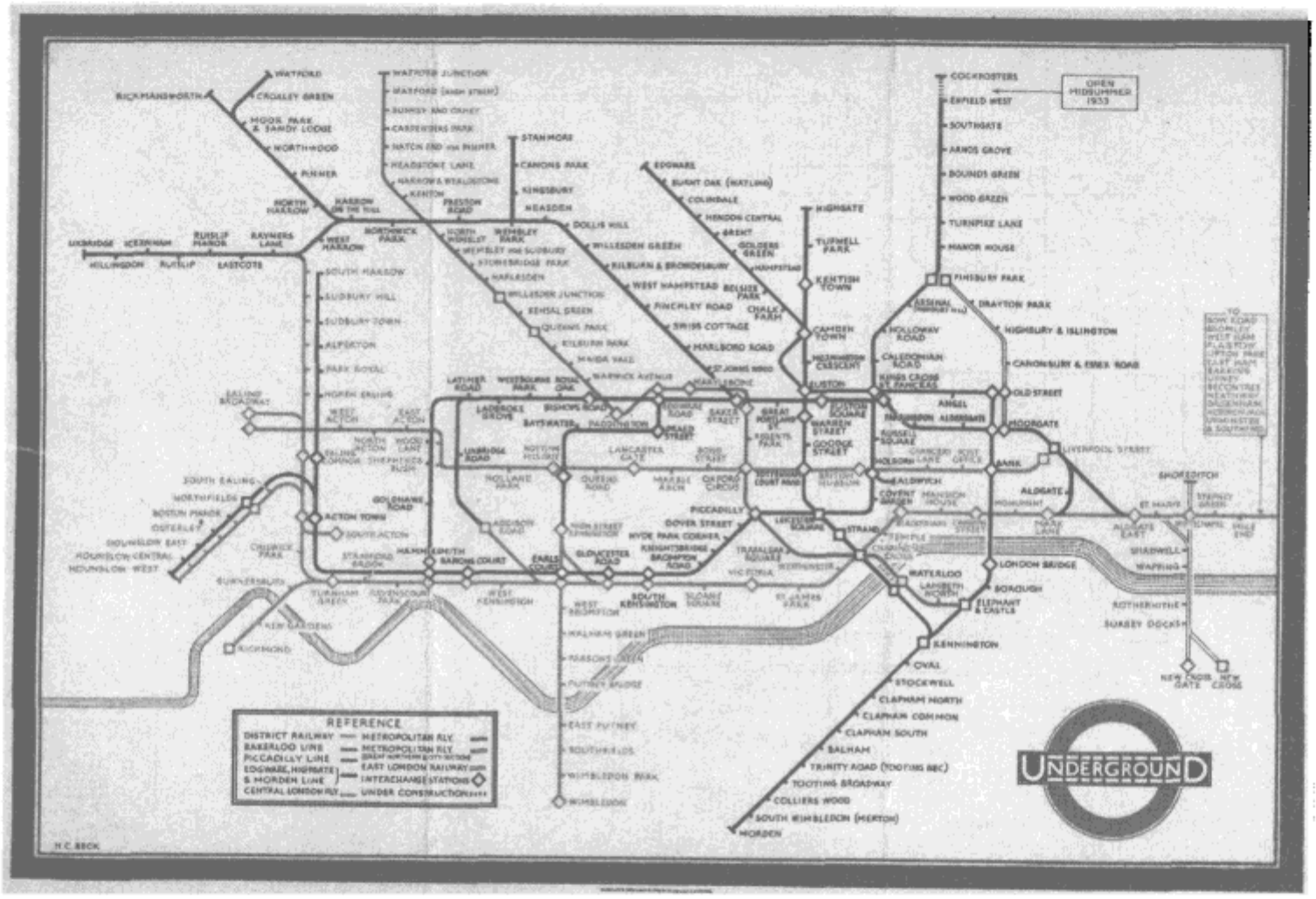


图5-2：Harry Beck的伦敦地铁图，它使得一个复杂的系统看起来简单优雅（1933版伦敦地铁图。伦敦交通博物馆收藏。已授权使用，见彩图25）

伦敦是一座中世纪的城市，因此其街道布局是随机的。穿过弯曲的十字路口，你所在的街道的名称就变了。它没有一个带有编号的网格来作为参照（像纽约那样），在这个城市中行走很容易迷失方向。Beck地图的天才之处在于它摆脱了随机复杂性，以泰晤士河作为地上可视化（和地理）的唯一参考点。基于这个原因，地图的布局是标志性的：当你想到伦敦，你很可能会想到地铁图。但是即使像我这样一个设计专业学学生，在那时也并没有对该地图的形式风格做进一步思考——它使用起来是如此简单方便，让人感觉出门旅行毫不费工夫。

有了这张小而有效的地图，以及“想去哪就去哪”、可以无限次使用的地铁月卡，我每天都可以乘地铁在伦敦穿梭。我轻松自在地去任何地方，充分利用伦敦这座伟大的城市中的资源。伦敦地铁图如此快速、清晰地传递信息，成为我的经历中不可分割的工具和组成部分。它使得我在伦敦仅仅生活了几周之后，就有这样的感觉：伦敦是“我的”。多么奇妙、强大的感觉！

事实上，我对这个宝贵的工具如此 “依恋”，在我逗留的最后时期，即离开这座城市之前，我去当地的地铁站买了一张新的地铁图，并在我回到纽约后把它装裱了起来。

纽约之“殇”

当你离开6个月重新回到家乡时，你会用新的眼光看待一切。当我回到纽约时，看到了纽约地铁图——真正地看到它——从我长大至今第一次看到它。我暗自思量，和伦敦的相比，纽约地铁图的设计很糟糕。

记得当时我对于纽约地铁图的看法刚好是Beck地图的反面：尺寸太大，看起来杂乱无章，而且非常不直观。我意识到这张地图在很多方面成为了使用我们伟大的纽约地铁系统的障碍，这和伦敦地铁图刚好相反。伦敦地铁图的简单性是理解和使用伦敦地铁的“金钥匙”。

然而，即使是作为一名设计师，即使曾在一念之间想要创建自己的地铁图，我肯定也很快地打消了这个念头。当时是在20世纪70年代，我不是那种拿着丁字尺的建筑师。对于任何非经验丰富的制图员来说，在那个没有计算机的年代，承担这种任务需要经过的训练和付出的时间都是不可想象的。

在我的设计生涯中，纽约地铁图的不足一直驻留在我脑海里。和绝大多数纽约人一样，我很少使用地铁图，也从来不带它。其部分原因是它太大了：和一个用做插页的公路线路图一样大。万一我需要借助该地铁图的一些信息去某个新的地方，我会从车站的免费地铁图中撕出一块六英寸大小的方形图，然后把剩余部分扔到垃圾箱中！我经常看到一些旅客很辛苦地携带着这张庞大的地铁图，并很为他们感到难过，这也使我回想起了自己学生时代在伦敦的美好经历。

好的工具衍生更好的工具

现在，“快进”到多年后的某个晚上，我带一个外地客户在市中心的一家餐馆吃饭。在我们等候地铁时，他私下告诉我纽约地铁“吓坏”了他。我很惊讶：20世纪70年代到90年代的犯罪现在已经从地铁系统消失了，我还对我们崭新的空调车和清洁的地铁站感到

自豪。但是，随着我们沿着市中心的路进行交谈，我意识到他的恐惧在于不能理解系统的复杂性：所有的线路和连接。那个时候，我意识到他的问题也是由于该地铁图设计得不够友好的缘故。这位客户经常旅行而且文质彬彬；如果他认为纽约地铁系统很吓人，那真正的原因是该系统的地铁图在交流上存在问题。

在那个时刻，这张地铁图重新潜入我的意识中，就再也没有离开。

那是2002年，我有了自己的设计机构和员工，我们每个人的电脑里都安装了当时最伟大、最优雅的图形设计工具。我意识到在现在这个时期，使用像Adobe Illustrator这样的图形设计程序，只需要一个人就可以创建属于他自己的地铁图！于是，我挑战自己重新设计纽约地铁图。

尺寸只是一个因素

当我决定利用周末试着动手做一个新的地铁图时，我考虑的第一个问题就是尺寸。因为纽约地铁系统的站点数几乎是伦敦的两倍，我决定采用两倍的伦敦地铁图的大小空间来制作纽约地铁图。（即使把伦敦地铁图的大小加倍，结果也只不过是现有纽约地铁图大小的五分之一。）

首先，我采用官方的城市交通管理局（Metropolitan Transit Authority，MTA）的纸质地图，如图5-3所示，用剪刀裁剪，然后以更有效的方式重新粘贴起来（一块块地用透明胶带粘起来），只是为了看看会产生什么效果。当我成功地裁掉原来地铁图的一半大小时，我觉得很受鼓舞。56个巴士的弹出框和其他非地铁信息都消失了！然后是创建一张实际的地铁图这项繁重的工作。我把所有的地铁站名字和线路都输入到Illustrator的文档工具中。两个月之后，瞧！我有了一张自己的、小得多的地图！我把地图折叠起来，很轻松地把它放到钱包里，带着它，并展示给所有的朋友。他们对大小很满意，但是没有人真正愿意使用它，因为它还存在很多设计上的问题，使得它难以使用。

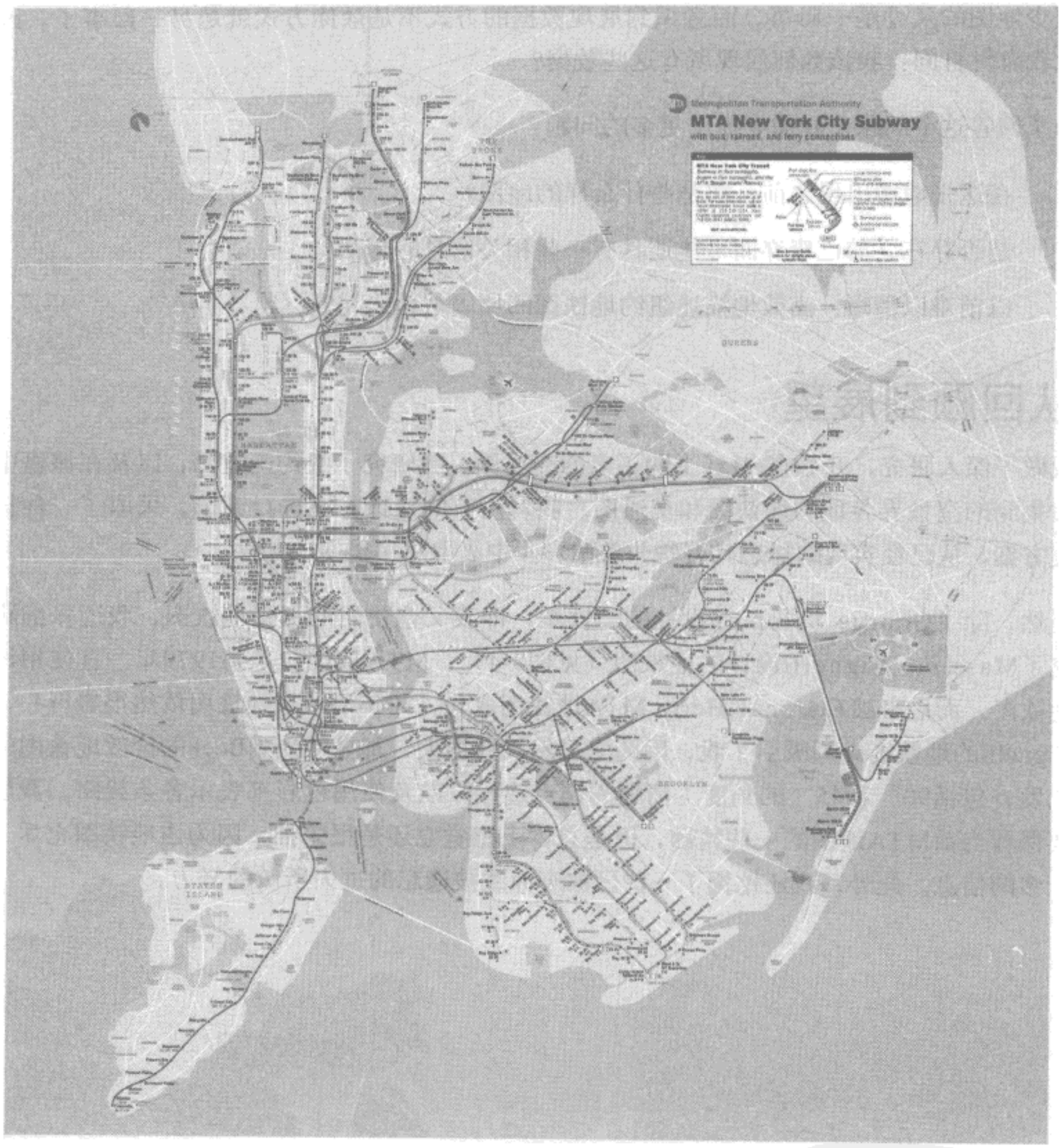


图5-3：MTA纽约城市地铁图的2004版，基于Michael Hertz的设计。除了其视觉上的复杂性，地铁图本身缺失的、不完整的信息使得用户不得不依赖于右下角复杂的图形说明（而在地铁里，坐在座位上的人们刚好挡住了这些信息）。但是，在地铁站，该信息展示在大海报上，也难以阅读，因为它离地面的高度往往小于18英寸（纽约城市地铁图。城市交通管理局收藏。已授权使用，见彩图26）

减少地图的尺寸是一回事，而意识到展现数据的方式不是最佳方式就是另一回事了。因此我自忖自问：我该如何展现所有这些数据？

为了回答这个问题，我需要提出更多的问题：

- 在这张地图出现之前，都是些什么样的地图？
- 是否存在之前已废弃而可能还具有一些相关信息的想法？
- 以前难以清晰、高效地描述纽约地铁图的原因是什么呢？

从回顾到展望

我做了深入研究，开始在eBay上购买老的交通图。我研究了纽约街道图，以及在旅途中收集到的全世界各地的地铁图和交通图。我对所有的设计方案进行筛选，采取了一种折衷方案，从已经实现的思想（有些非常精彩）中汲取尽可能多的想法。

当然，除了George Salomon设计的地铁图，即我父亲使用的那张地铁图，我还仔细研究了Massimo Vignelli设计的地铁图（见图5-4），MTA从1972年到1979年一直使用该地铁图，而后来被Tauranac-Hertz MTA地铁图取代（30年后，该地铁图依然很盛行）。Vignelli的地铁图立刻吸引了我，因为它虽然尺寸很大，却显然受到Beck的伦敦地铁图的启发，包括90°和45°的角度，清晰的站点连接，以及使用色彩来表示各条线路。我想要保留当前MTA地图的一些精髓，但是总体上感觉它还是很笨拙，因为该地铁图充斥了太多的信息。此外，我还挖掘了一些已被废弃或被遗忘的过去所做的努力。

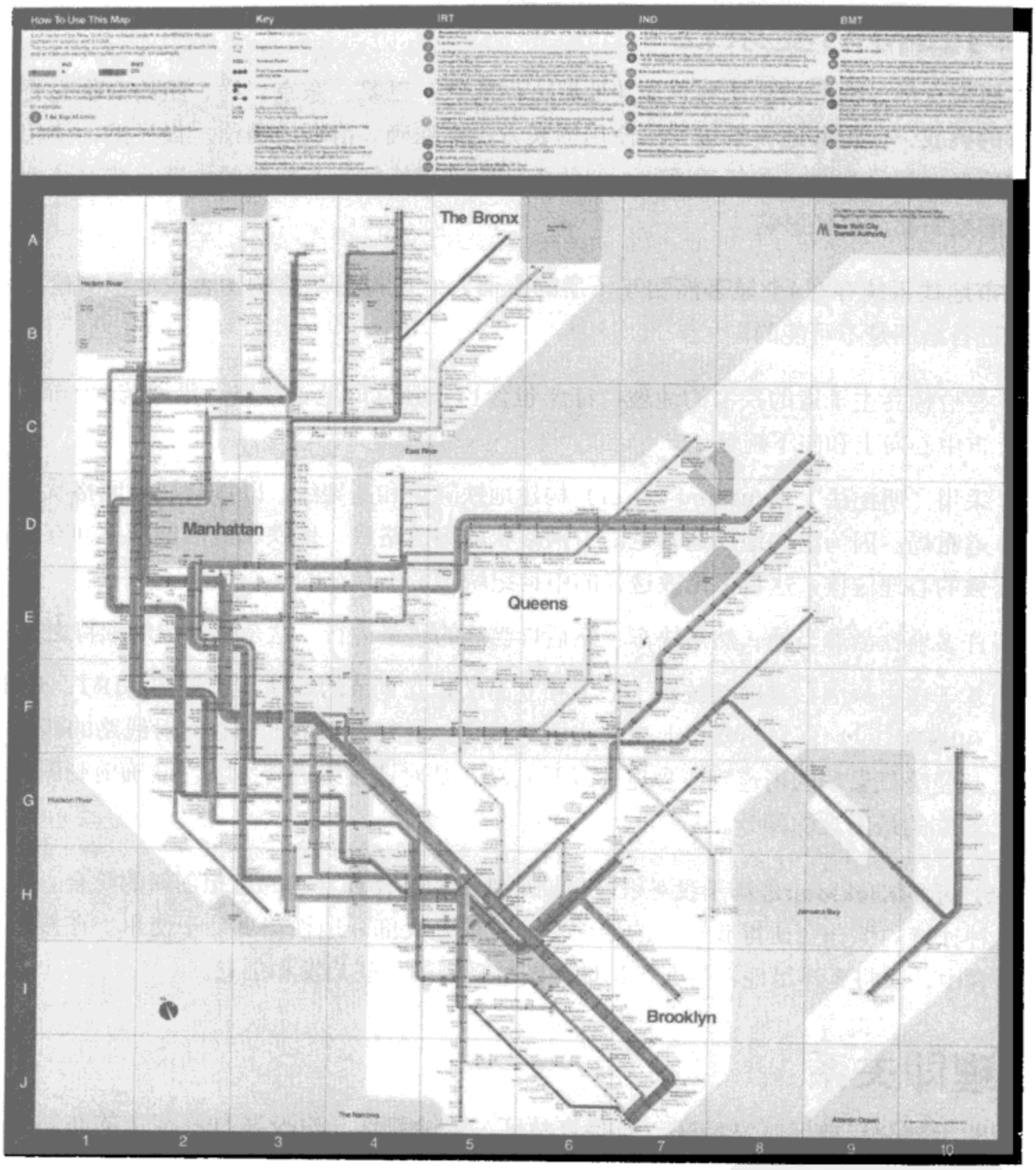


图5-4：Massimo Vignelli设计的1972版MTA纽约市地铁图。该风格在地理上扭曲得很混乱，但是它是设计上的一个为人称道的里程碑（1972版纽约市地铁图。MTA纽约市地铁收藏，已授权使用，见彩图27）

纽约独特的复杂性

深入研究之后，我开始意识到正如其他城市如伦敦、巴黎、东京一样，纽约面临它自己独特的挑战，使得其地铁系统无法使用图表方式来准确、清晰地描绘。很显然，使用纯粹的地形图测绘方法也是行不通的；纽约独特的地理特征及其网格状街道系统对其地铁系统的绘图都产生了影响。

纽约市地铁系统存在4个显著而相互矛盾的方面，它使得用严格的图表或地形测绘来成功地进行绘图是不可能的。

- 曼哈顿岛主干道的狭窄的地理特征，包含17条独立的地铁线路，沿着跨6个市区的市中心向上和向下蜿蜒。
- 采用“明挖法”（cut and cover）构建地铁隧道和高架线，以符合城市网格状的街道布局。因为纽约地铁通常是沿着网格状的街道路线，地铁和地面地形之间存在很强的心理链接，这在像伦敦这样的中世纪城市是不存在的。
- 许多地铁沿着当地、然后快车、然后再当地的线路运行，这是该系统的独特之处。
- 基于历史原因，当前系统源于三大独立而又相互竞争的地铁系统机构（IRT、BMT和IND[译注1]），它们整体上相互协调得不好。（三大竞争机构之间对线路的纠缠，从曼哈顿繁华的街道到布鲁克林以及长岛，是对该系统进行清晰、准确地制图时所面临的最困难的部分。）

图5-5所示的KickMap是基于我对很多早期地铁图的选择和自己的思想创新的综合。我相信这种独特的综合会使得我设计的地铁图比先前绝大部分地铁图更易于使用。在接下来的内容中，我将更详尽地探讨在设计地铁图时受到的一些启发和创意。

地理即关系

纽约的大部分行政区（皇后区、布鲁克林区、曼哈顿区，以及某种程度上的布朗克斯区）都已经由于城市街道的规划方式已经在地铁系统上存在网格。这使得地面上的地理不仅仅是一个直观的起始点，而且也是用户体验的一个组成部分。了解你的地理位置（以第42街道和第七大道为例）把你置于网格中，使你易于判断距离和位置。这使得在纽约地铁图中出现的很多地理错误（一个臭名昭著的例子是Vignelli地铁图把第50街道和百老汇地铁站放在第八大道的西部，而不是放在东部）非常明显且易于发现。

译注1：IRT（Interborough Rapid Transit）、BMT（Brooklyn-Manhattan Transit）和IND（Independent Subway）是20世纪40年代三大地铁运输机构，如果你想了解更多，请访问*http://www.nycsubway.org/faq/briefhist.html*。

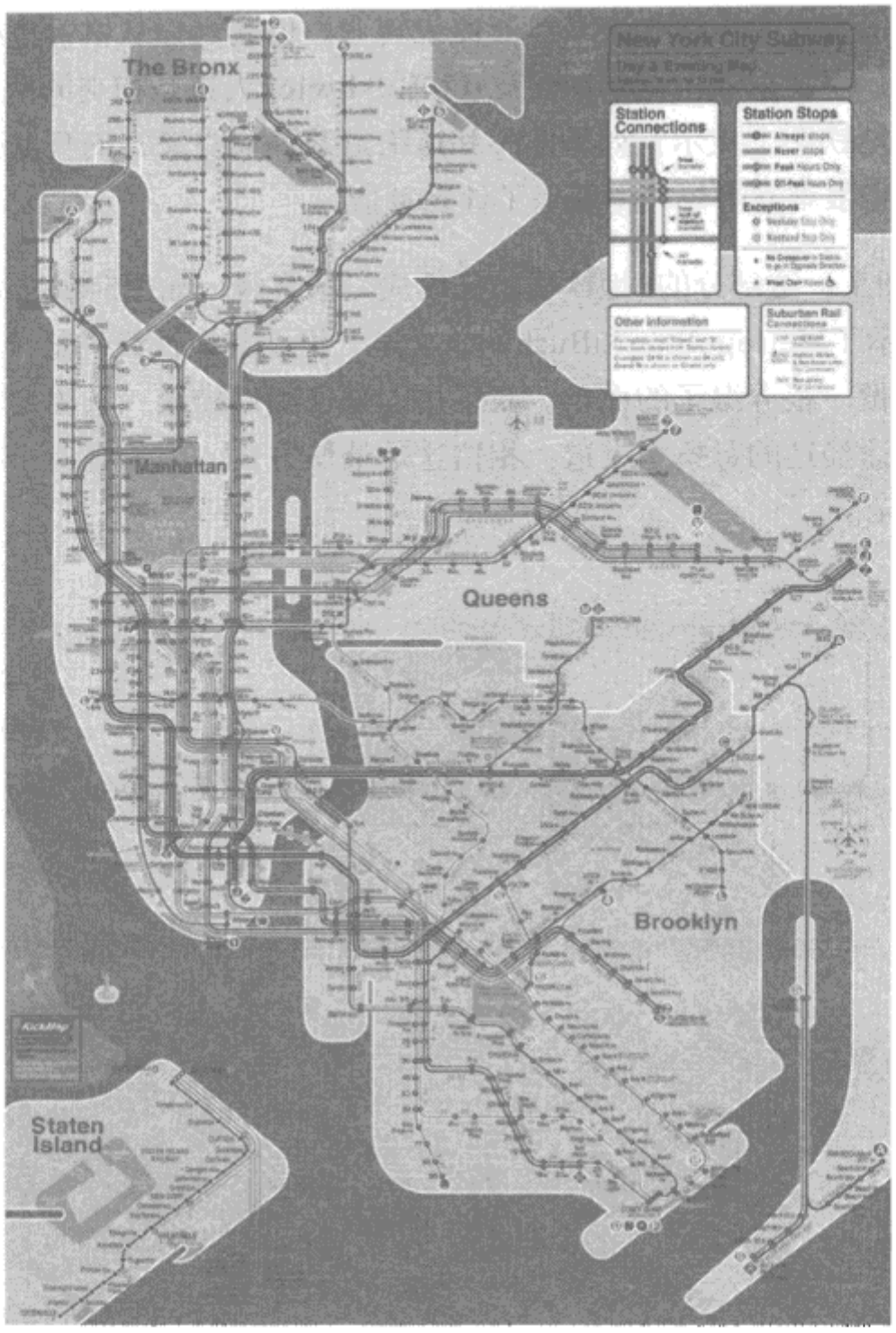


图5-5：2007年新版的KickMap地铁图（见彩图28）

对于纽约地铁图的一些早期版本，很难相信那些设计者曾经作为日常的城市生活真正地乘坐过地铁。他们做出的很多决策和地铁的现实情况脱节。作为设计过程的一部分，我乘坐地铁，去每个我不熟悉的主要交汇点。

在纽约，地上交通和地下交通之间存在很密切的关系，因为当地铁乘客离开地铁时，他们还需要继续旅途，因此地铁图尽可能清晰地表达出这种关系是很重要的。否则，会导致乘客产生迷失方向的不舒适之感。

囊括基本点

以布鲁克林的地铁L号线路为例。作为一名地铁乘客，你在拥挤的旅途中，并没有真正

注意到地铁线路沿着主要街道和交叉口弯曲或转弯。但是当你从格雷厄姆大街的地铁站出口离开地铁时，很显然Metropolitan大道和Bushwick大道是相交的两条主干道。为什么这一点没有在地铁图上显示？如果你不知道街道是如何交叉，而且从地铁出来后只看到某个标识，你将很难弄清楚究竟发生了什么事情。

在Vignelli地铁图上，这部分的L地铁线被描绘成一条直线，如图5-6a所示。Hertz地铁图（见图5-6c）显示了Metropolitan和Bushwick大道，但是其地铁线只是“敷衍性”地描了一条线，看起来像一根弄湿了的面条。我采用的是仔细地描绘一条固定格式、准确的地铁线，标明了沿途经过的每条主大道，相信这是最佳方式，因为它对于乘客是最有帮助的，如图5-6b所示。

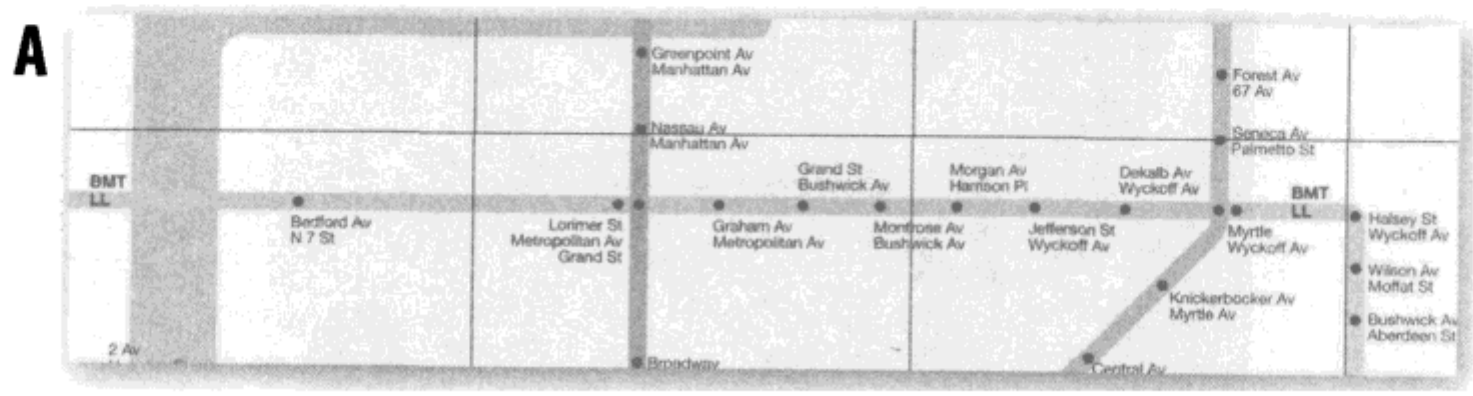


图5-6：布鲁克林的地铁L号线路的一部分：a) Vignelli地铁图，b) KickMap地铁图，c) Tauranac-Hertz地铁图（见彩图29）

相反，为了帮助乘客理解，我在制作地铁图时，有时对地理地形风格上做了一些简化。

举个例子，皇后区的主干道是皇后大道（Queens Boulevard），最初是跨越5个不同农场的道路，因此它蜿蜿蜒蜒地从皇后大桥东部穿过。近期的地铁图并没有贴切地捕获皇后大道和地铁的关系，这些地铁图或者完全忽略了它（如图5-7a所示的Vignelli地铁图）或者掩盖了它（如图5-7c所示的当前的MTA地铁图）。在我制作的地铁图上，我把皇后大道作为直线，如图5-7b所示。我这么做的原因是用户可以很容易理解路线，而沿线旅途中可以明白我所做出的这种“折衷”的意义——沿着一条地铁线路乘坐，然后换乘转到另一条地铁。在这种情况下，7条地铁沿着皇后大道运行，直到在罗斯福大道转向离开，地铁R/V/G/E/F号线路一直通向百老汇，然后在东部折回到原有路线。我所采取的展现风格可以使用逻辑来更好地表达地铁和皇后大道的关系，而在Vignelli地铁图和当前的MTA地铁图上，这些关系都不是很明显。

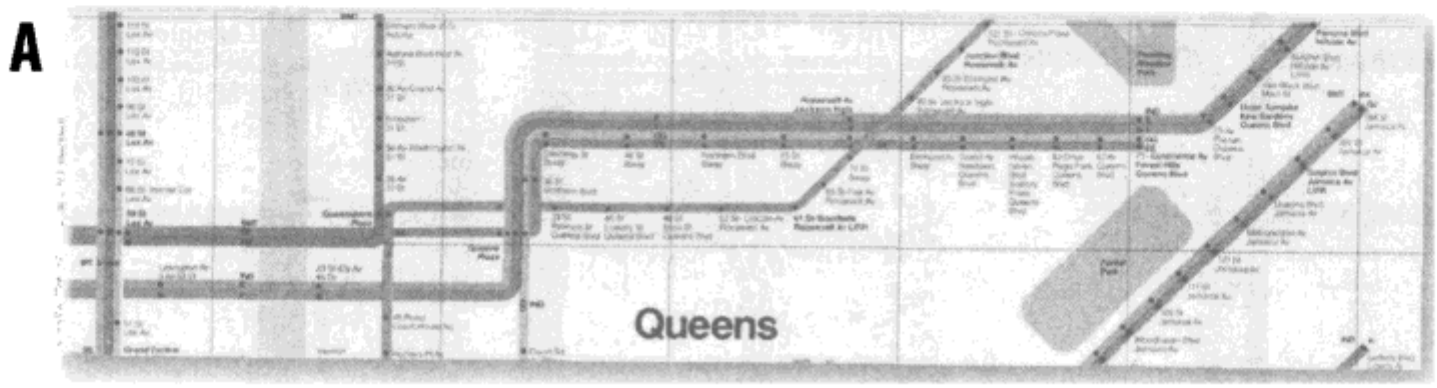


图5-7：沿着皇后大道的地铁线路在制图上的折衷：a）Vignelli地铁图，b）KickMap地铁图，c）当前MTA地铁图（见彩图30）

我觉得另一个需要清晰地显示的“折衷”是曼哈顿的中城[译注2]的第42街道，其中4/5/6线从帕克大道蔓延到列克星敦大道，如图5-8所示。沿着曼哈顿的中城或者默里山广场步行的旅客需要知道应该去哪个街道才有地铁入口。Vignelli地铁图把它作为直线，掩盖了其中的变换，它依赖文本来表达道路变换信息，而当前的MTA地铁图充其量只是表意很不清晰，而且看起来较乱。而在我所设计的地铁图中，用户应该去哪里是很清晰的。

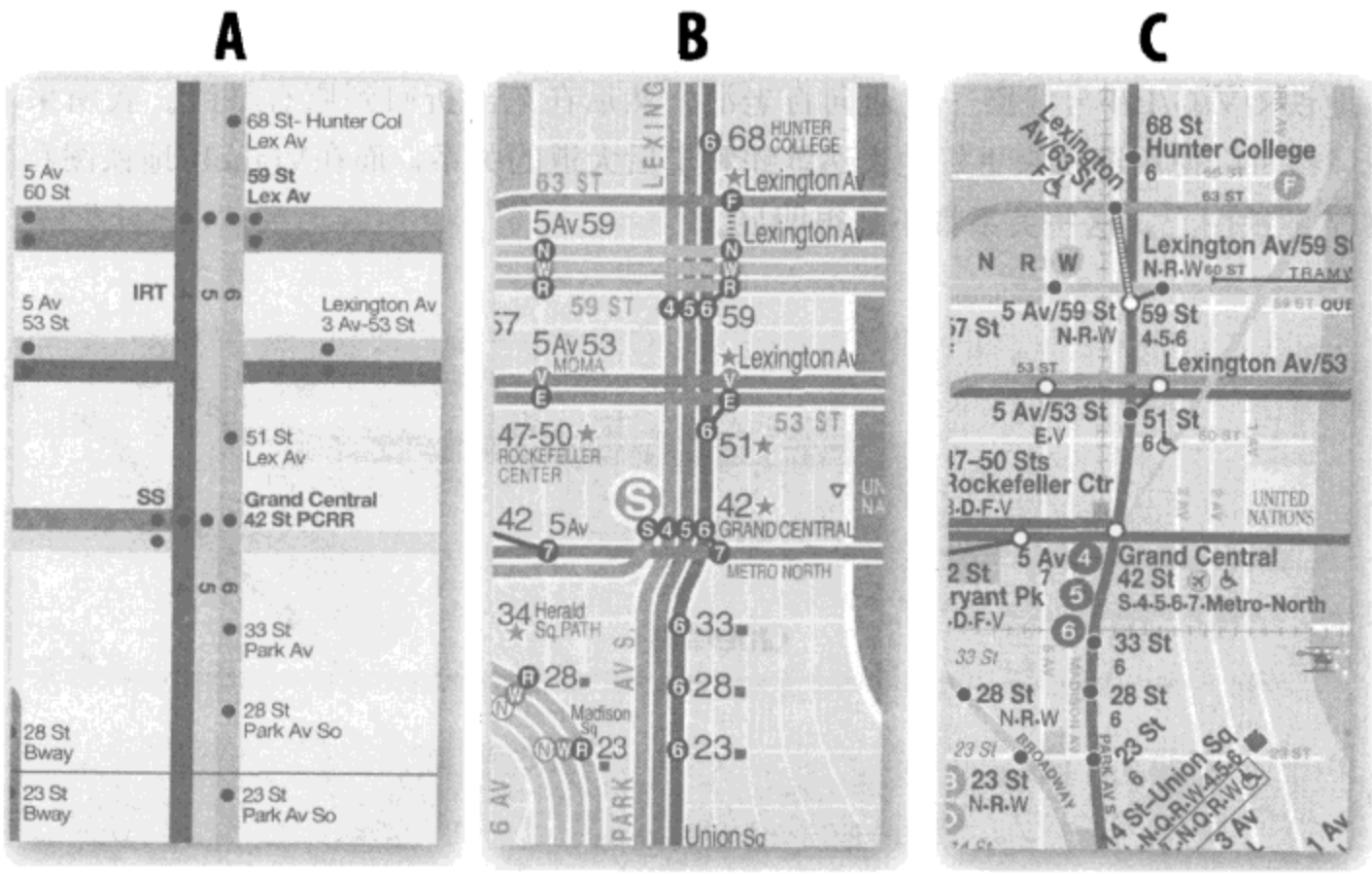


图5-8：曼哈顿的地铁4/5/6号线：a）Vignelli地铁图，b）KickMap地铁图，c）当前的MTA地铁图（见彩图31）

去除混乱

虽然在地铁图上显示地面上的一些地形很重要，但我觉得显示时去除掉一些地下信息也是很重要的。在地铁系统中，有些地铁站位于地铁隧道的交叉点和重叠点。显示这些交互信息，对于那些试着做某些修补的城市工人或者公共事业公司来说可能是很重要，但是作为普通的乘客，它们只会带来视觉干扰。我试着通过在地铁图上清晰地对这些线路进行分离，使得这些线路不会重叠，从而减少干扰。以布朗克斯的地铁4号线和地铁5号线的不同描述为例；当然，MTA的路径描述可能是准确的，但是它们在显示上很混乱，乘客不需要真正地看到那些具体细节来理解他们要去哪里。

译注2：中城（Midtown），是美国曼哈顿的中心区，指的是从曼哈顿的14街以北到59街为止。

对地铁线着色

地下的地理地形特征很重要，但是使用户能够理解应该坐哪一条地铁线去某个地方是更重要的。

1967年，MTA改变了之前所采用的和Salomon和先前的地铁图一样的三色地图，开始使用不同的颜色来表示不同的地铁线。然而，这种改变对于简化系统没有什么帮助。本质上，MTA地铁图还是包含26条线路，每条线路使用随机不同的颜色，使用一种颜色表示一条线路这种方法除了能够表示给定线路的连续性以外，并没有真正地给用户提供任何信息。Vignelli地铁图（见图5-10c）继续使用这种颜色表示体系。

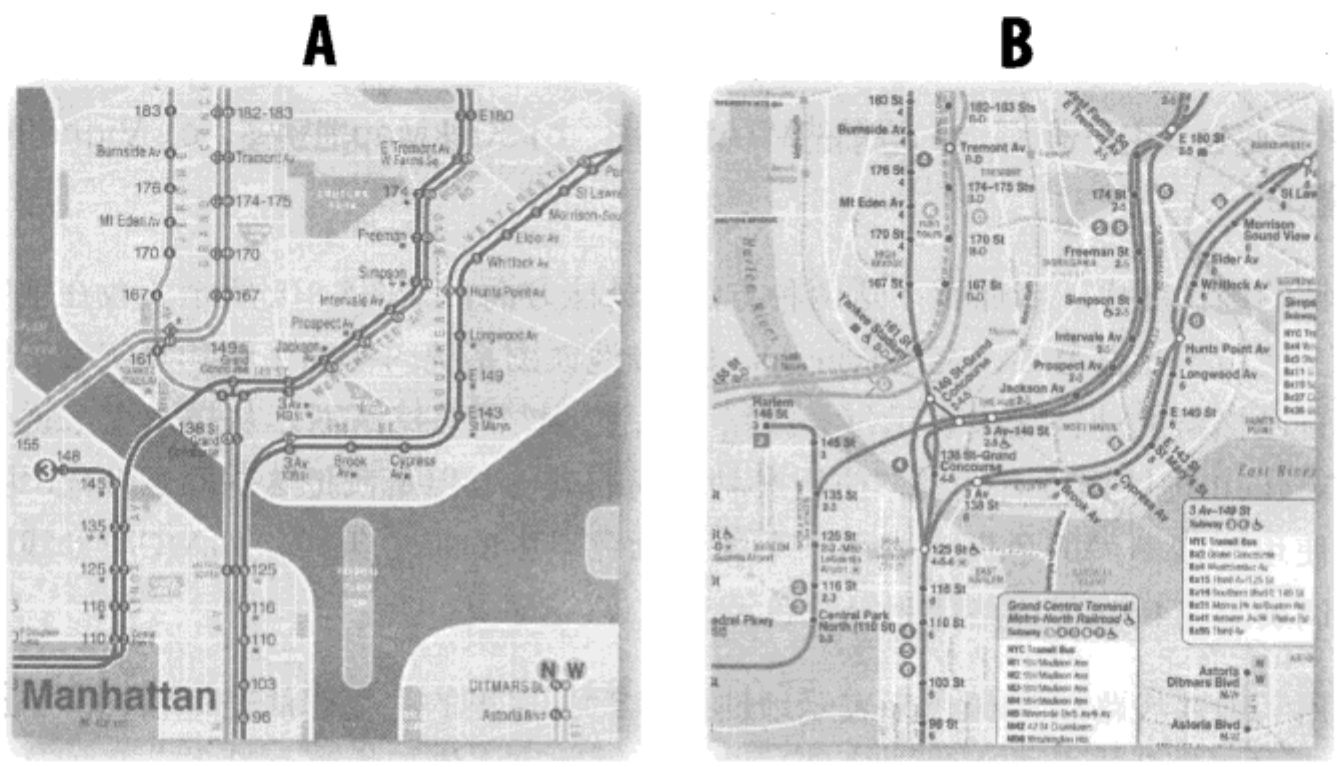


图5-9：地铁4号线和地铁5号线：a）KickMap地铁图，b）当前的MTA地铁图（见彩图32）

Tauranac-Hertz（即当前的MTA）地铁图试着把多条地铁线重叠成一条线来简化系统表示，但实际上它使得乘客需要对地铁系统的理解变得更加复杂，正如现在你需要看每个地铁站标注的文本才能知道某条地铁线是否在某个站点停下；如图5-10a所示。

Tauranac-Hertz地铁图的正确之处在于它对使用相同地铁轨道的一组地铁线进行颜色编码。举个例子，地铁A/C/E号线路全部都是蓝色显示的，而地铁4/5/6号线路全部都是绿色显示的。如果你查看从曼哈顿北部到南部的所有“主干”线路，颜色变换从蓝色到红色、橙色、黄色、绿色，产生一种光谱效应。这些颜色易于记忆，而且帮助乘客辨别哪一条地铁线将会带他们去想要去的地方。

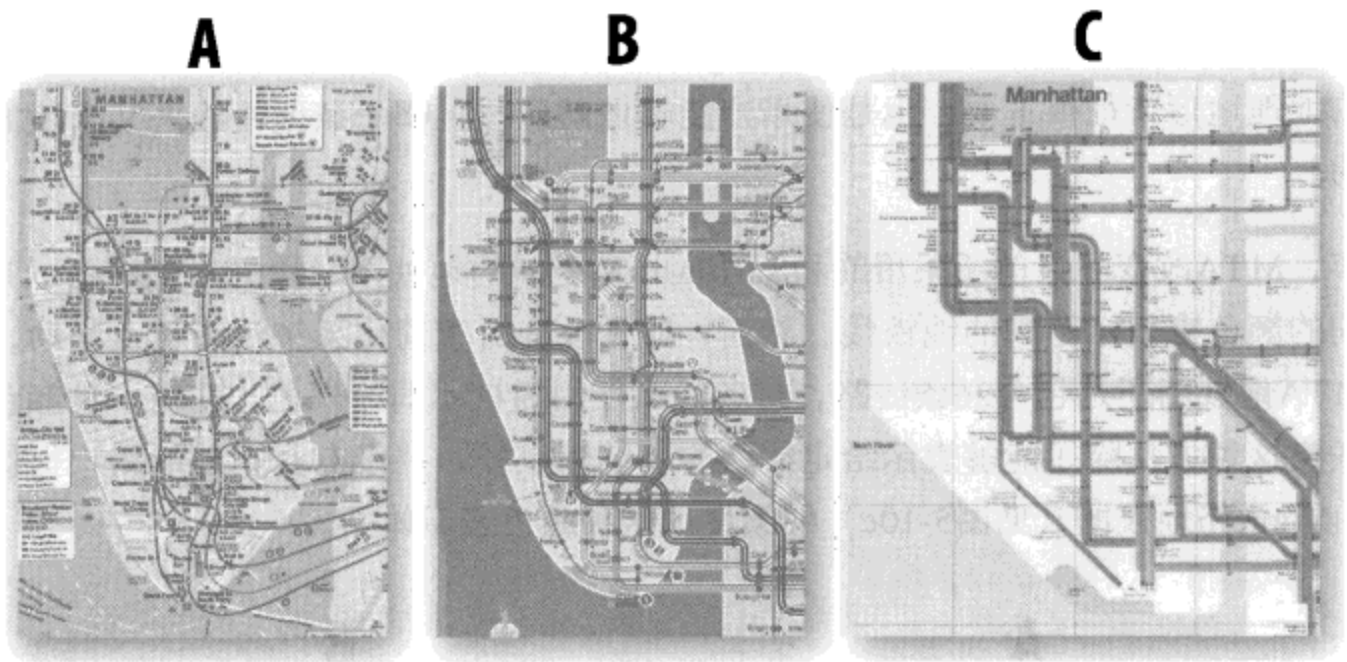


图5-10：曼哈顿“主干道”：a）当前的MTA地铁图，b）KickMap地铁图，c）Vignelli地铁图（见彩图33）

在我设计的地铁图中，我保留了这两种方法的最佳方面；如图5-10b所示。我在地铁图的主干线上使用了光谱色彩，突出Tauranac-Hertz地铁图系统内在的优雅和真实性，但是通过使用自己描绘的地铁线来表示每条线路可以使地铁图保持清晰。从技术上来说，我的做法和Vignelli地铁图没有区别，使用26种不同的颜色，但是我把这些颜色分成6到7组颜色系，使用深浅不同的色调来表示一个给定颜色系中的每一条线，如A/C/E号线路使用蓝色色调表示，4/5/6号线路使用绿色色调表示等。

我还利用了地铁线路ID和颜色来表示地铁站点[注2]。这里的主要想法是地铁图应该能够易于扩展，而不只是满足乘客的可读性。在一条地铁线上停下的每个站点，我把该地铁站的名字放在圆点内。通过这种方式，用户可以很容易准确地看到哪个地铁在哪个站点停下，而不需要去查看每个站点名字的地铁线列表。使用不同的着色点能够使读者一目了然看清该地铁是否总是停止在某处或者有特定条件，比如工作日/周末或高峰时期/非高峰时期的限制。

最后，纽约大约有80个地铁站点，如果你错过了某个站点，你不能仅仅只是出站，再方便地切换地铁方向。我通过在地铁名字旁边放一个小的红色方块来突出地理位置，表示那些需要转向换乘的乘客，他们不想离开地铁站，穿过街道，在街道另一面重新进入站点。当前的MTA地图显示了纽约的所有直升机机场，但是没有给乘客提供上面这个简单且重要的地铁信息——这样的优先显示很是让人困惑。

注2：　这是我在设计地铁图中的一个突发灵感。

我相信，总体说来，这些决策突出了使KickMap地铁图比它之前的那些地铁图更有用的创新点。

砍掉“鸡毛蒜皮”的东西

这些决定对我来说很容易，但是其他选择则更困难。但是我真正需要保留哪些地理特征？我应该使用哪些角度？我应该包含多少公共汽车和轮渡信息？

因此，在创建完满足我初始目标的构思后（如图5-5所示），我决定完善自己设计的地铁图，并体现了自己学到的所有知识点。我感到很兴奋。

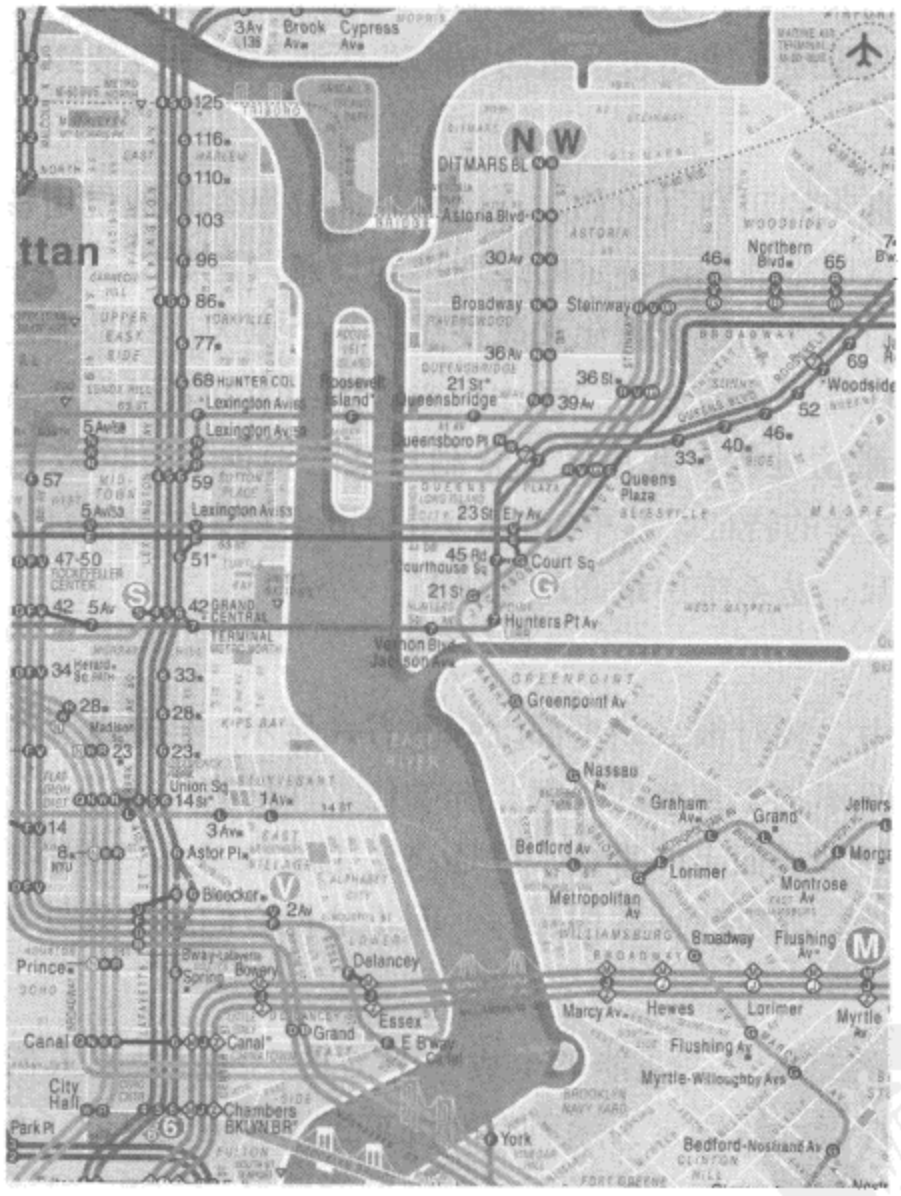


图5-11：我设计的测试版地铁图；我放了很多信息在该版本地铁图中，然后再修改它（见彩图34）

实践尝试

在汽车行业，构建所谓的“测试骡子”（test mule）是很常见的，它是模型或者试验性生产的汽车，塞满了每个可能的实验性特征，该模型经历了一些列的驱动测试来确定应该删除哪些特征（因为它不是基础必须特征或者工作不太理想）。我对自己设计的地图

使用了相同的“测试骡子”策略：我首先创建了一个版本，如图5-11所示，把我可能想要的所有特征都放到该版本中。Photoshop的Illustrator制图工具的图层特征在这里非常有用；我在这张地图中放置的很多东西最终都被删掉或修改。

测试版地铁图允许我们评价很多不同的折衷方案，比如：

街道网格

我想要在地铁图中显示街道的结构，而不干扰其他地铁信息。你将注意到测试版地铁图比最终版的设计包含的街道和街道名字要多得多。

海滩

我觉得一片绿色天地对于人们是重要的，纽约人应该能够乘坐地铁去海滩，而不是开车。我的测试版地铁图还包含纽约市的游泳池，但是我最终决定删除它们。

海岸线特征

很重要的是，真正的用户（比如，我妈妈）能够很容易地使用该地铁图，她一点都不在乎我在测试版地铁图中特定的地理详细信息（比如Steinway小溪或Wallabout海湾）。这是对地铁图进行简化和风格化的一个原因。但是我还希望能够有一些好的方面，使得任何一个地铁图“怪才”或者纽约爱好者可以欣赏。因此，有些地方我任由自己的激情驰骋。我决定充分利用某些地铁的好的效果，因此我包含了像Gowanus运河这样的特征，其中Smith 9号街道车站穿过该运河，在显示上必须去除它（高达91英尺，是系统中最高的车站）。

角度设计

在最后的设计中，我对很多角度都以标准方式显示，但是为了显示清晰，我有时做了一些修改。我不拘于角度的局限。标准化是件好事，但是我的目标是利用标准化使得乘客可以理解地面上的事情。我还决定把地铁站点名字都放在水平线上，保持一致以便于阅读，如伦敦地铁图那样，而不是把它们塞得到处都是。

桥梁和隧道

我做这个项目的目标之一是能够生成一个工具，可以鼓励人们乘坐地铁而不是开车。由于这个原因，我决定去除所有的汽车桥梁和隧道（除了标志性的布鲁克林大桥）。我希望乘坐地铁旅行的经历可以尽可能地整洁简单，不会吸引人们去开车，而是鼓励人们乘坐地铁。

我设计中做出的许多选择是基于以下原则。

用户只是平凡人

存在一些纽约标识可以帮助乘客辨别方向，这让人放心。在某种程度上，它们表示一些熟悉的事物，地铁图可以是富于情感的。因此，我觉得保留这些图标可以增强地铁图作

为工具的用户友好性。我设计的不是地理上十分精确的地形地图，而是情感和地理上相对准确的地铁图——曼哈顿看着像曼哈顿，中央公园是绿色的，哈德逊河是蓝色的，而地铁站点相互之间以及和街道的相对位置都是准确的（比如德兰街在包厘街的东部）。

同样为了以人为本，我在地铁图中包含了一些有名的标志——自由女神，爱丽丝岛雕像，布鲁克林大桥。而且我不仅仅只是通过名字标签来显示它们；实际上显示的是大家所熟悉的它们的形状，如20世纪30年代后期的地铁图一样[注3]。

由小区组成的城市

当我乘坐地铁去看望母亲时，我不是去第95街的地铁站看她；我是去她家里看望，在布鲁克林的瑞奇湾区。这正是纽约的一个重要特征：它是由各个小区组成的城市，而且纽约当地人一提到这座城市，就想着这些小区。这正是我们的参照系：比如说，我们从华盛顿高地区到瑞奇湾区。

当前的MTA地铁图包含一些小区名字，但是和地铁站点名字相比，它们只不过是用深蓝色显示的单词，对地区的描述没有什么价值。不存在信息层次。通过对小区进行颜色编码——至少在19世纪40年代以前，纽约市地铁图就开始用这种方式了——以不显眼的方式（采用柔和的色调），用白色文本来显示标签，而地铁站点名字是以黑色文本显示，因此不会造成视觉干扰，通过这种方式，我能够在地铁图上提供多层次的信息显示，而不影响地铁图的清晰和功能特征。

同样，这些元素实际上是在Illustrator工具中，通过不同的数字图像层创建的。它使得我可以通过不同方式显示不同小区，从而确定哪些小区是真正需要显示的，并制作出显示不同小区名字的不同版本的地铁图。

一种尺寸并不适合所有场合

我相信分离功能对于任何有用的可视化或工具都是很重要的。分层显示的另一个好处是它允许我们后期为用户界面定制地铁图。iPhone 和iPad的应用提供了KickMap地铁图，随着用户对地铁图进行缩放，KickMap地铁图的详细信息会自动变化。地铁图除了作为应用，乘客在很多不同的场景下也会查看地铁图：有可折叠的打印版，挂在地铁站的大幅面版，贴在地铁车厢上的（在座位右后方，因而你需要从某位乘客的缝隙中查看），以及贴在网上的。当前，你从每个地方得到的是基本相同的地图，但是实际上不应该如此：在每个场合下，应该有一个稍微不同的版本，它根据当时的特定环境进行了优化。

注3：　我原来想放上帝国大厦，但是它会影响中城的展示，而且我一直以来的目标是设计一个真正简单实用的地铁图！

每个地铁图版本都应该有自己的设计，根据其所在的场合进行定制。举个例子，挂在地铁站的大幅面版，应该能够显示各个小区，但是在地铁车厢中是供乘客做出决策参考的，如是否需要在下一个地铁站下车。因此，在地铁车厢中的地铁图就不必提供所有的公交信息了？

场合也不仅仅只是物理上的。晚上11点以后，纽约的26条地铁线减少到19条。因此，除了白天/夜间的KickMap主地铁图，我还设计了如图5-12所示的夜间地铁图。不是依赖在图下角包含大量文字、难以阅读的图形来说明的一张固定大小的地铁图，而是给乘客提供夜间地铁图（不仅仅是在iPhone上，而且在地铁图车厢上也提供）。

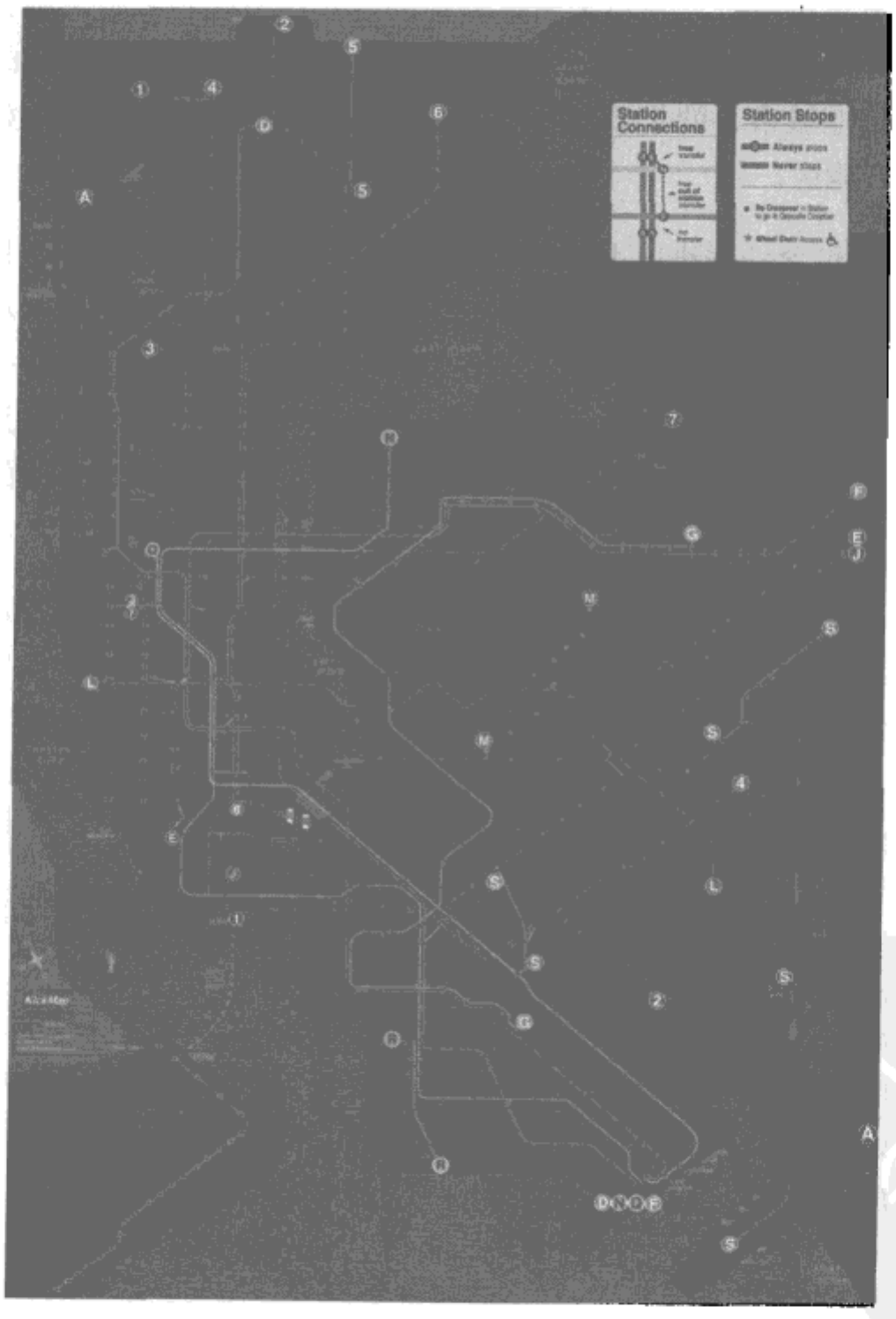


图5-12：只显示在夜间11点到凌晨6:30运行的地铁线路的KickMap地铁图夜间版（见彩图35）

在设计纽约的夜间版地铁图时，我对白天/夜间版本地铁图进行了简化，删除了大部分的街道和小区信息，因为它们看起来很冗余。此外，我非常喜欢Beck的伦敦地铁图的简洁美观，为了和它一致，把纽约的夜间地铁图也设计得很简洁。

结束语

最后，我确实认为KickMap地铁图实现了我绝大多数目标：使地铁线和连接尽可能地清晰以便于查看；当乘客离开地铁站时，提供清晰的显示信息，以使他们能够知道自己在哪里，从而使地铁对所有人显得友好热情。

然而，我的主要目标是把我设计的地铁图放到地铁乘客手里。MTA拒绝了我的设计后，我找到了另一种方式来分发它，通过Applie的iTunes——为iPhone、iPod Touch和iPad提供两个应用程序：一个免费的，一个付费的。

我做出的所有选择都是为了使用户体验尽可能的无缝和愉悦。显然，我激起了很多人的共鸣，超过25万（而且还在增长）的用户通过iTunes下载了KickMap地铁图。这是件好事，但是我仍然希望KickMap地铁图——或者一些更好的地铁图——能够取代当前地铁系统的地铁图。我希望人们使用我们的无与伦比的24小时地铁系统能够感到舒心，甚至幸福。地铁系统很复杂，但是如果人们知道乘坐地铁可以变得多么简单，（如果地铁图成为人们的好朋友[注4]，而不是障碍）地铁乘坐量将会增加。最终，它不仅仅有利于地铁系统本身，而且有利于所有在这里生活、工作、参观和呼吸的人们！

注4：我想很多人对于作为纽约伟大象征的地铁图充满热情。地铁图显示了地铁作为一种动态的毛细血管系统滋润着这座城市。这不仅在人们的观念上，而且在历史上亦是如此：建立地铁是为了以低廉的运输成本往返于中央商务区，并惠及新的居住区，从而这座城市可以继续繁荣发展。

第6章

飞行模式：深入探索

Aaron Koblin和Valdean Klump

天空中也有道路。虽然我们肉眼看不见它们，但是它们确实是存在的：独特的、定义严格的道路，每天有成千上万的飞机沿着这些道路飞行。作为独立的个体观察员，我们可能永远都无法猜测出这些情况，但是对原始的飞行数据所做的绘图却为我们展示了另一面（见图6-1）。

“飞行模式”（Flight Patterns）是我在2005年开始启动的一个项目，它是对美国和加拿大的民航运输进行可视化。它以两种媒介方式存在：静态图像，它追踪在24小时之内美国和加拿大机场抵达和离开的飞机；视频图像，描述了和静态图像一样的同一份数据的运动状态。在本章中，我将向你展示其中一些图像，并探讨用于渲染这些图像的技术。我还会分享一些想法，探讨我为何觉得该项目如此吸引人心，以及为何希望你也能有同样的感受[注1]。

注1：本章的所有图像都可以从网上获取高清图像，因此，如果你对这些图像很感兴趣，我推荐你访问我的Web站点，可以查看这些图的最佳效果：*http://www.aaronkoblin.com/work/flightpatterns/*。在该站点，你可以对可视化进行缩放，查看飞机高度、型号和制造商的彩色显示方式。你还可以查看飞行数据的动态视频。

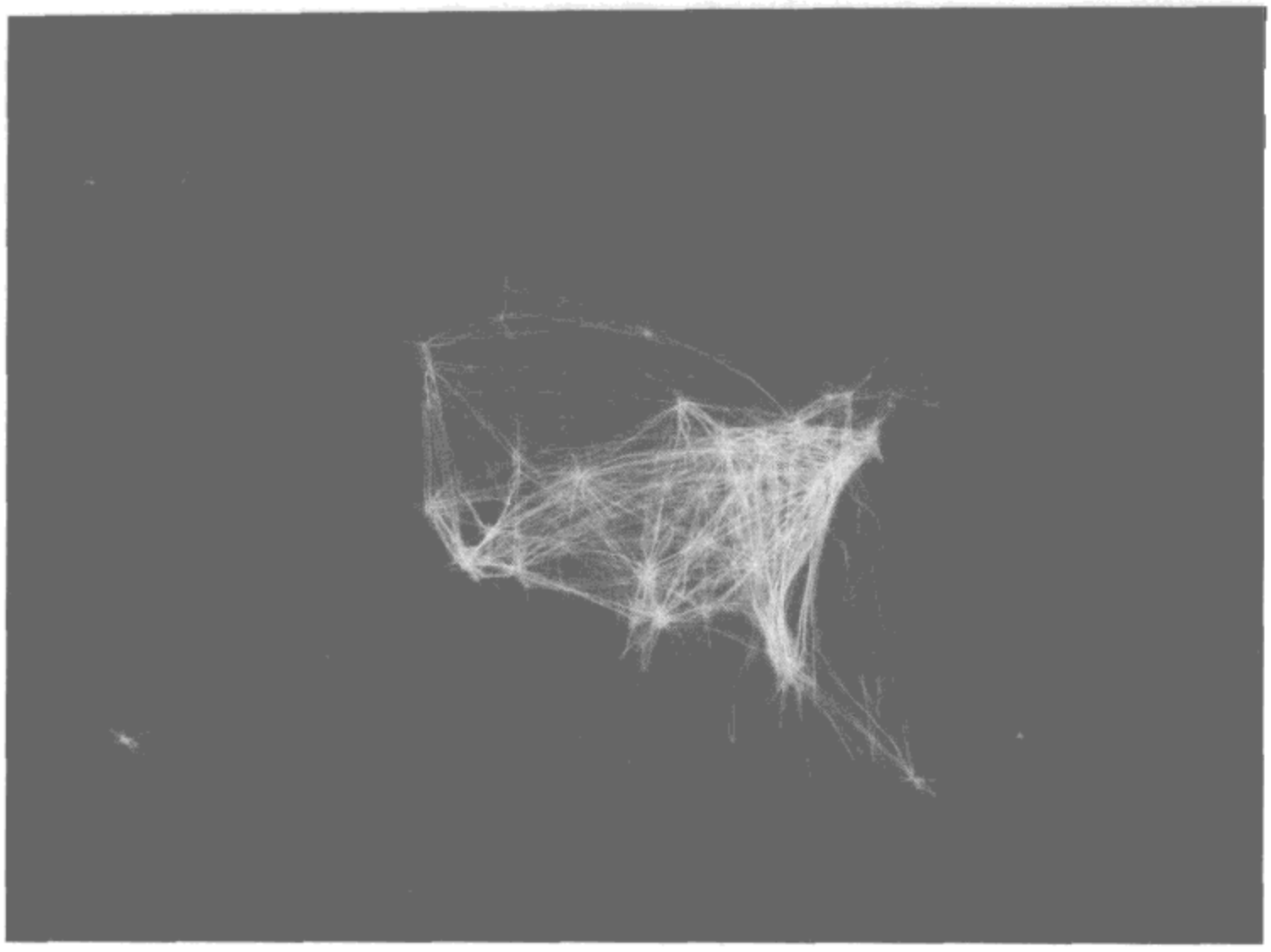


图6-1：“飞行模式”，飞机抵达和离开美国和加拿大机场时的飞行地理位置的数据可视化（见彩图36）

首先，在我看来，这个可视化拥有两个最为重要的特征：第一个特征是所有飞机往往沿着完全相同的飞行路线飞行。当我最开始对数据进行渲染时，我期望看到的是接近机场的飞机是紧密排列组合在一起，而且不同机场之间的飞机分散度很高。但是，实际情况却刚好相反：不同机场之间的飞行路线通常是聚集在一起，而只是在飞机准备降落或起飞的阶段，其飞行路线才会趋于分散（见图6-2和图6-3）。

仔细想想，这是非常有意思的。天空是无限敞开的，没有任何自然的限制，因此飞机可以选择任何路线飞行。但是当查看“飞行模式”时，看起来就像是有张地图悬挂在空中，它类似于空中高速公路系统，不同目的地之间有指定的路线。你甚至可以给其他飞机“让路”。

图6-2：图6-1所示的“飞行模式”的部分特写图，说明了我所期望的通过数据能够看到的：指向各个方向的航线（见彩图37）

为什么会是这样呢？说实话，我也不太确定。这些航线可能只是最高效的飞行路线，或者——我想更有可能是——这些航线是由很多因素来决定的：如飞机的自动驾驶系统、政府飞行线路管制、运营商的航道方向、海空控制系统、在人口密度高的地区的飞行限制规则、风向气压等气象因素。无论如何，我认为“飞行模式”所显示的趋势都很震撼人心，因为它显示了一个完全开放的空间的逻辑组织。正是由于这个原因，我选择“模式”作为这个项目的名称。

“飞行模式”的第二个显著特点是它使得我们能够对浩瀚的美国和加拿大的航空系统进行可视化。在我看来，这正是数据可视化的价值所在。我们无法通过查看天空或者原始数据来完全了解美国和加拿大的航空体系，但是我们可以通过可视化来了解它们。对这些航道统一进行可视化显示，它们所展示给我们的方方面面要超出其各个部分单独显示的总和：这些可视化为我们展示了一个系统，而且我相信这个系统是美丽的。该系统显示的不仅仅只是航道，而且是关于人类的地理种群，更广泛地说，它显示了我们人类所期望的旅程。

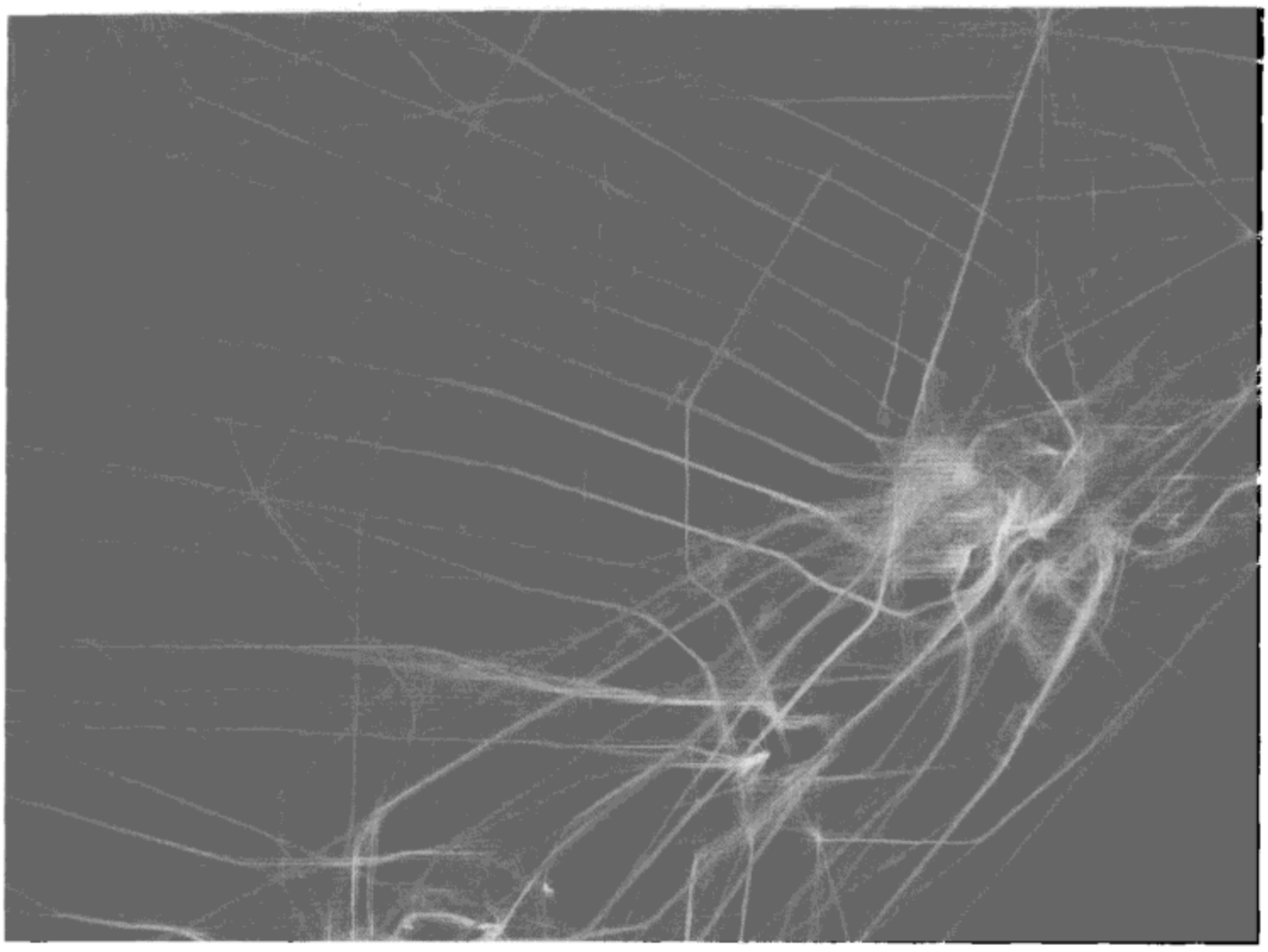


图6-3：“飞行模式”的另一个特写图，说明了我所发现的共同的方面：清晰、明亮的线条表示有大量飞机紧密跟进的航线（见彩图38）

技术和数据

“飞行模式”是使用编程语言Processing[注2]创建生成的。Processing是特别适合于数据可视化的一种编程语言。获取到飞行数据（这一步一直都是关键环节）之后，我写了一个简单的Processing程序，把每个数据点的纬度和经度翻译成在计算机屏幕上显示的二维地图的一个点。同时，为每个点添加了选择性的色彩来表示一些信息，如高度和飞机型号。然后，我把这些图片以TGA文件格式[译注1]导出。

对这些视频进行可视化有些棘手。如果以运动着的数据点的方式来展示飞机，这种方式无法展示每次飞行的变化。因此，采取的方法是在相邻的两个数据点之间画一条线，并在设定的时间间隔后（3分钟或5分钟，取决于数据集），在整张地图上增加4%的黑色不

注2：参考*http://processing.org*。

译注1：TGA，也称TARGA，是一种结构较简单的图像文件通用格式。你可以访问*http://en.wikipedia.org/wiki/Truevision_TGA*和*http://local.wasp.uwa.edu.au/~pbourke/dataformats/tga/*了解更多信息。

透明层。这意味着时间越久的航道会随着时间的推移而逐渐消隐到背景中，通过这种方式有助于显示飞机的飞行进度。

“飞行模式”中使用的数据是“行业飞机状态显示（Aircraft Situation Display to Industry，ASDI）”[译注2]订阅的处理后的版本，是一份由美国联邦航空局（FAA）发布的包含了所有的民航记录[注3]的数据。该订阅只有和航空业有关的公司才能获取。多亏了同事Scott Hessels，我获得了2005年的28个小时的飞行数据。这个可视化的最初版本是“天体力学”项目的一个成果，这个工作是我和加州大学洛杉矶分校（UCLA）“设计 | 媒体艺术”项目的Gabriel Dunne一起合作进行的。

我工作中使用的初始数据集是2005年3月19日到20日的飞行数据，它包含141 029次航班。每3分钟取样一次，总共包含6 871 383个数据点。3年后，即2008年，我和《Wired》杂志合作获取到了另一份数据。该数据来源于2008年8月12日到13日，包含205 514次航班。每分钟取样一次，共包含26 552 304个数据点。

获取到的数据是从ASDI订阅的，每个数据点包括以下信息：

- 维度
- 经度
- 高度
- 飞机制造商
- 飞机型号
- 时间戳
- 航班号

如果你对于查看一些具体数据感兴趣，目前FAA以XML格式提供了一些ASDI的订阅数据的样本，可以通过*http://www.fly.faa.gov/ASDI/asdi.html*访问。

色彩

“飞行模式”没有使用复杂的地图制作技术：简单地对数据进行绘图，让数据本身说话。然而，在讲述相同的航道上的不同“故事”时，色彩起着至关重要的作用。图6-4到图6-9给出了一些例子。

译注2：ASDI是通过美国交通局提供的数据流服务。你可以访问*http://en.wikipedia.org/wiki/Aircraft_Situation_Display_to_Industry*来了解更多。

注3：“民用”指的是FAA追踪的所有非军用的、商业的和私人的航班。

图6-4：在这张地图中，用色彩表示高度，纯白色表示飞机在地平面上（见彩图39）

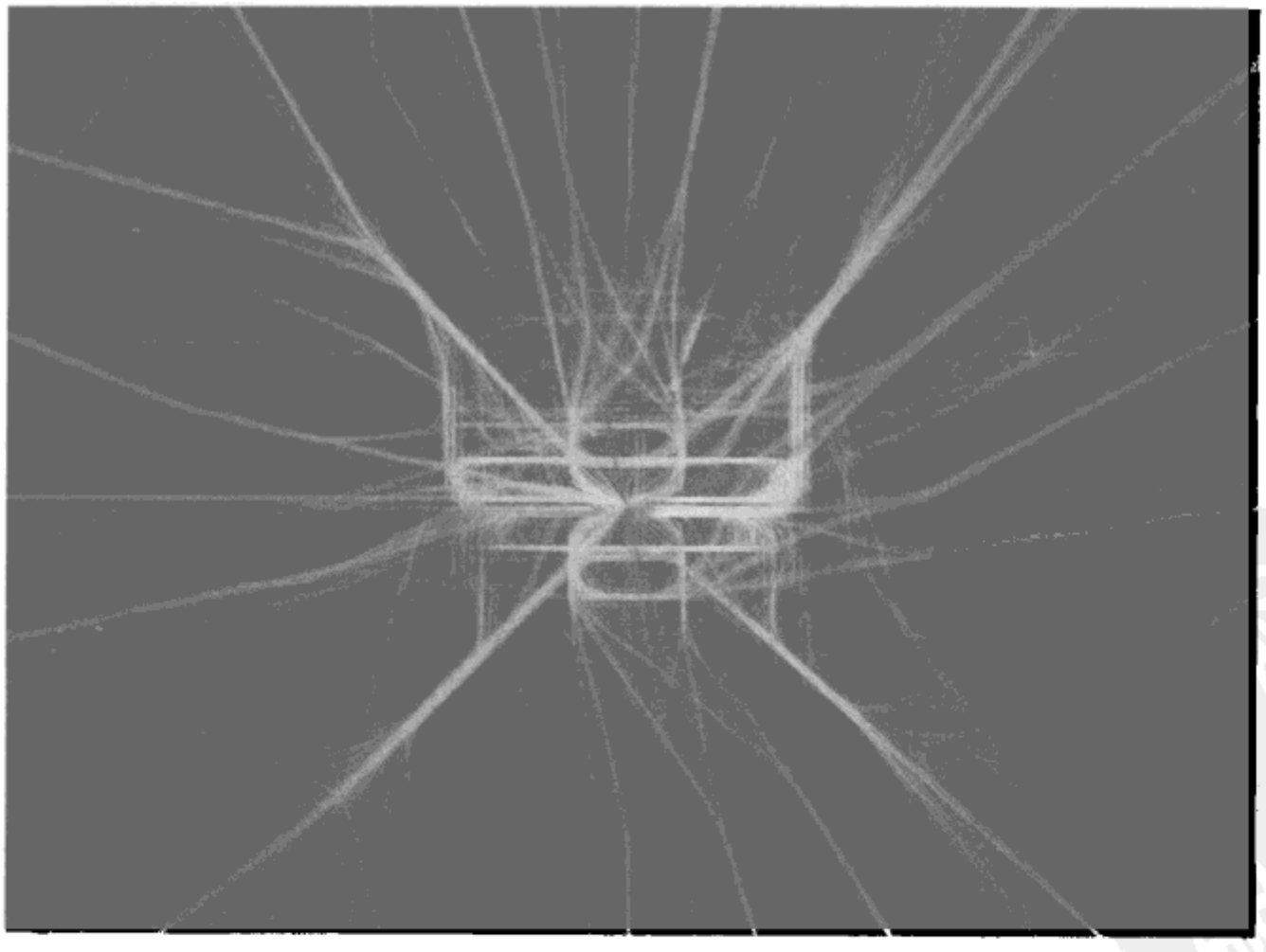


图6-5：Atlanta机场的一个特写图，清晰地显示了飞机跑道的布局（同样，色彩表示高度，见彩图40）

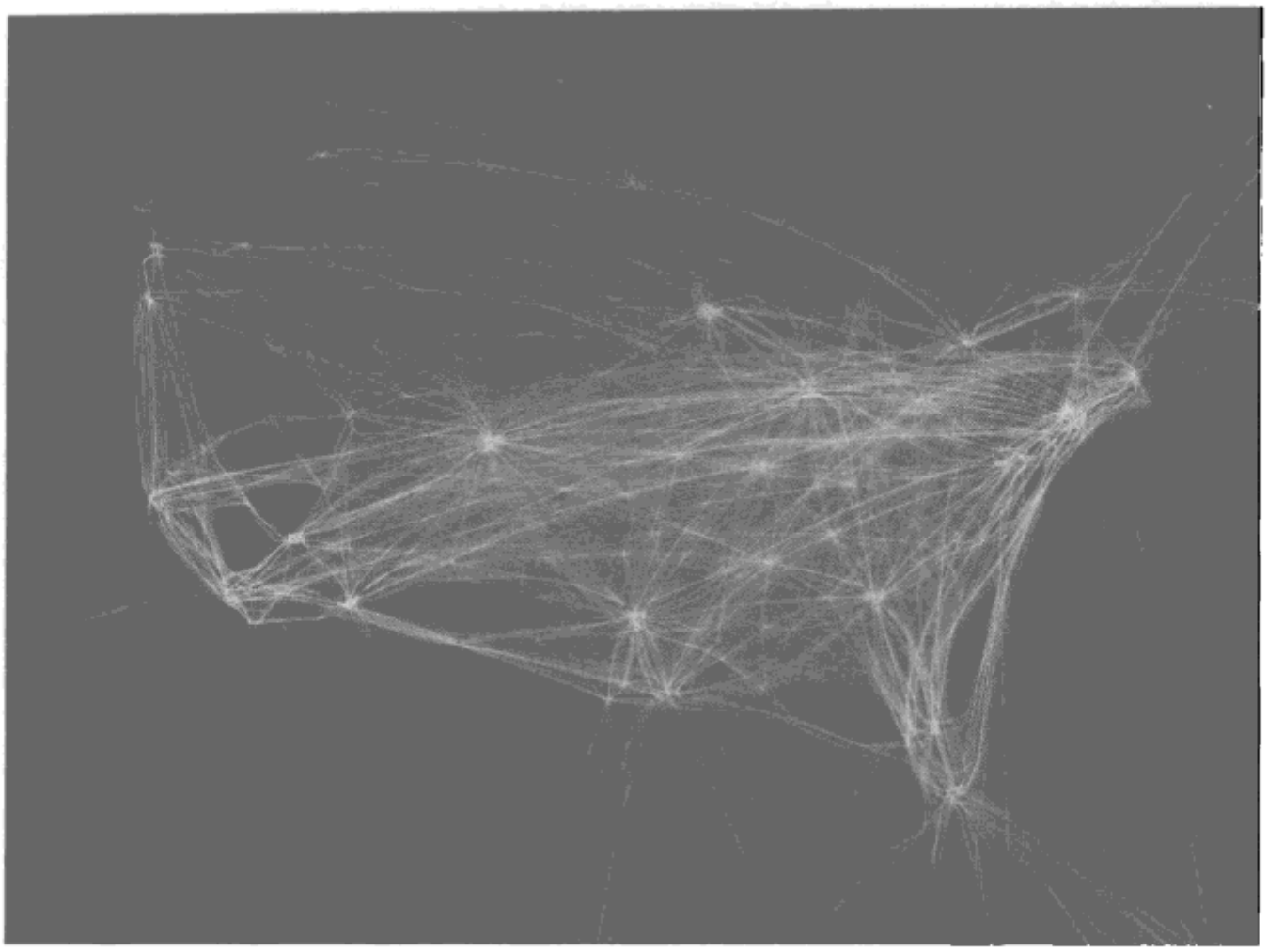


图6-6：在该图中，使用色彩来区分不同型号的飞机（见彩图41）

图6-7：单个型号的飞机的地图，只显示了飞机Embraer ER J 145支线的航班飞行路线（见彩图42）

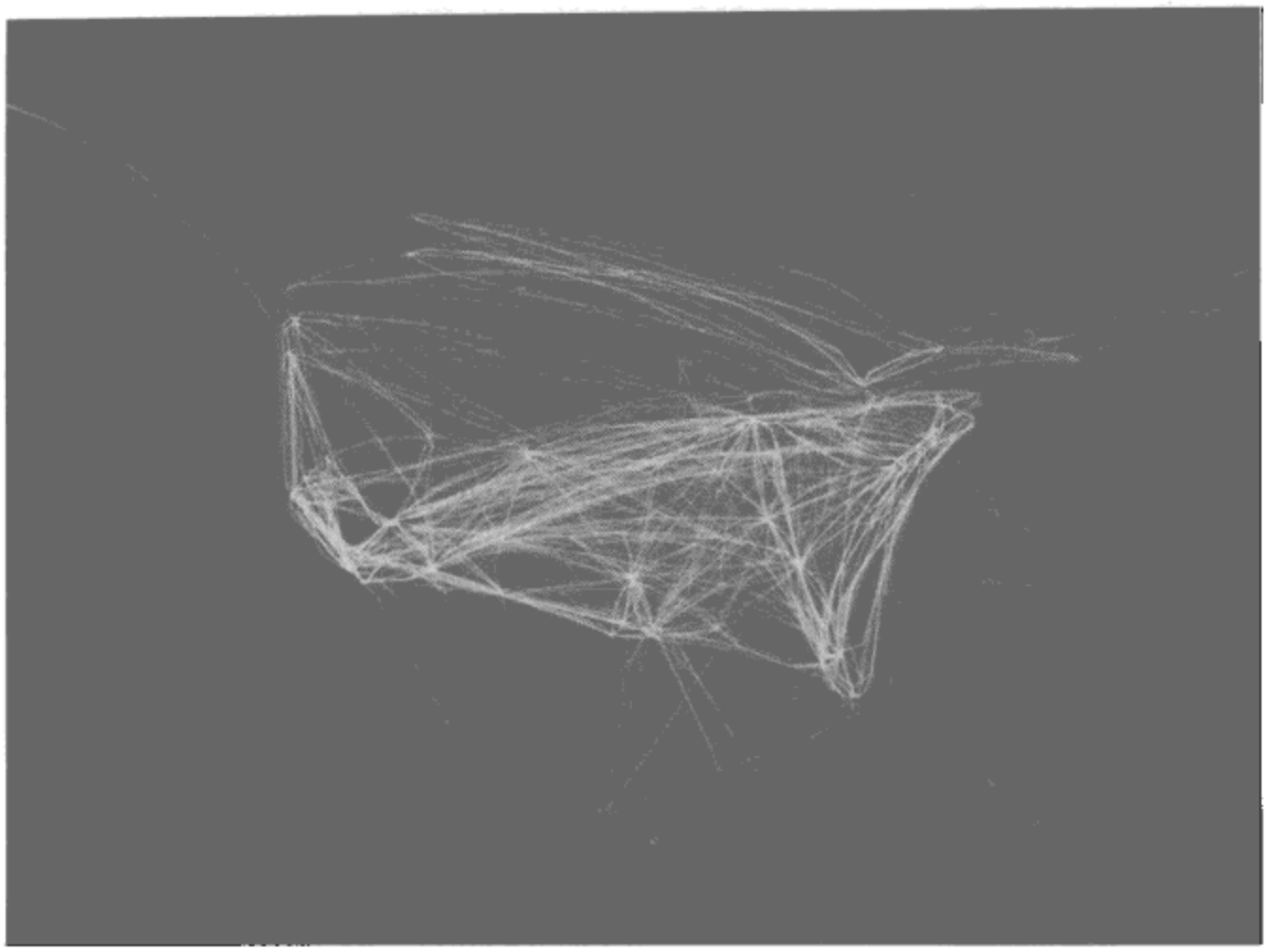


图6-8：另一个型号的飞机的地图，只显示波音737飞机的航班飞行路线（见彩图43）

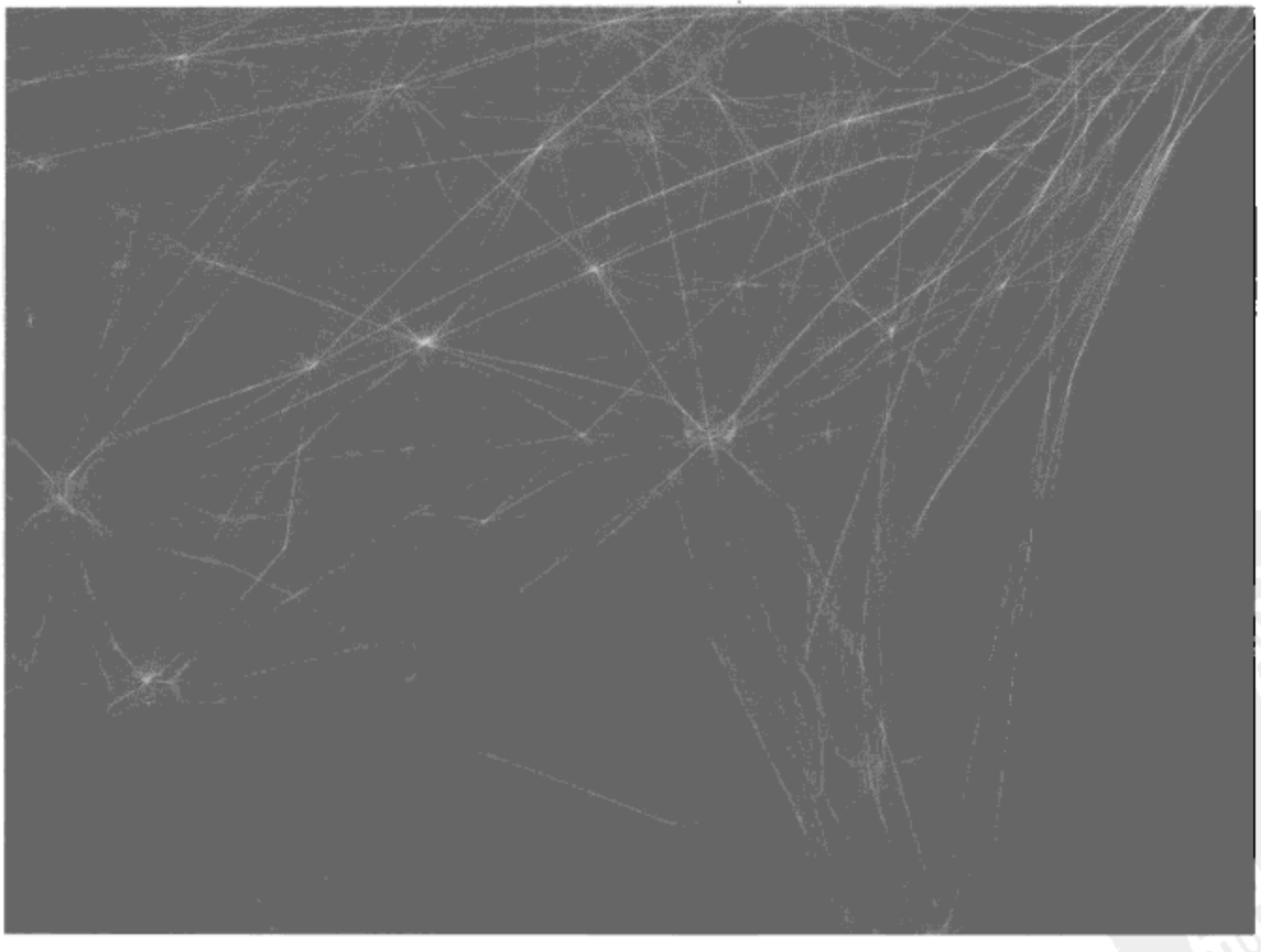


图6-9：在这张地图上，不同的色彩表示飞机的起飞和着陆：橙色表示正在降落的飞机，蓝色表示正在起飞的飞机（见彩图44）

动向

在动向方面，“飞行模式”揭示了新的信息，包括随着时间变化的飞行方向和飞行中的飞机的数量。可视化夜以继日地追踪着每个航道，以便显示一个国家如何进入“梦乡”以及如何在翌日“醒来”（见图6-10和图6-11）。

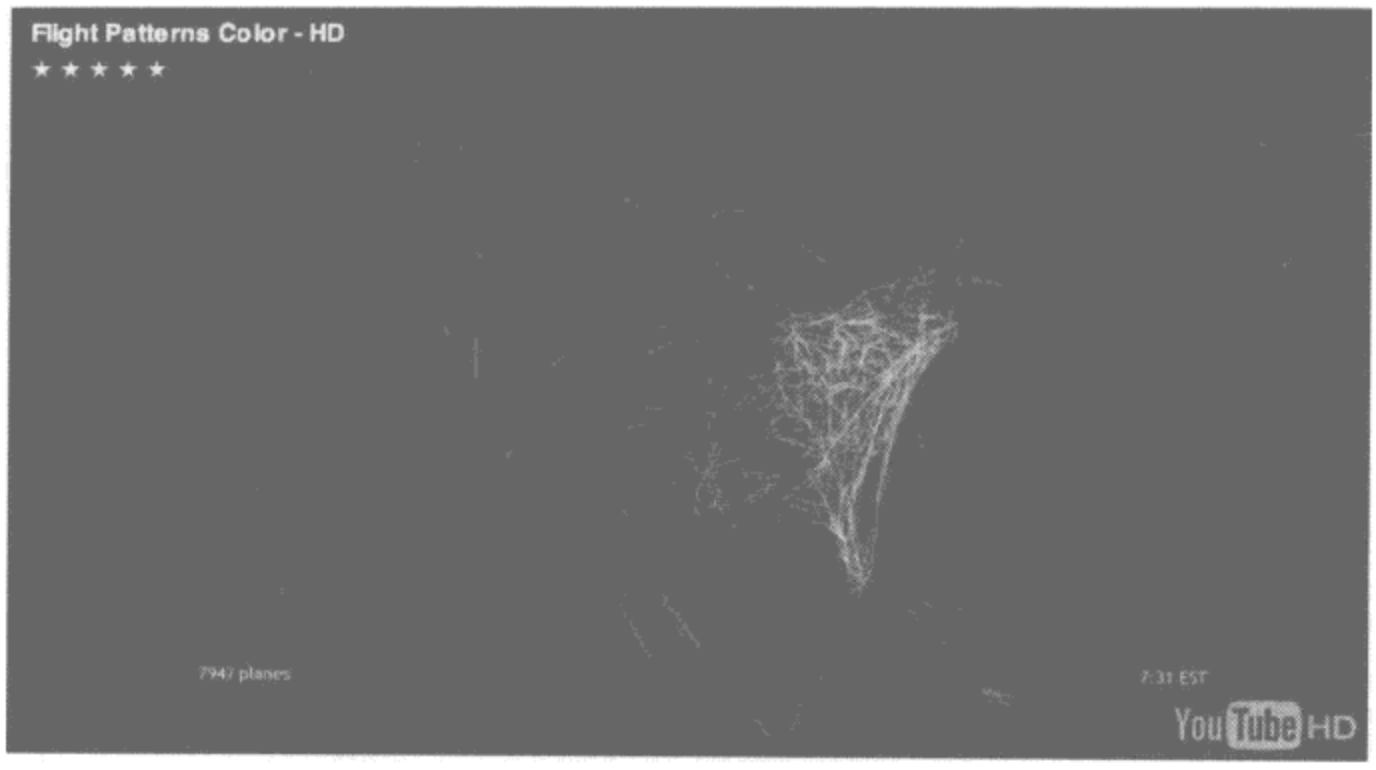


图6-10：东海岸“醒来”的图像：这是一幅静态图像，2005年3月20日美国东部标准时间早上7:31，显示了东海岸的高活动性，以及西海岸的虚拟静态性（除了从夏威夷起飞的向北飞行的一些红眼航班，见彩图45）

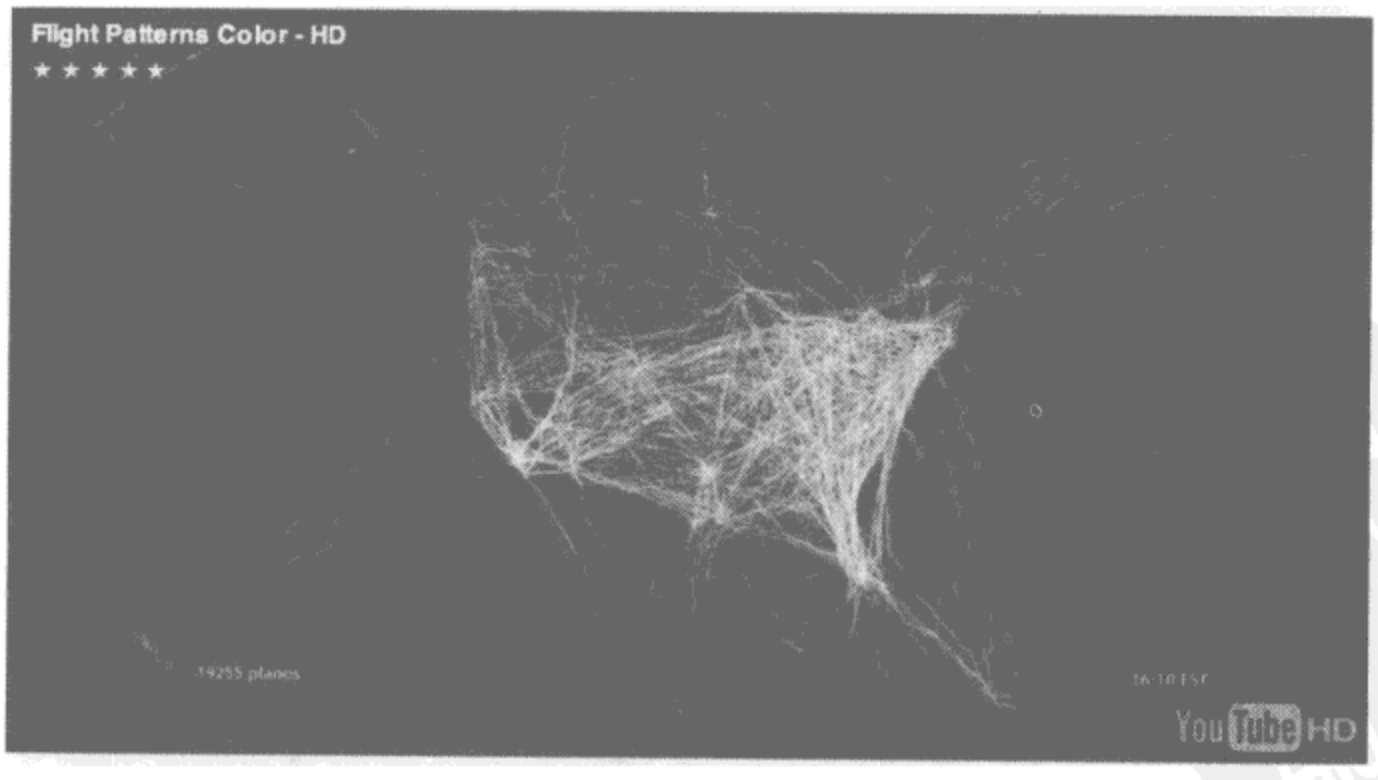


图6-11：美国东部标准时间下午4:10，我们看到一个非常不同的境况：此刻，航空最高峰达到了19 255架飞机（见彩图46）

在我的网站上有一个三维可视化视频，它描绘了三维投影面的z轴维度。为了在陆地侧面可以识别出z轴，我对维度做了些夸张显示，这样有利于生成稠密、有趣的可视化。然而，在印刷上显示的效果不好。感兴趣的话，我推荐上网看一下。

异常和错误

和很多数据集一样，我在飞行模式中使用的数据包含很多错误和异常，我删除了其中一些。举个例子，为了在数据集中寻找最快的航线，我识别出一个航班可以在6分钟内穿过整个美国——这显然是一个错误。另一个航班在穿过美国时，曲曲折折地沿着“之”字形（不可能的）由北向南的航线飞行——另一个明显的错误。我把这两个航班的数据都删除了。

还存在一些其他异常，然而，我把这些异常保留了下来。举个例子，北大西洋的航道看起来很曲折。我倾向于在可视化中保留这些数据，因为显示来自欧洲的航班是很重要的。我不知道为什么会存在这些错误。可能是飞机设备或者ASDI的处理出现了故障，或者是数据提供商导致的错误。在长时间思索之后，我决定保留数据原样。此外，当查找最短的航班时，我发现3000多个航班在没有离开机场时就报告了它们的地理位置，我也保留了这些异常。

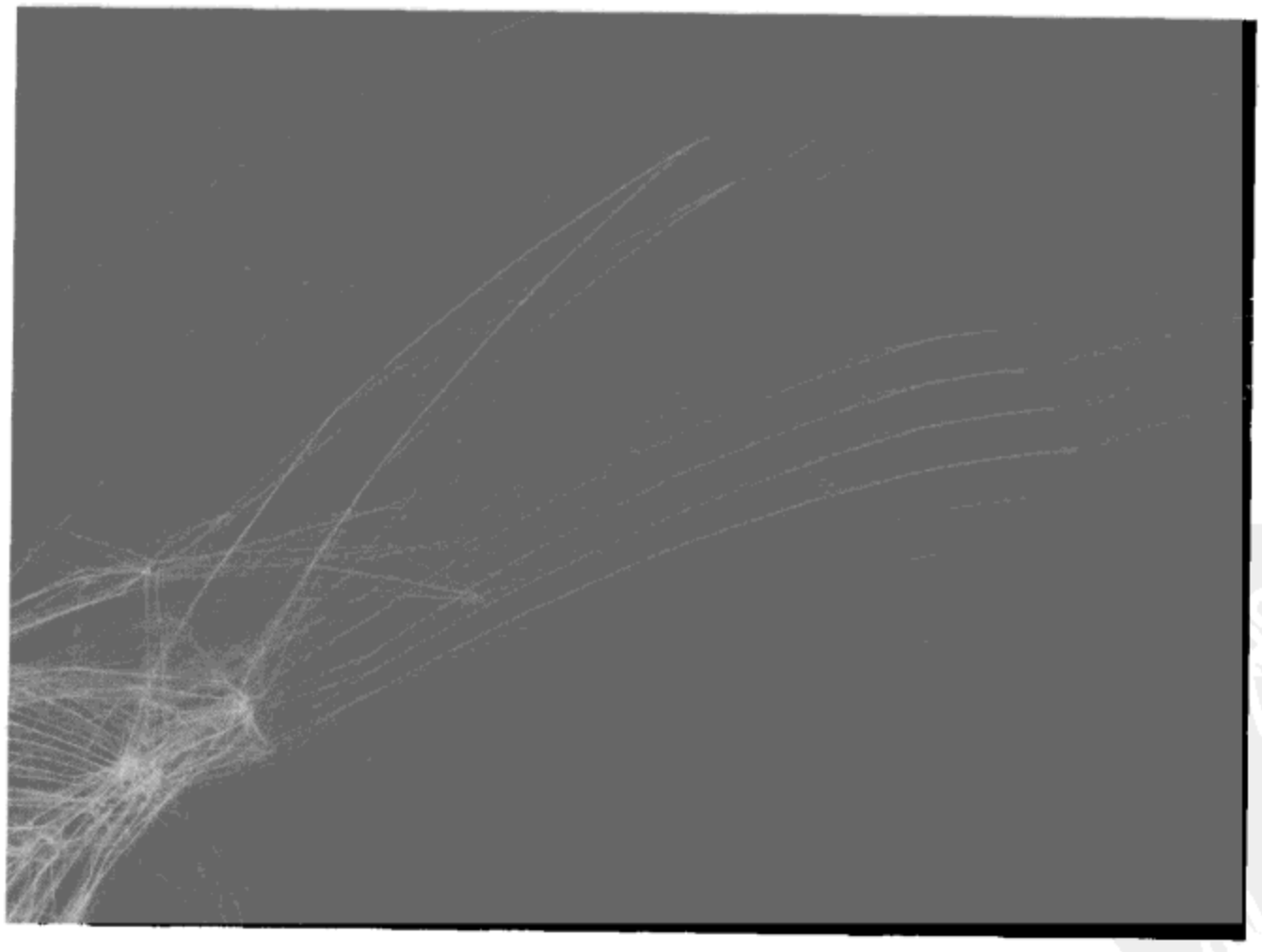


图6-12：北大西洋的飞行线路显示了数据中的一些异常（见彩图47）

如果你仔细查看该可视化，你将会注意到一些有趣的特征。一个明显的例子是美国内华达州的禁飞区域（见图6-13）。然而，这些禁飞区域看起来并没有完全禁飞：可以观察到有很少量的航班穿过黑色的太空。

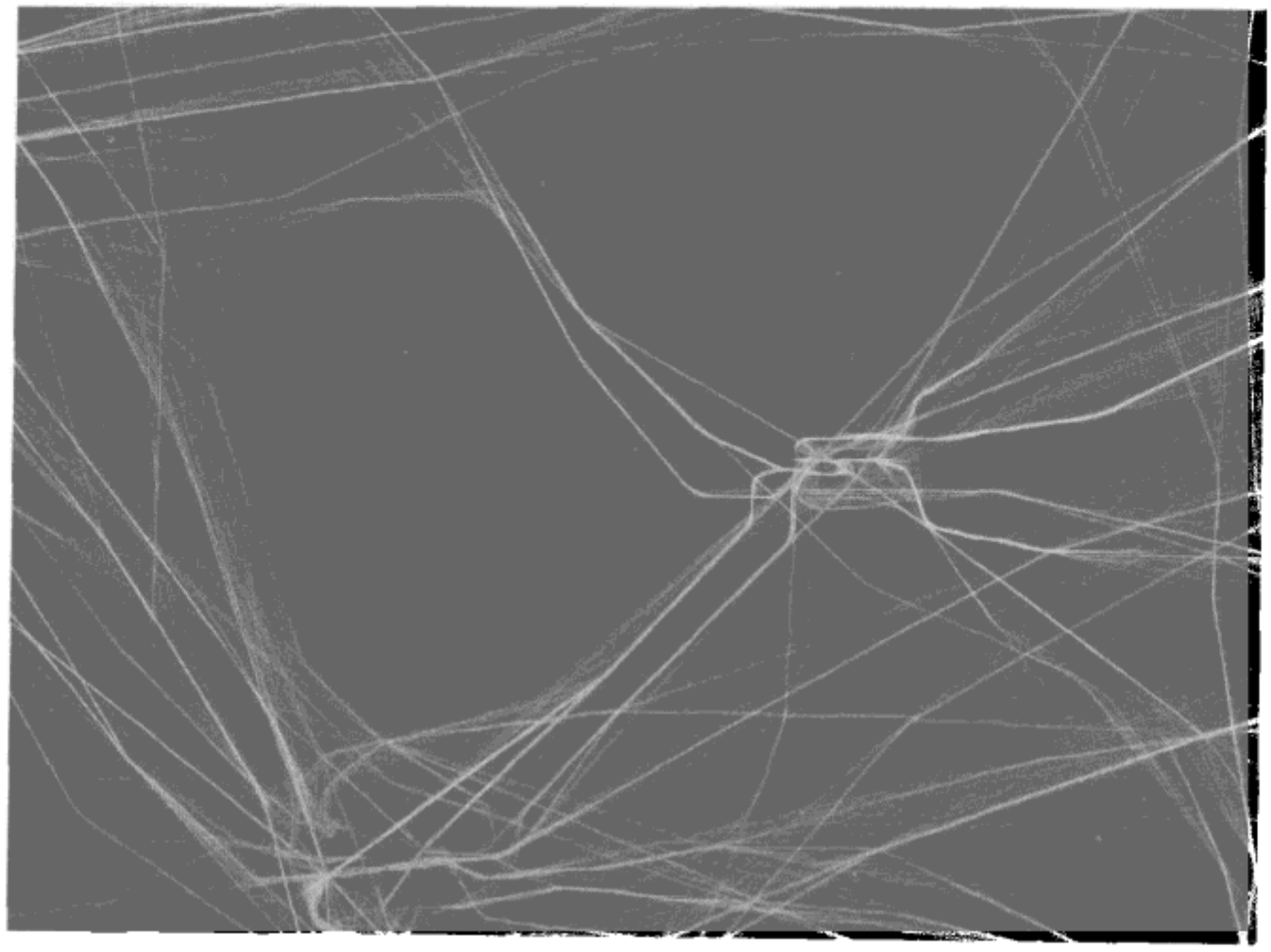


图6-13：美国西南部禁飞区的一个特写（见彩图48）

每当你处理大量的、有机的数据集时，你会发现数据中存在错误和异常。我认为去思考如何处理这些数据是很重要的。对于每一种数据的处理，我都扪心自问，通过对数据操纵，我是否会破坏数据的完整性？如果答案是肯定的，最好还是维持数据本身的完整性。对于存在明显错误的数据，把它们全部删除。如果存在某些原因，使得你应该保留异常而不是删除它们（那应该调查它们，查找出其中隐藏的有趣的故事）。

结束语

“飞行模式”是一种简单的数据可视化，而且这种简单性使得它更有吸引力。首先，该项目显示了空中交通系统地图，据我所知，在此之前它从未被公开可视化过。其次，可视化易于理解，虽然它完全是由数据生成的——可视化中根据机场创建的节点与我们对北美地理特征的理解保持了一致（见图6-14）。相似地，正如我们所期望的那样，最稠密的航道位于人口密度最高的地区。

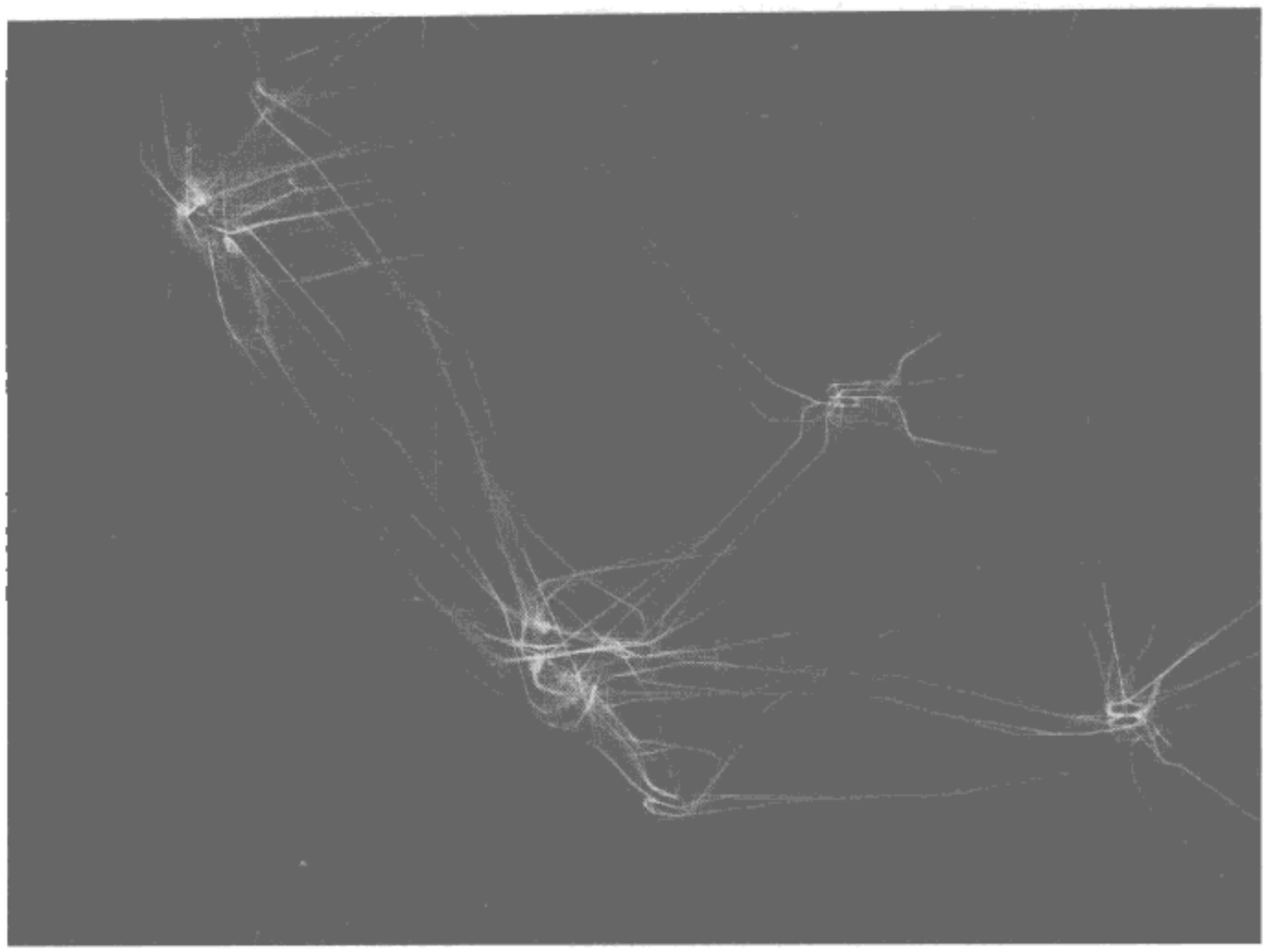


图6-14：美国西南部的一个特写图——你可以识别出几个机场呢（见彩图49）

最后，我觉得“飞行模式”之所以非常引人注目正是因为它很鼓舞人心。这一方面可能是由于和地图关联的特殊的感情，但是通过显示航空的有序性以及揭示飞机从一个地方如何到达另一个地方，“飞行模式”显示了一个逻辑系统。而当我们坐在离地面34 000英尺的机舱的16A位置时，我们只不过是浩瀚的天空中非常渺小的一部分。我觉得能够看到一个规模如此庞大的系统工作运行如此良好很是鼓舞人心。在美国和加拿大每天就有超过200 000个航班，我们真正地在空中“修路”，从出发地到目的地，每个航班都承载了成千上万人，安全记录非常高。因此，“飞行模式”不仅仅是数据可视化，它也是对当代空中旅行创造的奇迹的展览。

致谢

“飞行模式”的思想和启发归功于加州大学洛杉矶分校的两个同事：Gabriel Dunne 和 Scott Hessels。在2005年，我们启动了一个艺术项目Celestial Mechanics (http://cmlab.com)，它用于描绘运动中的航空和航天系统。该项目的一小部分工作是致力于处理航空飞行数据，这些数据正是我用来构建“飞行模式”的原始数据。感谢加州大学洛杉矶分校的Mark Hansen以及《Wired》杂志（尤其是Carl DeTorres）为这些图像的进一步制作所需的数据提供的帮助。

第7章

你的选择揭示你是谁：社会模式的挖掘和可视化

Valdis Krebs

数据挖掘和数据可视化密不可分。在数据中挖掘复杂的模式并且对它进行可视化，可以便于人们利用计算机的计算能力和人类的思维能力，对可视化有进一步理解。如果对数据挖掘和可视化善加利用，它们可以成为伟大的组合，能够催生出高效复杂的数据处理和模式识别。

在本章中，我们将探索一些数据集，挖掘出隐藏于数据背后的人的行为。基于活动出席率和对象选择构造出的模式，将为我们了解人们参加活动和选择对象的思考和行为提供线索。通常，简单的行为和选择就可以揭示出我们是谁，以及我们像谁。

早期社交图

在20世纪30年代，一组社会学家和种族学家做了一个很小的“数据挖掘”实验。他们的实验目标是描绘出美国南部小城镇的一组女人的社交结构图。他们使用的数据集是当地报纸上公开发表的数据。该数据集很小：18个女士参加14个不同的社交活动。

他们在想：我们能否弄清这组女士的社交结构（我们称之为社交图）？为了这个目标，他们提出了以下问题：

- 谁和谁是朋友？
- 她们属于哪些社交圈？

谁在社交圈中起到了关键作用？

识别网络结构通常会涉及“攻击性”的采访和调查。是否有可能只通过检视公共行为来推导出网络结构？真正的问题是：人们所做出的公开的选择能否揭示你是谁以及你像谁？能够看透人类系统、组织和社区内部真正的关系，是理解不同群组如何交往及其成员的行为方式的核心。社区网络分析（SNA）是当前流行的一门社会科学，它可以用于市场营销、改进组织有效性、构建经济网络、追踪疾病爆发、揭露欺骗和腐败、分析在线社交网络中发现的模式以及干扰恐怖分子的网络。SNA技术还可以揭示“南方女性”数据集中的基础网络结构，我们很快将对此了解更多。

SNA在20世纪早期作为社会人际学的方式产生。Jacob Moreno[译注1]对他所在的学校的朋友关系（或称社交图）的绘图在社会学历史学家之间很流行，商业学者开始转向20世纪早期对著名的Hawthorne工厂的工人关系[译注2]，以及后期的“Bank Wiring Room”员工间的交互关系的研究。图7-1说明了“Wiring Room”中员工间的朋友关系的连接图。

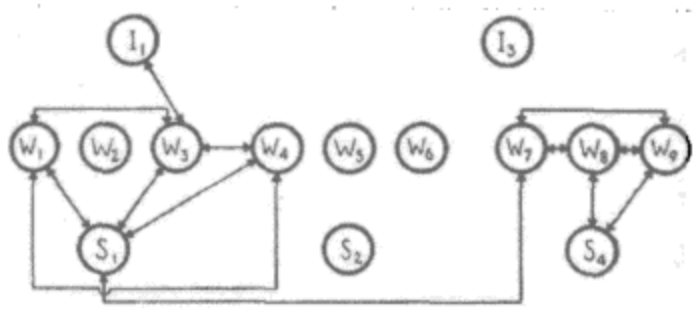


FIGURE 43
FRIENDSHIPS

图7-1：20世纪早期对员工间工作流的研究的社交图

SNA把人类系统映射成节点和连接。节点通常代表人，连接用于描述人们之间的关系或者交互的流。连接是有向的。如果节点只有一种类型——举个例子，如Moreno的朋友关系和对Hawthorne的“工人”研究中的节点，所有的节点都代表人——这种方式被称为单模式分析。

然而，“南方女性”研究初始于一个稍微复杂一些的社交分析：双模式。有两种节点——人物和活动（事件）——连接表示哪些人参加了哪些活动。图7-2显示了包含了两种数据模式的社交图。左边的蓝色节点表示受研究的女性，而右边的绿色节点表示参加的每个活动。用圆圈表示人，方框表示活动。

译注1：Jacob Moreno是一名社会科学家，他是心理疗法的创始人。更多详见：*http://en.wikipedia.org/wiki/Jacob_L._Moreno*。

译注2：1924年至1932年，人们对Hawthorne的工人进行研究，发现了工业管理上的霍桑效应（Hawthorne effect），即工人等会因受到研究人员的关注而增加产量或提高成绩。

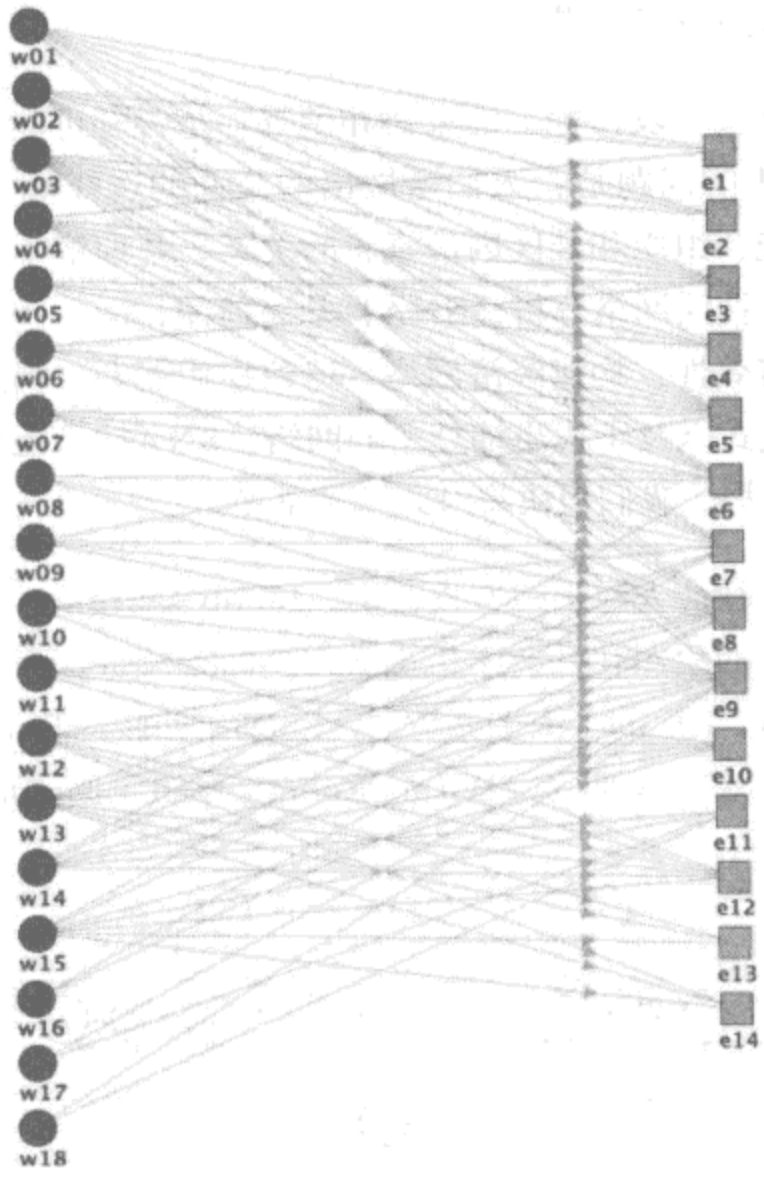


图7-2：“南方女性”社交活动数据集的双模式视图（见彩图50）

该图可以得出不同类型的结论，比如：

- 3号女士参加的活动多于18号女士。
- 参加8号活动的人数最多。

除了这些简单的现象，双模式视图很难揭示出其他任何明显的模式，比如这些女士的社交结构或者不同活动之间的关系。为了进行更深入地观察，我们使用一种流行的社交网络分析技术：把节点转换成连接，这种技术可以把双模式数据转换成单模式数据。在第一次转换中，我们将把活动节点转换成连接：

> X女士和Y女士所对应的节点是连接的，因为她们都参加了活动Z。

两个女士一起参加的活动越多，她们之间的关联就越强。我们还可以把重点转移到活动网络：

> 如果有个女士C同时参与了活动A和活动B这两个活动，那么A和B两个活动节点之间将有一条连接。

参加两个活动的女士人数越多，则这两个活动之间的关系就越强。当把双模式网络转换成单模式网络时，有很多方法可以计算节点之间的连接的强度。在这个例子中，我们使用了最简单的方法：对共现度求和。

活动网络如图7-3所示。两个活动之间的关系越强，其线条越粗。也就是说，参加这两个活动的女士越多。SNA软件的网络组织方式是使用改进的图形布局算法来确定两个人之间的连接关系：网络中一个节点的位置是通过它的连接以及这些连接的连接决定的。

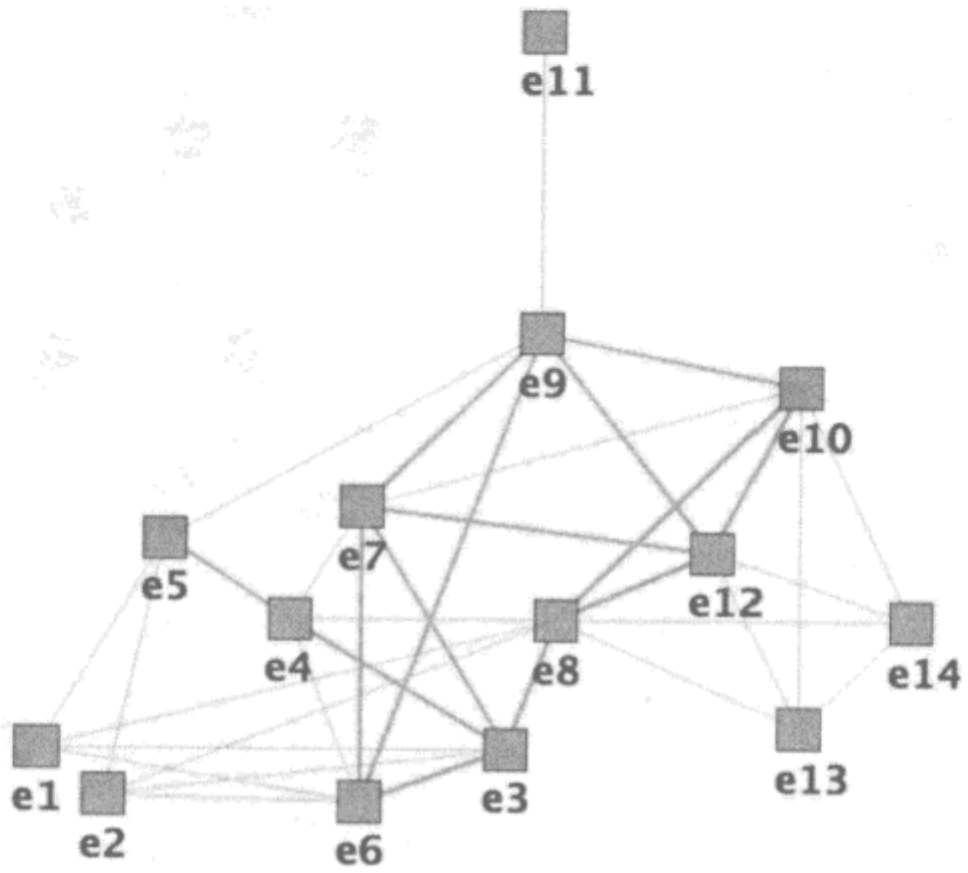


图7-3：基于人们共同的参与数建立起来的活动布局图

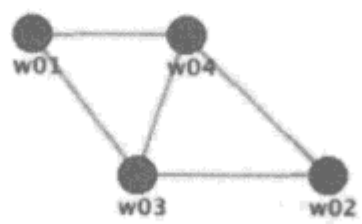
连接度较高的节点位于布局图的中心，而连接度较低的节点则在图形的四周。因此，在社交日历表中，哪些活动是最重要的可以一目了然。然而，到目前为止确实还没有一张图片展示了我们最感兴趣的事情：这个小城镇的自然形成的社交网络。为了探索出这个网络，我使用了“逐步纳入”的方式，首先专注于结构图中最强的关联，然后逐渐降低阈值来找出网络中的弱关联，允许更多人和已经存在于结构图中的人进行关联。这种方法通常忽略数据中的弱关联，而把它们作为社交网的噪音数据而排除掉。在这种方式中，小的数据集上的弱连接的排除操作必须十分小心。而在有数百万节点和数百万种选择的数据集中，调整社交噪音数据的条形图往往就不需要十分精确。

采用五分制，5表示两个节点之间的连接最强，1表示最弱，开始使用逐步纳入的方法，从强度=5的连接开始。换句话说，识别出参加活动最多的女性。图7-4说明了基于活动出席率的最强连接。

我马上就看到了两个聚类：一个聚类包含了1号、2号、3号和4号的女士，另一个聚类则

包含12号、13号和15号的女士。我使用两种不同的颜色对节点进行着色，从而区分开每个聚类分组的成员。

接下来，包含下一强度级别的连接：强度=4的连接。其结果是每个聚类内部各自增加了一些新的节点，但是不存在能够将两个聚类连接起来的节点。如图7-5所示的，我们还是只有两个完全独立的分组。



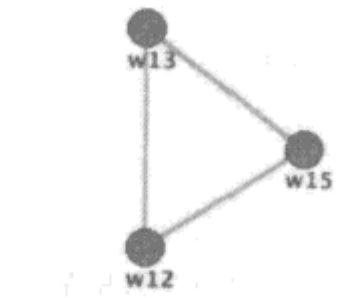


图7-4：基于同一活动出席率的女士之间最强关联（见彩图51）

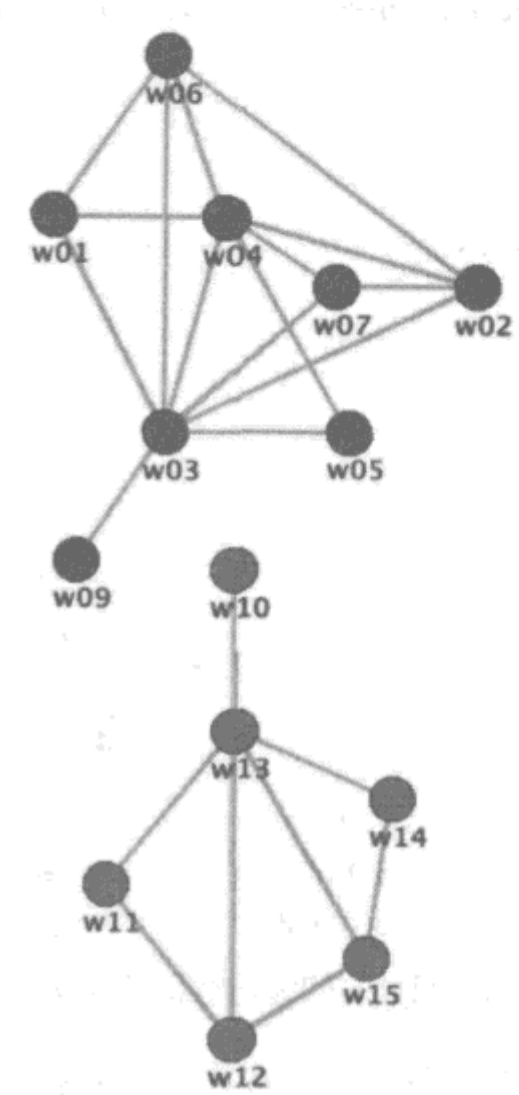


图7-5：参加相同的社交活动的女士之间强度最高的两级连接（见彩图52）

包含强度=3的连接之后，显示出将两个分组桥接在一起的连接，如图7-6所示。以下情况在绝大多数的社交结构图中是很常见的：强度最大连接出现在一个分组内部，而强度较弱、频率较低的连接出现在两个分组之间。在每个分组内部还存在一些强度更弱的连接，说明在一个给定分组内，不是所有的人都和这个组的所有节点都有强连接。

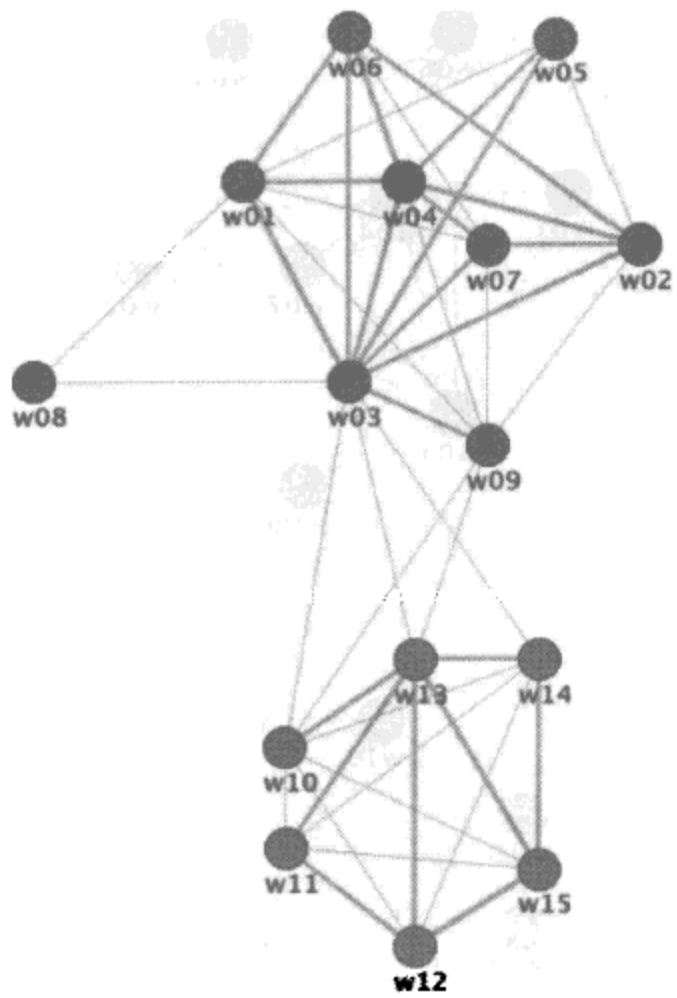


图7-6：通过对弱连接“逐步纳入”进行桥接的两个分组（见彩图53）

我们的社交结构依然缺乏一些节点：16号、17号和18号女士。使用逐步纳入算法，这些节点不满足之前给出的任何纳入标准。或许这3位女士是城镇中新来的，或许是她们较不善于社交，参加的活动较少，导致确定她们的关系更复杂。当我把阈值降低到强度=2的连接时，这3位女士也被连接到社交网络中。现在，所有人都连接到了网络中，而最初的两个聚类结构还保留着。16号女士是唯一的不能明显归属于某一个聚类的节点；她对两个聚类的连接都同样不频繁。因此，我把她归于不属于任何一个聚类，用紫色表示。最终的自然社交网络图如图7-7所示。

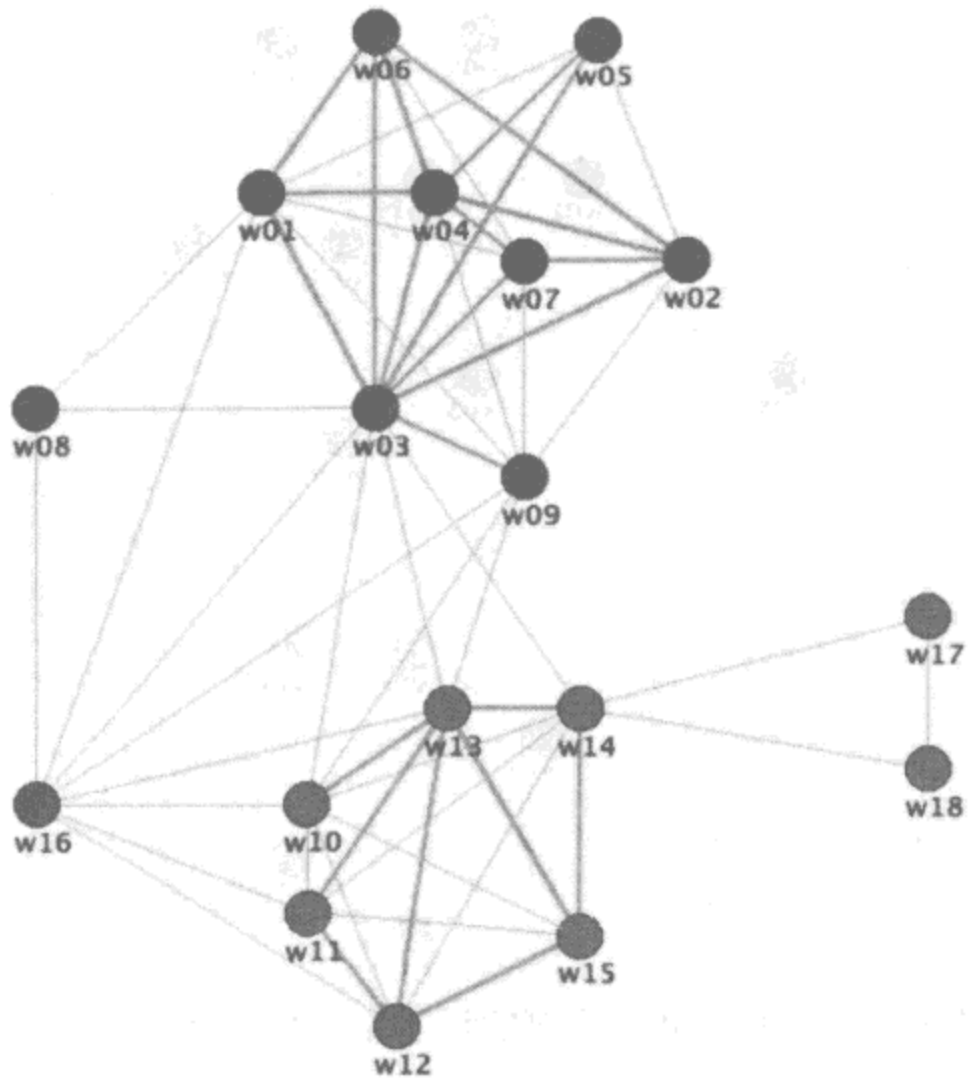


图7-7：基于在社交活动中共同的出席情况建立起来的女士社交图（见彩图54）

基于当地社交活动的出席率，所有18个女士都已经被相应地置于社交网络中。该社交网络揭示了和该小镇相关的社交结构的一些有趣的方面：

- 存在两个显著不同的社交聚类。
- 两个聚类之间是连接的。社交重叠说明了两个聚类之间的兴趣和关系存在一些可能的共同性。
- 产生各种不同的网络角色。有些女士起着连接作用，对两个聚类有桥接作用，而其他女士则表现为是聚类内部的核心成员，只和组内的成员有关联。

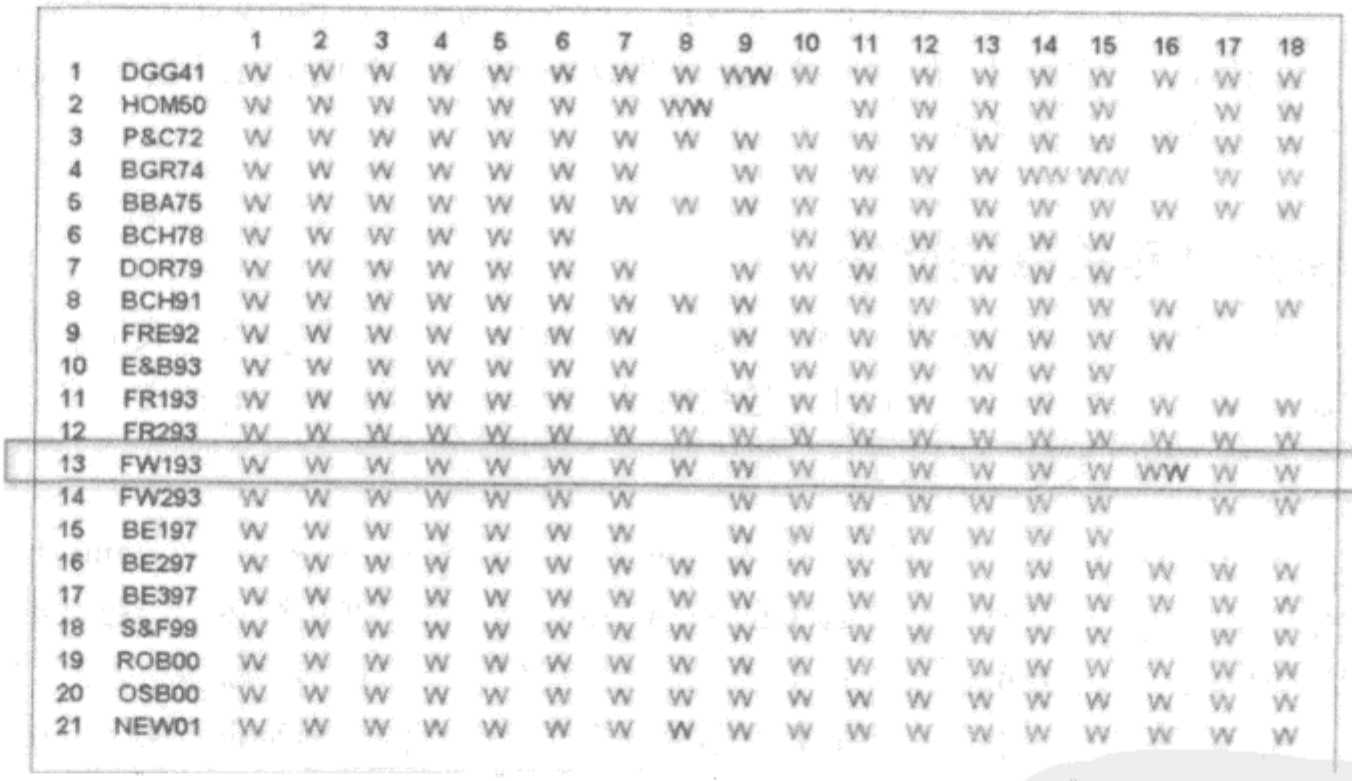
如图7-7所示的社交图可以用于市场营销或者口头传播活动。在该简单的例子中，除了可以收集到给出的这些信息之外，通常可以收集到更多的信息，但是仅仅从这些仅有数据中我们依然能够推导出一些信息。

- 6号女士可能不会受到12号女士的言谈举止的影响。
- 4号女士可能在蓝色聚类内有最高的内部影响。她可能增强了其所在分组内的当前每个成员之间的关联强度。
- 蓝色分组的9号女士是“黏合剂”，即对两个分组起桥接作用，而且可能给分组带

来新的思想和观点。她至少和分组内的一个成员（4号女士）有很强的关联，而3号女士在其所在的分组内又起着重要作用，这对于9号女士很有好处。给一个分组带来想法的人通常需要至少有一个在分组内起着关键作用的同盟。

- 16号、17号和18号女士可能是小镇新来的，或者不是“黏合剂”。她们可能知道分组内发生的事情，但是她们可能不清楚分组内部真正的私密信息，因为她们和各个分组的连接都很弱。

不同的数据挖掘算法通常会产生不同的结果，即使是对于如上所述的小的数据集。在过去几年，不同的社会学家和网络科学家重新检视了这个有趣的小数据集，应用新数据集来查看出现了什么模式。图7-8显示了21个最流行的研究结果。我们的结果和13号Linton Freeman 的研究结果相匹配（Freeman 2003）：1号～9号的女士在一个分组，10号～15号和17号、18号的女士在另一个分组，16号女士同时属于两个分组。Freeman在建立社交网络分析（Freeman 2004）中起到了关键作用，而且在建立一些早期的网络衡量标准上所在的工作尤其重要，这些标准至今还很流行（Freeman 1979）。



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	DGG41	W	W	W	W	W	W	W	W	WW	W	W	W	W	W	W	W	W	W
2	HOM50	W	W	W	W	W	W	W	WW			W	W	W	W	W		W	W
3	P&C72	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
4	BGR74	W	W	W	W	W	W	W		W	W	W	W	W	WW	WW		W	W
5	BBA75	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
6	BCH78	W	W	W	W	W	W				W	W	W	W	W	W			
7	DOR79	W	W	W	W	W	W	W		W	W	W	W	W	W	W			
8	BCH91	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
9	FRE92	W	W	W	W	W	W	W		W	W	W	W	W	W	W	W		
10	E&B93	W	W	W	W	W	W	W		W	W	W	W	W	W	W			
11	FR193	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
12	FR293	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
13	FW193	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	WW	W	W
14	FW293	W	W	W	W	W	W	W		W	W	W	W	W	W	W		W	W
15	BE197	W	W	W	W	W	W	W		W	W	W	W	W	W	W			
16	BE297	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
17	BE397	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
18	S&F99	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		W	W
19	ROB00	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
20	OSB00	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
21	NEW01	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

图7-8：网络科学家对“南方女性”社交活动数据集的研究结果（Freeman 2003，见彩图55）

大多数的研究得出的结论都是很一致的，所有的研究都发现数据中有两个很不同的聚类。然而，对于哪些成员属于哪个分组并不是完全一致，尤其是8号～18号的女性。该表可以很好地显示成员分组，但是它无法揭示网络角色和社交距离。图7-7所示的社交图确实清晰地显示了社交结构的细微差别，显示了网络中的失败点——也就是说最可能发生故障的点。举个例子，如果把3号女性移开，网络将会有很大变化。查看4号女性和9号女性对于3号女性的离开将会如何反应会很有意思。

Amazon的书籍购买数据的社交图

Amazon.com允许用户轻松容易地访问网站以获取经过汇总的购买数据（对交易数据进行聚集，防止对个人信息的识别）。Amazon提供的书籍购买数据形成了和图7-3的活动网络类似的网络数据集。在Amazon网站，人们不是参与相同的社交活动，而是通过购买相同的书籍而相互关联。在这两种情况下，人们被关联在一起的原因都是因为有些人和另一些人做出了相同的选择。

在每个商品页面，Amazon都提供以下信息：

> “购买该商品的顾客还购买了……”

当人们购买两个商品时，在这些商品之间就形成了关联。人们购买相同的商品越多，这些商品之间的关联就越强，同时购买相同商品的概率也就越大。虽然通常情况下是用节点来表示人，但是在这个案例中，Amazon的顾客是用网络中的连接表示，而他们购买的商品是用节点表示。因此，Amazon能够生成一个网络，可以提供其顾客选择和偏好的显著信息，而不会暴露关于顾客的任何个人数据。该网络揭示了顾客的偏好模式，同时还保留了其隐私性。只需要很少的数据挖掘和一些数据可视化，我们就可以对Amazon的顾客的习惯和选择有很深的了解。

确定特定书籍关联的网络

人类网络的一个最基本的规则是“物以类聚，人以群分”。朋友的朋友变成朋友，同事的同事成为同事。在整个社交圈产生了连接的紧密聚集。对于可视化的社交网络，我们发现其中存在“物以类聚，人以群分”的情况。

我们一起来查看Amazon上一本流行的计算机书籍：Toby Segaran和Jeff Hammerbacher的《数据之美》。除了其他信息，该书的Amazon网页提供了书籍描述、出版详细信息和“同时也购买了”书籍的清单。这个清单给我们提供了关于该书的哪些信息？作为一个网络方面专业的学生，我对该书的好奇不仅仅在于该页面的“同时也购买了”的清单（作为网络中的第一层子节点）。我想知道如果我点击清单中给出的每个书籍的链接，并将新页面中的“同时也购买了”的书籍链接添加到一个网络中，将会发生什么情况（作为网络中的第一层和第二层子节点）。

对网络的动态性的了解关键在于能够感知到围绕这个单个节点的连接所具有的关联模式，或者是在一个具有相同兴趣的社区内部或者周围的关联模式。我希望能够弄清我的书籍的兴趣点所产生的关联网络。了解这些关联可以对网上邻居（围绕该书的网络）有深入的理解，它可以帮助顾客做出更明智的选择。

当研究自学习网络时，从焦点节点向外扩散追踪网络中的两层节点是社交网络分析的一个通用步骤。自学习网络允许人们查看谁是他们的网上邻居，他们是如何相互关联以及该结构可能如何影响到人们自身——焦点节点。

当我收集了《数据之美》的“同时也购买了”的书籍列表，我在思考：

- 在书籍以及书籍和书籍之间的关联中，我能够发现什么主题？
- 《数据之美》这本书的读者还对其他什么主题感兴趣？
- 《数据之美》最终是否可能成为庞大的、广泛关联的聚类的中心，或者成为一个具有其中某种兴趣的一个独特的社区的一部分？

图7-9显示了与《数据之美》这本书相关的书籍所连接起来的网络。每个节点表示顾客在Amazon上购买的一本书。通过一条灰色线条把顾客所购买的书籍连接在一起，其中箭头指向“同时也购买了”的书籍。红色节点表示O'Reilly出版社出版的其他书籍，而黄色节点表示其他出版社出版的书籍。

在这个网络中，一本书的优势不在于其拥有的关联的数量，而在于这些关联的指向。网络的黄金规则和房地产是相同的：位置、位置、还是位置。在房地产，真正重要的是物理位置：地理位置。在网络中，则是虚拟位置，由围绕节点的连接模式决定。

图7-9的节点通过连接到“同时也购买了”的书籍，在图形空间中具有了自组织性。这种特性使得相似的书籍可以自组织在一起形成相似主题的聚类，它揭示了在这些书籍聚类背后的兴趣社区。在图7-9中，两个分组很明显地通过主题紧密关联：

- 图的右下角分组都是关于程序员和编程。
- 图上方的分组是关于语义Web。

虽然图7-9中出现了聚类，但是这些聚类没有我们将要看到的聚类那么明显，这些聚类之间相互混合、交叠，尤其是那些关于现代编程方法和过程的书籍。

在图7-9中，除了相似主题的聚类，还存在关于出版社的聚类，由彩色节点表示：红色书籍连接到其他红色书籍，黄色书籍连接到其他黄色书籍。这意味着喜欢O'Reilly出版社书籍的人们倾向于购买O'Reilly出版社的书籍。在节点尺寸上，大小相似的节点形成弱连接模式。尺寸大的节点，在图表中不受局部影响，连接到其他尺寸大的节点，而中等大小尺寸和小尺寸的节点通常相互连接。这是我们在人类网络中经常看到的一种模式——“物以类聚，人以群分”。虽然我们看到的模式并不是Internet的物理结构，但是很多小的节点连接到一些大的节点上，生成一个明显的星形模式。人们通常把该模式称之为无尺度网络（scale-free network）。

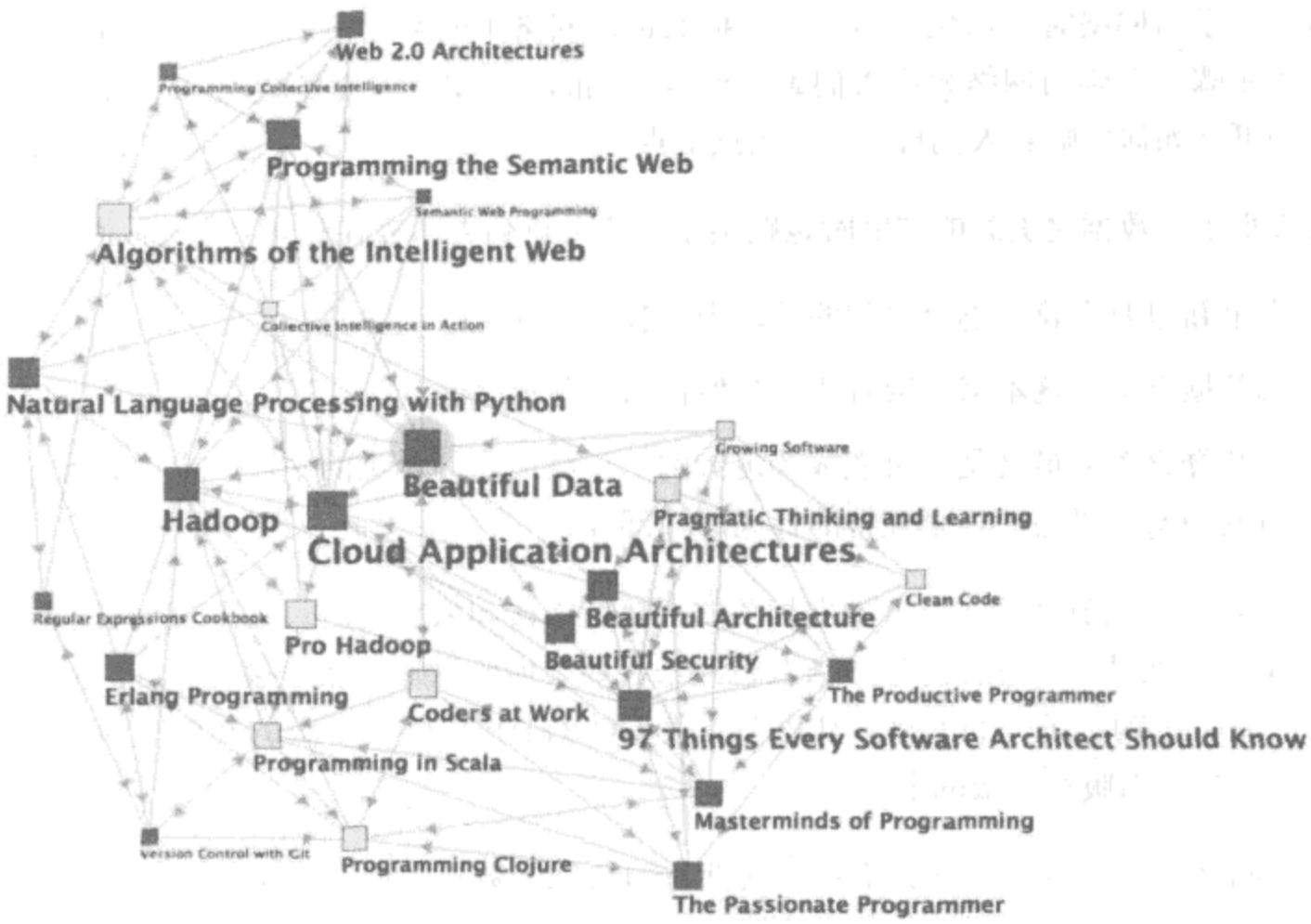


图7-9：《数据之美》一书的“网上邻居”（见彩图56）

下一步，为了查看哪个节点在连接网络中的位置良好，我检查了每个节点/书籍的网络度量。因为这是一个有向网络，和万维网很相似，我采用类似Google的PageRank来计算影响指标。这些指标是通过同时使用每个节点的有向和无向连接来计算的。正如在Web上，连接更重要的节点产生的影响也越大。这些衡量尺度并不能说明销售量或者数量所能表达的流行度；相反地，它们表达的是成千上万的Amazon顾客的同感——“物以类聚”的书是哪些。基于“同时也购买了”的模式的兴趣社区中，节点越大，其在社区中的影响力也越大。

另一种常见的网络测量方式是“结构等价性”。这种测量措施揭示了哪个节点在网络中发挥了相似的作用。等价的节点在网络中可能是相互可取代的。作为一个作者，我不希望自己的书能够被很多其他书籍所替代。然而，作为一名读者，我喜欢有多种选择。在图7-9中，和《数据之美》一书的连接模式最相近的两本书籍是《云计算架构》（Cloud Application Architectures）和《Programming the Semantic Web》。

Amazon提供的另一个增值服务是用户提交的书评。考虑购买特定书籍的读者可能会从其他读者提交的很多书评中受益。遗憾的是，这些书评可能分布很不均衡：一个拥有庞大的个人社交网络的作者，其在Amazon上的最新著作能够快速得到几十个甚至更多的书评，而不受欢迎的作者可能刚好相反。因此，仅仅基于读者的评价来购书可能会产生误导作用。

一本书的网络图比起读者的个别书评可能更能说明应该购买哪些其他书籍。连接到很多其他相似书籍的书能够揭示出花钱购买这些书的读者的很多客观性选择。当然，购买这种行为不是随机的；它是基于思考和比较所做出的决定。做出购买决定就是最佳的评论，即使它没有写一个字书评。

我给出的书籍网络图是为了消除网络中的不相关的节点而设计的（即连接度很低的节点）。图7-9所示的网络图显示了“3核网络”，其中每个节点的连接度至少为3的网络。为了达到这个目标，所有入度=1或入度=2的连接都被删除。这些节点生成了其他兴趣社区，它们表示新书或者非常老的书籍，或者包含“3核网络”社区中很少的“同时也购买了”的连接。

把结果付诸实践

这些兴趣社区地图通过其他消费商品也可以在相似的范围内工作。如果我对一项产品、一位作者、一名艺术家、一个年份、一个品牌、一部电影或者一首歌曲不熟悉，我希望能够通过其“同伴”——它的“网络邻居”来判断。以下是需要问的关于该节点的一些相关问题：

- 什么节点指向该节点？
- 它属于哪个社区？
- 它在社区中起核心作用吗？
- 它在社区中起桥梁作用吗？
- 是否存在等价替代品？

看起来，作为Amazon的顾客，我可以通过查看页面中嵌入的信息来做出更明智的决定——网络内部的“上下文”——Amazon销售的不同的兴趣社区的各种商品。其他厂商，比如Netflix公司和Apple公司的iTunes，可能在推荐一部电影或者一首新歌或者一名新艺术家之前也做类似的分析。通过收集成千上万的顾客信息以及他们所做出的选择信息，把这些信息组织起来，卖家就可以形成如图7-9所示的“产品-产品”的网络，甚至类似图7-7所示的“人-人”网络。这两张映射图都表示了可能的影响模式，以及对顾客做出“购买/出租/下载”的原因。

以下是我们根据Amazon分析提取的一些网络经验规则：

- 如果有两本结构相同的非小说类书籍，你阅读了其中一本后，可能不会急于阅读第二本，因为第二本书所覆盖的信息很可能和第一本书相同。另一方面，对于小说类书籍，你可能希望阅读大量的结构相同的这类书籍（对于这些网络“惊悚片”总是乐此不疲！）

- 如果你喜欢A、B和C三本书，想读一些类似的书籍并找到哪些书籍同时连接到A、B和C。你只能通过网络图看到这些连接；无法在Amazon的单个列表中看到它们，除非你打开3个浏览器窗口，并且自己比较这些列表。
- 如果你想阅读一本关于主题X的书籍，找到在主题X的书籍聚类中，网络影响分值最高的书籍。这遵循Google的PageRank方法，而且可能找到一本口口相传、优秀的书籍。
- 如果你想要查找的书籍没有现货，那么可以找到一本和该书结构相同的其他书籍。这些书将提供相似的内容，而且可能是有货的。

一本书的作者和/或评论员可以用自己已有的书籍网络知识把一本书放到网络中的某个空隙中。出版商可以对不断变化的书籍网络进行评审，这些书籍网络可能会周期性变化来适应市场。当然，Amazon依然是一个大赢家：它拥有所有的数据，以及目前尚未利用的对数据进行分析和发现所开发的丰富的应用。

政治书籍的社交网络

对Amazon的书籍网络进行可视化不仅可以帮助我们选择购买哪些书籍，而且还为我们在特定兴趣领域内提供对更深远的趋势的深刻洞察。其中一个成熟的、值得探索的领域是政治。Amazon的购买模式往往反映了全国范围的政治信仰和选择的调查结果。

如果Amazon报告相同的顾客经常购买两本书，那这两本书就是有关联的。在通过我的社交网络分析软件InFlow 3.1.[注1]对“同时也购买了”的数据进行填充之前，我不会对节点进行排列或着色。InFlow软件包含一个算法，能够基于每个节点的连接对节点的布局进行排列。一旦该软件找到某种新兴模式，识别出任何聚类，我就会对每个聚类的书籍进行审查，然后观察这些书籍是否会自然地聚集成蓝色、红色或紫色（该着色方案遵从2000年美国总统竞选时期流行的“红色表示保守派”、“蓝色表示自由派”的习惯风格；紫色是红色和蓝色的组合，常常用于描述落在这两个“派别”之间的书籍）。

从2003年开始我就一直从事政治书籍的购买模式的社交网络分析。不出所料，从第一次映射，我就发现两个很不相同的政治聚类：红色表示那些阅读了右倾书籍的，而蓝色表示那些阅读了左倾书籍的。在2003年所做的网络分析中，我发现只有一本书把红色和蓝色聚类连接起来。颇具讽刺意味的是，这本书的名字叫《What Went Wrong》，如图7-10所示。

注1：参考*http://orgnet.com/inflow3.html*。

图7-10：对2003年的政治书籍的划分（见彩图57）

2004年的映射图（见图7-11）是在2004年美国总统竞选之前几个月构建的，有几本书把这两个聚类连接在一起。同样，至少对于销售较好的书籍，左右阵营之间很少存在交叉：每个党派的人似乎阅读越来越多支持他们现有的思想框架的书籍。这并不是说没有同时阅读红色和蓝色书籍的读者，但是这样的读者看起来是少数。我只查看Amazon的畅销书籍，通常情况下也会查看这些书籍的“同时也购买了”的书籍列表，重点查看最频繁和紧密联系的书籍连接（正如人人网络中的强连接）。对Amazon数据的更深入的分析（如果Amazon允许的话）可能会揭示出红色和蓝色书籍中更弱更不频繁的连接。我期望看到少部分人阅读两个党派的书籍——很多可能是在学术行业、教学或者选择了两个党派都介绍和讨论的课程。

我使用2005年到2007年的Amazon数据继续创建这些政治书籍映射图，我依然还是得到同样分明的红色/蓝色划分。书籍会随着时间变化，但是全局的网络模式依然保持不变。该模式连接强度如何？为了对这个问题进行测试，我对自己的数据收集方法进行了实验——连接度强的模式是由于我的测量方法所生成的结果吗？不是！不考虑数据收集方法，只要我遵循为人们所接受的实践方法——比如“滚雪球式抽样”（snowball sampling）（Heckathorn 1997）——其结果就显示了强连接的红色和蓝色聚类。有时不同的方法会导致一些新的书籍混入其中，但是全局模式还保持稳定。出现的政治书籍网络模式对于数据收集方法和截断并不敏感，意味着该模式是强模式，而且具有持久性。

2008年，随着美国总统大选的临近，我决定对政治网络捕获若干快照。随着大选日越来越近，网络会如何变化？我从3个关键时刻捕获网络：

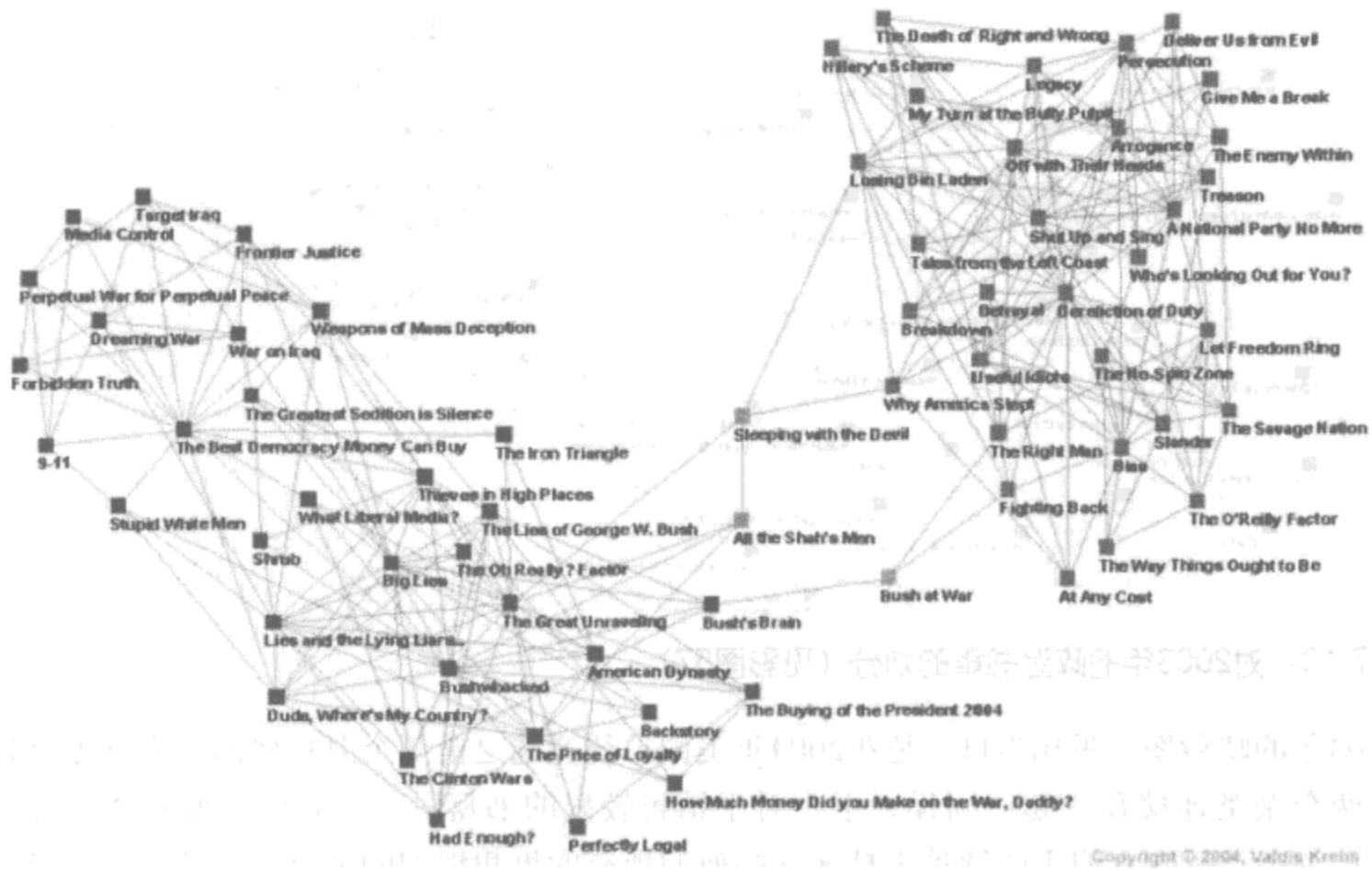


图7-11：对2004年政治书籍的划分（见彩图58）

- 在初选结束时。
- 在最后一场拉选票大会后。
- 在11月选举日临近前。

我预计红色/蓝色划分还会持续存在，但是不知道在总统选举过程中是否会出现有趣的模式。

在2008年6月，在初选确定各个政党的主候选人后，我采用了党派政治辩论的预测模式。在2008年1月的Iowa党团中，奥巴马表示：“我们不是一个由红色和蓝色表示的州的集合，我们是美利坚合众国。”而麦凯恩宣传其紫色表示的“独立”的根源。但是书籍数据会给我们提供什么信息呢？

图7-12是在2008年6月创建的。作为一个小实验，我增加一种新的颜色：浅蓝色。根据Amazon的销售数据，这些书籍和其他蓝色表示的聚类有交集。但是查看这些书籍的标题和作者，它们并不适合普通的蓝色主题和先前迭代的蓝色节点。在该时间点，比起红色表示的书籍读者，流行的保守派、独立派和自由派都和蓝色表示的读者有更多的连接。只有George Will把红色节点表示的人们和美国政治界的其他人桥接起来，而“老保守派”和“新保守派”之间存在分裂，其中比起“新保守派”，“老保守派”在2008年夏天立场和进步派更一致。

图7-12：2008年6月的政治书籍购买模式（见彩图59）

2008年8月，出现了一些反奥巴马的书籍。一本新的亲奥巴马的书，奥巴马自己作序，也在预发布和Amazon中销售。图7-13显示了谁在阅读这些书籍。那本亲奥巴马的书《Change We Can Believe In》，是坚实地属于蓝色聚类，表示已经购买亲奥巴马书籍的人们也会购买这本书。相似地，反奥巴马的书籍——《The Obama Nation》和《The Case Against Barack Obama》——也主要是那些已经购买了反奥巴马书籍的人们所购买。然而，其中一本反奥巴马的书籍和紫色聚类中的其中一本《The Late Great USA》连接。可能是一些犹豫不决的选民，对当前国家现状不满，一直在阅读该书来确定对奥巴马的看法吗？

对于麦凯恩，无论是支持还是反对，都没有相关书籍列于Amazon的最畅销政治论战书籍之列。人们是否在选举过程中对他已经有足够多的了解，还是人们对他不感兴趣？图7-13所示的书籍之间的连接模式映射图说明了在2008年夏末最有影响力的政治书籍是《What Happened》和《The Post American World》——都没有提到当前的选举！《What Happened》是由布什的前新闻秘书写的，但只有蓝色聚类中的读者购买它。

图7-13：2008年8月的政治书籍购买模式（见彩图60）

社交网络分析和数据挖掘/可视化为我们提供两类成果：

- 预期的和未预期的结果和观点。
- 正面和负面的结果和观点。

这两个分类存在交集，如图7-14所示。在参与的数百个社交网络分析项目中，我发现客户通常最喜欢观看他们没有料想到的结果——未预期（尤其是负面未预期）模式，而且这些模式会引发一些问题。

正面预期	正面未预期
负面预期	负面未预期

图7-14：社交网络分析的“发现矩阵”（见彩图61）

使用图7-14的发现矩阵，我们一起来查看最后一张图。在2008年10月底，随着总统竞选

逐渐接近尾声，我又查看了一下人们购买的政治书籍以及生成的模式。图7-15显示了预选网络图。在该图中出现了一些未预期的模式，以及一个预期模式。

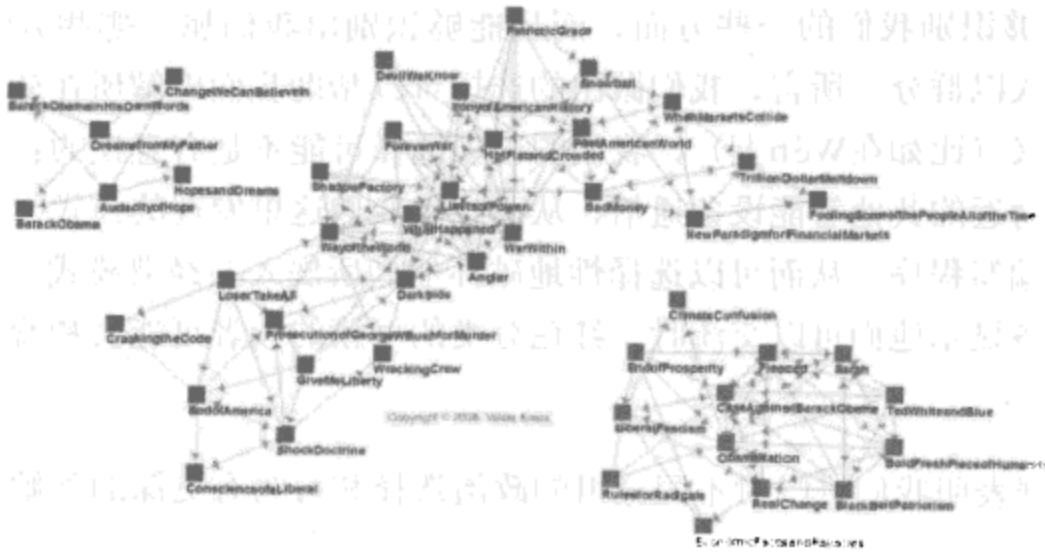


图7-15：在2008年11月竞选前几周的政治书籍购买模式（见彩图62）

和之前所有的映射图不同，在红色聚类和蓝色聚类之间不存在把它们桥接起来的书籍——这两个聚类是完全分开！红色聚类和蓝色聚类没有任何共同之处！这种模式体现了两极分化之间存在的鸿沟，以及在竞选活动过程中表现出的深深的敌意。没有讨论到政治问题和大经济问题。这种模式可以归类为基于每个竞选活动的日常行为的负面预期模式。

图7-15所示的可视化还说明了右倾读者一直购买社区组织者的重要书籍《Rules for Radicals》。而该读者群曾经嘲笑社区组织！为什么右倾读者会购买这本通常只受左倾读者欢迎的书？是否是右倾读者试图找出为什么奥巴马的竞选活动基于社区组织原则能够如此成功？这是一个未预期模式，而该模式应该归属于正面还是负面模式取决于你站在哪一边。

最后一个未预期模式是那些购买亲奥巴马书籍的人们没有购买其他政治书籍。“关于奥巴马”的聚类和包含政治辩论的其他聚类不相连。该模式可能说明这些读者只对奥巴马和本届竞选感兴趣，而不是一般的政治问题。

从本届预选政治书籍网络图中还发现一个预期模式。从2004年开始，注册的民主党人要多于共和党人，因此直观感觉是存在更多的蓝色书籍。相反地，右派专注于更少的书籍来宣传其消息（书籍网络图并不能反映销售的书籍的数量，因此有可能是右派读者实际上购买了更多数量的书籍——我们无法知道，因为Amazon没有给出这些数据。）这可能可以看做两个党派的正面预期模式，但是原因不同。右派可能理解为其方法更集中，而左派可能理解相反，认为缺乏不同的观点。相反地，左派可能正面地评价其书籍种类的多样化，认为表示不同的观点，而右派可能认为它表示信息分散不集中。

结束语

正如本章所给出的可视化所示，我们的选择揭示了我们是谁以及我们喜欢谁。我们做出的决定不仅能够识别我们的一些方面，而且能够识别出我们属于哪些分组。正如谚语“物以类聚，人以群分”所言，我们做出的选择可以帮助我们理解所在分组的其他成员的行为。在将来（比如在Web上），我们的很多选择可能不是有意识的：我们的智能手机可能可以和附近的其他智能设备通信，从而找出我们这里发现的模式。少数大胆的人可能会为设备编写程序，从而可以选择性地破坏他们所嵌入的经典模式——举个例子，当两个人的设备显示他们可以交流时，红色分类的书籍的读者可能会和蓝色分类的读者交谈。

Amazon的数据表明我们可以对不同分组的政治选择和行为有更深的理解，而不需要知道属于这些分组的任何个人信息。不需要透露任何私人数据，我们就能够理解基于书籍购买的大规模的政治模式。更让人惊奇的是，这些数据和用于显示它所创建的简单的可视化，与代价很高的全国范围的选民调查一致。花费一个小时对Amazon数据的收集和映射能够使我们获得一些和花费数千小时收集和分析选民调查和采访数据一样的洞察。Pareto的“80/20法则”[译注3]在此很适用：我们获得了80%的洞察，而花费的时间远远少于20%——合理结合数据挖掘和数据可视化的高回报！

参考文献

1. Davis, Allison, B.B. Gardner, and M.R. Gardner. 1941. *Deep South: An Anthropological Study of Caste and Class*. Chicago: University of Chicago Press.

2. Freeman, Linton C. 1979. *Centrality in social networks: I. Conceptual clarification*. *Social Networks* 1: 215–239. *http://moreno.ss.uci.edu/27.pdf*.

3. Freeman, Linton C. 2003. “Finding social groups: A meta-analysis of the southern women data.” In *Dynamic Social Network Modeling and Analysis*, eds. Ronald Breiger, Kathleen Carley, and Philippa Pattison. Washington, DC: The National Academies Press. *http://moreno.ss.uci.edu/85.pdf*.

4. Freeman, Linton C. 2004. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Vancouver, Canada: Empirical Press. *http://aris.ss.uci.edu/~lin/book.pdf*.

译注3：80/20法则，又称帕累托法则，指的是在众多现象中，80%的结果取决于20%的原因。更多详见*http://en.wikipedia.org/wiki/Pareto_principle*。

5. Heckathorn, D.D. 1997. "Respondent-driven sampling: A new approach to the study of hidden populations." *Social Problems* 44: 174–199.

6. Mayo, Elton. 1933. *The Human Problems of an Industrial Civilization*. New York: MacMillan.

7. Moreno, Jacob L. 1934. *Who Shall Survive? A New Approach to the Problem of Human Interrelations*. Foreword by Dr. W.A. White. Washington, DC: Nervous and Mental Disease Publishing Company.

第8章

美国参议院社交图（1991～2009）的可视化

Andrew Odewahn

2009年初，很多新闻报道都在关注两党合作的弊端。尽管绝大多数报道只是典型的“人云亦云”之类的文章，其中一篇文章引起了我的特别注意。《Slate》杂志的副主编Chris Wilson发表了一篇伟大的文章，在这篇文章中，他使用了对亲和性数据进行投票和图形可视化的方式来帮助说明参议员Arlen Specter的“换党”事件（Wilson 2009）。图表显示了两个大的党派聚类（民主党用蓝色表示，共和党用红色表示），两个党派之间通过几条细线连接，这些细线代表了一贯跨党派投票的一些参议员[注1]。Specter正是这些参议员中的一位。

这篇文章让我想到了如下几点：第一，通过定量的证据来说明本质上定性的事情真的很酷。可以一目了然的是，参议员Specter身上正发生一些有趣的事情，预示着他正在背离原来所在的党派。这件事情使我对于新闻报道中的其他事件是否也存在类似的证据感到很好奇。举个例子，很多报道聚焦于各种参议院联盟（“十四人帮”（Gang of Fourteen）、“新英格兰温和派”（New England Moderates）和“南方共和派”（Southern Republicans））以及他们如何力挺或阻挠此提议或彼倡议。

基础公民学知识会使你相信参议院和众议院不同，国家创始人设计它的目标正是为了抑制类似上述情况的联盟。这是一个简单的机构：总共100个参议员，每个州每6年选举两

注1：在这里，需要说明的一点是，“图表”指的是一些节点和边的集合，而不是以(x,y)坐标表示的数据点绘图。

个参议员作为代表。各个州的选举交错举行，因此大约每两年会对三分之一的议员重新选举一次，这意味着参议院联盟会发生变化，但变化不会太剧烈。虽然可能发生参议员们更换党派、退休甚至在任期间去世，这些事件发生的概率很低。最后，任期本身就为参议员带来很大优势。一旦任职，现任参议员们很少会因为投票被罢免。

我对自己是否能够使用图形可视化来描绘出一幅广阔的图像感到好奇，通过这个图像可以显示参议院的组织结构随时间的动态变化情况。如果关于“高校故事”是事实的话，即参议院本质上是一个保守的团体，通俗地说就是倾向于排斥改变，那么这个图形应该会保持相对稳定。如果不是事实，那么可视化展现很可能使人们对2009年发生的塑造了美国的一些难以置信的重要事件以及记者们报道这些事件的方式产生一些深刻的见解。

在本章中，我将介绍如何应用投票数据对这些问题进行可视化探索。首先，介绍生成可视化所需的基本步骤。其次，展示最终结果，讨论在我所研究的时间跨度为18年的期间内图像是如何变动的，并提供一些历史背景信息，对参议院的“高中公民教育”的优点的观点做出一些结论。在此之后，我将讨论为什么该可视化是美丽的（而不仅仅是有趣的），同时探讨在可视化过程中带来的种种缺点。最后，我将分享在完成这个可视化过程之中领悟到的一些收获，希望可以应用于你们的日常工作之中。

创建可视化

我是按照Wilson的文章中给出的可视化基础指南开始工作的：

- 节点代表参议员；每个节点有一个数值标签，一个标签对应一个参议员，参议员是按照字母序进行排序的。
- 节点是基于其对应的参议员的党派倾向进行着色的。采用标准规范，蓝色表示民主党，红色表示共和党。（我还使用了绿色表示独立党，黄色表示原始数据中不包含的党派。）
- 如果两个参议员在选定的时期内投票相同的概率超过65%，他们所对应的两个节点就通过一条边连接起来。

此外，我决定对图表的方向进行调整，这样民主党议员所对应的节点在左边，而共和党的在右边。另外，由于期望可以了解参议院是如何演化的，因此，我根据几个有意义的时间帧对数据进行分段，并为每个分段数据创建一个可视化图形。

我选择使用立法会会期作为基本的时间单元。一个立法会会期持续两年，开始和结束时间都是1月3日，通常被称为“国会”。每届国会都按序进行连续编号。比如，第104届国会历时时期始于1995年1月3日，终于1997年1月3日；第105届国会则是始于1997

年1月3日，终于1999年1月3日。各届国会都依此类推。（在写本章时正值第111届国会期间。）

选用会期作为基本单元有两个原因。第一，它是最短的一致的时间段。参议院是一个动态团体，其成员在任何时候都有可能发生变化，尤其是在选举年份，因此，如果使用超过两年的时间周期，会因为需要根据投票记录中途产生新的参议员而使得关系混乱。第二，更显而易见的是，这个时间周期正是报告数据的周期，因此这是一个非常方便的选择。

完成这些初步选择之后，构建可视化还需要3个步骤：收集关于参议员以其投票的原始数据；计算描述这些参议员的关联度的亲密度矩阵；把信息输入到GraphViz（一个图形可视化工具包），把关系图形化成一个图像。以下各节将深入描述其中的每个步骤。

收集原始数据

我的可视化需要两种主要类型的数据：关于个别参议员的元数据（名字、党派等），以及在一段时期内他们的投票记录。刚开始，由于很多大的政府数据网站（data.gov、thomas.com等）通过订阅发布消息，缺失历史信息看起来是一个主要障碍。国会中的一次特殊投票会被发表，但是难以及时追踪完整的投票记录。

幸运的是，我发现网站GovTrack（*http://govtrack.us*），该网站宣传自己为“追踪国会的人文项目”。虽然它在很大程度上提供和其他大的政府网站一样的数据，它还（除了其他以外）通过非常有意义的聚集函数，将订阅的信息转换成追溯到1991年的XML文件，部分数据可以用于预测本届国会。因此，我的项目包含第102届国会之前的国会的所有记录，但是在1991年前的数据是不完整的。你可以免费从“Source Data”（源数据）[注2]页面下载任意或所有的数据。该网站有非常好的文档说明，清晰地描述了如何下载数据及其结构。

在GovTrack，参议员的元数据保存在文件people.xml中。在这个站点上，该文件存在两种版本：当前文件，包含当前正在国会就职的工作人员信息；历史文件，包含任何曾经在国会就职的人员信息。在这个项目中，我使用的是历史版本。

在这两个文件中，关于个别参议员（或众议员）的信息显示在<person>元素中；每个人有一个唯一ID，在整个GovTrack数据集中，一个人的ID号都保持一致。关于党派的信息是保存在子元素<role>中。举个例子，以下是John Kennedy的数据项，他既是众议员又是参议员（当然，他还是总统）：

注2：　参见*http://bit.ly/4iZib*。

```
<person id='406274'
   lastname='Kennedy' firstname='John' middlename='Fitzgerald'
   birthday='1917-05-29' ... >
   <role type='rep'
      startdate='1947-01-01' enddate='1948-12-31'
      party='Democrat' state='MA' district='11' />
   <role type='rep'
      startdate='1949-01-01' enddate='1950-12-31'
      party='Democrat' state='MA' district='11' />
   ...
   <role type='sen'
      startdate='1959-01-01' enddate='1960-12-31'
      party='Democrat' state='MA' district='' />
</person>
```

GovTrack中的投票数据是按照两年的立法会议组织的。投票时根据唱票来记录的，即当参议员在面临的一个问题上一起投“是”或“否”。在一次会议过程中，通常有几百轮唱票。

GovTrack把每一轮唱票以XML文件形式记录下来。举个例子，下面这个列表是唱票文件*s1995-247.xml*的一段摘录，它是在第104届国会上做出的一轮投票，决定是否由允许贝尔公司提供交互本地访问和传输区（LATA）商业移动服务。（其中一些投票非常无聊。）注意，每个<voter>元素都有一个id，该id可以重新链接到*people.xml*文件中：

```
<roll
   where="senate" session="104" year="1995" roll="247"
   when="802710180" datetime="1995-06-09T11:03:00-04:00"
   updated="2008-12-30T13:34:55-05:00"
   aye="83" nay="4" nv="13" present="0">
   ...
   <voter id="400566" vote="+" value="Yea" state="MN"/>
   <voter id="300016" vote="-" value="Nay" state="WV"/>
   <voter id="400559" vote="-" value="Nay" state="WA"/>
   <voter id="300011" vote="0" value="Not Voting" state="CA"/>
   <voter id="400558" vote="0" value="Not Voting" state="GA"/>
   ...
</roll>
```

这些文件（历史“people”文件和所有的不同种类的唱票文件）包含我想要的所有数据。然而，*people.xml*文件有6MB多的数据，整个GovTrack数据集中有几千轮唱票，我希望这些数据能够以更便捷的格式保存。因此，我写了一些脚本，只抽取可视化需要的部分数据，把它保存到SQLite数据库中。模式如图8-1所示。为了简单起见，我把一个党派基于最近的<role>进行赋值，后来回想时对该决定一直觉得比较纠结。

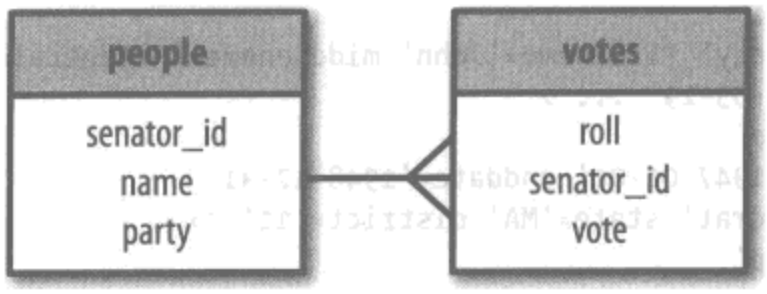


图8-1：表示可视化所需的原始数据的简单的数据库模式

计算投票亲和性矩阵

随着原始数据被糅合成更灵活的格式，我已经准备好计算亲和性矩阵的问题，亲和性可以表示图中的各条边。这需要构建一个亲和性矩阵，如图8-2所示，它可以计算不同参议员做出相同选票的次数。我可以使用该矩阵来替代边界条件。

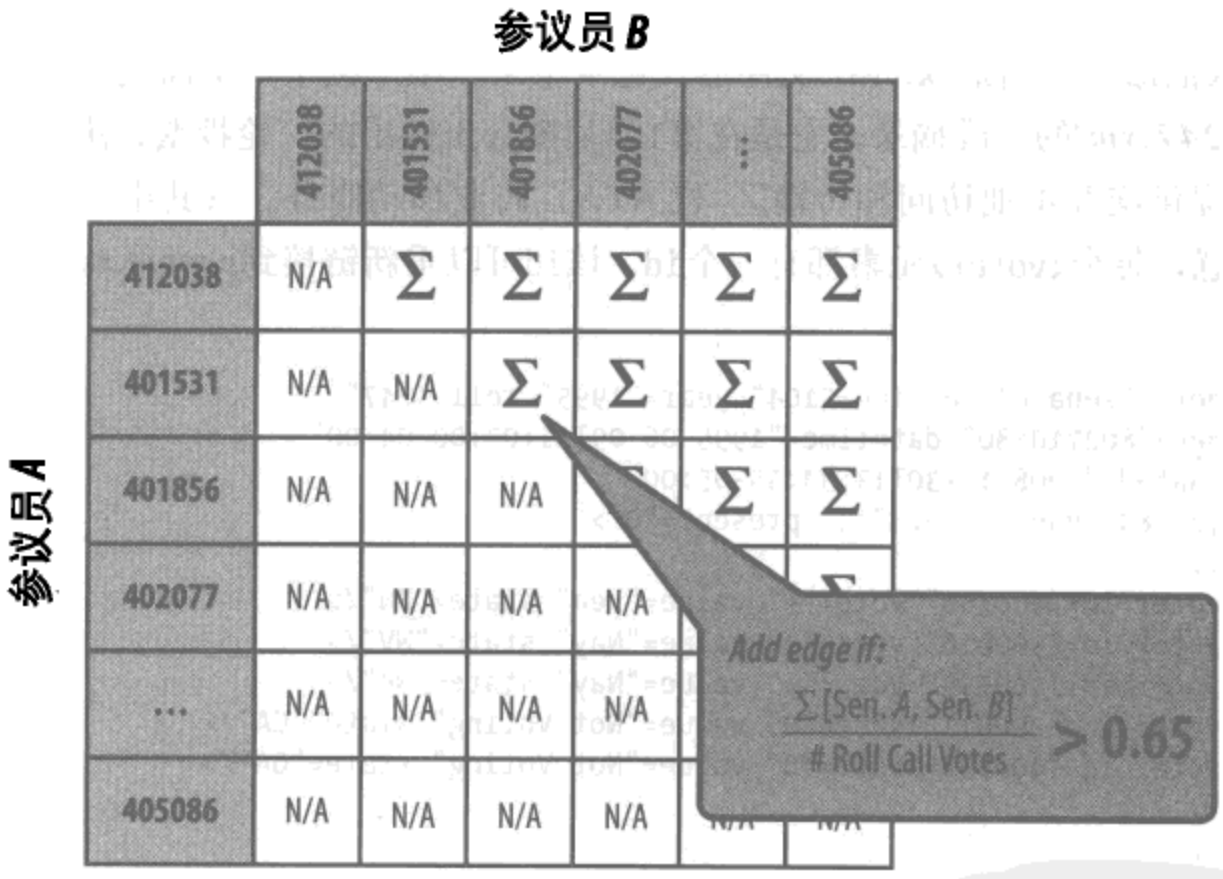


图8-2：亲和性矩阵

以下伪代码说明了基本逻辑：

```
# Select all distinct roll calls from the vote table
roll_list =
   select
      distinct roll
   from
      votes
# Process each roll call vote in roll_list
for roll_idx in roll_list:
   # Process "Yea" votes, then "Nay" votes
```

```
    for vote_idx in ["Yea", "Nay"]:
       # Find the senators that cast this vote on this roll call
       same_vote_list =
          select
             senator_id
          from
             votes
          where
             roll = roll_idx and
             vote = vote_idx
       # Now tally all the pairs of senators in the list
       for senator_a in same_vote_list:
          for senator_b in same_vote_list:
             affinity_matrix [senator_a, senator_b] += 1
             affinity_matrix [senator_b, senator_a] += 1
# Translate the raw matrix into edges
N = length(roll_list) # Represents the number of votes in the session
for senator_a in affinity_matrix.rows:
   for senator_b in affinity_matrix.columns:
      if (affinity_matrix[senator_a,senator_b] / N) > 0.65 then:
         add an edge between Senator A and Senator B
```

因为这是一个相当密集的运算集，我把结果保存在数据库中的另一个表中。

使用GraphViz对数据可视化

最后一步是把所有这些数据——参议员的元数据和投票记录——转化成一系列图片。GraphViz（*http://www.graphviz.org*）是一个开源的图形可视化包，是适合该工作的理想工具。

图形可视化是对各种不同的布局算法的研究，这些算法对图形中的节点和边进行抽象表示，并转化成一张图片。我使用GraphViz的“neato”布局算法[译注1]，其工作方式是通过把节点模拟成带正电的粒子，把边模拟成张力。节点互斥，而边把关联的节点拉到一起。刚开始，所有的节点都是随机置于一个平面上，算法模拟推力和拉力这些制衡来为每个节点计算最终表示“最佳”全局布局的*x*坐标和*y*坐标（由于这个原因，这样的算法被称为“力导向布局”（force-directed layout）算法。）。图8-3说明了该布局算法的思想。

译注1：想要更多了解“neato”布局算法，可以通过以下链接下载其文档*http://www.graphviz.org/Documentation/neatoguide.pdf*。

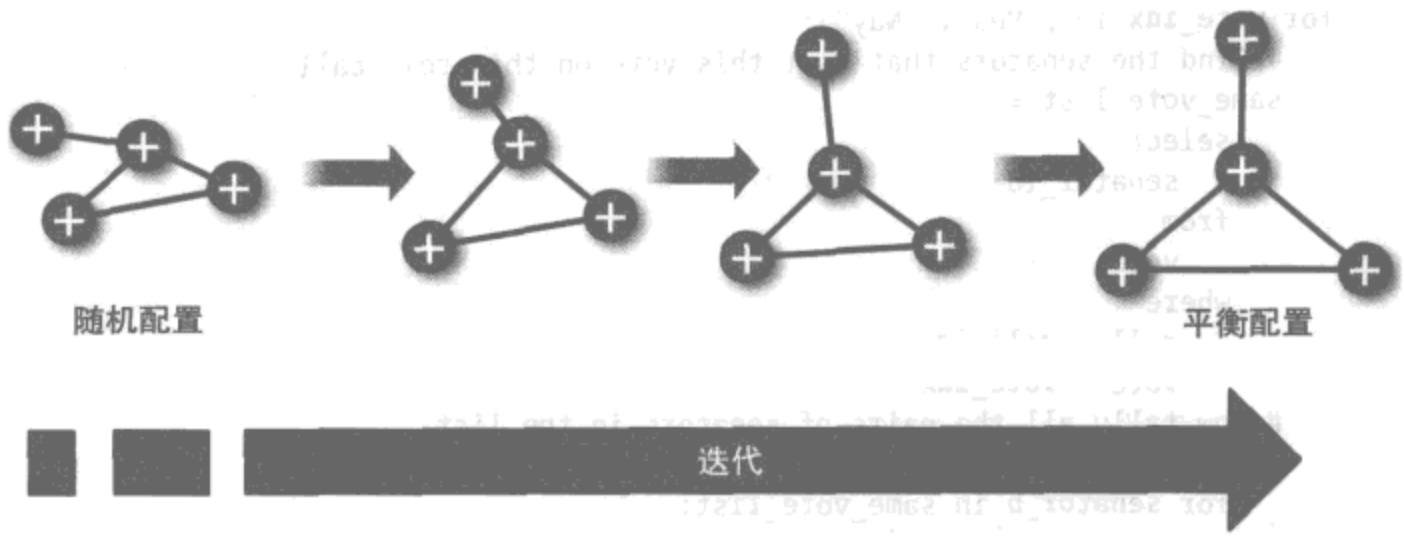


图8-3：Neato，GraphViz中的力导向布局算法，把节点模拟成带正电的粒子，边表示成张力

从该过程产生的结构和基础数据的连接密度成正比。因此，一组紧密连接的参议员应该创建一个子聚类，该子聚类排斥其他子聚类。另外值得一提的是，因为子聚类控制边是否存在，基于选票亲和性分配给边的临界值决定了图中观察到的聚类的程度。一个非常低的值（如20%）将会导致相对较少的子结构，因为一个会议上的很多选票通常都是例行事项，绝大多数参议员都会同意。相反地，一个很高的值（比如95%）将会导致生成碎片很多的图形，因为只有强连接的节点对才会出现；该图看起来就像一个偶尔连接的随机点集合。临界值65%看起来是这些紧张的竞争之间的最佳平衡。

一种称为DOT的语言描述了GraphViz的节点和边。DOT是直截了当的：使用唯一标签表示节点，边是通过使用→标识符连接两个或者更多的节点标签来表示的。各种不同的其他属性（颜色、标签等）是通过把它们放置在其修改的对象的方括号中来定义的。

以下是DOT文件的一个例子（Gansner、Koutsofios 和 North 2006）：

```
digraph G{
    a[shape=polygon,sides=5,peripheries=3,color=lightblue,style=filled];
    c[shape=polygon,sides=4,skew=.4,label="helloworld"]
    d[shape=invtriangle];
    e[shape=polygon,sides=4,distortion=.7];
    a -> b -> c;
    b -> d;
}
```

图8-4显示了在GraphViz中生成的相应图片

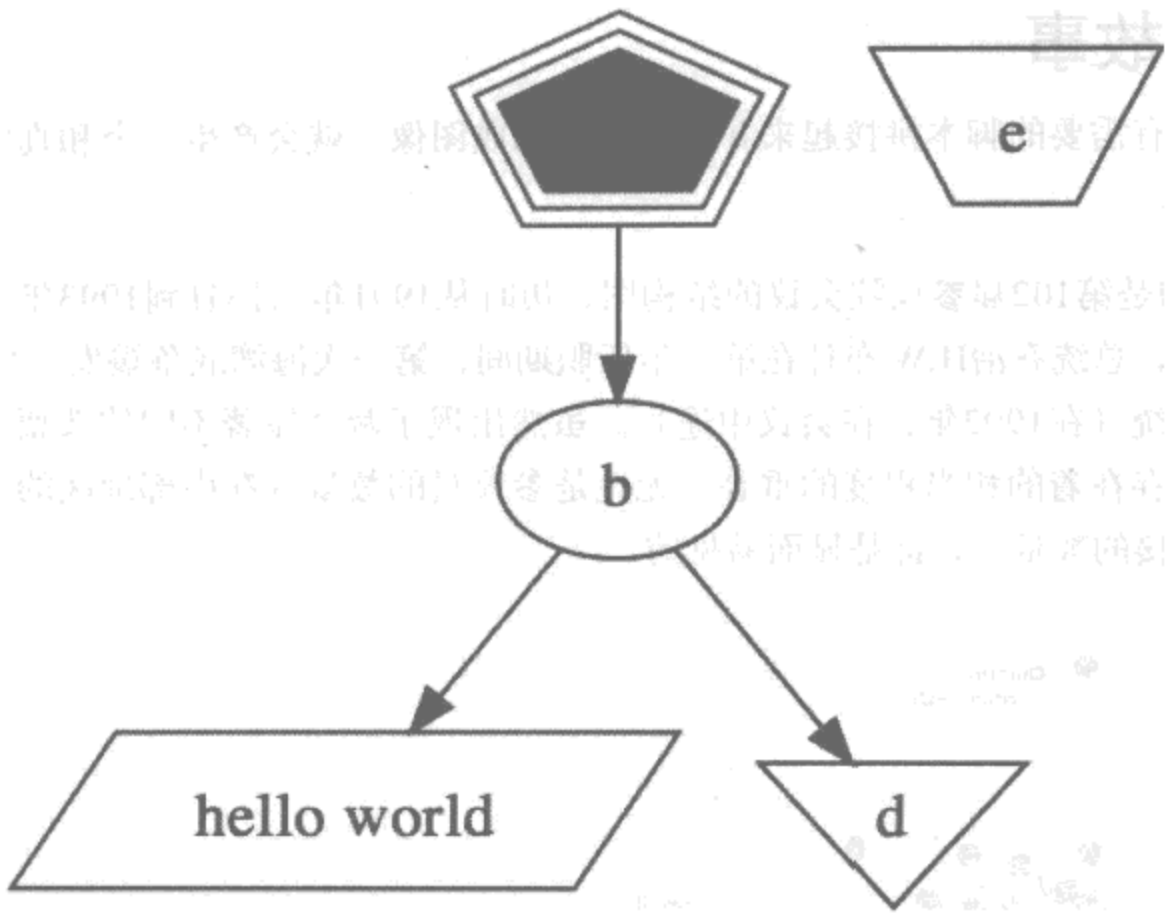


图8-4：GraphViz生成的样本图片

因此，为了对参议院数据创建可视化，我需要创建一个DOT文件，作为GraphViz软件的输入。这需要实现另一个脚本，对所有信息进行打包，保存到之前创建的数据库中——参议员ID、按字母序排列的标签列表、基于党派的节点色彩以及亲和性矩阵中的边——然后把这些数据传给模板引擎，该引擎会生成一个DOT文件来表示。以下是模板：

```
1 Digraph {
2
3 #for $senator in $vote_data.nodes:
4    $senator['id'] [
5       shape="circle",
6       style="filled",
7       color = $senator['color'],
8       label = "$senator['label']"
9       fontsize = "128",
10      fontname = "Arial",
11   ];
12 #end for
13
14 #for $e in $vote_data.edges:
15     "$e['senator_a']" -> "$e['senator_b']" [arrowhead = none];
16 #end for
17 }
```

需要注意的是，第3行和第14行的for循环是用于对节点和边重复进行循环。粗体显示的是在每次迭代中会被取代的变量。

产生的故事

一旦我把所有需要的脚本拼接起来并把它们转化成图像，就会产生一个和真实情况非常一致的故事。

图8-5显示的是第102届参议院会议的结构图，历时从1991年1月3日到1993年1月3日。在这届会议中，总统乔治H.W.布什在第一年任职期间，第一次海湾战争爆发；后来比尔·克林顿当选总统（在1992年，在会议中途）。虽然出现了两个显著不同的选票分块，在中心分块之间存在着的相当程度的重叠，无论是参议员的数量（在中部地区的节点）还是边（交叉连接的数量），都是显而易见的。

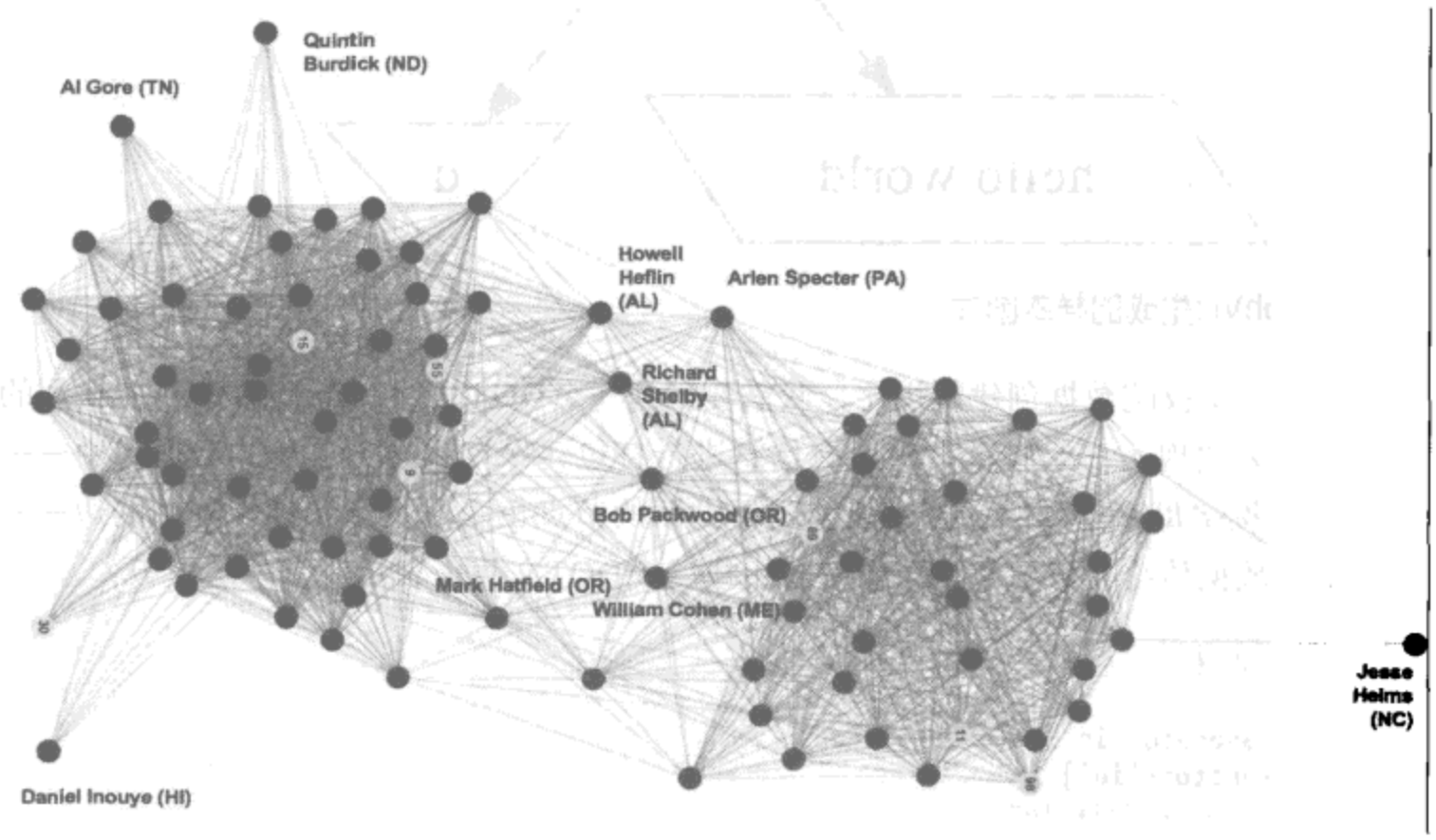


图8-5：第102届参议院会议的结构图（历时从1991年1月3日到1993年1月3日，见彩图63）

图8-6显示了第104届参议院会议的结构图，即仅两年后。该结构图（和前两年的会议结构图）表示“共和党革命”（Republican Revolution），在这期间共和党在近40年来首次重新夺回众议院和参议院的权力。这一时期党派关系非常紧张，经历了政府被解散、按共和党“和美国合约”投票以及在俄克拉荷马城Murrah联邦大楼爆炸案这些事件。参议院的可视化图说明了党派之间存在很深的分歧，两个党派都锁定在分离的、紧密的小圈子中。

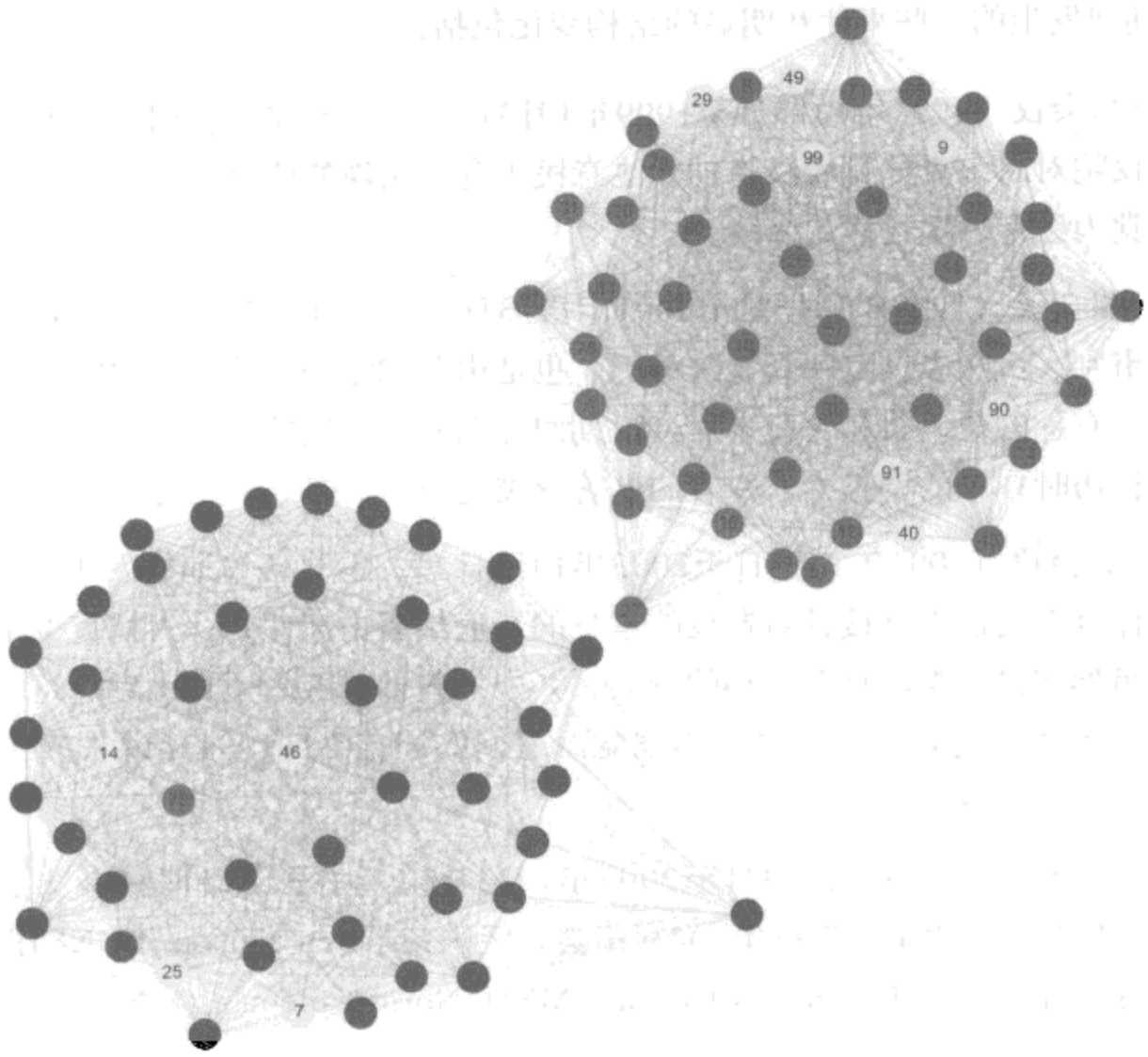


图8-6：第104届参议院会议结构图（从1995年1月3日到1997年1月3日，见彩图64）

图8-7显示了随后六届会议的可视化图形组合。

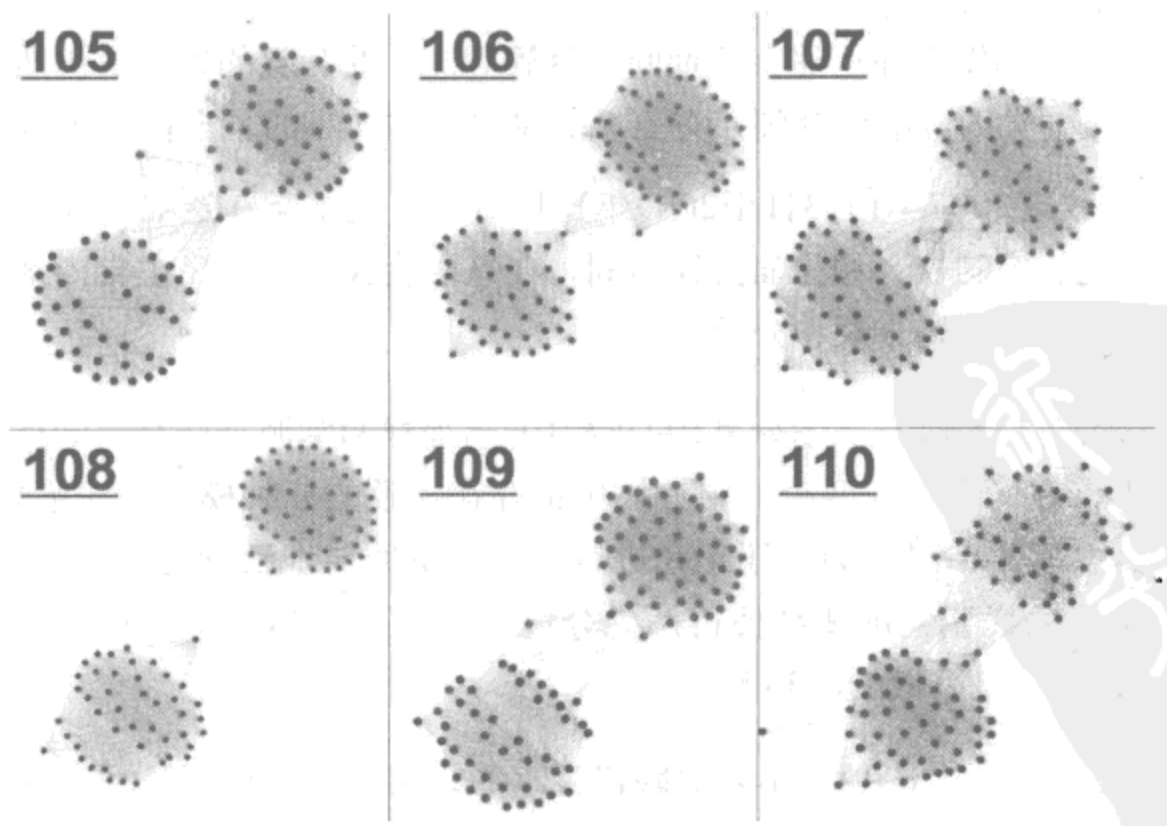


图8-7：从第105届到110届参议院会议的结构图（历时从1997年1月3日到2009年1月3日，见彩图65）

这些会议期间发生的一些事件和明显的结构变化包括：

- 第105届会议（1997年1月3日至1999年1月3日）。在本届会议期间，由共和党控制的众议院对总统克林顿表决弹劾。注意民主党中出现的明显的分裂，在那段时期，民主党内经常出现这样的分歧。
- 第106届会议（1999年1月3日至2001年1月3日）。该期间在参议院对总统克林顿弹劾的审判。虽然参议院和众议院相似，也是由共和党控制的，参议院最终投票无罪释放。有趣的是，共和党在本届会议期间存在界限分明的重大分裂；这是在对共和党进行历时18年的调查中，少有的显著分裂之一。
- 第107届会议（2001年1月3日至2003年1月3日）。本届会议期间发生了“9·11”袭击事件（以及后来直接针对参议院本身的炭疽热袭击案件）；伊拉克战争也授权通过。虽然在民主党内存在很小的分裂，其中一些参议院趋于自由党，这一期间在党派中产生了一股新生的力量，不同党派之间的连接比自1991年以来的任何时候都多。
- 第108届会议（2003年1月3日至2005年1月3日）。本届会议期间爆发了伊拉克战争。本届会议几乎是回退到第104届国会，区别在于Ben Nelson (D, NE)投票支持由Olympia Snowe (ME)，Susan Collins (ME)和Norm Coleman (MN)组成的规模很小的温和共和党派。而其余的共和党依然保持紧密团结，民主党内依然存在小分裂。
- 第109届会议（2005年1月3日至2007年1月3日）。共和党的多灾多难时期——Tom Delay和Jack Abramoff丑闻，Terry Schiavo案例的决裂性投票，以及对卡特里娜飓风非常糟糕的回应（“你干的什么鬼工作！”（You’re doin’ a heckuva job, Brownie!））都发生在这届会议期间。尽管如此，共和党的参议员仍然非常团结。相反地，民主党内部继续分裂，有更多的参议员转向小的、自由派团体。
- 第110届会议（2007年1月3日至2009年1月3日）。民主党在这届会议期间获得众议院和参议院的控制权。和往届会议不同，在这届会议期间，民主党内部看起来非常统一，而共和党开始分裂和分散。

虽然图8-7显示的会议都没有显示如第102届和104届那样两党派之间存在的巨大分裂，在过去6届会议中，在一个（或两个）主分块中都存在一致的分裂模式。在第111届国会的最初6个月中（在写本节时，会议还正在进行）甚至更明显地延续这种模式。如图8-8所示，第110届会议的民主党的团结使得两党分块几乎达到均匀。共和党显示了其组成以保守派为核心，外围是分散的温和派。

因此，看起来数据中是支持2009年夏的联盟故事的。实际上，至少从1991年始，参议院一直是不断变化的地方，有变化的联盟、党派甚至是决定关键决策方向的个人。

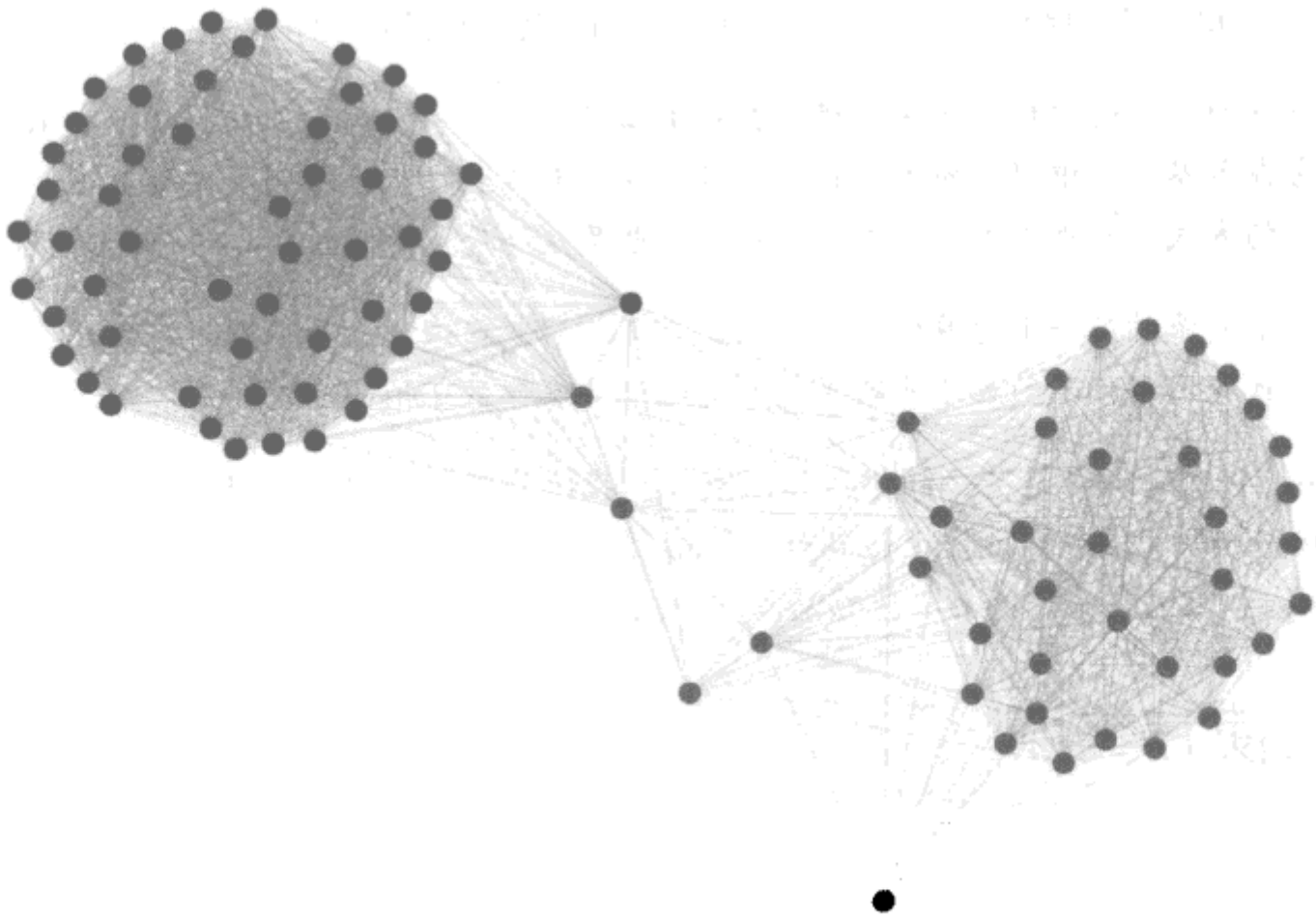


图8-8：第111届参议院会议最初6个月的结构图（2009年1月3日到2009年7月1日，见彩图66）

当然，回想起来，这几乎算不上什么新闻。这种交替联盟模式可能又回到了最初的美国成立时期，正如乔治·华盛顿在1796年的《告别演说》中给出的告诫，如图8-9所示。

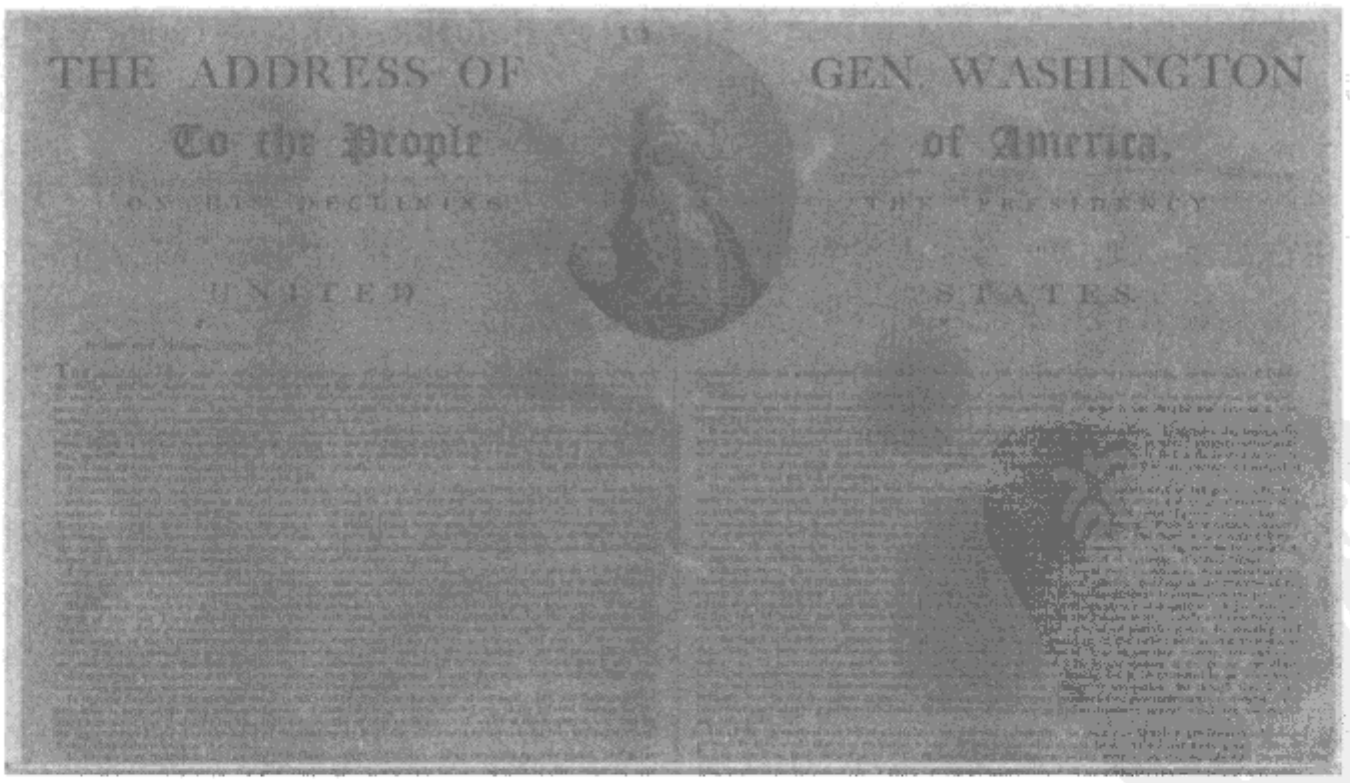


图8-9：乔治·华盛顿的1976年《告别演说》（从国会图书馆的珍藏版和特别收藏部门获得授权使用[注3]）

注3：参考*http://en.wikipedia.org/wiki/George_Washingtons_Farewell_Address*。

以下是我们的首届总统关于不同政党形成党派的趋势所说的话[注4]：

> 遗憾的是，这种精神深深地根扎在人类心灵的激情中，和我们的本性是分不开的。它在各界政府中以不同形式存在，多多少少有点被镇压、控制或压制；但是它以流行的方式，被作为第一优先级，而确实也是政府最大的敌人。
>
> 一个党派对另一个党派的交替控制，受复仇思想所激化，党派纠纷很自然，在不同年代和国家犯下的最可恶的罪行，莫过于其本身最可怕的专政。

华盛顿的告诫针对的是“不同年代和国家”，我认为该警示同样适用于今天。因此，虽然2009年的联盟故事可能还比较新鲜，其最基础的模式实际上已经久经考验了。故事中的不同人物来来去去，而故事依然是同一个故事。

什么使它美丽

当编辑请我参与本书的写作时，我的第一个想法就是“可是我做的图太丑了！”。标签随着时间变化，有时显得有些歪，并且划分党派的方式明显地存在一些不准确的地方。（我很快将会详细描述一些决策上的失误。）但是当我进一步思考这些问题时，我确定这项工作中做出的最基础的决策是正确的，因此它使得其他一切都可以挽回。

选择相关参议员之间的网络连接作为可视化框架是创建美丽的可视化的关键因素。可能查看其原因的最佳办法是把它和其他描述进行比较，它们描述的是相同的事物，只是以不同的方式展示而已。考虑图8-10，这是McCarty、Pole和Rosenthal（2008年）给出的党派指数的时间序列图。

现在，这张图表绝对没有错误，而它非常出色地说明了在20世纪70年代中期保守主义在共和党中占据重要地位。当你考虑它如何清晰地反映了尼克松总统的“南方策略”的影响，该策略利用人们对公民权利的担心，把曾经坚实的南民主党转变成共和党的堡垒，你就会发现该可视化非常有趣。然而，虽然该可视化所表达的意思非常清晰，但是它没有提供任何其他因素引发读者共鸣，因而需要做一些研究才能了解其背后的故事。

该可视化和社交图可视化不同。举个例子，知道了每个点表示一个参议员，你很自然地会好奇：“那个很不合群的人是谁？”然后欣喜地发现，他就是那个“特立独行”（mavericky）的约翰·麦凯恩。在这个可视化图中，你会发现：党派中的崛起在图上并不是简单的一条直线，而是两个对立的党派的相互竞争，由中间少部分人连接起来；两党合作在第104届国会时期的彻底破裂，双方形成了严实的自我防护；根据每个党派成员在不同时期对外部事件做出的反映，可以发现其党派分块内的内部冲突等，这些发现都很让人惊奇。

注4：　参考*http://avalon.law.yale.edu/18th_century/washing.asp*。

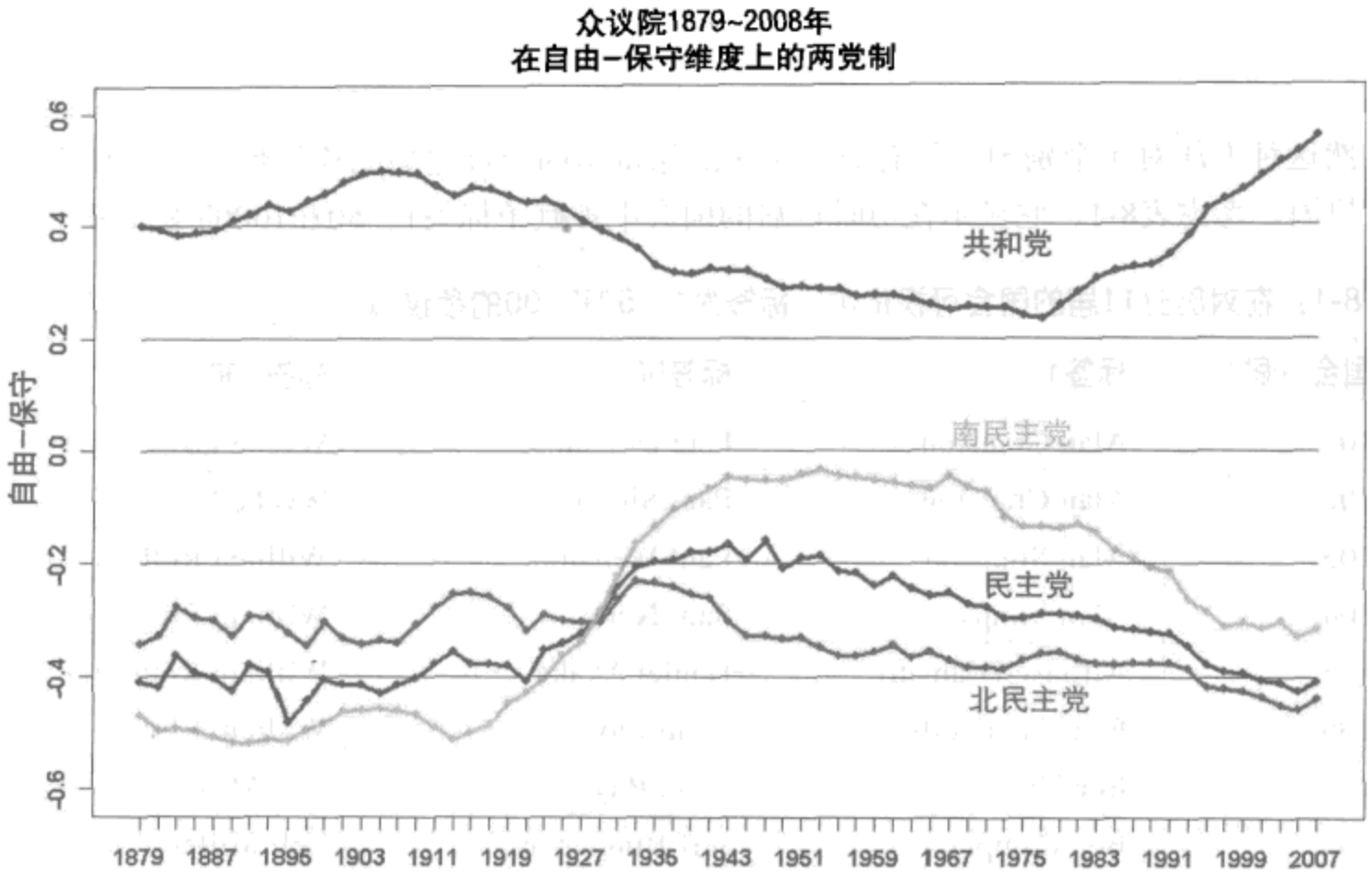


图8-10：一个很有意思但不是特别美丽的两党制可视化（见彩图67）

该可视化可能会使人们产生共鸣，这也是其美丽，而不仅仅是有趣的原因。线条图可以说明一个事实，而且可以非常清晰地达到这个目的，但是它很少可以激发你去参与探索更多的信息。就像一个好的故事，美丽的可视化应该能够吸引你，引出问题，并激励你去探索和发现。

如果能够在可视化中激起用户共鸣，用户将会忽略一些其他方面的瑕疵。而我的可视化激发了用户的一些共鸣。

什么使它丑陋

虽然我对于自己的可视化图形最终的显示效果很满意，事后反思，还是有些方面我本应该改掉。绝大多数问题源于对数据做了太多的假设，我将在下一节介绍。

标签

可视化的一个主要目标是揭示参议员之间的全局结构，而不是透露具体个人细节。虽然有时知道一个特定节点代表谁是有用的，例如，当一个节点看起来是党派之间的中心“桥梁”或连接（比如Olympia Snowe或Ben Nelson），或者偏离任一党派（如约翰·麦凯恩）。我希望能够快速识别这些“有趣”的节点，而依然保持专注于全局模式。我采

取的解决方法是按字母顺序给每个参议员赋予一个标签，然后在相应节点上使用这些标签。

虽然这种方法对于个别国会很有效，它无法保留不同会议之间的连贯性。为了查看其中的原因，考虑表8-1，它显示在历时11届的国会中被赋予标签1、50和100的参议员。

表8-1：在对历时11届的国会可视化中，标签为1、50和100的参议员

国会（届）	标签1	标签50	标签100
101	Alan Cranston	Pete Domenici	Wyche Fowler
102	Alan Cranston	Paul Simon	Wyche Fowler
103	Alan Simpson	Paul Wellstone	William Roth
104	Alan Simpson	Sam Nunn	William Roth
105	Alfonse D'amato	Daniel Akaka	William Roth
106	Ben Campbell	Evan Bayh	William Roth
107	Ben Campbell	Evan Bayh	Zell Miller
108	Ben Campbell	Jeff Bingaman	Zell Miller
109	Barack Obama	John Ensign	William Frist
110	Charles Hagel	John Thune	Wayne Allard
111	Kirsten Gillibrand	Joseph Lieberman	Tom Udall

理想情况下，每个参议员在他出现的所有图形中的标签应该是相同的。然而，快速扫描一眼以上这个表，就可以发现我给出的方法在这一点上做得多么不好。比如参议员Joseph Lieberman从1988年开始一直是康涅狄格州的参议员。按简单的字母排序，他在11届国会可视化图形中的标签分别是50、54、59、65、66、73、76和77。而其他参议员亦是如此，除奥巴马外。这些参议员绝大多数都在参议院中任职多届，但是在我的系统中，给他们赋值的标签却是非常不一致。

更好的系统应该是创建一个列表，代表在历时11届国会中的所有的参议员，然后基于该列表对每个参议员赋值一个唯一ID。当然，其中的折衷是我将需要100个以上标签，但是这一点是可以接受的，尤其是如果该列表是按每个参议员的第一个选举年而不是字母序排序。另一个解决方式是创建一个动态、交互的可视化，其中（举个例子）每个用户可以用一个节点悬浮式表示，可以弹出窗口表示额外的元数据。然而，由于我是为了打印而设计的可视化，这种方法对于我来说不可行。

旋转定向

除了给参议员打上标签，我希望可视化是有方向的，这样民主党显示在左侧，共和党显

示在右侧。按照既定习俗，其思想是一致的标签可以给各种不同图表带来一致性。然而，事实证明由于Neato布局算法的本质原因，该策略实施很困难。

前面描述的“力导向”过程是揭示隐藏在抽象图形内的复杂结构的很好的方式。然而，因为它依赖于特定的随机性，它无法每次产生相同的结果：虽然总体结构是相同的，旋转定向会有非常大的区别。举个例子，图8-11显示了对一个简单图形的3种不同、但等效的布局。

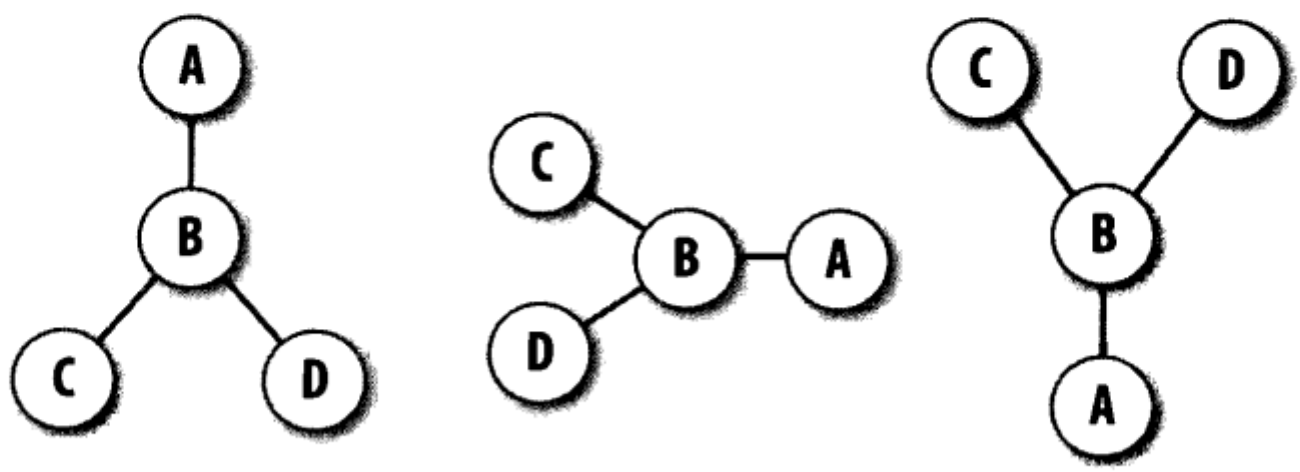
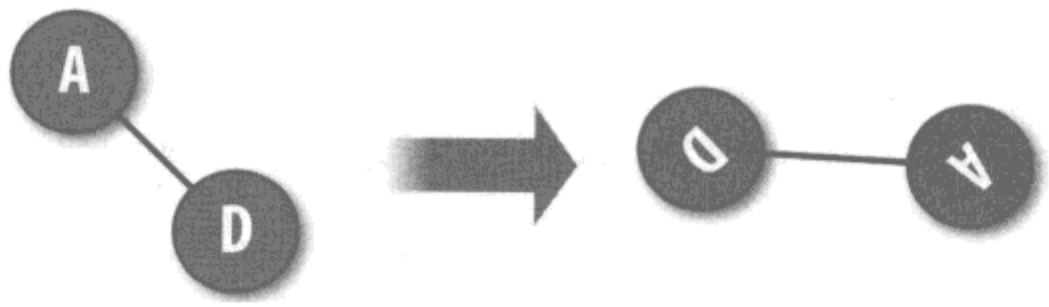


图8-11：对于相同图形的3个等价的“力导向”布局

最后，我采取的办法是打开图像文件，手工对它们进行旋转。虽然这种临时解决方式达到了期望的旋转定向，它带来的“副作用”是也旋转了标签文本，使得整体看起来有点奇怪。图8-12的原理图说明了其原因。



初始图像　　旋转后的图像

图8-12：对图形布局算法中的原始图进行旋转，使得民主党在左侧，共和党在右侧，其结果是引起标签上产生一些“副作用”（见彩图68）

回想起来，如果投入时间从编程上解决旋转定向问题将是更好的策略。举个例子，我本来可以增加一个步骤来计算两个聚类之间的质心，然后计算整个图形绕着质心的旋转角度，这样可以生成我所期望的旋转定向。这个额外的步骤在后面的运行中可以省去很多努力，但是在当时显得有点矫枉过正。

党派

最后一个主要的不足是由于一个愚蠢的假设：因为参议员很少改变党派，可以假定每个参议员最近的党派关系对于所有图形都适用。在我的可视化中，这个错误却显得非常醒目。

比如，再一次以参议员Joseph Lieberman为例，他在2006年民主党初选中失利给反对派候选人Ned Lamont后变成无党派。以下是他的个人文件信息*people.xml*条目：

```
<person id='300067' lastname='Lieberman' firstname='Joseph' ... >
   <role startdate='1989-01-01' enddate='1994-12-31' party='Democrat' .../>
           <role startdate='1995-01-01' enddate='2000-12-31' party='Democrat' .../>
   <role startdate='2001-01-01' enddate='2006-12-31' party='Democrat' .../>
   <role startdate='2007-01-01' enddate='2012-12-31' party='Independent' .../>
   ...
</person>
```

正如你所看到的，参议员Lieberman在改变他的党派之前18年一直属于民主党参议员。然而，该文件的最后一条信息表明他是无党派，因此我在自己的ETL（抽取、转换和加载）过程中认为他是属于无党派。其结果是在第102届到109届的国会可视化中，他一直被（错误地）显示成绿点，在一片“浩瀚的”蓝色显示的民主党中。

为了避免这个问题，在设计上，ETL过程本应该是基于GovTrack网站[译注2]提供的数据的<role>元素的范围来检查党派。在旋转定向问题上，这一点在当时看起来似乎是不必要的。但是事后回想，它可以作为对不熟悉的数据作出“简单假设”的前车之鉴。

结束语

我将分享几条自己通过该项目积累的一些经验来结束本章，希望能够使你在工作中有所受益：

做好准备，花费很多时间做数据整理

当我发现GovTrack网站时，我以为这个项目会变得轻而易举。毕竟，数据都在那里了，整整齐齐地以XML文件格式打包。然而，实际上把这些原始数据真正转换成该项目可用的数据格式需要很长一段时间。我估计花在该项目上有80%的时间仅仅是数据转换——抽取我想要的那部分数据，实现数据库装载程序和模式，编写脚本计算数据的亲和性矩阵，这些花费的时间都多于创建DOT模板时间。这显然是非常普遍的现象，因此如果你发现自己正争扎于处理项目中的数据问题，不要泄气，看起来这是一个必不可少的过程。

译注2：GovTrack网站记录美国国会信息，详见*http://www.govtrack.us/*。

尽可能实现自动化

当你第一次处理数据时，你很可能会匆匆做出一个快速但龌龊的解决方案。所以，你写了很多shell脚本、SQL语句，可能还需要在Excel上做一些操作来获得你期望的结果数据。如果你100%确定真的只使用一次数据集，这么做是合理的。但是实际情况很可能是，如果你的工作是成功或有趣的，你很可能想回过头来做些修改，重现它或者做些改进。而当发生这样的情况时，你会发现自己搔搔脑袋，自忖自问："我刚才是运行什么脚本来计算呢？"因此，即使你可能只是认为你做的是一次性的可快速解决的项目，也值得花些时间去开发自动化脚本，并写一些最基本的文档。将来你会因此感谢自己的。

仔细想清楚你将如何表示时间

因为人们往往是对事情在过去如何发生了变化或者它们在未来将会是什么样感兴趣，一定要想清楚你将在可视化中如何表示时间。有时时间是明确表示的，如图8-10中的时间序列；有时它们是在背景中体现出来的。比如，在该项目中，随着时间的运动效果是通过屏幕上的图像变换来表达的。在任何情况下，正如在电影中那样，给人清晰的、通过时间推进的感觉将会使你的作品更有吸引力。

决定什么时候才是"足够好"

在前期花些时间整理数据以免后期遇到一些很尴尬的问题，虽然这一点是很重要的，知道什么时候是"足够好"也是很重要的。除非你是致力于一个真正需要完全精确性的系统（比如喷气飞机的平板显示），通常"早发布、常发布"是更好的。向别人展示你的工作，得到他们的反应，看是否达到你所期望的答复，然后不断迭代。

以记者的方式处理问题

本书的很多其他章节都提出：一个伟大的可视化需要讲述一个故事。我总体上持赞成观点。然而，这种思想的本质是创建可视化的人们是故事叙述者。在我看来，那个人就像戴上了创造故事的"魔戒"，而人物和场景适应情节，完善这个故事。我认为"记者"是更贴切的比喻，而不是"故事叙述者"。记者讲述故事，但是它（理想上）是一个客观的故事——记者的目标是一点一点地揭示现实，理清混乱的复杂性，并试着把它们编织成一个完整的画面。最后，你的可视化中的故事对数据中基本事实的"忠实度"是真正决定美丽的根源。

参考文献

1. Gansner, Emden, Eleftherios Koutsofios, and Stephen North. 2009. "Drawing graphs with DOT." *http://bit.ly/4GlYAp*.

2. McCarty, Nolan, Keith T. Poole, and Howard Rosenthal. 2008. *Polarized America: The Dance of Ideology and Unequal Riches*. Cambridge, MA: MIT Press. *http://polarizedamerica.com*.

3. Wilson, Chris. 2009. "The Senate Social Network: Slate presents a Facebook-style visualization of the Senate." *http://bit.ly/FD5QY*.

第9章

鸟瞰图：搜索和发现

Todd Holloway

搜索和发现是信息检索的两种方式。搜索是一种众所周知的方式，百度和其他Web搜索引擎都是很好的例子。虽然搜索引擎也包含发现，但是还有一些更为直接的发现系统，比如Amazon的商品推荐和Netflix的电影推荐。

这两种检索系统的共性之处在于引擎背后提供支撑的系统可以非常复杂。系统提供的结果可能不仅仅依赖于查询的内容和返回的结果，而且依赖于系统用户的集体行为。举个例子，你在Netflix上对电影进行评价以及为电影提供的具体评价将会影响到系统向其他用户推荐哪些电影；而在Amazon，顾客评价、购买一本书，甚至先向购物车添加了一本书然后又删除了它，都会影响到系统给其他用户的推荐。相似地，使用百度时，当你点击了一条搜索结果，或者没有点击某条结果，这些行为都会影响到以后的搜索结果。

这种复杂性的一个后果是系统行为变得难以解释。我们主要依赖于性能指标来对检索结果的成功或失败进行量化评估，或者找出系统的哪些变化比其他的效果更好。这些指标可以使系统得到不断改进。

理解系统行为的另一种辅助方法是使用信息可视化。借助可视化，我们有时可以获取单纯根据指标所无法获取的一些认识。在本章，我将介绍一个实例，借助特定的可视化技术为系统的动态特性提供一些宏观视角。我们接下来要分析的第一个系统是一个搜索引擎，YELLOWPAGES.COM。目标是获取该网站的用户查询行为的“鸟瞰图”，这可以用于改进系统本身的设计。我们要查看的第二个系统是根据“Netflix奖”数据集构建的

电影推荐，Netflix举办的一百万美元的预测模型竞赛最近刚刚结束。该可视化可以帮助我们理解基于用户偏好的发现模型所存在的一些本质问题。

可视化技术

本章描述的技术都是关于比较相同类型的事物项——如第一个例子中的查询以及第二个例子中的电影。其前提很简单：我们将把待比较的事物项放在页面上，相似项彼此之间很紧密，而不相似的事物项距离很远。这个前提假设是基于Gestalt的相似度原则，该原则认为当两个事物项被紧紧放置在一起时，人们往往会认为它们属于同一个组。

因此，创建这些可视化的第一步是定义清楚是什么使得两个事物项之间相似和不相似。它可以是任何方面。在前文的Netflix奖的例子中，我们可以将两部电影的相似性定义为用户的评分。使用用户评分来定义相似度是很有道理的，但我们还可以选择如风格、演员这样的电影属性来定义相似度。

一旦定义了相似度，需要对它们进行坐标化，把这些相似度值转换成二维或者三维坐标。有两种方式可以实现坐标化。第一种方式是使用一个公式，把高维空间映射到二维或者三维空间。另一种方式是把各个事物项看成图表的节点，相似的节点通过边进行连接。因而，坐标化就是试着把连接着的节点放置在相邻位置，而把不连接的节点放置在不相邻的位置。在本章中，我们将使用后一种基于图形的方法，并探讨所需要的特定工具和算法。

完成坐标化以后——也就是说，在给事物项赋予特定的坐标值之后——这些事物项的表示（在后面两个例子中，采用的是简单的圆圈表示）会被放置到坐标系的相应坐标中。创建可视化的最后一个步骤包含标签放置（这一点相当有挑战）以及做出各种各样的其他分析。

YELLOWPAGES.COM

直到最近，使用打印版的电话簿找人和查询服务仍然司空见惯。其中的服务部分被称为“黄页”（Yellow Pages）。在这些黄页上，企业按类别进行分组并按字母序进行排列。一切都很简单。

YELLOWPAGES.COM（见图9-1），是我所在的公司AT&T的一个Web站点，是一个现代化企业搜索引擎，其最基本的目标和打印版一致。很明显，它虽然是在线版本的，但并不是局限于只能通过和打印版一样的方式分类和字母序来组织数百万的企业。

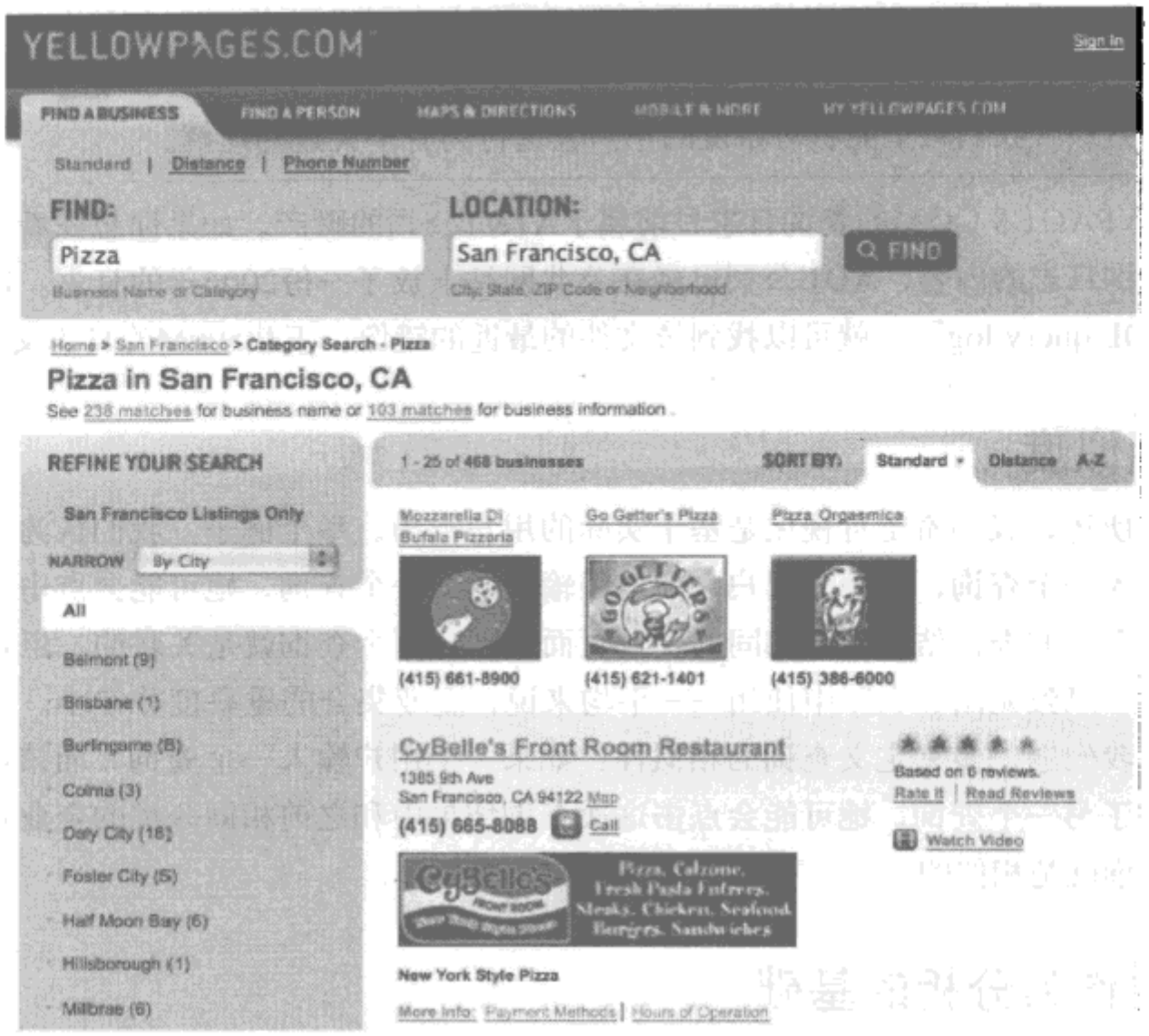


图9-1：YELLOWPAGES.COM：一个本地企业搜索引擎（见彩图69）

事实上，设计或改进这种搜索引擎的部分工作涉及理解应该如何为一个给定的查询组织企业列表，以及在该组织中应该包含哪些企业特征。为了达到这个目标，查看用户的行为是有帮助的，因为该行为可以对我们的直觉做出验证或否定。

查询日志

YELLOWPAGES.COM保留了在网站上执行的每个查询日志，因此它可以使用这些数据来改进服务。以下是2008年12月的查询日志中词频最高的5个查询：

1. Restaurants（餐馆）
2. Movie theaters（电影院）
3. Pizza（比萨）
4. Walmart [sic]（沃尔玛）
5. Animal shelters（动物收容所）

前5项综合了“浏览式”查询和“搜索式”查询，前者是人们在分类范围内进行浏览

（如餐馆），后者是人们搜索特定企业（如沃尔玛）。我们将使用日志中的查询作为可视化的“事项”，将基于用户执行这些查询的行为的相似度来对这些事项进行坐标化。通过这种方式，我们希望能够对系统的用户查询行为有个大致的理解。

YELLOWPAGES.COM的查询日志目前属于AT&T公司的财产。如果你想查看主流搜索引擎的查询日志的内容，AOL公司已经在公共网站上放了一份2006年的日志。通过百度搜索“AOL query log”，就可以找到该文件的最近的镜像，下载500M的日志文件。

分类相似度

正如之前所述，我们希望可视化是基于实际的用户行为。举个例子，我们认为，如果一个用户输入一个查询，如果该用户之前已经输入了另一个查询，她可能会点击这次查询结果中和之前的查询结果中的相同的企业页面，则这两个查询就是关联的。但是，数据太稀疏了，以致无法在实际中使用——平均来说，企业集合的重叠度非常小。为了解决稀疏性，我们退一步来定义查询的相似性：如果一个用户输入一个查询，而该用户之前已经输入了另一个查询，她可能会点击这次查询结果中和之前相同分类的企业页面，则这两个查询就是相似的。

可视化作为分析的基础

在AT&T应用研究所，我们构建了很多工具来分析查询。其中一个工具是预测模型，它试图确定一个查询是否是为了参考一个特定企业的信息（如Walgreens）或者浏览一组企业信息（如药店）。我们可以在可视化基础上应用这些预测来获得“搜索式”和“浏览式”查询的分布的总体概览。可以使用很多可视化编码来显示一个查询属于哪一种。最明显的一种，即我们所采取的方法是对节点进行着色：在我们的可视化中，绿色节点表示预测认为是对特定业务搜索的查询，而其他查询则是用黑色节点表示。可能会存在一些不正确的节点着色，它们显示的是该特定预测模型中存在的误差。

图9-2用绿色节点显示 “Goodwill”和“Salvation Army” 查询，其含义是预测上认为（而且是正确的）这些节点属于对特定企业的查询。

图9-2：在我们的可视化中，“搜索式”查询用绿色显示（见彩图70）

可视化

图9-3显示了最终的可视化结果。它显示了从2008年12月开始查询频度最高的4600条查询。当查看这种类型的可视化时，应该记住的是它没有坐标轴。所有位置都是相对

的——相似的查询毗邻，而不相似的查询距离较远。每个圆圈表示一条查询。一些圆圈的上面标注了查询项。圆圈大小和标签大小都是基于查询在日志中出现的次数。采用这种方式，可以使得频繁的查询在该可视化中可以“一目了然”。

图9-3：在YELLOWPAGES.COM网站上，查询频度最高的4600条查询（见彩图71）

查看图9-3，很容易识别出该系统最经常被使用的领域。“Restaurants”（餐馆）这一条查询“脱颖而出”，而零售商如“Walmart”（沃尔玛）和“Best Buy”（百思买）的查询也很频繁。对餐馆和零售商的查询很频繁可能不足为奇，因为YELLOWPAGES.COM是一个企业搜索引擎。可能相对难以预测的是底层在大区域范围内包含社区相关的查询，包括搜索“公立学校”、“教堂”和“公寓”。

这种类型的可视化很大。无法把它打印在一页纸上；显示它的最佳方式或者是把它作为大海报进行打印，或者作为在计算机屏幕上可缩放的版本显示。为了使可视化可缩放，可以把它加载到如下应用中，如百度地图、Gigapan或微软的Seadragon。

因为这个可视化将要被收录到一本书中，我们将通过对一些特定方面进行放大和讨论的方式来进一步审视它，并获取一些新的理解。

图9-4所示的是对看起来与社区较为相关的企业的查询集合进行放大显示。看到像该可视化这样对真正的用户行为的描述可能会给搜索引擎工程师留下深刻印象，可能验证了他对系统使用情况所持有的观点，或者给他带来惊喜，甚至是启发他做出设计上的一些变动。

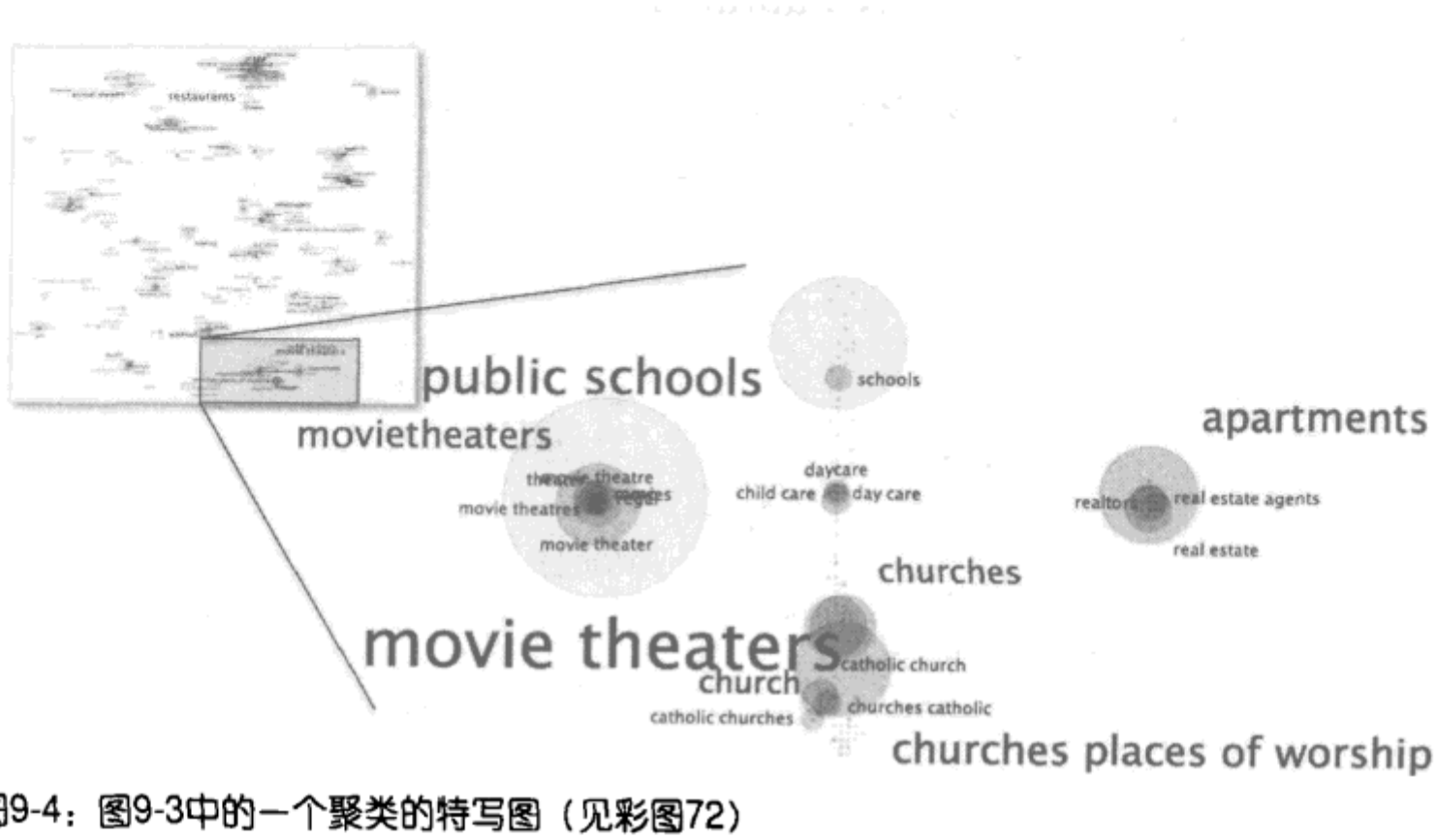


图9-4：图9-3中的一个聚类的特写图（见彩图72）

看起来对图9-5所示的聚类进行特征化相当简单明了，但是还是存在很多方面值得指出。注意对GameStop的常见但不相同的拼写；人们可能期望不论是哪一种拼写，用户都可以对搜索结果做出相同的行为，因而应该也期望这些查询能够在可视化中毗邻显示。可能可视化中最有趣的是典当行相关的查询与书店和游戏商店相关的查询的相似度。用户做出什么样的查询和点击行为可能会生成这种模式？

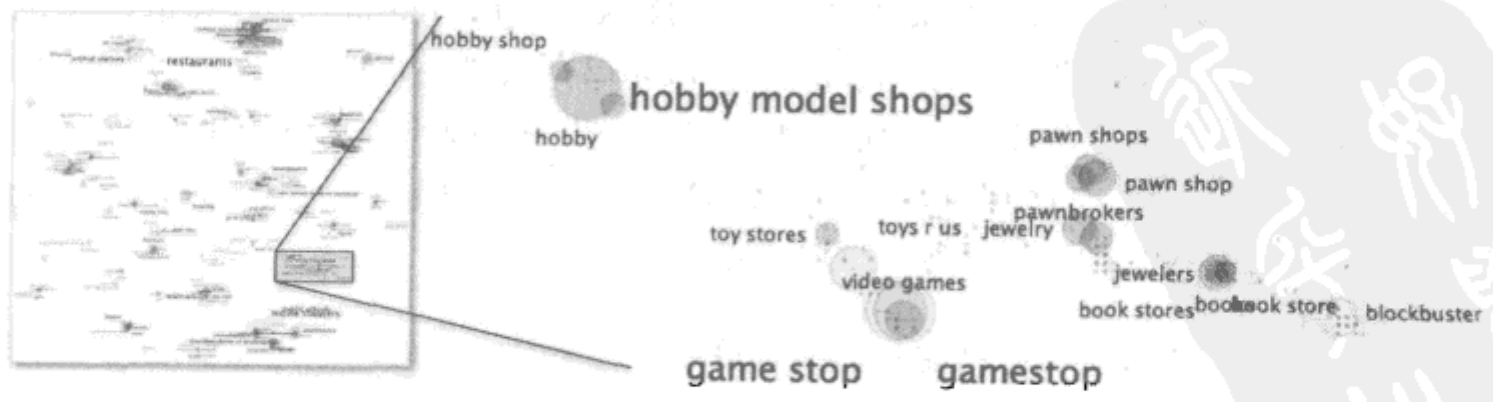


图9-5：大规模的兴趣相关的商业聚类（见彩图73）

这种可视化技术是强大的，因为不仅仅是单个聚类内的相似度可以帮助我们提高认识，

而且多个聚类之间的相似度也同样可以加深理解。图9-6中存在两个聚类，一个是关于零售药店，另一个是关于酒店，它们在可视化上毗邻。这意味着用户无论是搜索药店还是酒店，往往会点击相似的企业。但是在打印版的电话簿中，这两类企业分别只存在于两种不同的分类内部，而搜索引擎却可以考虑这些行为的关联，生成搜索结果。

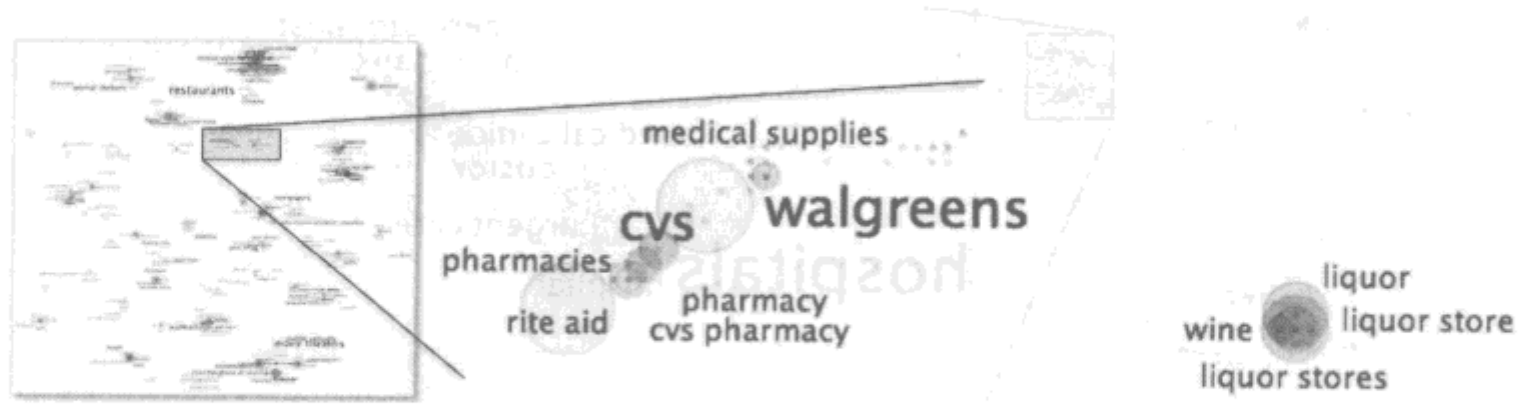


图9-6：两个毗邻的集群：药店和酒店（见彩图74）

这种可视化技术的优缺点

纵观了其中一种可视化，值得探讨的是这种可视化技术的优缺点。其最大的优点在于可扩展，而且是完全基于算法的。图9-3的可视化显示了4600个事物项，但是该算法可以扩展到处理几百万个事物项。（显然，为了有效地查看几百万个事物项，需要有一个可以平移和缩放的界面。）

该可视化技术的另一个优点在于它作为稳定、全局的基础平台，可以显示其他分析，而且工作良好。举个例子，我们使用绿色和黑色来区分“搜索式”和“浏览式”这两种不同的查询。我们可以很容易在该平台上应用任意数量的其他分析。可能显示提交特定查询的用户的平均年龄会很有意思，假设我们有这样的数据，或者有关于用户在输入某个查询之后还会使用该系统的预测。应用这样的预测可以帮助我们对系统的总体运行情况有个较全面的理解。

这种可视化技术的最大缺点（和对它的批评）是不支持精确比较。在这种可视化中，难以量化和解释特定事物项之间的关系；其他的可视化技术对于这种狭义的分析则是更有效的。这种可视化技术更偏向于技术，启发人们对数据集提出新的问题，或者提示人们某些问题的可能答案，而不是作为具体的问题答案来源。

另一个明显的缺点是当前社会尚未教育人们应该如何解释这些可视化。散点图、柱状图、饼图——人们当然了解这些图形，但是不了解大规模的图形可视化。

图9-7所示的一些有趣的聚类说明的一个技术问题是，难以对这么多的事物项添加标签。本章给出的可视化都是使用自动化标签算法，它对标签的位置放置进行优化来减少标签

之间的重叠。同样地存在一些重叠还是难以避免。也许随着技术的不断发展，创造性的新的解决方案将会解决这个问题。

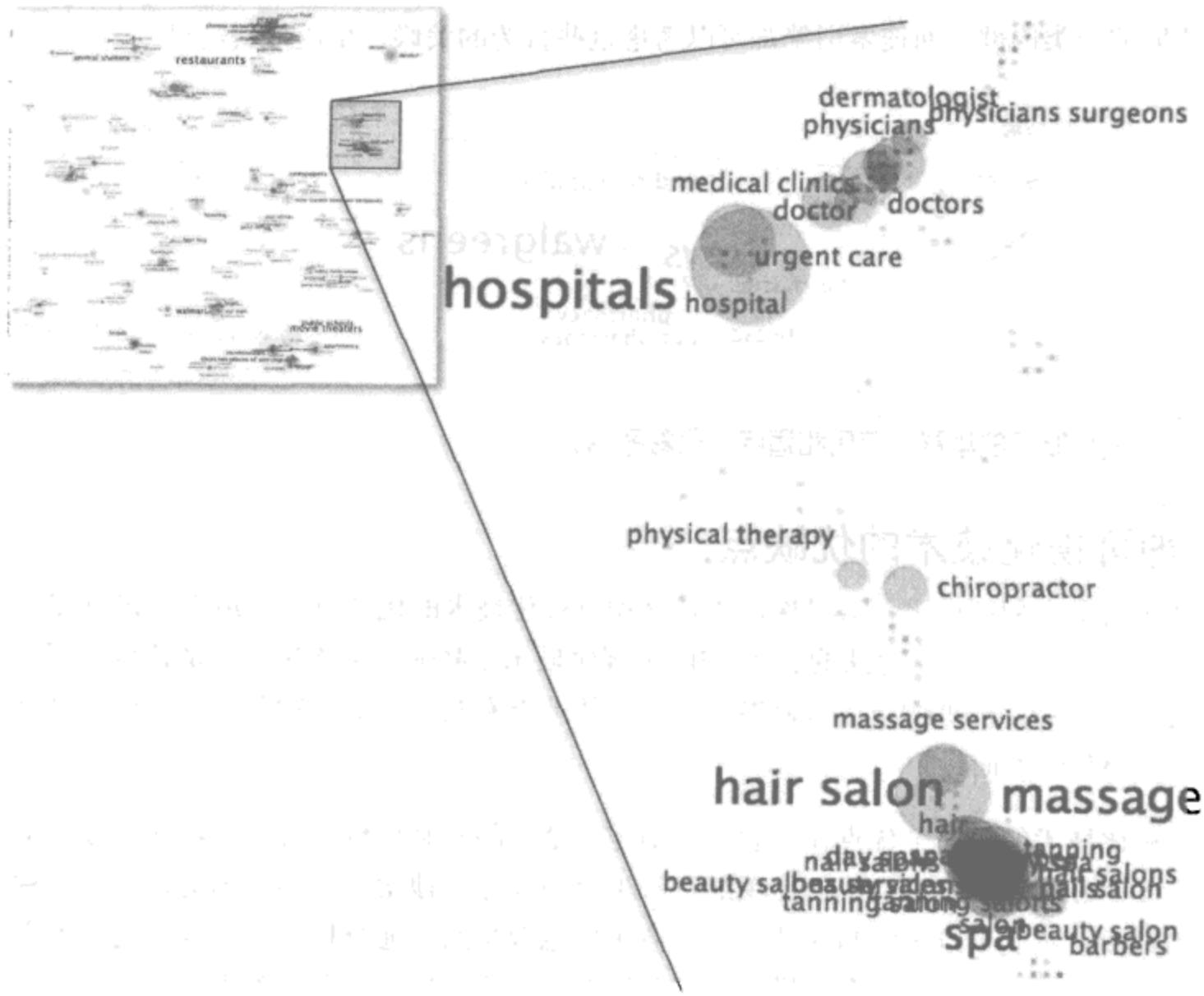


图9-7：一个标签难以阅读的聚类（见彩图75）

正如之前所述，这些可视化的最后一个问题是它们通常涉及从高维数据集降维成二维或三维数据集。在降维过程中可能会丢失信息，因此难以确定看起来很有意思的分组方式是真实地反映了关于数据集的一些有趣的方面，或者仅仅是该过程中产生的一个假象。

Netflix奖项

早就存在促使用户个性化定制自己的Web体验的愿景，而且人们为了实现这个目标也做出了许多努力。理想情况下，这种个性化定制可以促使Web服务足够了解你的偏好，从而帮助你查找餐厅、书籍、音乐、电影和其他你感兴趣的事情。

Netflix公司是通过邮件和在线出租电影的公司，它有一套推荐系统，尝试着给顾客提供合适的推荐。这些推荐是基于顾客评价高的电影以及具有相似偏好的顾客的评价高的电

影。在2006年夏，公司举办一场竞赛，给任何可以提高其推荐算法10个百分点的参赛者提供100万美元的奖金。作为本次比赛的一部分，Netflix发布了一个包含1亿个用户、对17 700部电影进行评价的数据集。该数据集可以通过UCI的机器学习库在线获取（*http://archive.ics.uci.edu/ml/*）。

从该数据集中构建一个发现系统的挑战在于一方面数据量太多，而另一方数据量又太少。使用简单的技术来解释这些数据甚至浏览它，则问题是数据量太多。然而，从做出准确的推荐的角度上看，其包含的数据比我们期望的要少。用户对电影的评价的分布远远没有达到均匀分布，即很多用户只对很少的电影做出评价，很多电影只有很少的评价。对于这些用户和电影，很难做出精确的预测。

偏好相似性

在很多推荐系统中，众所周知的相似性计算方式是计算余弦相似性。Linden、Smith和York（2003年）的文章中对该技术做了实用的介绍。

对于电影，从直观上考虑，这种计算方式说明了如果用户对一部电影评价很高、对另一部电影评价也很高，则这两部电影就是相似的；或者反之，如果用户对一部电影评价很低、对另一部电影评价也很低，则这两部电影也是相似的。

我们将使用这种相似度衡量方式来对Netflix奖数据集的所有17 700部电影生成相似性信息，然后基于该数据生成坐标转换。如果我们对于构建真正的电影推荐系统感兴趣，我们可以简单地推荐和用户评价很高的电影相似的电影。然而，这里的目标只是对这种推荐系统的动态性有更深的理解。

标签化

YELLOWPAGES.COM可视化比Netflix奖可视化更易于添加标签，其原因有很多，包括其节点更少，标签更短，但是最主要的原因是其节点是均匀分布的。虽然Netflix奖可视化中存在很多聚类，绝大多数电影只存在于其中很少量的聚类中。当我们只查看那些评价最多的电影，这种差异看起来则更加明显。

考虑两种不同的添加标签方法：

- 对最受欢迎的电影添加标签，随机对其他电影进行抽样。这种方法将得到包含最受欢迎的电影的聚类，但是由于这些聚类的密度很高，可能难以阅读这些标签。
- 把页面划分成网格，在每个网格节点位置对小样本的节点添加标签。这种方式可以确保所有聚类都包含一些标签。

对于图9-8中的可视化，使用第一种策略，因为它说明了电影总体和那些有很多评价的电影（用更大的圆圈表示）的分布都是非常不均匀的。然而，对于后续图形中对该可视化的放大展示，可使用第二种策略来增强可读性。

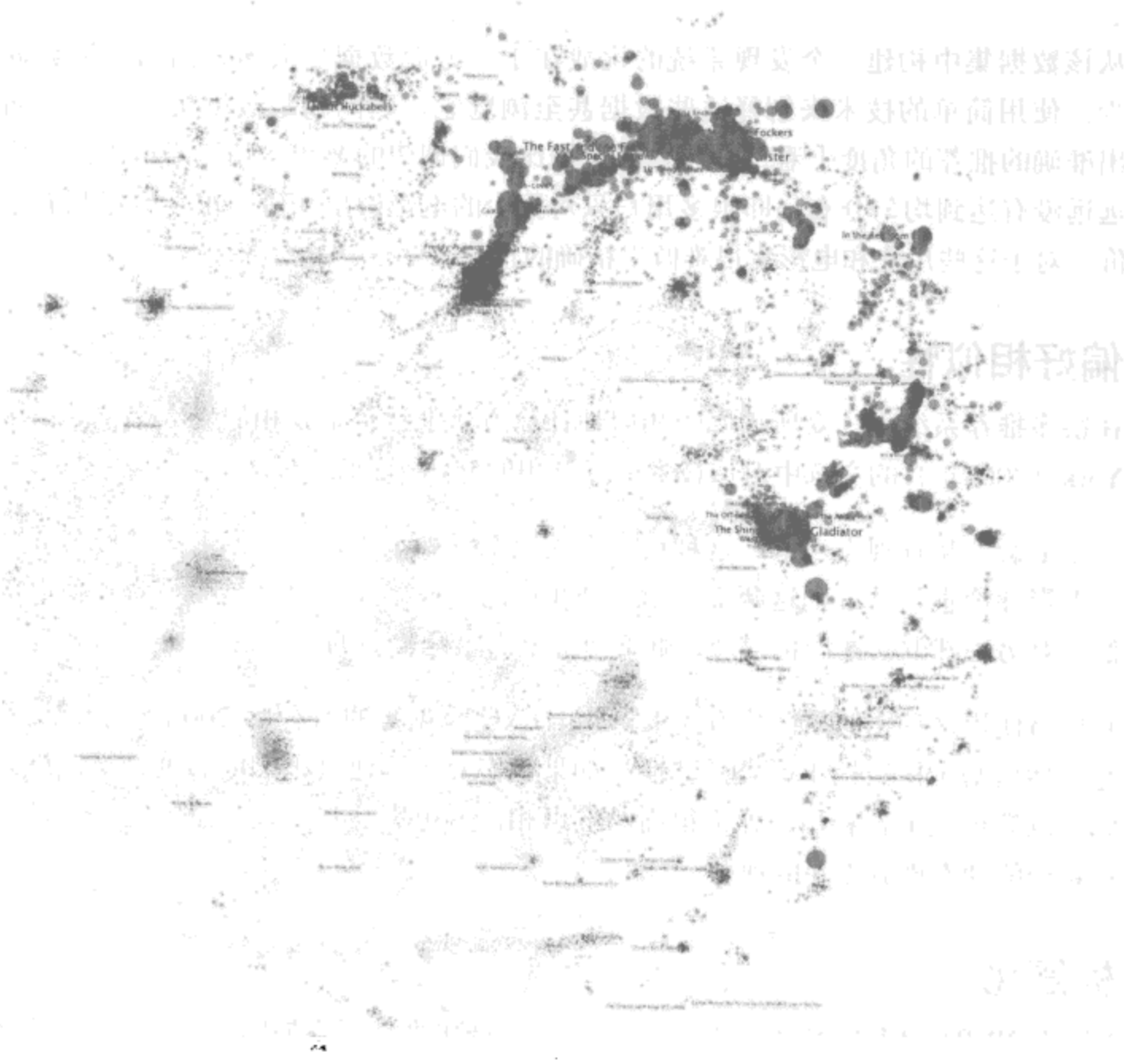


图9-8：Netflix奖数据集中对17 700部电影的可视化（见彩图76）

进一步研究

除了用户评价外，Netflix奖数据集中其他仅有的数据就是电影的标题和发布日期。然而，Netflix奖参赛者发现了潜在的属性，比如电影中的暴力程度或者用户的性别，都是预测偏好的重要因素。不足为奇的是，有些聚类似乎可以通过这些属性解释。然而，为什么其他从用户偏好生成的聚类却更难以解释？

我们将要查看的第一个电影聚类（见图9-9），包含如《星际迷航》（Star Trek）、《X档案》（X-Files）和《沙丘魔堡》（Dune）这样的标题，看起来归属于一个流派特征：

科幻小说。《银河追缉令》（Galaxy Quest）也是科幻小说，但它是讽刺科幻小说。侦探喜剧《神探阿蒙》（Monk）也属于这个集合，会显得非常怪异。然而，这是一个偏好聚类，而偏好绝不可能只通过流派这个因素来定义。这种不正常现象的其他可能的解释是给《神探阿蒙》打分的用户非常少（注意该聚类内其所表示的节点大小很小），因此把《神探阿蒙》归属于这个聚类可能是个错误；也就是说，它可能并不能反映Netflix用户的真正偏好。这一点不仅仅是创建该可视化的一个主要难题，也是Netflix奖竞赛的难题；根据很少量的已有用户评分来预测用户的偏好是非常困难的。

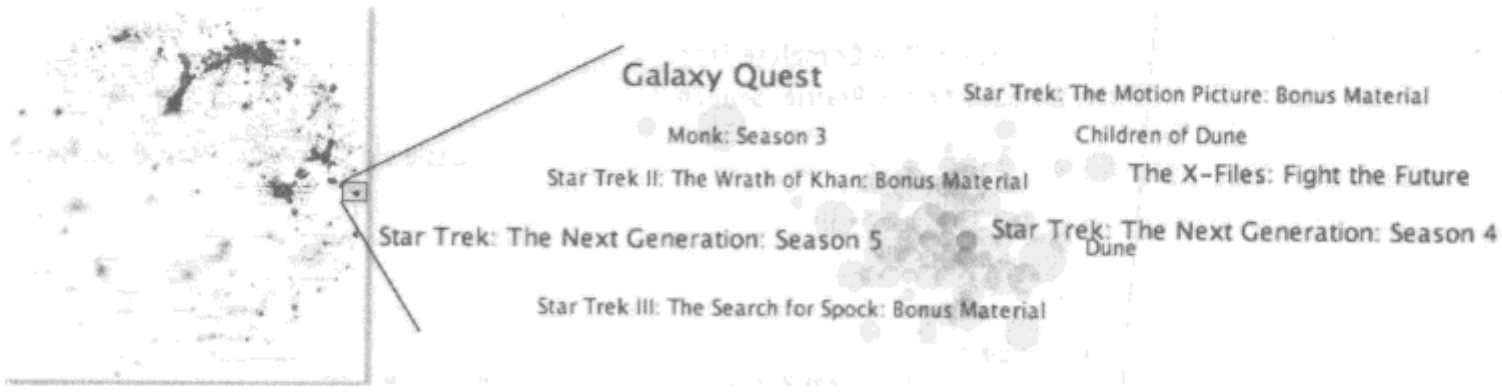


图9-9：科幻小说电影聚类

对其他聚类的解释则更有挑战性。考虑图9-10中的例子。可能给人的直观感觉是如《欲望都市》（Margaret Cho）、《双人秀》（The Man Show）、《洛基恐怖秀》（The Rocky Horror Picture Show）（三部都是很受争议的喜剧）可能会受到一群人的褒扬，却受到另一群人的唾骂，因此会看起来很混乱。但是如果是由于这个因素，为什么其他类似的幽默类型的电影没有包含在这个聚类中？为什么这几部电影之间的关系这么强，使得它们能够形成一个聚类而不是分布在其他聚类中？

图9-10：具有相似幽默风格的电影聚类

图9-11提供了聚类的另一个例子，直观上看，该聚类反映用户偏好是有意义的。如果我们能够获取到这些电影的其他属性，或者获取到对这些电影评价很高的用户信息，哪一种信息可能会帮助我们解释在这个聚类中显示的用户偏好？

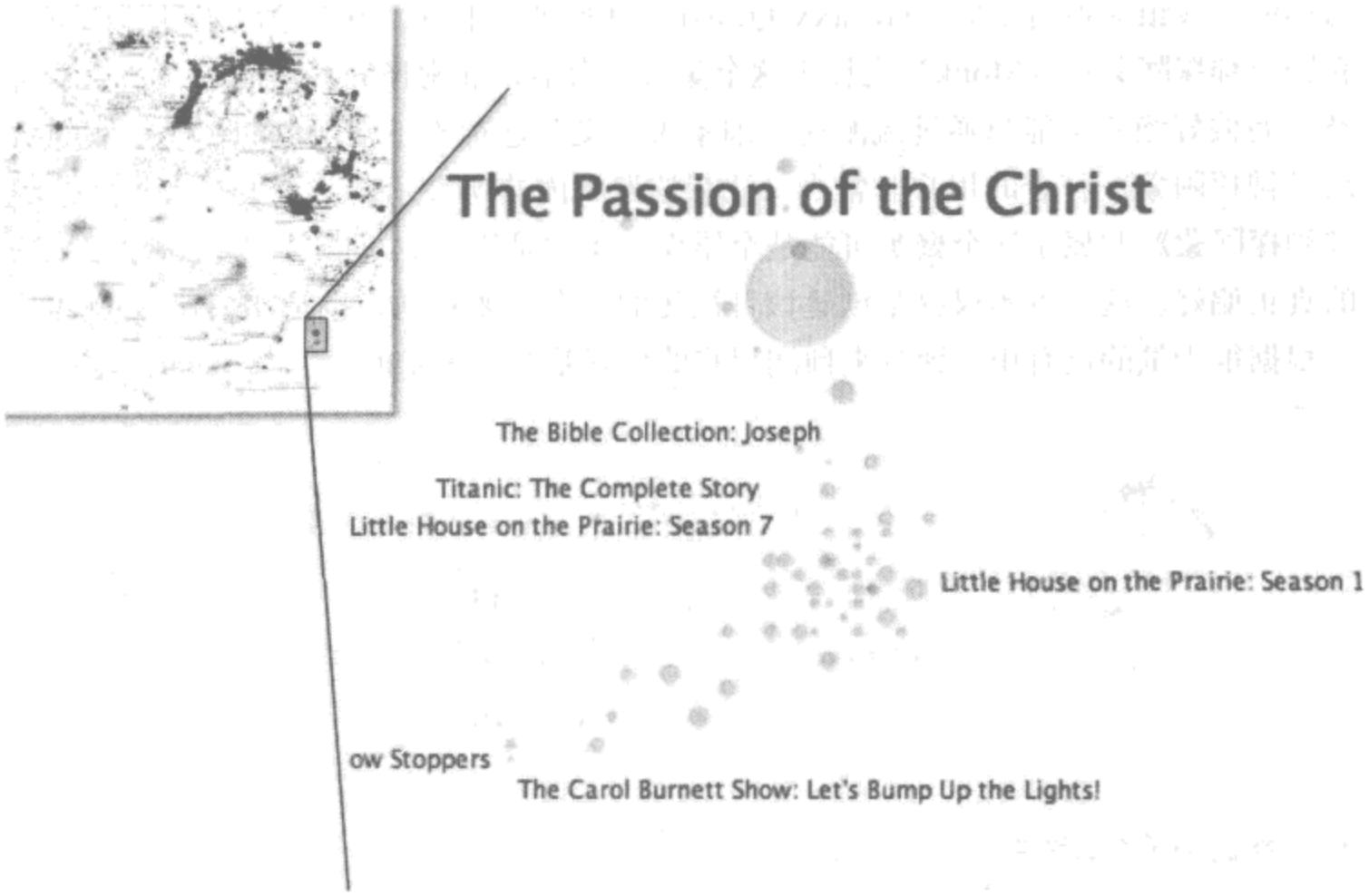


图9-11：“适合家庭的”电影聚类

解释图9-12中的聚类的其中一种方法可能是专注于一个事实，即这个聚类中的所有电影都是票房很高的动作片。即使有人认为《魔鬼代言人》（The Devil's Advocate）并不是一部动作片，其主演奇洛·里维斯（Keanu Reeves）出现在很多这样的票房很高的动作片中，因此预期他所主演的其他电影可能也会吸引观众。

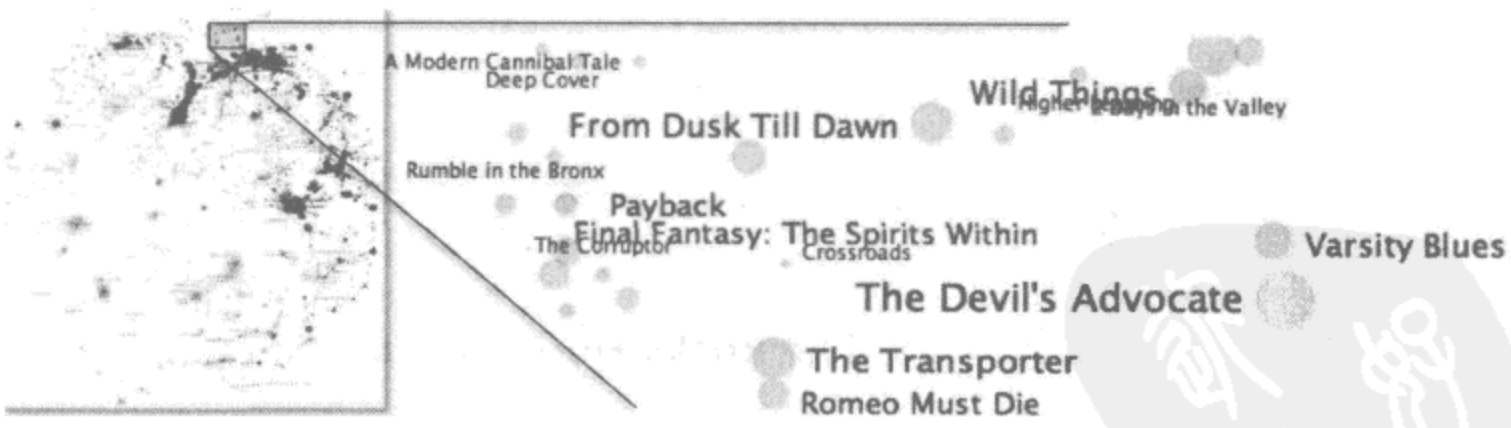


图9-12：动作片电影聚类

图9-13中显示的聚类更大，而且较难特征化，但是它还是很好地反映了用户偏好。绝大多数这类电影“让人感觉良好”；绝大多数是爱情故事。

图9-13：“让人感觉良好”的电影聚类

之前提到的一个问题是，系统提供的电影推荐可能对于那些尚未对很多电影做出评分的用户还不够好，因为系统还不知道这些用户的喜好。我们称之为冷启动（cold start）问题。实际上，对于那些对很多电影做出评价的用户，如果这些评价是分布在很多不同的场景中，那也会存在这个问题。举个例子，假设有个用户不是真正喜欢“让人感觉良好”聚类中的电影，但是为了和女朋友约会，开始租这些电影，然后基于每次约会的进展来对电影进行评价。如果他开始租影片自己看，为了发现他真正喜欢的电影，他可能没有对足够多的电影做出评价来反映其自己的个人偏好。更广泛地说，我们可以把这个问题看成是数据稀疏问题的扩展。

创建自己的可视化

你可能有兴趣以自己最喜欢的数据集来创建和本章给出的类似的可视化。存在很多工具可以用于达到这个目的。我们首先使用Perl来解析数据，计算相似性（当然可以使用其他语言来替代Perl）；然后把这些相似性结果传给Shawn Martin提供的免费软件“DrL软件”（*http://www.cs.sandia.gov/~smartin/software.html*）。DrL使用之前提到的图形方法，把这些相似性转换成每个节点坐标。DrL的优势在于它可以递归执行，因此坐标可以反映更高层次的组织。另一个可以取代DrL的很好的软件是GraphViz（*http://www.graphviz.org*）。

完成以上处理后，我们继续使用Perl把坐标和其他额外信息进行归并，如节点的尺寸、颜色和标签。最后，把处理完成的数据集传递给商业绘图库yFiles（*http://www.yworks.com/en/index.html*），yFiles对标签进行布局，把整个可视化渲染成一个png文件。yFiles

是一个非常有用的包，但是你可以略过这一步，比如直接使用Perl创建EPS文件，其代价是没有对标签进行布局。

结束语

本章给出的两个例子是可视化技术的两个非常简单的应用。如果你对查看这种类型的可视化感兴趣，在线的“地点&空间”（Places & Spaces）展览网站上包含很多（*http://www.scimaps.org/maps/browse/*），它是印第安纳大学的Katy Borner教授组织的大规模的可视化集合。

值得一提的是，这种可视化类型目前仍然是一个很活跃的研究领域。最近的前言发展专注于对该技术进行扩展，纳入一些约束条件。其中一个可以从增加约束条件中受益的领域是系统生物学，人们可能想要显示蛋白质之间的相互作用。其相似性计算可能是基于相互作用的蛋白质的数量。需要的约束条件可能是在一个细胞核内的某些蛋白质，对其以某个特定圆形区域的坐标显示；而对于细胞质内的蛋白质，则以更大范围的圆形区域坐标显示，并且不会和细胞核内的蛋白质重叠。同样，可以限制膜蛋白在一个圆圈上显示，而同时还是按相似度分组。像本章讨论的搜索和发现系统的可视化，这种可视化可以提供全局画面，有助于启发思考或者验证人们当前的直觉。这种可视化技术在其他领域的可能应用方式，作为练习留给读者思考。

参考文献

Linden, Greg, Brent Smith, and Jeremy York. 2003. “Amazon.com recommendations: Item-to-item collaborative filtering.” *IEEE Internet Computing* 7, vol. 1: 76–80.

第10章

从社交网络可视化的混杂之中寻找美丽的感悟

Adam Perer

我的目标始终是把符号并列、组合成为统一、
一致的整体来解释物质。
——Mark Lombardi，2000年

Mark Lombardi[译注1]可能堪称完美的网络布局算法。作为一位致力于揭露经济和政治丑闻的错综复杂的网络信息的艺术家，他努力绘制节点没有重叠、边很少交叉，而且连接平滑且弯曲的网络（见图10-1）。以计算方式创建的社交网络的可视化很少能够达到这种程度的优雅和感性。虽然高级的计算布局算法可能是以弹力和推动力的物理模型为基础，但是它们很少能够像Lombardi的绘图那样突出模式和趋势。本章详细描述我为了使用户能够使用可视化和统计的集成交互技术来深入研究混杂的社交网络所做的一些探索。

社交网络可视化

现代社会数字信息的增长开辟了数据分析的黄金时代。丰富的数据促使人们为了解释科学、社会、文化和经济现象，做出了更频繁的数据分析探索。虽然能够使用数据很重要，但仅仅做到这一点还是不够的，我们还需要能够理解模式、识别游离点和发现差异。现代的数据库太大了，人们如果没有计算工具的帮助将无法处理和使用数据。

译注1：Mark Lombardi是美国的概念派艺术家，其错综复杂的艺术作品主要展现在《Mark Lombardi: Global Networks》中。

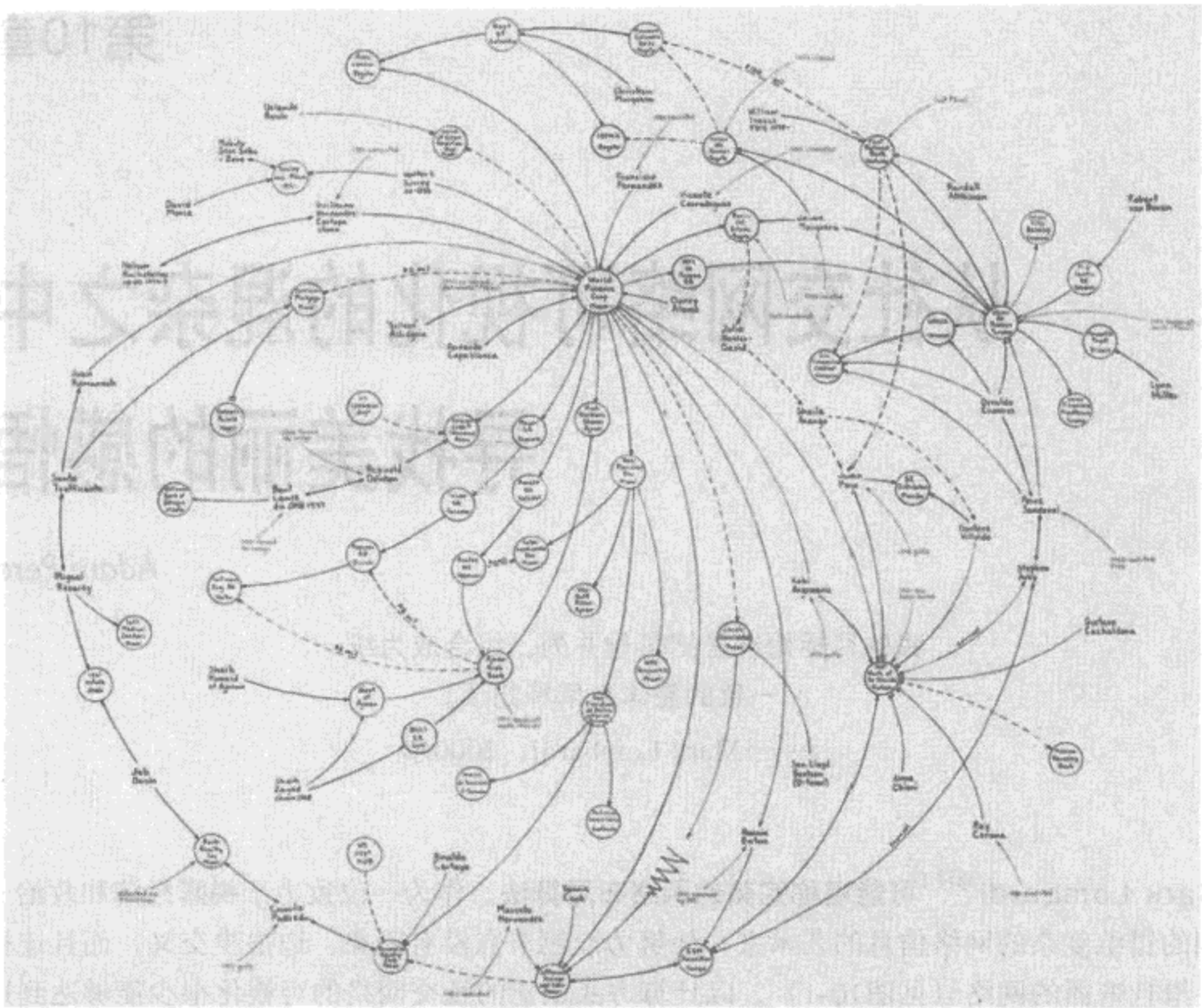


图10-1：艺术家Mark Lombardi手绘的一个社交网络的例子（“世界金融公司，迈阿密，佛罗里达州，1970年~1979年（第6版）”（1999）；纽约布鲁克林PIEROGI艺术馆授权使用，见彩图77）

最强大的感官接收器——眼睛，其“带宽”和处理能力远远高于嗅觉、听觉、味觉和触觉接收器。因此，信息可视化是充分利用人类最强大的感知系统的强大能力的有效方式。然而，选择有效的展现方式具有很大的挑战性，因而不是所有的信息可视化都可以达到相同的效果。不是所有的信息可视化都是为了突出对于分析师的任务而言重要的模式、差异和游离点，更进一步说，也不是所有的信息可视化都是为了“迫使我们去注意自己从未期望看到的事物”（Tukey 1977）。

数据分析中一个渐趋普遍的趋势是将相互关联的数据作为网络进行分析。网络分析不仅仅是查看数据的属性，还会关注数据和最终产出之间的结构关联。我的研究重点正是理解这些网络，因为在分析师看来，网络是热门的、新兴的且本质上具有挑战性的。网络总是难以进行可视化和导航，而且最大的问题是很难找到与任务相关的模式。尽管有这些挑战，网络分析依然深受社会学家、情报分析师、生物学家、通信理论家、

文献研究员、食物网生态学家以及很多其他专业人士的青睐。从最畅销的书籍，如Malcolm Gladwell的《The Tipping Point》（Back Bay丛书），Albert-László Barabási的《Linked》(Plume出版社)和Duncan Watts的《Six Degrees》(Norton出版社)，可以看出社会网络分析（SNA）的流行度日趋增长，同时也因为这些书籍的畅销而进一步促进了它的流行。无数的分析师希望能够分析他们的网络数据，但是只有很少成熟且广泛应用的工具和技术能够达到这一目的。

网络分析师注重研究不同因素之间的关系而非具体因素；这些因素可以解释社会、文化和经济现象，但它们之间如何联系和它们本身一样重要。在出现社交网络分析观点之前，很多分析师主要注重于内在的个别属性和被忽略的社会行为，也就是说，注重于个别因素如何交互以及它们之间的影响（Freeman 2004）。借助来自社交网络社区的更为新型的技术，分析师可以发现结构中的模式，见证资源或消息流在网络中的传输，理解个别因素如何受到周围环境的影响。

在实践中，社交网络可视化是混杂的，尤其当网络规模很大时。可视化在充分利用人类强大的感知能力上很有用，但是混乱的展现方式、边重叠以及不合法的节点标签通常会削弱可视化探索的价值。在这些情况下，交互技术对于理解这些复杂的静态可视化是有用的。内在属性是存在于数据集中的属性，比如性别、种族、薪水或受教育程度。通过节点和边的内在属性进行缩放、平移或过滤等交互可以简化复杂的可视化。不幸的是，对于复杂的网络，这些技术所能达到的可能也就仅此而已，无法挖掘出整个故事，尤其是在小世界网络中密度高的连接很少会没有交叉（van Ham 2004）。内在属性缺乏对社交网络分析师而言非常重要的结构化拓扑信息。我们的主要贡献是通过反映用户任务的计算属性来增强信息可视化。计算属性可以通过以下几种策略来计算：相关的重要性统计指标（如度或距离中心的程度），聚类算法或者数据挖掘。

充分利用计算属性的处理方式对于社交网络分析师尤其有价值，因为他们也开始意识到内在属性并不能揭示整个故事。实际上，社交网络分析师采取的方法是在探索时忽略内在属性，避免个人偏好，而只注重数据的结构化属性。对于社交网络分析师，计算属性可以通过一组丰富的统计方法来计算（从社会学到图形理论），因而可以使分析师大量挖掘他们所在网络的有趣的特征。分析师可能会寻找紧密结合的个人社区群体，或者是他们当中的信息传递员，或者是处于中心地位的强大个体；存在很多找到这些特征的复杂的算法。

大部分可视化工具的目的是把复杂的数据映射到易于理解的视图中。然而，很少有工具可以通过突出代表数据重要特征的计算属性来帮助用户进行可视化。用户可以在统计和可视化软件包中来回切换使用，但是这种做法可能会导致分析过程中数据流很低效，从而阻碍人们新的发现。

SocialAction是Ben Shneiderman和我一起创建的用于探索这些问题的软件工具（*http://www.cs.umd.edu/hcil/socialaction*）。通过集成统计和可视化技术，该工具可以即时提供有意义的计算属性，帮助用户快速利用二者的优点。SocialAction嵌入了统计算法来探测重要的个体、关系和聚类。该工具不是以经典的表格方式来表示统计结果，而是集成在网络可视化中，该可视化能够提供有意义的节点和边的计算属性。通过计算属性，用户可以很轻松地动态过滤节点和边并找到有趣的数据点。这些可视化简化了统计结果，有助于增进理解和发现如分布、模式、趋势、差异和游离点的特征。这些统计简化了对有时混杂的可视化的理解，允许用户关注统计上有意义的节点和边。在一个一致的接口内的这些丰富的交互可以提供流式的、高效的可视化分析系统，它使得用户可以从混乱的软件包的管理之中解放出来，从而可以将精力集中于深入考察数据并得出推论。我在后面将带你一起来看一看丰富的统计和可视化交互，但在此之前我们将首先探讨其之所以重要的原因。

谁想要对社交网络进行可视化

我在学术界和工业界的社交网络分析领域的研究工作都表明：在试图解释社交网络时，纯粹的统计分析是最常用的技术。虽然网络可视化在学术性文章和报告中很常见，但它们通常是在分析完成后为了和用户交流而创建的，并不一定是在探索性分析过程中所使用的。

在社交网络中使用可视化图像的历史在“Visualizing Social Networks”（Freeman 2000）中有介绍，其中包含了Jacob Moreno在1934年描述的最早的社交网络可视化例子。在图10-2中，三角形表示的节点是男孩，圆圈表示的节点是女孩。在不知道教室中每个人的详细信息的情况下，人们也可以很快地从该可视化图形中了解到：1）男孩和男孩交朋友；2）女孩和女孩交朋友；3）某个勇敢的男孩选择一个女孩作为朋友（虽然不是相互的，即这个女孩并没有选择该男孩作为朋友）；4）有两个女孩单独组成一个群组。该可视化图形典型地说明了一个合理的、结构良好的网络可以很好地解释个体的社交结构。

随着每个关系的数据维度的增加，社交网络数据会变得极端复杂。熟悉网络可视化的人可能会很同情那些负责统计的从业人员，因为当节点和边的数目很多时，设计一个有用的网络可视化非常困难。大规模的网络可视化通常是节点和边的交叉集合，而且几乎无法到达“NetViz Nirvana”（Ben Shneiderman创造的一个术语，用于描述能够看到每个节点以及可以通过它的边到达所有其他的节点）。网络可视化可能会提供聚类和游离点信息，但是总体而言，人们很难从这些复杂的可视化中得出更深入的感悟认知。

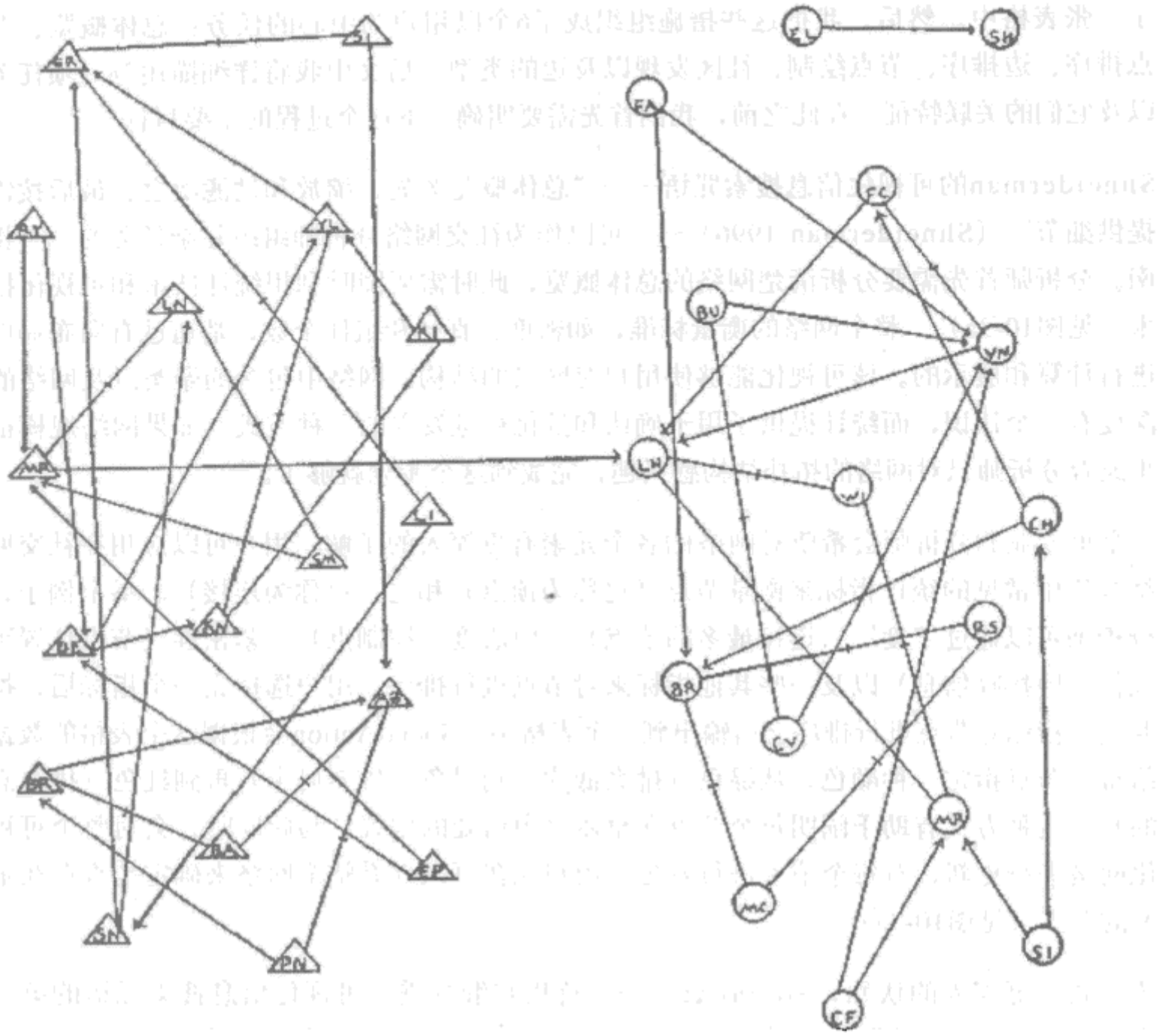


图10-2：最早的社交网络可视化之一：Jacob Moreno制作的四年级学生中的好友选择（Moreno 1934）

第一个原因是很难使用单纯的统计方法找到模式和趋势。第二个原因是网络可视化往往只是提供很少的见解，通常几乎无法提供任何的实用功能。因此，一名社交网络的研究人员应该做哪些事情？以紧密结合的方式同时利用可视化和统计技术，从而创造出美丽实用的可视化作品。SocialAction的设计即是秉着这个目标为中心。

SocialAction的设计

结构分析师提出了很多衡量方法来从统计的角度评估社交网络。然而，却没有一种系统的方式可以用来对这种网络进行解释，因为这些方式在不同网络中会有不同的涵义。这是有问题的，因为分析师希望确保他们没有忽略了网络中一些重要方面。为了使探索更加简单，我采访了几名社交网络分析师并查看了社交网络期刊，把最常用的措施汇总到

了一张表格中。然后，我把这些措施组织成了6个以用户为中心的任务：总体概览、节点排序、边排序、节点绘制、社区发现以及边的类型。后文中我将详细描述每一项任务以及它们的关联特征。在此之前，我们首先需要明确一下这个过程的主要目标。

Shneiderman的可视化信息搜索咒语——“总体概览为先，缩放和过滤次之，最后按需提供细节”（Shneiderman 1996）——可以作为社交网络分析师组织复杂任务的一个指南。分析师首先需要分析清楚网络的总体概览，此时需要同时利用统计技术和可视化技术（见图10-3a）。整个网络的衡量标准，如密度、直径和组件个数，是通过有向布局图进行计算和展示的。该可视化能够使用户对网络的结构、网络中包含的聚类以及网络的深度有一个认识，而统计提供了用于确认和量化视觉发现的一种方式。如果网络规模很小或者分析师只对网络的拓扑结构感兴趣，完成到这个步骤就够了。

一个更专业的分析师会希望对网络的各个元素有更深入的了解。用户可以应用在社交网络分析中常见的统计指标来衡量节点（也称为顶点）和边（也称为连接）。举个例子，分析师可以通过“度”（连接最多的节点）、中心度（控制点）、紧密性（节点位置可以很好地接收信息）以及一些其他指标来对节点进行排序。用户选择完一个指标后，按照这个指标对节点进行排序之后输出到一个表格中。SocialAction会根据这个表格的数据给每个节点指定一种颜色，从绿色（排名低的）到黑色（排名居中）再到红色（排名高的）。这种方式有助于阐明每个节点在整体之中所处的位置。与此同时，会对整个可视化网络进行更新，对每个节点进行着色。用户现在可以查看整个网络来确定是否存在重要的节点（见图10-3a）。

为了获取更深入的认知，SocialAction支持用户继续进行可视化信息搜索咒语的第二步——“缩放和过滤”。这是大多数其他社交网络分析工具包为“束手无策”的用户提供的一个方案。平移和拖放实际上无法真正地帮助用户找到信息：对网络中的某一块进行缩放会使用户无法了解全局结构，密集网络可能永远都纠缠在一起而无法解开。SocialAction允许用户通过自己控制的统计来驱动导航。用户可以使用范围滚动条，忽略不满足他们的标准的网络区域。通过对属性或者重要性指标进行过滤，并允许用户专注于他们所关心的节点类型，而同时简化了可视化，如图10-3b所示。

虽然分析师通过统计方法和可视化展现可以了解全局趋势，但是他们的分析通常是不完整的，没有理解单个节点所代表的涵义。和大多数其他网络可视化不同，在SocialAction中通常包含标签。字体大小和长度控制条允许分析师决定他们的重点。这与第三步中的可视化信息搜索咒语（“按需提供细节”）一致，用户可以选择一个节点来查看其所有的属性。在节点之上悬停也突出了每个节点的边和邻居节点，达到了找到感兴趣节点的NetViz Nirvana效果，如图10-3c所示。

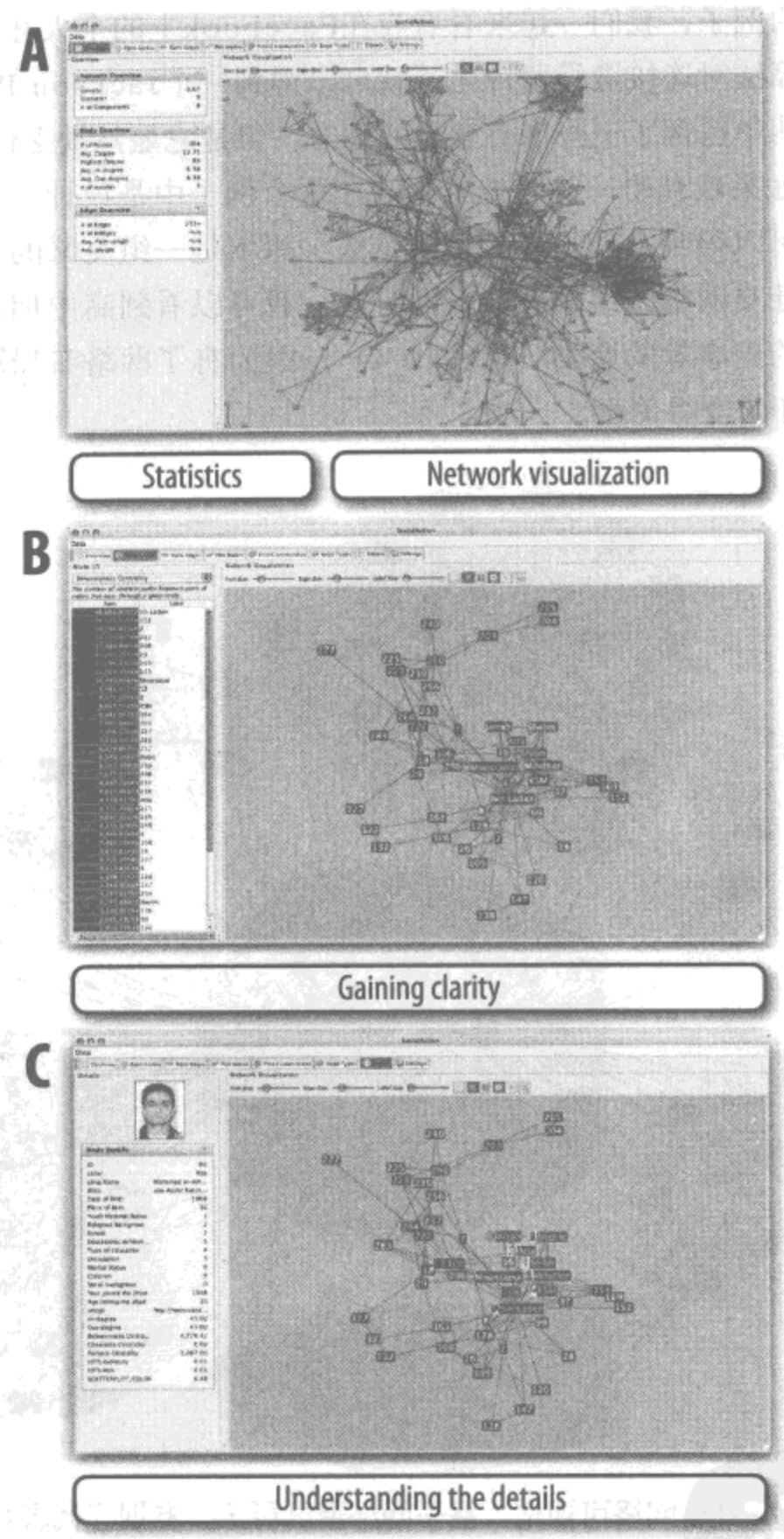


图10-3：a）该界面显示的统计部分允许用户选择统计算法，从而找到重要的节点、检测聚类等。而可视化部分是和统计结合起来的。根据节点排序对它们进行着色，红色节点是统计指标最重要的节点。b）使用统计算法查找控制点。用户使用动态滚动条过滤不重要的节点，这种方法简化了可视化，同时能维持网络中节点的位置和结构。c）标签通常被赋予权重优先级，这样用户可以理解数据代表什么。当用户选择一个节点时，会突出显示其邻居节点，并在左侧显示其详细信息（见彩图78）

另外，举个较轻松的例子，我们一起来看看我在Facebook上的个人社交网络。如果我使用标准的网络布局算法对连接进行可视化，就可以得到一个Jackson Pollack图——它看起来一团糟；虽然其中包含了一些很有意思的地方，但是它显然缺乏Lombardi图所具备的优雅性。然而，如果我利用一些统计方法（在这个例子中是设计一个聚类算法，用于检测社区），我就可以得到合理得多的输出结果。原本是一组交叉的节点和边，而现在却可以成为用于把社交网络分组成有意义的分类。我可以看到高中朋友、大学朋友、研究生朋友、在微软的同事等的聚类（见图10-4）。因为有了网络布局算法，一个原本没有任何意义的图像开始变得美丽。

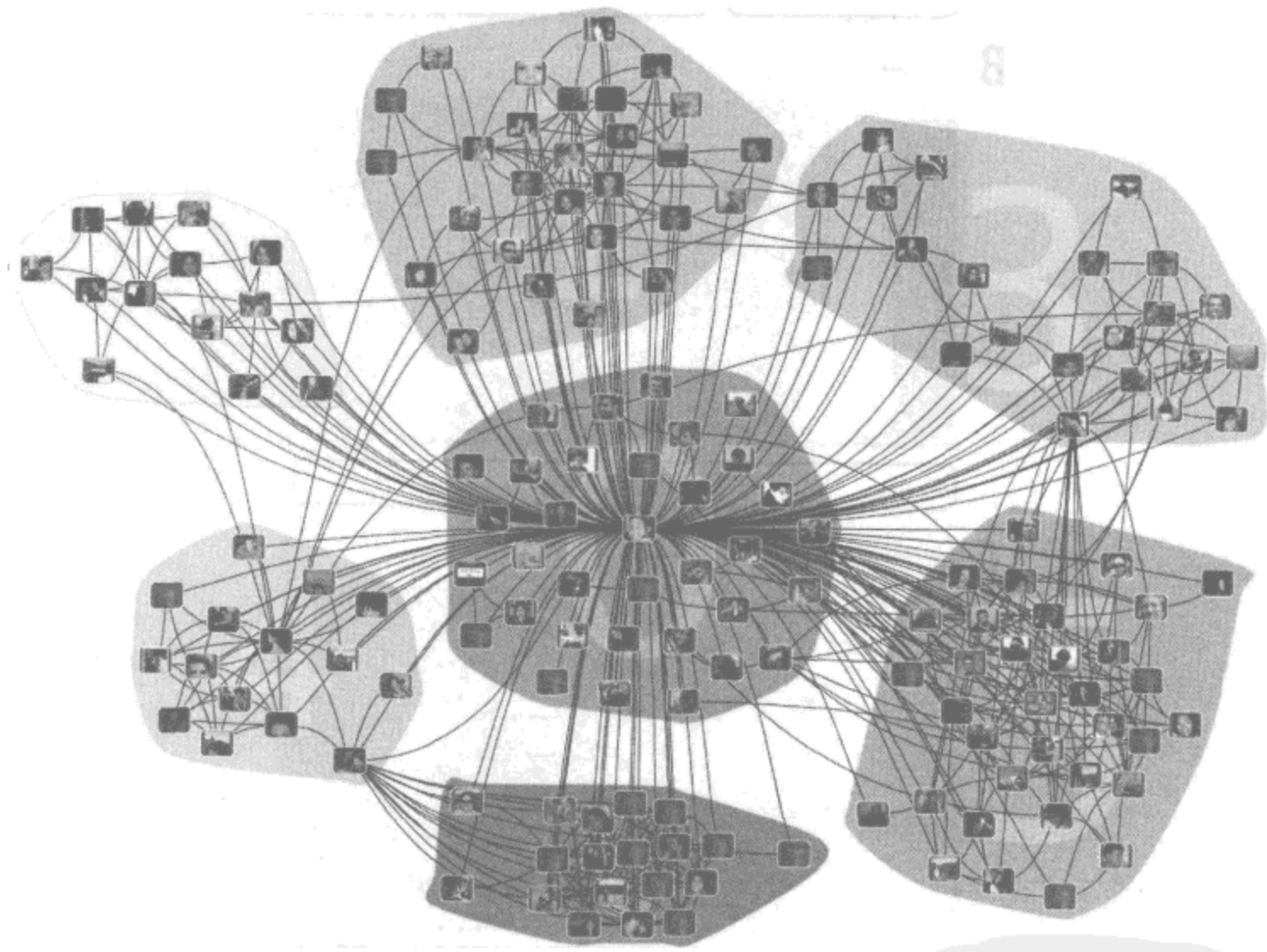


图10-4：我的Facebook社交网络可视化。基于网络聚类算法，发现了代表我生活中的不同方面的7个有意义的朋友社区。如果没有聚类，该网络就会由于有太多边而交叉在一起，导致无法提供任何意义（见彩图79）

总之，把统计和可视化技术结合在一起可以给出一套优雅的数据分析探索的解决方案。可视化简化了统计结果，改进了对模式和全局趋势的理解。而统计学又简化了对偶尔混杂的可视化的理解，允许用户专注于统计指标上重要的节点和边。

案例研究：从混乱到美丽

最终，是什么使网络可视化变得美丽？18世纪的苏格兰哲学家David Hume（1742）写道：

> 美不是存在于事物本身的品质中。她只存在于对美进行思考的人们的心目之中；而且每个人看到的美都是不同的。

然而，Hume对美的这个定义受到一些人的质疑。苏格兰副教授Henry Home（Kames爵士）认为美是可以被分解为一个理性的规则系统。

当谈到对基础数据的可视化时，我同意Kames爵士的观点。对于一个成功的可视化，其成功的衡量标准是，是否能够帮助人们产生对事物的认知。分析师可能是验证自己的直觉、检测异常或游离点，或者发现潜在模式。Virginia Tech大学的教授Chris North把认知特征化为复杂、有深度、定性、出乎意料和相关的发现。而对于有用的特征化，给人的印象是衡量认知就像衡量美丽一样复杂。传统的基于实验室的对条件进行控制的环境已经被证实对于很多科学试验是有效的，但是它们是否对于认知也有效？举个例子，如果我发明了新的展示或输入小工具，对条件进行控制的环境可以通过衡量学习时间、任务执行时间或者错误概率来比较两种或者更多不同的处理方式。典型的实验将会涉及20～60个参与者，每人进行10～30分钟的训练，所有参与者在1～3小时的时间段内都完成相同的2～20个任务。可以使用统计方法如t测试和ANOVA方法检查在均值上的显著区别。这些汇总统计是有效的，尤其当不同用户间存在较小的差异时。

然而，如果有人把认知分解成一组可衡量的任务，那结果会怎么样呢？第一个挑战是对于规模巨大的问题，分析师通常需要工作几天或者几周才能完成数据的分析，而且他们的工作过程几乎无法在基于实验室的条件可控制的环境下复现（即使在需要的时间段内可以有大量的教授参与）。第二个挑战是探索性任务在本质上就是无法明确定义的，因此告诉用户应该完成哪些任务与探索这一任务本质上就是冲突的。第三个挑战是每个用户都有自己独特的技巧和经验，这会造成执行结果差别很大，它会削弱汇总统计的有效性。在控制条件的研究中，异常的表现被认为是不幸的游离点，但是在案例研究中，这些特殊事件是有助于产出成果的关键事件，它会为发现提供认知基础。第四个挑战是我希望该工具具备更多的量化分析功能；我希望听到用户遇到的问题和挫折，以及他们那激动人心的成功故事。由于这些原因，我采取了结构化的、可复制的案例研究方法来确定SocialAction是否能够生成美丽的可视化。

以下各节概述了一些真正的分析师使用SocialAction对自己的数据进行可视化的一些案例研究。为了表达对Mark Lombardi的敬仰，我这里介绍他所做的关于政治和恐怖分子的秘密网络的研究。

参议院表决的社交网络

国会分析师对于研究美国参议院的各个党派很感兴趣。例如，《国会季刊》会对每个选票计数，计算多数民主党反对共和党的选票，然后计算每个参议员选票支持其政党的比例。这个指标可以有效地追踪不同年份每个参议员对其所在政党的忠诚度，但是它无法揭示整体格局的全局模式。

Chris Wilson当时是《美国新闻与世界报道》的副主编，对2007年美国参议员的选票模式感兴趣。Chris Wilson开始揭示数据集中参议员模式，包括战略、两大党派和地理联盟。他投入了很大努力来挖掘公共数据库中的投票数据，但是通过正常的分析方法无法找到任何不同的模式。Wilson相信社交网络分析能够产生其探索的结果。数据包含2007年最初6个月每个参议员的选票结果，从民主党开始，他们以多出一席的投票得到议院的控制权。可以依据选票的共现度（co-occurrences）来推导社交网络。

Wilson构建了一个这样的网络，当一个参议员和另一个参议员在一项决议上的投票立场一致，就用一条边把它们连接起来。每条边的强度是基于参议员之间的相同投票来计算的（比如，奥巴马和希拉里相同投票数为203，而奥巴马和布朗巴克的相同投票数只有59）。这样会产生一个非常密集的网络，因为存在一些无可争议的决议，所有参议员都投赞成票（比如，决议RC-20，一个表彰“地铁英雄”Wesley Autrey的英雄行为的法案）。所有参议员都连接在一起，结果生成一个看起来巨大的、复杂的网络可视化。

SocialAction允许用户根据重要性指标对边进行排序。Wilson使用该特征，通过动态过滤掉重要性排序低的关系来比较网络可视化。举个例子，图10-5显示了阈值为“180个选票”（约60%的选票相同）。即使对于这个非常低的阈值，党派间的关系还是很强，很可能选票和民主党一致的共和党参议员（如Collins、Snowe、Spector和Smith）也非常明显。这个可视化说明了在这个特殊的参议院中，虽然两个政党都有很强的党派性，共和党的党派性低于民主党的。

另一个意想不到的发现是随着阈值增加，民主党似乎比共和党更紧密团结，因为图中所示，民主党内的连接更密集，颜色更深。虽然每条边都有些透明，但是民主党内由于边的不断重叠产生了颜色很深的一团，而共和党内则相对稀疏得多。Wilson认为该交互可以生动地说明民主党在保持党内一致的决策会议中的成功，它是评审立法战略的一个重要方面。统计和可视化的结合使得该发现成为可能。

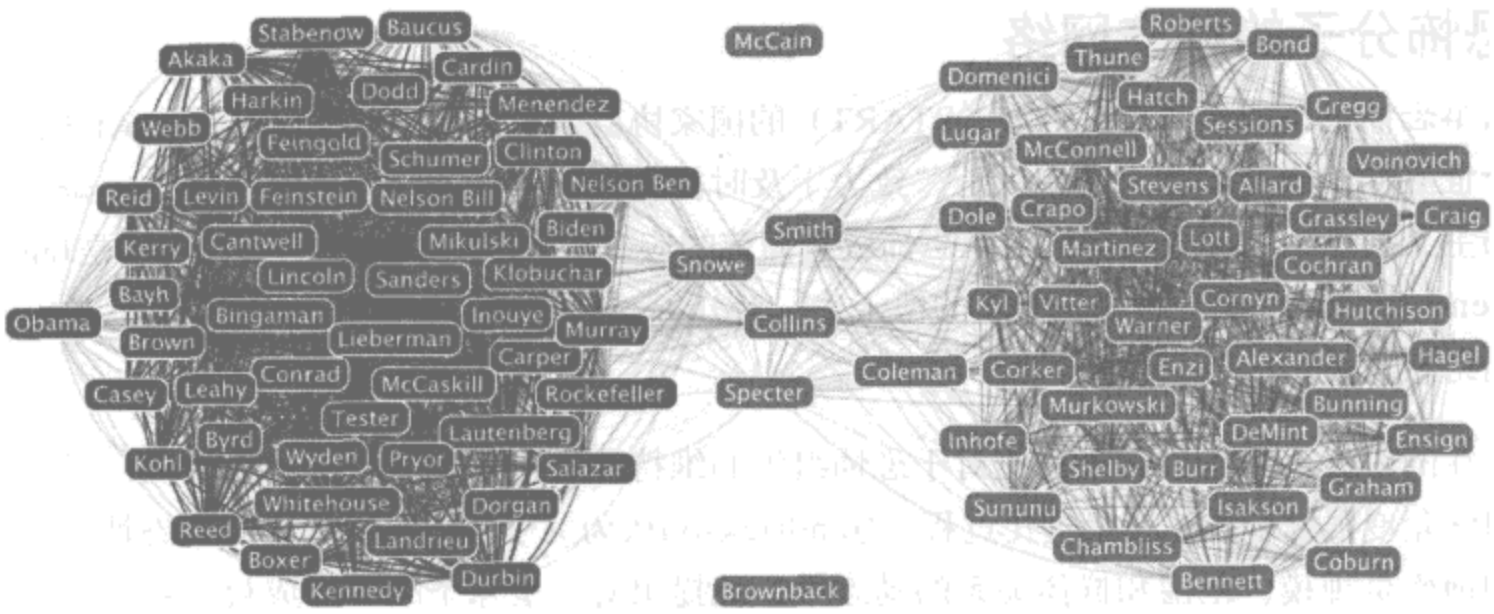


图10-5：该可视化说明了在2007年美国参议员的投票模式。红色表示的共和党显示在右侧，蓝色表示的民主党显示在左侧，另外还有两个独立派。连接表示投票记录的相似性，揭示了2007年民主党的党派忠诚度更高。4位来自东北各州的共和党通常投票支持民主党。麦凯恩和布朗巴克一起参加总统竞选，但是他们的相同选票数不足以把他们连接起来（见彩图80）

为了确定个别政客的投票模式，Wilson使用了SocialAction的统计重要性指标。对所有节点进行排序、对排序结果进行可视化，以及过滤掉不重要的节点，这样就可以带来很多新的发现。Wilson说，举个例子，介数中心性（betweenness centrality，BC）[译注2]统计之间是“量化衡量参议院的重心的很好的方式”。从SocialAction中可以明显地看出只有少数参议员是作为同事之间连接的中心。Wilson还可以使用SocialAction的交互聚类算法来“发现民主党之间在地理上的联盟”。这些发现只是在Wilson对SocialAction数据进行分析之前所未能发现的一些见解的几个例子。

SocialAction数据所揭示的一些发现给Wilson留下了很深的印象。统计和可视化的紧密连接帮助他发现并把调查结果在《美国新闻与世界报道》杂志和国会中报导，且使得人们可以易于理解。SocialAction受到了很多来自国内的关注，因此《美国新闻与世界报道》杂志能够复制其一些功能，为它的在线读者服务。完成该案例研究后，Wilson就去了《Slate》杂志，但他依然使用SocialAction进行调查报告。对SocialAction的分析还使得美国棒球大联盟（*http://www.slate.com/id/2180392*）的类固醇使用者的社交网络分析增加了交互特征，而且后期将会有更多的计划。

译注2：在网络分析中，存在4种广泛使用的中心性指标：degree centrality、betweenness、closeness和eigenvector centrality。如想要了解更多，可以访问*http://en.wikipedia.org/wiki/Centrality*。

恐怖分子的社交网络

从事恐怖主义和恐怖反应研究（START）的国家协会是美国国土安全中心。START有一个世界性的研究团队，其宗旨是“致力于及时提供指导如何粉碎恐怖分子网络，减少恐怖主义的发生，加强美国社会在恐怖威胁面前的应变能力。”该协会的一名成员是James Hendrickson，他是研究犯罪学的博士生，对分析“全球圣战”（Global Jihad）的社交网络感兴趣。

以往的研究已经指出了激进化对于恐怖组织的维持和宣传的重要性。虽然人们已经从心理学角度很好地描述了激进化过程，Hendrickson认为关于恐怖主义的团体动态性无法确切地衡量规模、范围和群体关系的动态性。他提出对“全球圣战”的成员关系的紧密程度和类型进行系统地比对，以评估他们是否可能参与恐怖袭击。Marc Sageman是START的一个访问学者，在为其后来出版的畅销书《Understanding Terror Networks》（宾夕法尼亚大学出版社）做调查研究时，收集了参与圣战的350多个恐怖分子的数据库。Hendrickson计划对这些数据进行更新并正式应用社交网络分析，并作为其博士论文的一部分。

Sageman数据库对每个恐怖嫌疑分子都包含30多个变量。这些变量表示不同的关系，包括朋友、家庭成员和教育合作关系。Hendrickson假设两个人之间的关系将极大地影响其参与恐怖袭击的可能性。他开始使用UCINET工具进行分析，可以对其中的一些假设进行分析。然而，他相信UCINET不利于探索和生成新的假设。最初，Hendrickson对于使用可视化技术来分析表示怀疑。他更喜欢量化证明统计的意义，而不是依赖于人们对图像的主观判断。然而，他说对SocialAction的可视化统计减少了他的这种担忧。

特别地，SocialAction的多样性特征有利于Hendrickson的探索。SocialAction允许用户分析不同的关系类型，而不会强迫用户下载新的数据集。可视化显示了选择的关系之间的边连接，但是节点的位置是保持稳定的，这样有利于理解。同时，统计结果也自动基于新选定的结构进行重新计算。举个例子，图10-6a只选择“圣战者”之间的关系。（和密集图10-3a相比，该图显示了关系类型。）这里的节点是通过出度和入度来排序的，因此红色节点表示其朋友最多。“圣战者”Osama Bin Laden和Mohamed Atta（因参与9·11事件，已经为广为人知）排名最高。然而，当涉及宗教关系时，出现了不同的“圣战者”核心人物；如图10-6b所示。

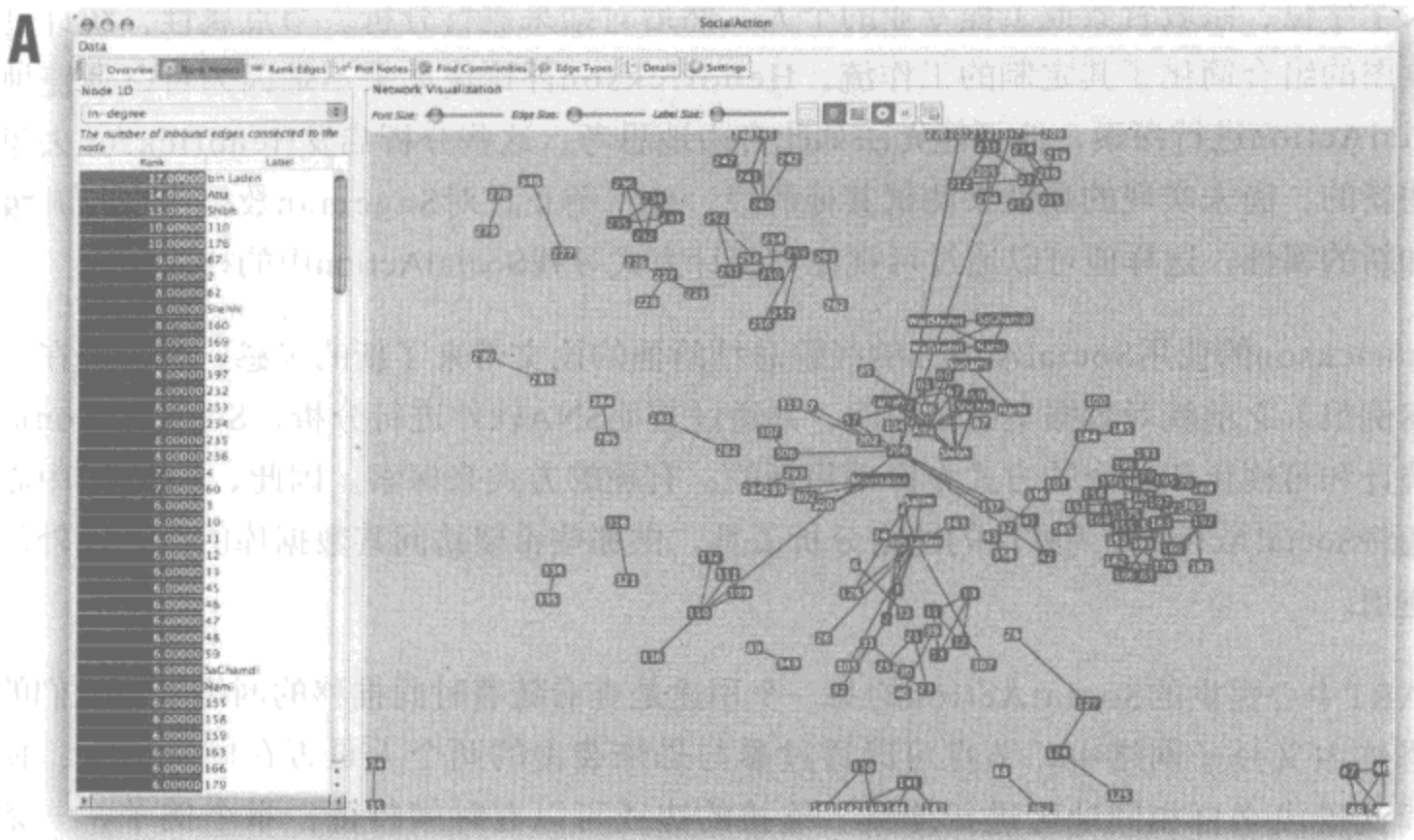


图10-6：“全球圣战”社交网络的多样性特征的演示。a）显示了朋友关系网络，其中bin Laden的知名度最高。b）显示了宗教关系，提供了对恐怖组织的不同的可视化展示方式（见彩图81）

在分析了节点的统计属性之后，Hendrickson开始对了解个人属性感兴趣。举个例子，他对于回答如下这样的问题感兴趣：“个人的社会经济地位或教育水平是否影响其在恐怖主义网络中的地位？”当然，社交网络数据不允许做因果推导，但是它可能会显示相关性。如SocialAction中的统计排序，用户可以基于属性进行排序。Hendrickson过滤掉没

有大学学位、宗教背景或工程专业的个人，然后对结果进行分析。节点属性、统计过滤和绘图的组合简化了其定制的工作流。Hendrickson评论说如果不是因为可以便捷地对SocialAction进行探索，他可能无法如此自由地思考。这些分析启发Hendrickson去思考一些新的、尚未实现的属性来测试其他假设。他目前正在对Sageman数据库进行升级，添加新的属性，这样他可以通过可视化和统计方式寻找SocialAction中的模式。

Hendrickson的使用SocialAction的经验总结给他的论文带来了新的灵感。虽然他在研究该案例很久之前就对数据集进行访问，并通过其他SNA软件进行分析，SocialAction提供的统计和可视化相结合的方式允许采用新的、有趣的方式来探索。因此，START中心有兴趣将SocialAction作为默认的网络分析工具，供那些希望访问其数据库的内部和外部用户使用。

START中心提供的SocialAction的另一个用途是查看随着时间推移的网络。在他们的全球恐怖主义分子网络中，节点可以通过参与恐怖袭击的两个人是否在同一地区、使用相同武器或来自相同地区进行连接。连接的边还可以有时间特征。举个例子，一条边可以表示某个年份的恐怖袭击。使用的边的类型取决于该分析师想要回答什么类型的问题。除了网络图以外，用户可以看到一个堆栈图，如图10-7所示。每个节点通过一条边表示，每一列表示一种边类型。每个列的节点密度表示节点在该类型的边的网络中的排序。节点颜色是基于其在所有边类型的总体排序来确定的。

在图10-7中显示了两个堆栈直方图，它表示恐怖分子网络随着时间推移的演变。这种特殊的网络有两种类型的节点：恐怖组织和他们组织的进行恐怖袭击的国家。国家节点按字母序排列，如图10-7a的堆栈图，而所有的恐怖组织都出现在图10-7b中。每个年份的节点深度是基于节点在网络中的出入度。节点是根据它们的出入度来着色的（红色表示出入度很高，绿色表示很低），在高峰年份时添加标签（在1992年有一个明显的恐怖袭击高峰）。从该图中可以解释各种趋势，比如意大利在前些年有很多不同的恐怖组织袭击，而印度在后面的几年达到恐怖袭击高峰。

由于恐怖组织的数量要远远大于国家，图10-7b有一些难以理解。然而，这些可视化是交互的，而且用户可以通过名字对它们进行过滤。因此，如果一个分析师输入“Armenia”这个单词，只有包含该词的恐怖组织节点才会被现实（比如“Armenian Secret Army for the Liberation of Armenia”（为亚美尼亚解放的亚美尼亚秘密军），“Justice Commandos for the Armenian Genocide”（为亚美尼亚种族灭绝的正义突击队））。

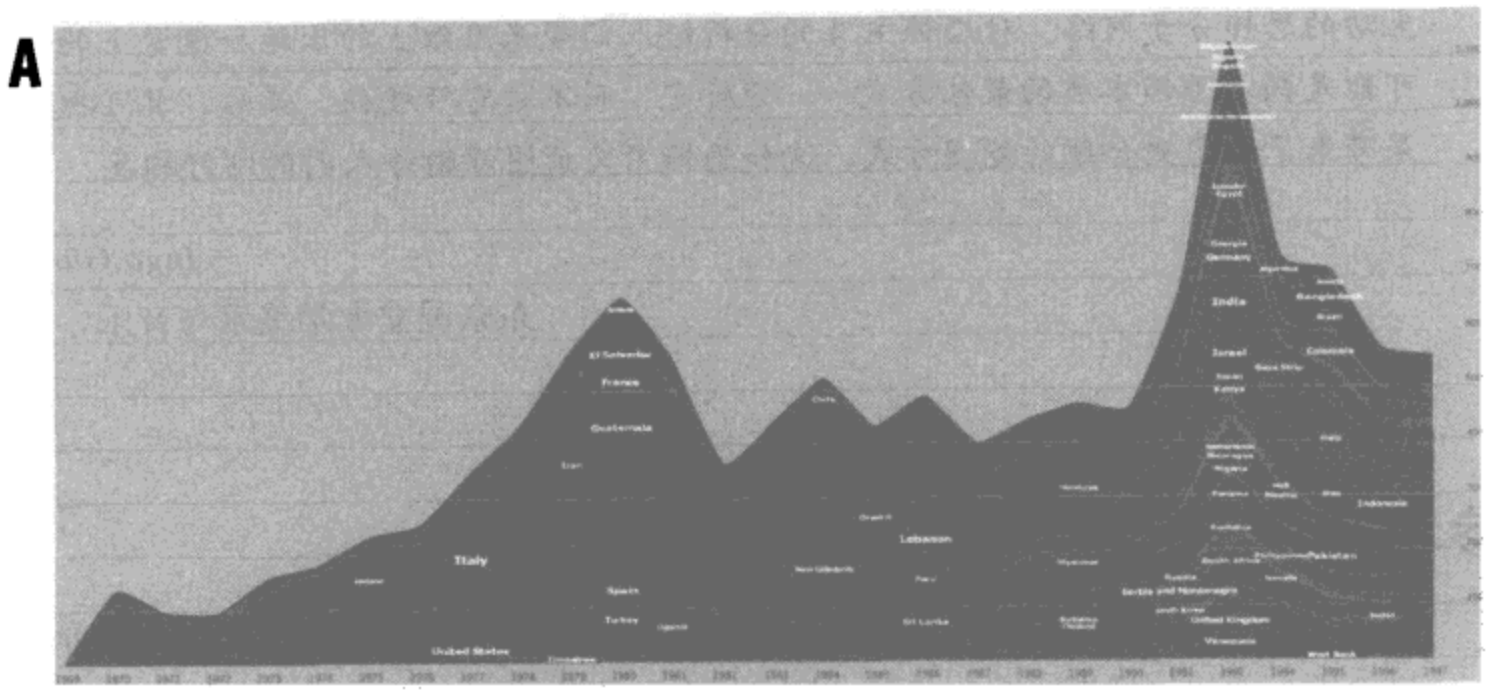


图10-7：突出两个演化网络的时间趋势的堆栈图。a）显示了国家节点的演化，b）显示恐怖组织节点的演化（见彩图82）

2007年，图10-7所示的时间可视化在纽约科学展览馆展示，作为网络动态可视化竞赛的一部分（*http://vw.indiana.edu/07netsci/*）。我将引用一个突出了SocialAction的某些目标的评语来结束本章，也许它正是道出了创建可视化之美的本质内涵：

> 网络是最佳的阅读，如果它们不仅“技术上准确”、视觉上吸引人，而且采用了一种渲染方式，为读者创建了一种景观。这种渲染方式给外行的观众架起了一座桥梁，带领他们进入专业领域。“数据领域之旅”变得如此让人舒服，它们可能很快就会出现在你附近的旅行社的特定目的地。Perer的可视化效果为我们展现了无比

生动的恐怖分子网络。对恐怖主义的分析给人们带来思维上的乐趣和视觉上的舒适可能是揭示恐怖本质的最佳方式——分析它，而不被它吓唬住。最后，其可视化效果带来了期望更合理的处理方式，这和恐怖主义试图灌输给人们的刚好相反。

—*Ingo Günther*
东京国立大学美术与音乐，日本

参考文献

1. Freeman, Linton. 2000. "Visualizing Social Networks." *Journal of Social Structure*. *http://www.cmu.edu/joss/content/articles/volume1/Freeman.html*.
2. Freeman, Linton. 2004. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Vancouver, BC, Canada: Empirical Press.
3. Hume, David. 1742. *Essays: Morale, Political, and Literary*.
4. Moreno, J.L. 1934. *Who Shall Survive?* Washington, DC: Nervous and Mental Disease Publishing Co.
5. Shneiderman, Ben. 1996. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations." In *Proceedings of the IEEE Symposium on Visual Languages*. Washington, DC: IEEE Computer Society.
6. Tukey, John. 1977. *Exploratory Data Analysis*. Boston: Addison-Wesley. van Ham, Frank. 2004. "Interactive Visualization of Small World Graphs." In *Proceedings of the IEEE Symposium on Information Visualization*. Washington, DC: IEEE Computer Society.

第11章

美丽的历史：对维基百科可视化

Martin Wattenberg 和 *Fernanda Viégas*

在维基百科的最初几年，我们创建了一些可视化来说明在线百科全书如何运作。本章将带你重温我们的创建过程：从最初的草图设计开始，到解决方案的实践直至科学论文的发表。在这个过程中，你将领略到：在所有步骤中使用真实数据工作的重要性；初始阶段使用粗糙、原始的可视化的好处；最后一点，发现可视化只是一个庞大的分析系统中的一个环节。本章所讲述的故事还说明了从感知某个领域有可能可以从可视化中受益，到确定可视化应该做到什么程度，直觉能够给成功的可视化项目带来指导作用。

描述分组编辑

故事起始于2003年。我们两个人在IBM的协同用户体验研究实验室工作，该实验室研究人们如何一起在线工作。我们发现在互联网上正在兴起一些新的协作模式，于是想对它们进行研究。我们有很多选择，那时正值“Web 2.0”刚刚开始兴起，而维基百科更是让我们格外着迷。

2003年，也就是在线百科全书诞生两年，很多人还不知道这个网站，而那些知道它的人却对这种开放的编辑模式持严重怀疑的态度。我们自己也抱有一定的怀疑，但是发现很多文章都很有意思且很有用。到底发生了什么？这样随意的过程怎么能够产生高质量的产品？除了这些最初的好奇，这些困惑感往往是一种“丰富”的研究领域的标志。我们

决定进行调研。维基百科上的文章为何能够拥有这么高的品质？为什么我们没有在维基百科上看到存在于很多在线社区中的疯狂、愚蠢和幼稚的行为？

数据

为了回答这些问题，我们需要有更多的了解。第一步是找到原始数据（正如在我们的任何一个可视化项目中所做的）。对于维基百科，其数据并不是数据库中的一个数值表，而是由各种版本的文档和编辑历史组成的一个集合。维基百科创始人最初做出的一个英明的决策是为每个页面给公众保留一个完整的版本历史。正如我们最终所认识到的，它对于维基百科的适应能力有着至关重要的影响——但是随着我们展开调查，主要感觉还是为可以使用这些数据而感到非常高兴。

这种喜悦之情之中很快就夹杂着一丝茫然。手工过滤这么多的数据开始变得让人困惑。数据库中存在由于数据过于丰富带来的一丝“尴尬”，因此现在引入一些可视化技术正当其时。

对于一个普通读者，维基百科仅仅是一个庞大的文章集合，和传统的百科全书很相似。但是在维基百科的内部，其结构是复杂的。因为大多数人现在知道，每个页面上有一个链接，读者可以通过该链接编辑文本。另外两个受到关注较少的链接被标记为关于讨论和历史。点击前一个链接会进入对话页面，读者和编辑可以在该页面中探讨一篇文章。这些页面内容丰富，从关于页面内容的讨论到寻求家庭作业帮助，表示的是维基百科的“非内容”页面。然而，到页面的编辑历史的那个链接马上引起了我们的兴趣。

编辑历史（见图11-1），包含了指向所有前期版本的完整文本的链接的列表，同时提供了关于作者的信息、编辑时间以及评论。评论是可选的，它是给作者一个机会来解释本次编辑的目的，但是编辑时间和作者这两个信息是自动写到日志中的。如果某个编辑没有登录到系统，则记录该用户的IP地址来取代其用户名。

维基百科的编辑历史在2003年已经很大了，而到今天则更是达到了巨大的地步。当然，不同文章所做的编辑次数很不相同。当我们最初开始梳理时，关于“Microsoft”的那篇文章共有198个版本（总共是6.3MB的文本），而关于“Cat”的那篇文章却只有54个版本。最开始，我们写了一个程序直接从该网站上下载编辑历史。但是，我们很快意识到这是一种很不友好的方式，因为它会给维基百科的服务器带来压力，因而，我们使用了维基百科网上免费提供的一份大文件。如果你想对其中的任何数据进行可视化，最好的方式就是自己下载一份该文件快照的最新版本[注1]。

注1：参考*http://en.wikipedia.org/wiki/Wikipedia:Snapshots*。

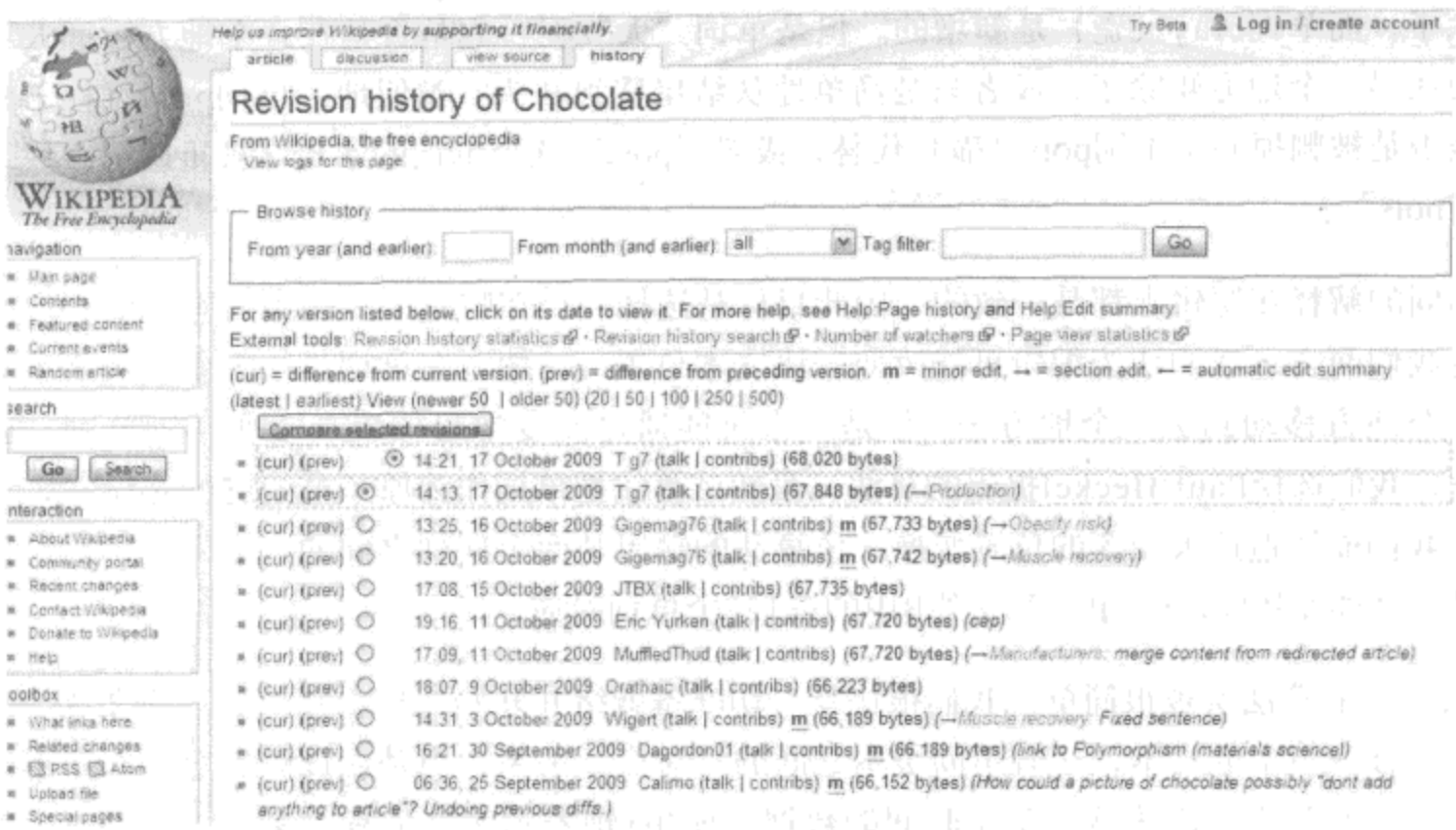


图11-1：维基百科上关于“Chocolate”条目的讨论页面：该页面列出了给文章所做的每一个修改，包括谁做的编辑，什么时候做的等

历史流：对编辑历史进行可视化

维基百科可以显示几组不同版本的差别，突出所增加和删除的文字，但是我们希望能够看到一篇文章随着时间推移的所有编辑的总体概览。为了达到这个目的，我们引入了一种新的称为“历史流”（history flow）的可视化技术。

即使我们手中有数据，我们也无法直接开始编写图形代码。我们需要自己计算出相邻的文章之间的差别。找出两篇文章的差异出现的位置以及内容间的具体区别，这看起来像个日常的运作程序，类似于普通用户使用的应用程序Microsoft Word以及开发者们使用的类似版本控制软件这样的开发者工具。但是这种做法实际上比看起来更灵活一些，虽然（可能也正因为）这个问题已经被人们研究了很长时间，最终发现不存在最佳的方式来实现这个功能。

目前的挑战在于不存在唯一的某种方式能够描述文本之间的区别。举个例子，考虑以下两个句子：

- “行动迅速的棕色狐狸跳过了大柱子（The quick brown fox jumped over the big post）。”
- “大的棕色的狐狸跳过了瓷壶（The big brown fox jumped over the clay pots）。”

大多数算法能够告诉你和第一个句子相比，在第二个句子中，单词quick（迅速的）被删

除了，而单词clay（瓷）是新增的。但是单词“大”呢？它是否是在一个地方被插入了而在另一个地方删除了，或者只是简单地从结尾移到开头？类似地，单词post（柱子）是否是被删掉并由单词pots（壶）代替，或者“post”这个单词的字母序被重新组合成了“pots”？

不同的解释在逻辑上都是一致的，因此目标是选择一个在特定上下文中有意义的算法。对我们而言，我们认为编辑可能改变一块文本位置——把一个单词或一个句子从文本的一个地方移动到另一个地方——但是不太可能通过改变字母位置来改变个别单词。因此，我们选择Paul Heckel提出的算法，虽然该算法把单词作为原子单元进行处理[注2]，它使我们能够追踪大段落的位置变换。该算法的输出是两个序列之间的一组对应，其形式是“文件A中的第5个单词和文件B中的第127个单词对应。”

Heckel的算法实现很简单，我们很快就一切准备就绪并开始分析。对于每一篇文章，我们有每个版本的文本，还有不同版本之间的“对应”关系。但是应该如何对不同版本进行展示？首先，因为这是基于时间的数据，使用*x*轴表示次序是有意义的，把第一个版本放在左边，第二个版本放在右边等。这种方式适用于查看一篇文章的编辑历史，因为文档中每个位置都像一条“河流”上的不同“湍流”。刚开始，*x*轴只是表示序列化信息，每个版本是一个坐标点，不同坐标点之间的像素数相同；然后我们加入一个根据编辑时间的设置版本间距离的选项，因此间隔很短的版本之间在空间距离上也很紧密。这两种查看数据的方式后来都被证明是很有用的。

接下来，我们需要对文档位置和段落之间的对应关系进行编码。我们决定使用竖线描绘版本，其长度与每个版本的长度相对应。实际上，*y*轴对每个版本内部的文档位置进行编码。一旦我们做出这个决定，就很容易知道应该如何在一个版本到另一个版本间画线来描述匹配关系了，如11-2所示（它是我们在开始编码前在白板上手工描绘的一个素描）。

我们第一次计算出的版本看起来大致如图11-3所示，它描绘了单词Abortion（流产）在2003年的页面编辑历史。该图看起来有些丑陋且让人费解，但是存在一种清晰的结构，甚至是某些特征使我们开始怀疑代码中是否出现了问题。举个例子，你会注意到版本4中有一条明显的间隙。我们手工检查了数据，确定这并不是代码的错误：我们看到的版本是被一个恶意用户删除掉了文章的大部分内容。啊哈！该可视化已经开始把我们的注意力吸引到该文章的编辑历史的一些重大事件中。

注2：　从技术思想上看，该算法工作如下：首先找到在每个序列中只发现一次的词项单元（token），然后把这些匹配扩展到更大的连续分块中。

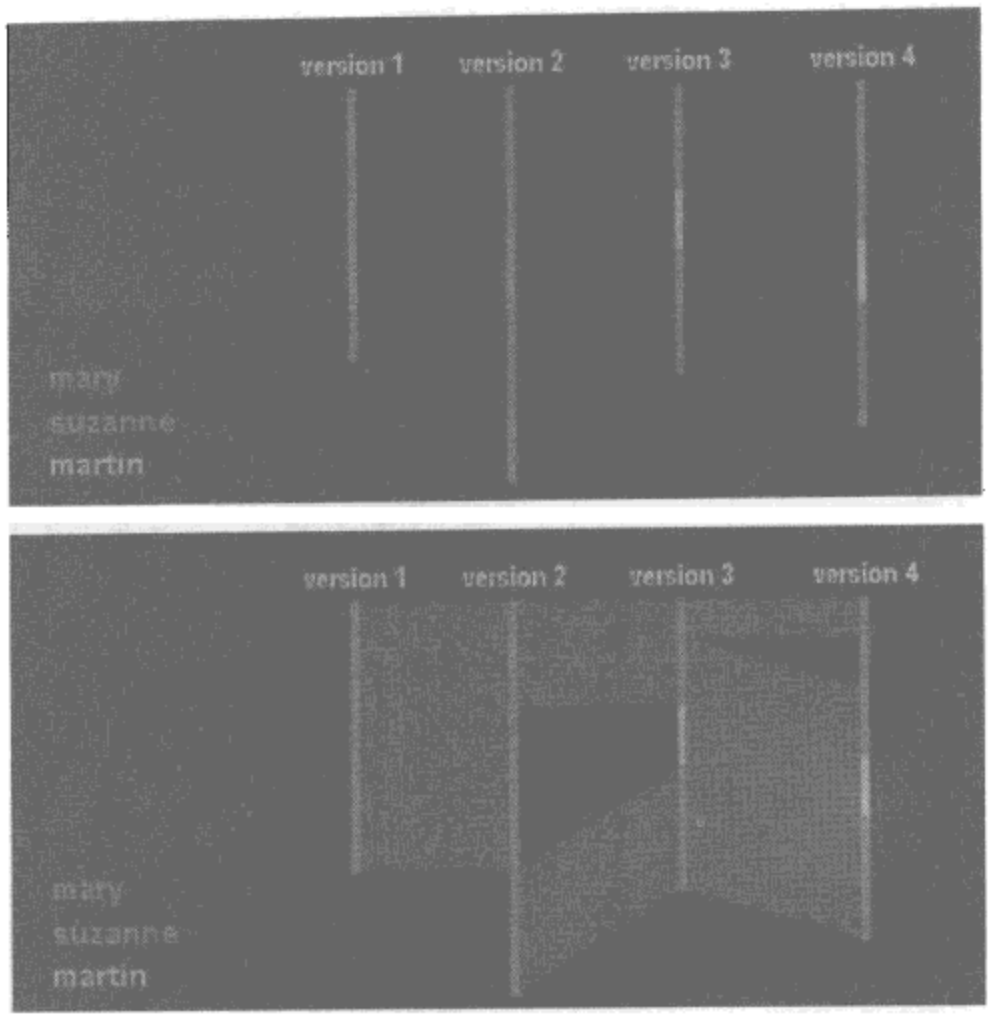


图11-2：历史流的可视化机制示意图（见彩图83）

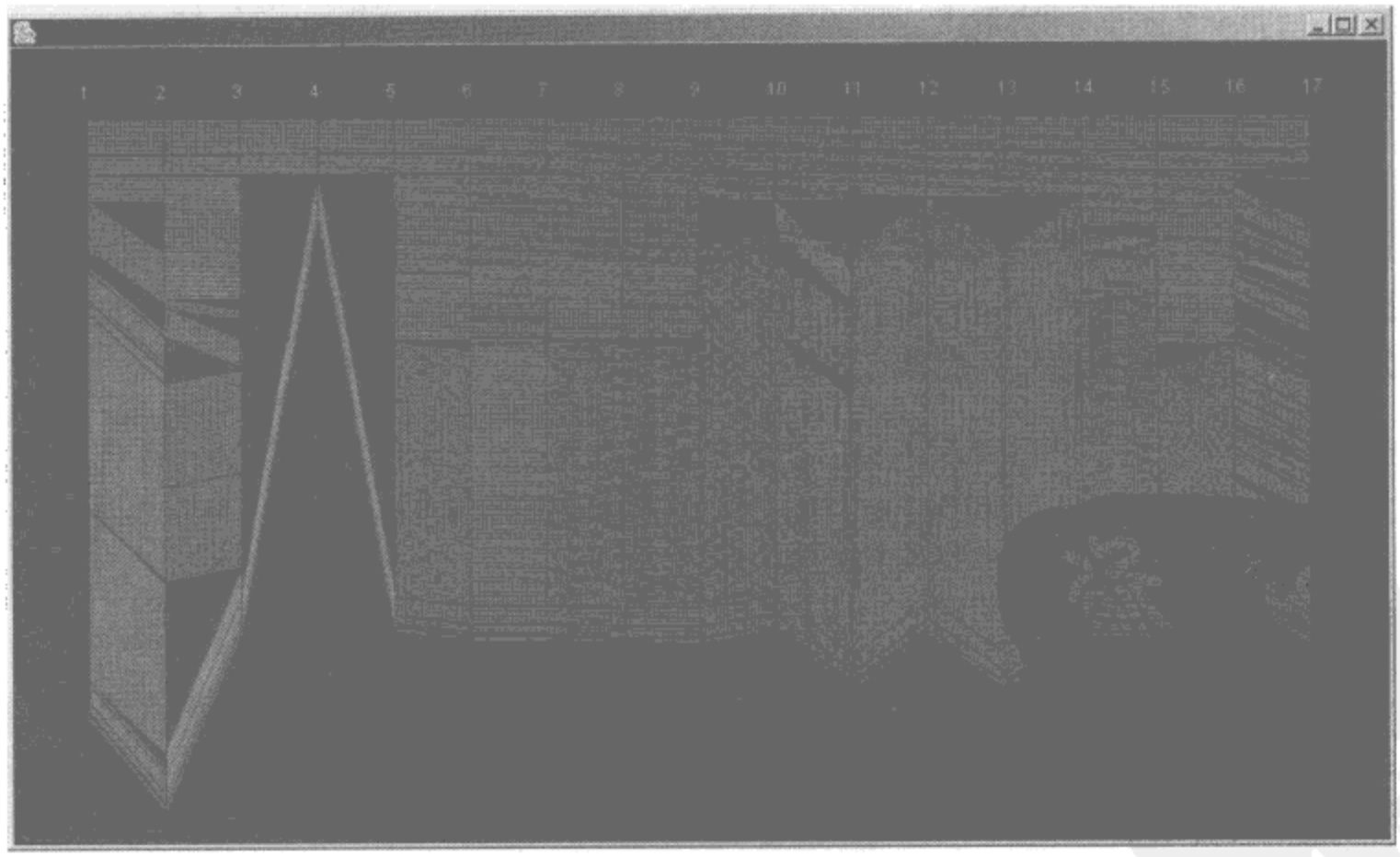


图11-3：历史流的一个早期可视化版本，通过简单的线条对连续版本中都完整的文本片段进行连接

由于通过手工方式查看原始数据源很繁琐，我们很快增加了一个特性，能够在面板右侧

显示每个版本的原始文本。这在可视化开发中很典型：在获取到对原型可视化的总体概览之后，能够查看详细信息通常是一种很好的方式。这不仅是用户通常想要的特性，而且提供了一种重要的方法来检查可视化概览的正确性。可视化结构还是难以阅读，因此我们决定进行相应的“填充”，即对每对平行线内部进行填充。图11-4显示了填充结果。

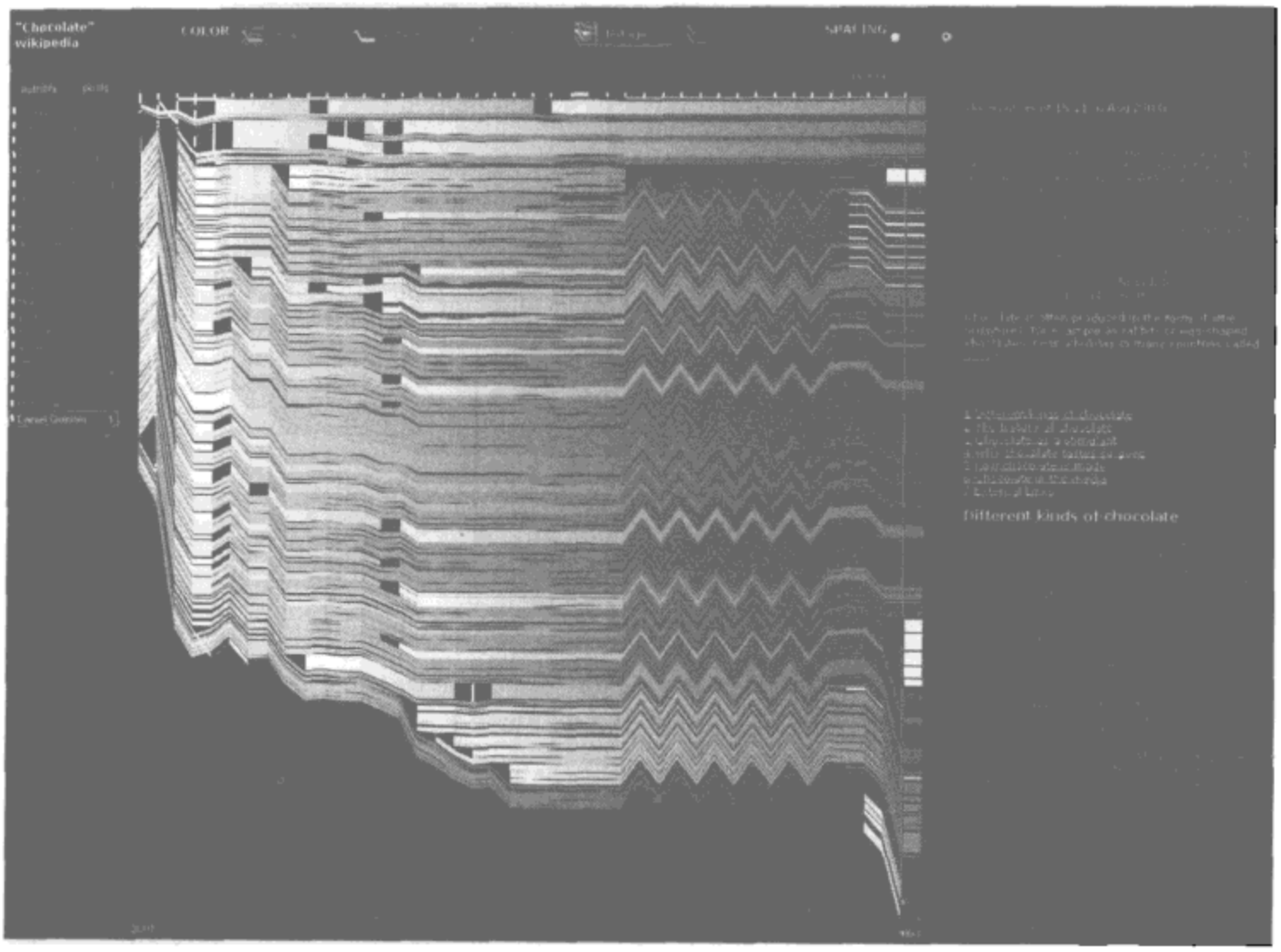


图11-4：历史流图显示Wikipedia上的“Chocolate”条目的相关文章的文本在不同年份的变化：颜色更深的分块表示时间更早的文章（见彩图84）

结果图片易于理解，而且看起来也没有那么复杂。实际上，我们现在认为存在自然的方式来呈现另一种变量，通过对连接相应文章的多边形进行着色。

编辑年份

在网站上的编辑历时很长的页面是否比历时短的编辑页面质量更高，我们对这个问题很感兴趣，同样感兴趣的是是否可以以任何其他方式对编辑质量进行区分。年份是一个简单的数值变量，使用灰色来描述是有意义的，如图11-4所示。这是我们增加的第一种彩色效果，它存在两个优点：一是说明了年代这个维度；二是深浅变化的灰色实际上使得整张图形变得更加清晰易读。这可能和人们的直观感觉有所不同，但是属于可视化中的常见现象：增加额外信息实际上可以帮助我们理清复杂的流程图。

著作权

然而，我们的真正目标是要找到群组编辑背后的驱动力。因此，我们需要对著作权进行描述。我们拥有必要的数据，因为每次编辑都包含了著作权信息（登录的编辑人员的用户名，或者匿名贡献者的IP地址）。我们应该如何给每次编辑分配颜色？我们希望有多种颜色，这样可以区分开不同的贡献者，而且我们希望任意一个贡献者在不同页面的颜色都相同。同时，我们希望能够区分开匿名的和登录的贡献者[注3]。

我们最后决定采用不同的编码方式，通过该方式应用软件会为每个用户选择鲜明、饱和的色彩。用户的色彩实际上并不是随机的，而是基于对每个作者名字的Java“散列码”（hashcode）。这种技术实现可以确保每个作者的色彩在流程图中保持一致，而且存在很广泛的色彩变化空间。对于匿名编辑，我们选择浅灰色来表示。

整体视觉效果很显著，如图11-5所示。这样，用户可以对包含很多匿名编辑的页面（显示一片灰色）和完全或主要由登录用户编辑的页面（充满彩色显示）之间的区别一目了然。当一篇文章的编辑工作主要是由一些编辑完成时，也可以很容易区别。为了把作者名字和色彩关联起来，我们在屏幕左侧增加了一些说明。

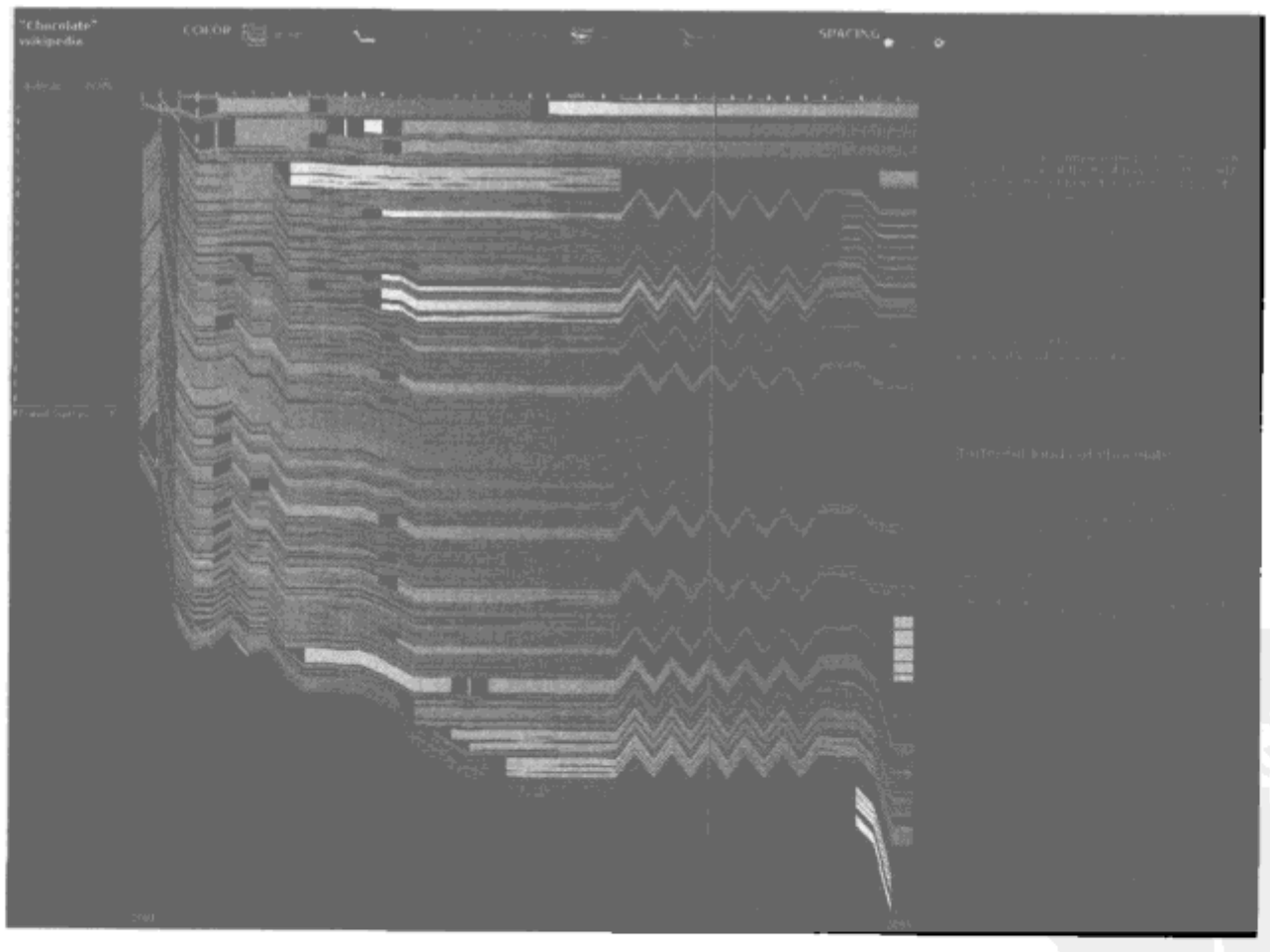


图11-5：历史流的彩色显示：每种颜色表示某个作者所编辑的文本（见彩图85）

注3：对匿名用户基于其IP地址来分配不同的颜色看起来可能有欺骗性，因为地址和实际用户之间没有明显的关联。不同的人们在不同时间通过公司网络登录可能会显示相同的IP地址，相反地，同一个人从不同IP地址进行编辑也很寻常。

作者个人

接下来，我们希望当只查看做者个人的贡献时能够更加简单。为了这个目标，我们把作者的故事做成可点击：选择一个作者，对流程图进行着色，因此被选定作者所做出的贡献会采用很鲜亮的奶油色来突出表示，而流程图的其他区域在显示上则颜色更深（见图11-6）。我们在采取该措施之前尝试了一些其他方案。保持被选定作者用很鲜亮的颜色显示而其他作者用较暗淡的颜色表示，达到这种效果的另一种做法是使用白色表示被选定的作者，但是这种方式并不能突出选择，反而会让人费解，因为主视图中的灰色带表示的是匿名编辑。

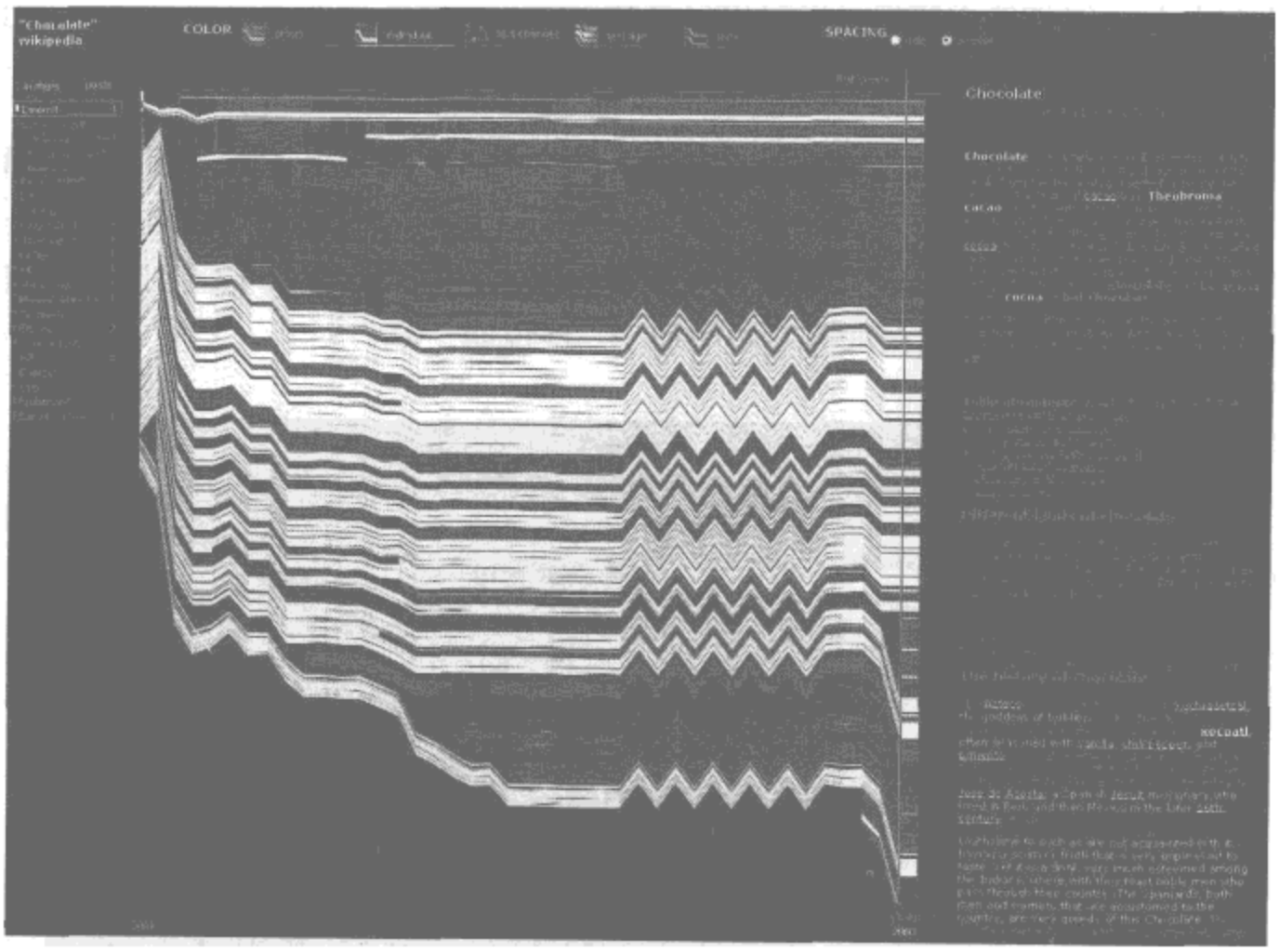


图11-6：奶酪模式显示的流程图，表示单个作者随时间所做出的贡献（见彩图86）

然后，我们增加了一些其他小的特征和编码，但是实际情况是开发速度开始放慢，因为程序变得很有意思[注4]。实际上，它可能是太有意思了！我们不再一直写代码，花了很多时间看一篇又一篇的文章，着迷于各种各样的模式。这对于可视化开发始终是个好兆头，而从身边走过的人常常被我们屏幕上的图片所吸引，开始停下来和我们长时间地交谈。

注4：还是存在很多其他方面我们还没有探索。当Ben Fry独立创建了一个历史流图版本“revisionist”来显示“Processing”的环境演化过程时，我们看到了一个这样的并行可视化世界。不是增加颜色和交互性，他采用全局的方式，使用优雅的曲线和在y轴上文档位置的变化，使得可以很容易追踪到各种不同的变化。

可视化允许我们很快地了解参与一篇文章编辑的不同编辑人员、每个人所做的改动甚至是做出最后的决定上产生的分歧。我们按捺住了对无数的文章进行可视化的冲动，决定至少在那个时候，可视化工作已经完成了。显然，它满足了我们初始的目标，采用协作模式看起来对于调查很有保障。接下来，我们把注意力转到使用它来获取科学上的结果。

历史流的实际作用

随着我们对文章的研究，我们开始采用了探索模式。在查看了一个又一个的流程图之后，我们开始慢慢地认识到什么是正常的，什么是怪异的。我们还开始看到一些不同类别的行为变化，如“编辑战争”，在这些“战争”中，一些编辑不断地撤销别人的修改，在可视化显示上是很醒目的之字曲线图。更重要的是，我们开始跟踪该图片给我们提供的一些线索。

如何追踪可视化线索，从定性研究转移到定量研究的一个很好的例子是，我们对一些经常被恶意篡改的文章如“Abortion”条目的调查。从图片中可以很清晰地看出恶意篡改通常只会在站点上保留几分钟的时间。当查看每个版本都显示一样大的历史流图时（见图11-7），我们看到特征化的黑色裂纹表示恶意删除；当通过编辑时间对版本进行显示时，这些裂纹通常就会消失（见图11-8）。

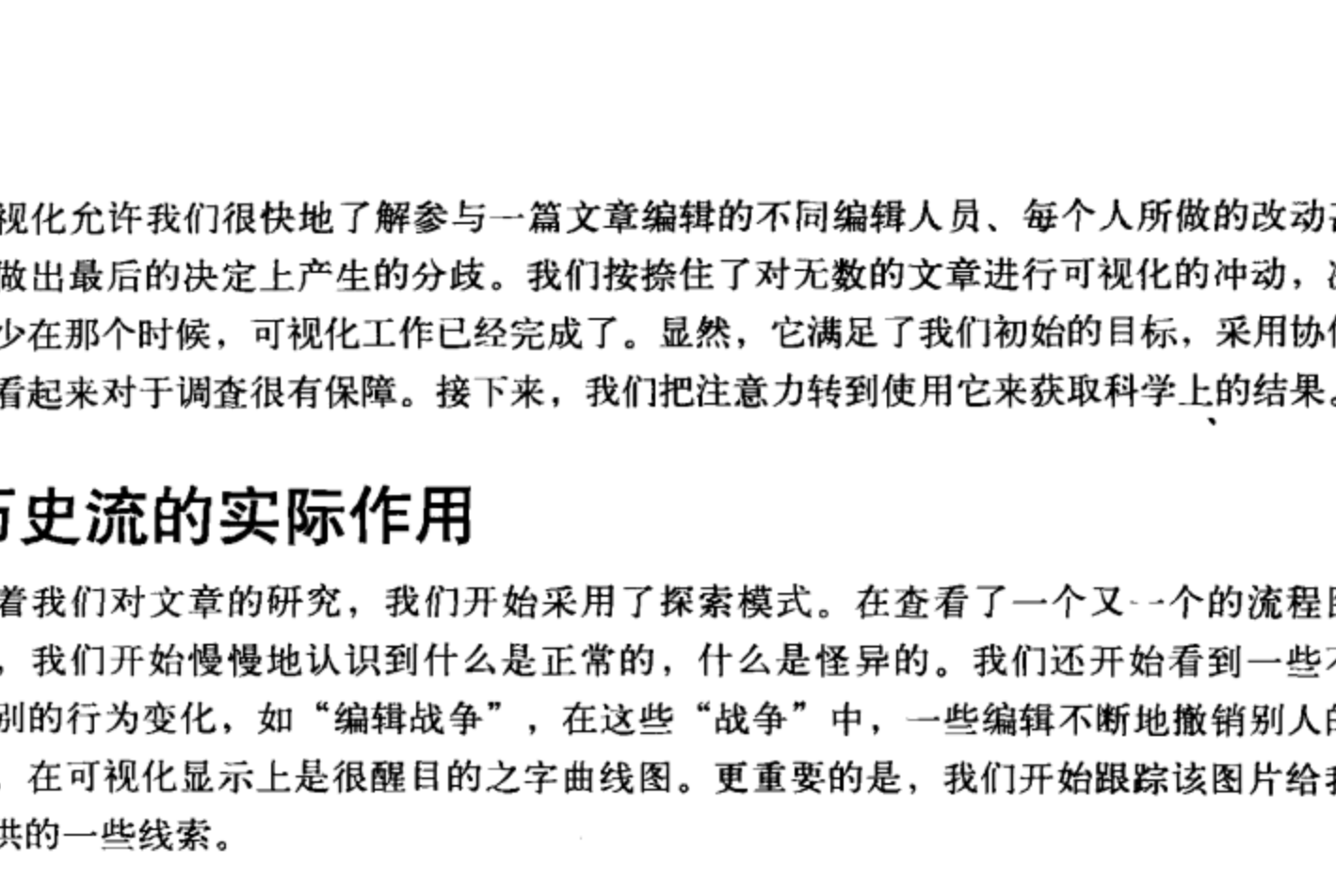


图11-7：“Abortion”条目的页面的编辑历史，显示了空间上等价的不同版本——黑色裂纹表示“恶意删除”，即某个用户把某篇文章的所有内容都删除掉的恶意行为（见彩图87）

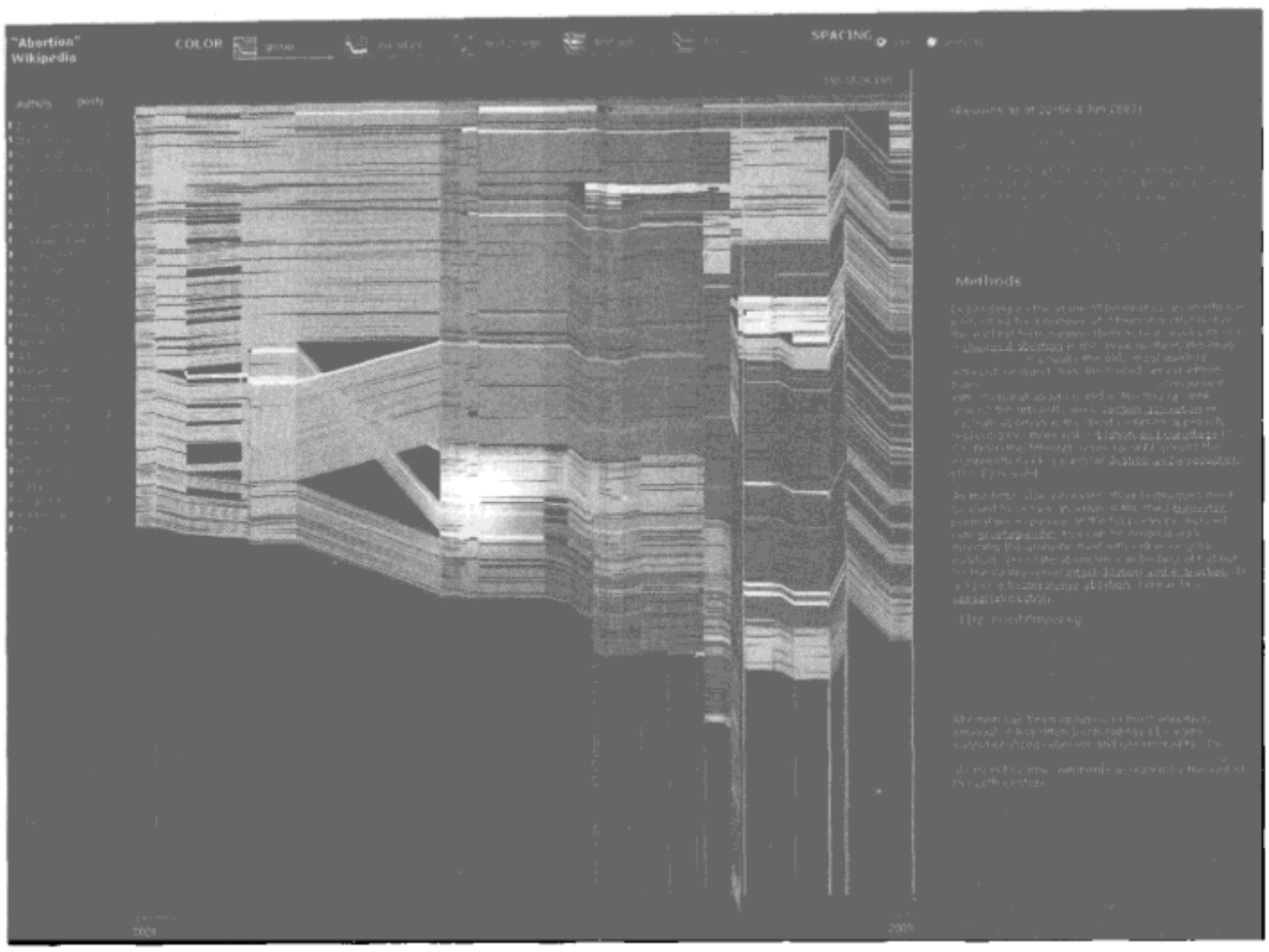


图11-8：“Abortion”条目的页面的编辑历史，显示了按时间进行划分的不同版本（见彩图89）

即使多次发现这种模式，然而它也并不能构成科学依据。可能我们想到的文章刚好是特别有争议的或者监管良好的。为了说明恶意破坏和快速修复实际上非常普及，我们需要考虑更多的页面。为此，我们对整个维基百科的编辑数据库进行扫描。在同事Kushal Dave的帮助下，我们创建了一组标准可以识别出特别明显的恶意破坏[注5]，并实现了一个程序，可以检查所有满足这些标准的编辑。结果发现是绝大多数这种恶意编辑在几分钟内就被撤销了，说明了维基百科的编辑对于变化有密切地审查。

对结果进行沟通

对主观印象进行统计确认是我们所面临的最后一道难题，并且这种统计确认方式提供了令人满意的解决维基百科的初始问题的方式。我们没有看到破坏性行为的证据的原因不是因为这种行为不存在，而是因为它往往可以很快地从公众视野中消失。我们记录下了这些结果，并提交了一篇科学论文，但是我们对其研究并没有就此停止。

注5：　我们是通过寻找文章长度显著减少以及页面中存在低俗文字来判断的。这种方式当然无法识别所有的恶意破坏、但是它所挑选出的编辑确实绝大多数是恶意的。

除了添加不同的科学例子来支持我们的理论，还存在一些数字可以很容易地解释我们的结果。反过来，可视化从深度和细节上给这些数字增加了可信度。我们发现这些结果存在很多科学界领域外有意思的地方。一方面，那些不熟悉维基百科内部运作模式的人很快就被在线编辑、公共百科的神奇所吸引。另一方面，那些了解开源编辑风格的研究人员则会惊叹于其图像的清晰度和瞬间所能够展示的信息的丰富性。历史流证明了对在线社区进行可视化所蕴涵的价值同时满足人们对文化的兴趣和科学的研究。

染色图：一次对一个人进行可视化

2006年，我们重新访问了维基百科。百科全书人气很旺，我们想找出参与的贡献者的更多信息，尤其是那些贡献了很多编辑的核心活跃用户。他们是如何分配时间和精力的？我们对于数据是否匹配Yochai Benkler的“对等生产”（peer production）模式特别感兴趣，这种模式的行为包括从维基百科的创建到Linux的创立。

我们和一个非常有才华的实习生Kate Hollenbach一起决定对网站的管理员（admins）、享受特权（如阻止其他用户或删除页面）的超级用户的编辑历史进行分析。管理员通常在站点上有很长的编辑历史，而且代表的是维基百科社区的核心贡献者。

我们为了了解该数据做出的第一个尝试是创建了一系列的图表和图形来表示随时间变化的活动水平。创建活动图本身很简单。显示该数据的标准方式是一个线条图，x轴表示时间，y轴表示编辑次数。我们制作了一系列的这种图表，它们虽然很清晰但是我们感觉其信息量还不够丰富。和历史流图不同，我们通过该尝试没有发现意外的模式或者是可以启发新的调查的线索。

其中一个问题是简单的图表概括了太多的数据；成千上万的编辑压缩成单一的数值时间序列，最终导致我们必须删除重要的信息。我们面临着可视化项目中一个典型的抉择：随着我们对数据进行探索，我们应该“以多低的距离来飞行”[译注1]呢？不存在先验知识可以预先确定是否存在有趣的小规模的模式。但是既然我们从“30 000英尺的高空还无法识别任何东西，我们只能选择飞得更低。”

显示所有数据

为了能够更接近“地面”，我们决定查看每个编辑人员编辑过的各个页面。对维基百科进行编辑是重复、复杂的业务，我们觉得需要在可视化中反映这一点。其挑战在于一些管理员贡献了10万多份编辑！（最活跃的用户在两年中平均每隔10分钟就执行一次编辑。）很少有可视化技术能够把这么多的数据点显示成一个可以理解的图片。

译注1：即应该以什么样的粒度来研究。

然而，我们的可视化技术在渲染大数据集上非常有优势。在学术文献中为人所知的一系列方法是关于像素填充，它是把每个数据点表示成一个像素或者最多表示成一个很小的矩形。像素填充可视化方式是尽可能地把信息打包到屏幕中，而其稠密性往往会带来一种缥缈的美丽。实际上，艺术家Jason Salavon把整部电影显示成一组像素的美丽的作品启发了我们去实现进一步的探索[注6]。

应用这种技术，我们把管理员历史中的每次编辑在屏幕上表示成小矩形。把这些矩形放置在分块内，按时间次序从左到右、从上到下查看。然后，由于空间位置显示的是序列化信息，我们只能采用一种方式：色彩。对于所有由像素填充的可视化，按照定义，确实如此。通常，颜色是由表示数值维度的梯度来定义的。挑战在于那些最重要的变量——文章标题和编辑评论——都是原始文本。

为了把这些文本片段转换成色彩板，一种自然的方式是尝试我们在历史流中使用的相同的散列编码技术。当我们应用该技术后，我们确实开始看到模式：一个编辑多次处理相同页面会显示成一条彩色块，而在其他情况下，我们一点都看不到重复，这表示对很多页面进行编辑的编辑人员通常只是对每个页面做了一处改动。虽然现在看到的细节比以前多得多，我们还是觉得有用的信息被隐藏起来了。一方面，文章名字的结构不是由散列编码来捕获的。通常，相关的文章以相同的短语开始（如“List of”或“USS”）。我们意识到这种结构可以通过字母序着色方案来保存，其中每个字符串的首字母确定其颜色。图11-9解释了着色方案，而图11-10则说明了如何构造流程图。

图11-9：对在维基百科编辑评论中发现的单词的色彩示例（见彩图89）

我们所看到的

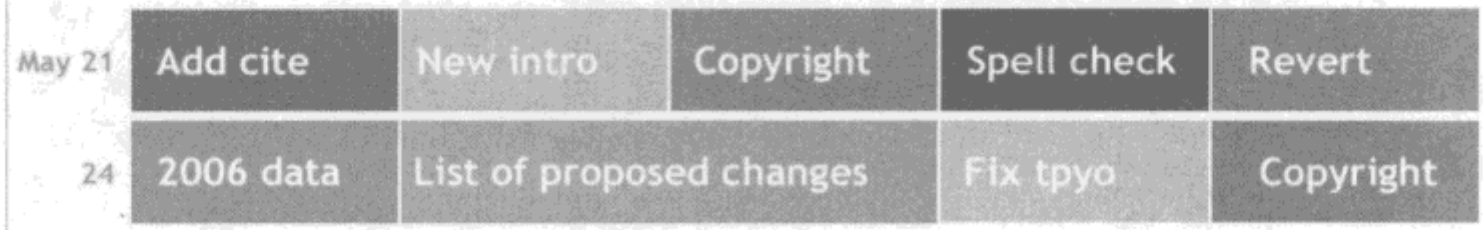
一旦我们采用这种新的配色方案，这些图片就成为焦点。虽然编辑历史依然很复杂，而且需要仔细查看，我们看到了更多类型的模式。以下几张图像大体说明了我们所查看到的。

注6：2000年，Salavon描绘《泰坦尼克》为“有史以来票房最高的电影（The Top Grossing Film of All Time）（1*1）”。每部电影画面被显示成一个点，其色彩是所有画面色彩均值。

a

时间	页面	评论
May 21, 9:32 am	Sphere	Add cite
May 22, 10:56 am	Sphere	New Intro
May 22, 1:23 pm	sphere.png	Copyright
May 22, 2:54 pm	Helix	Spell check
May 22, 3:00 pm	Mathematics	Revert
May 24, 11:21 am	Fields Medal	2006 data
May 24, 11:25 am	Talk:Fractal	List of proposed changes
May 24, 11:27 am	Talk:Fractal	List of proposed changes
May 25, 10:13 am	Sphere	Fix tpyo
May 25, 10:23 am	mercator.png	Copyright

b 块状视图



c 块状视图（压缩、无标签）

图11-10：对每次编辑的用户评论的可视化构建彩色图（见彩图90）

图11-11显示了由两种主要色彩组成的文章-标题编辑历史。我们发现这些编辑和births（出生）与deaths（死亡）这两个单词对应。典型的标题是“1893年出生”。该编辑所做的是给不同年份页面增加关于著名人物的出生和死亡信息。

图11-11：对birth（出生）和death（死亡）相关的文章的编辑（见彩图91）

有些编辑发现了自己感兴趣的主题，并坚持致力于该主题。图11-12看起来像一个紫色海洋，该颜色对应于前缀为“USS”或“United States Ship”。该编辑致力于编辑那些描述美国海军特定船只的页面。

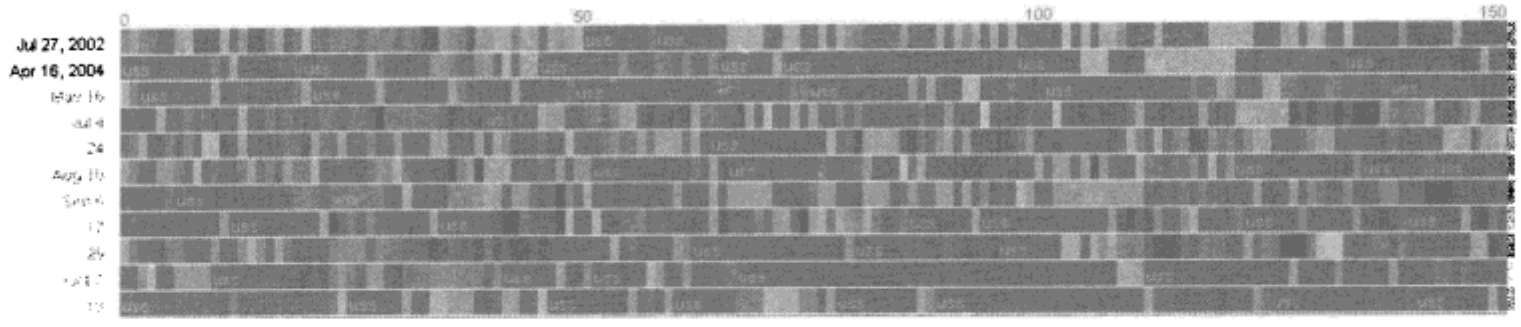


图11-12：超过1000次的编辑，绝大多数是针对标题以“USS”开头的文章（见彩图92）

查看了这些图之后，我们开始习惯于紧密和随机的颜色数组，偶尔被完全相同的色彩干扰。因此，当看到图11-13中一些区域的颜色形成一条彩带时，我们感到大吃一惊。

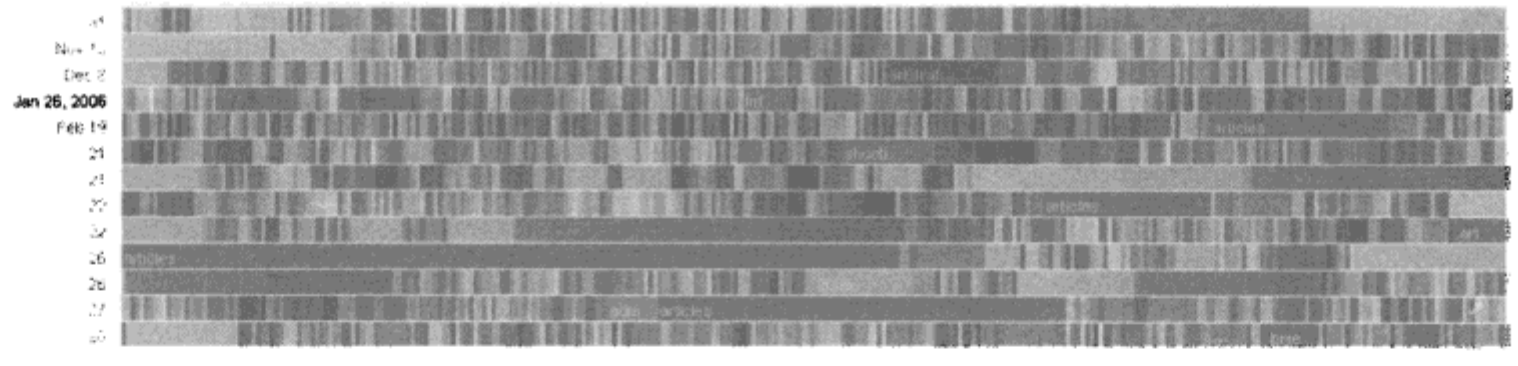


图11-13：彩带（见彩图93）

这种带来视觉冲击的模式表示了按字母序排列的文章标题。虽然从有时会出现短字母序模式，但是我们看到了很多长字母序模式，有的非常长。这是值得研究的一个非常好的先例。为什么会发生？它又会对维基百科带来什么样的影响？

有些彩带看起来很微妙。而其他的则看起来如图11-14。谁能够做到如此有序的编辑？当我们查看用户页面时，发现是由一个“机器人”完成的：设计了一个软件程序，用于执行自动编辑。在这种情况下，这些编辑包含了大量的关于地理位置的文章的基础分类。

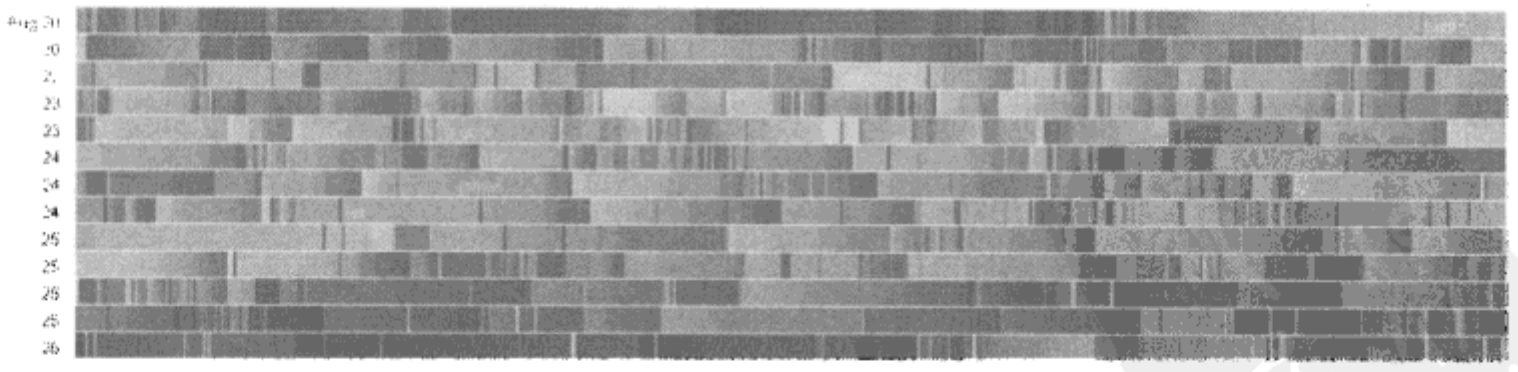


图11-14：“机器人”（见彩图94）

分析数据

对于历史流，我们决定通过统计学方法对一些视觉印象进行验证——举个例子，彩带问题表示按字母序的编辑。首先，我们写了一个程序可以识别这些序列，并根据出现频度

计算概率，验证它不是随机事件。然后，我们进一步研究。如果很多用户是以字母序来编辑的，是否表示标题按字母序排列在前的文章吸引了更多的注意力？这看起来可能有些编辑会乐观地开始长期编辑很多页面，而最终却只是半途而废。为了测试假设而完成数据收集后，我们发现在文章标题和编辑次数之的字母位置间存在反向关联，这证实了我们的直觉，以字母“a”开头的文章的被编辑次数要远比以字母“z”开头的文章多。但是，这种关系也不是绝对的，举个例子，以字母“L”开头的文章，由于其包含的列表数目最多，其编辑次数也最多，但是这种关系还是足以作为统计上的一个重要参考。

这些彩带使我们更细致深入地查看编辑是如何使用列表来组织他们自己的以及别人的工作。这种现象和Benkler的“对等生产”理论是一致的，在该理论中，工作被划分成小的单元，人们可以自己分配时间。可视化促使我们对初步研究的问题得到满意的解决。

结束语

正如我们的故事所示，创建可视化会面临错误的开始和死胡同。但是，虽然道路是曲折的，但它并不是随机的。我们给出的两个例子都遵循一致的过程，它是我们通过对过去的几十次可视化不断进行完善得出的。以下是我们在所有的可视化项目中总结出的3条基本准则：

采用真实数据

获取到好的数据往往既困难又折磨人。不论是谈判获得数据库访问权限的法律合同还是写一个程序从Web中获取信息，为可视化获取原始材料是很困难的。可能由于这个原因，很多人会尝试多任务并发，甚至是在他们还处在获取原始数据过程时，就开始设计可视化。根据我们的经验，这种做法绝大多数情况下是错误的。举个例子，在Chromogram项目中，只有在查看一组相关的文章标题时，我们才意识到按字母序的着色方案可能是有意义的。

尽早并经常进行可视化——但是知道该什么时候开始

对于其他类型的软件开发，迭代开发是很重要的。每个项目都开始于一系列的设计草图。对于历史流，这些草图最终慢慢发展成为最终的可视化。而对于Chromogram项目，我们抛弃了所有的草图，从另一个思维角度查看数据。在每一种情况下，我们都对细节粒度进行了调整（多大“粒度”）。对于历史流，增加对不同作者的着色区分和编辑年份的指示说明突出了可视化视图的重心。而对于Chromogram项目，在把数据以可能的最细粒度展示前我们没有获取到任何信息。迭代并不能一直持续，因为我们需要注意自己已经做的所产生的效果。历史流和Chromogram这两个可视化项目都可以进一步完善，但是它们都达到了我们所期望看到的那个阶段。

注意更大范围的过程

可视化仅仅是更大范围的分析链中的一个步骤而已。在整条分析链中，起始于一个问题（为什么维基百科可以工作？）或者是一个模糊的调查领域（这些维基百科的编辑是如何做到的？），然后是分析、文档记录和结果展示。一个良好的可视化会注重整个过程中的链接，对正确的信息进行编码来驱动最初的调查，并保持正确的思维角度，从而促进后期的分析以及对结果的交流。

第12章

把表转换成树：把并行集发展成意义深远的项目

Robert Kosara

学术软件项目往往会从一个初始想法有机性地发展成复杂、难以使用的项目，从而可以显得足够新颖，用于发表论文。一些特征通常是在最后一分钟才被添加，其目的仅仅是为了能够给论文“润色”，而几乎不考虑如何集成这些特征或者如何改变程序的基础架构以适应这些特征。

结果是很多程序都是被杂乱地拼凑在一起，bug很多而且坦白说看起来很让人尴尬。结果是这些软件并没有和论文一起发布，这导致产生一个最根本的可视化问题：再生性在理论上是可能的，而在实践中很少能够做到。很多程序和新技术也是从零开始开发构建，而不是基于已有的技术。

解决这种问题的最佳模式是尽可能早地发布软件，然后再不断完善和重构它，这样软件可以反映项目的全局设计目标。然而，很少有这么做的，因为重新实现（或者彻底重构）没有带来什么学术价值。相反地，人们的做法是启动下一个项目。

“并行集”（Parallel Sets）最初的原型实现（*http://eagereyes.org/parallel-sets*）和上述方式并没有什么区别，但是为了把学术思想转变成真正应用，我们需要制订一个项目规划。因此，基于经过长时间酝酿总结出的对必要的内部结构的一个更好的理解，我们开始重新思考并重新设计它。在这个过程中，我们不仅给项目增加了工程化思想，而且修改其生成的可视化来理清其基础思想。

分类数据

学术文献中描述了数以百计的可视化技术（每年增加更多），但是只有少数特定的技术使用了分类数据。这种数据只包含一些特定意义的数值（和连续的数值数据不同，数字代表本身）。例子包括经典的普查数据，如性别（男性或女性）、种族、建筑类型、使用的取暖燃料等。实际上，分类数据对很多真实世界的分析任务是至关重要的。我们最初设计该技术的目的是源于一个庞大的客户调查，该调查包含99个多选题，发给近10万的接收者。调查问卷询问人们如清洁剂和其他家用物品这样的日常消费品，以及如家庭收入、孩子个数、孩子年龄这样的人口问题等。即使在可以收集到准确的信息的情况下（如年龄），该调查也会把结果值组合成不同的分组，这些组合可以用于后期的分析。这使得可以对所有维度进行严格分类，而使用传统方法几乎无法可视化。

在这章中，我们将使用描述关于在泰坦尼克号上的人们的数据集作为例子来说明“并行集”。如表12-1所示，我们了解每个乘客的旅行舱等级（一等舱、二等舱、三等舱旅客或工作人员）、性别、年龄（成年或小孩），以及是否幸存。

表12-1：关于泰坦尼克号的数据集

维度	值
舱位等级	一等舱/二等舱/三等舱/工作人员
性别	女/男
是否幸存	是/否
年龄	小孩/成人

实际上只有3种可视化技术可以真正在分类数据上工作良好：树形图（treemap）(Shneiderman 2001)、镶嵌图（mosaic plot）（Theus 2002）和并行集。其原因是在数据的离散领域和大多数可视化变量（位置、长度等）的连续领域之间存在不匹配。当只有一些维度是连续的时，把分类数据作为数值的方式是可以接受的，但是当所有数据都是分类数据时，这种方式会变成完全无用的（见图12-1）。虽然绝大多数的数值数据集的自然分布使得收集至少和数值一样多的粗略分布是可行的，但是这种方式对于当只存在很少的不同的值完全分布在相同的数据点之间时，就完全不可能获取分布情况。

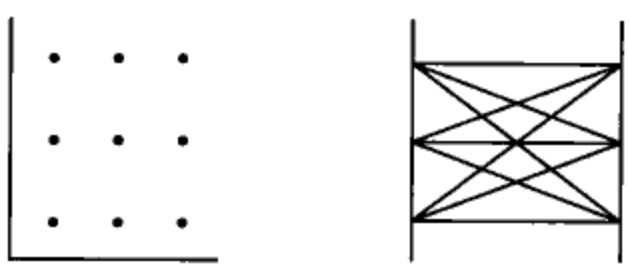


图12-1：利用经典分类数据可视化技术：散点图（左）和并行坐标（右），这两种方式带来的大量数据点重合导致即使采用一些技巧（比如，数据点抖动技术）也无法提供多少信息

并行集

“并行集”或称ParSet（Bendix 2005，Kosara 2006），它是一种可视化技术，专门用于描述分类数据。当和分析用户调查数据的专家交谈时，我们意识到他们咨询的大多数问题不是基于单个人的调查回复，而是基于这些回复的分类，或者是集合和交集。对于有3个小于5岁小孩的父母，有多少人会购买名牌洗衣粉呢？或者，换句话说，集合A中有多少人也在集合B中？泰坦尼克号上有多少一等舱乘客幸存（在“舱位等级”维度有多少对应值是“一等舱”，而在“是否幸存”维度有多少对应值是“是”）？他们当中有多少是女性（有多少人在“性别”维度是“女”）？

这种方法意味着不需要绘制数以千计的代表个人的数据点，我们只需要显示数据中存在的可能的集合和子集，以及这些集合的大小。如果这些集合的数量和相对大小总是相同，我们推测我们甚至可以证明该技术和实际数据集无关。

ParSet不是把数据显示成集合，而是深受“并行坐标”（Parallel Coordinates）（Inselberg 2009）的影响，后者是一种流行的对高维数值数据的可视化技术。平行轴布局使得对“树形图”和“马赛克图”的阅读和比较更简单，尤其是随着维度数量的增长。为这种布局设计有效的交互也更简单。

并行集的第一个版本（见图12-2）首先是基于分类，然后是基于交集。对于每个轴，我们把每个分类显示成一个盒子，其大小和每个分类所表示的数据点的比例一致。从统计学而言，这种显示方式被称为边缘分布（或边缘概率）。每个轴基本上是一个柱状图，每个柱状是倾斜的，而不是彼此相邻并竖直显示。

只看图12-2的柱形图，很容易发现工作人员是泰坦尼克号上最大的分类，三等舱人员居次。一等舱的人数比三等舱要少很多，但是实际上比二等舱的人数要多。很显然的一点是船上大部分是男人（接近80%），而整艘船上大约只有三分之一的人幸存下来。

使用色带连接一起出现的分类，例如，显示一等舱和女性这两个集合相交的概率，这样可以算出一等舱中女性乘客的比例。色带使得并行集不仅仅是一堆柱形图：它能够使用户同时看到几个轴的分布，可以允许用户识别和比较不同的模式，否则有些模式将很难被发现。

在泰坦尼克号这个案例中，在不同分类中，女性很明显地分配不均。虽然一等舱中有接近50%的女性，而二等舱和三等舱中男性的比重要远远超过女性。船员95%以上是男性。虽然色带显示很有用，它们也存在一些问题。必须对色带宽度进行排序，越宽的色带应该越先描绘，这样细的色带可以显示在上面，不会被其他色带掩盖掉。此外，当存在很多不同的分类时，往往会存在很多色带，结果导致这些色带可视化显示上很密集，人们难以阅读和与之交互。

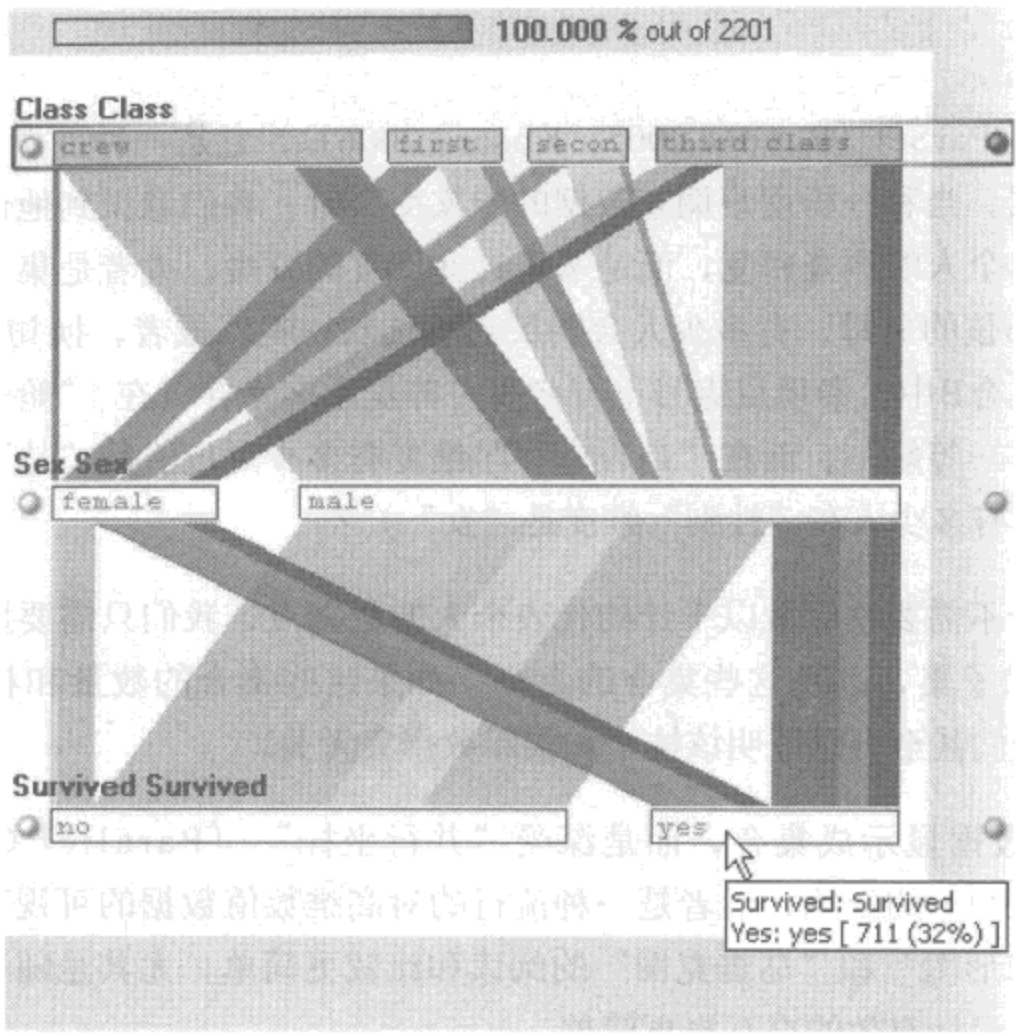


图12-2：原始的并行集设计（见彩图95）

交互是“并行集”的一个重要方面。用户可以使用鼠标显示来查看真正的数字，而且可以对分类和维度重新排序，给显示增加（或者删除）维度。还存在通过大小对轴上的分类进行排序的方法，以及把分类组合成更大的分类（举个例子，增加一个由所有的乘客组成的分类，可以更好地和船员进行比较）。

可视化重设计

并行集需要我们通过很多实验才能确定的一个方面是关于如何对一个轴到另一个轴的色带进行排序。我们想出了两种看起来很可行的排序方法，称之为“标准式”和“捆绑式”。标准式只根据上方的分类对色带进行排序，它可以形成分枝结构，但是带来的问题是当包括大量的维度和分类时，可视化显示会非常密集。捆绑式对位置在上方和下方的分类都进行分组，尽量使色带平行显示，这意味着它会对部分色带进行垂直隔离。

我们开始重新实现该技术以寻找好的可视化结构的表现方式，在进行了一段时间之后，我们才意识到自己一直在看的结构是一个树型结构（这是“标准式”的方式）。整个数据点集合是该树的根节点，而且每条轴把数据集划分成轴上的分类（见图12-3）。色带显示了树状结构；节点看起来和预期的不一致，因为我们在每条轴上收集这些节点来形成柱状图。

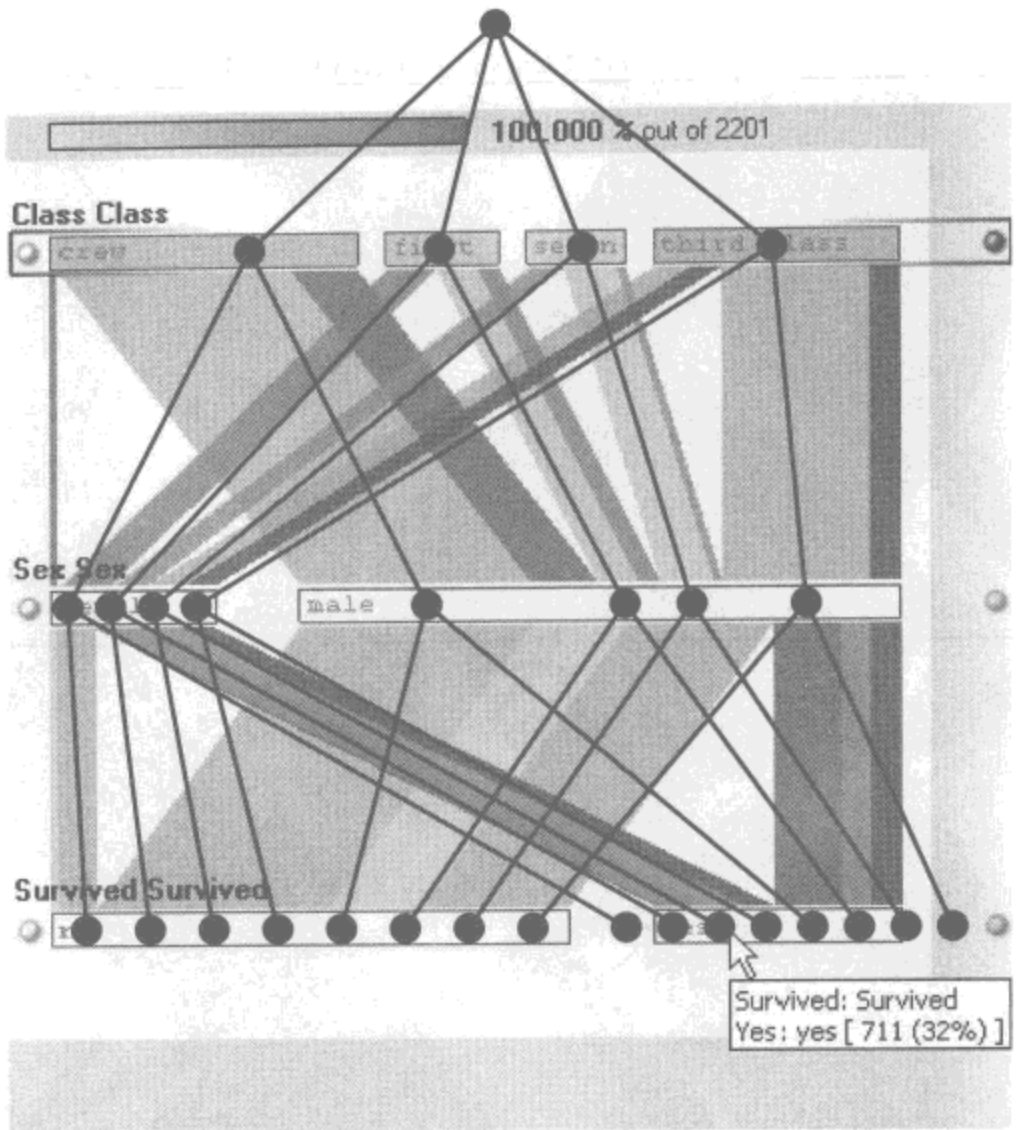


图12-3：并行集中的树状结构：每一层节点都被收集生成柱状图，色带连接不同的节点（见彩图96）

我们重新实现但没有对可视化做任何重大改变，但是树状结构的想法扎根在我的脑海里。因此，有一天我问自己：如果我们减少块状显示，主要集中于条状显示，结果会是什么效果？其结果是生成了一幅更清晰的树状结构（见图12-4）。

一种简单的变化已经把关注焦点从分类方框变换成条形树状结构。在新的设计中，当用户沿着线条点击鼠标时，方框依然会存在（提示用户可以点击交互），但是这只是个手段罢了。我们真正感兴趣的核心信息在于把分类方框划分成了多个子集。

除了增强结构清晰性，新的设计还更好地利用了字体来体现维度层次和分类标签，而且视觉效果显得更为优雅。

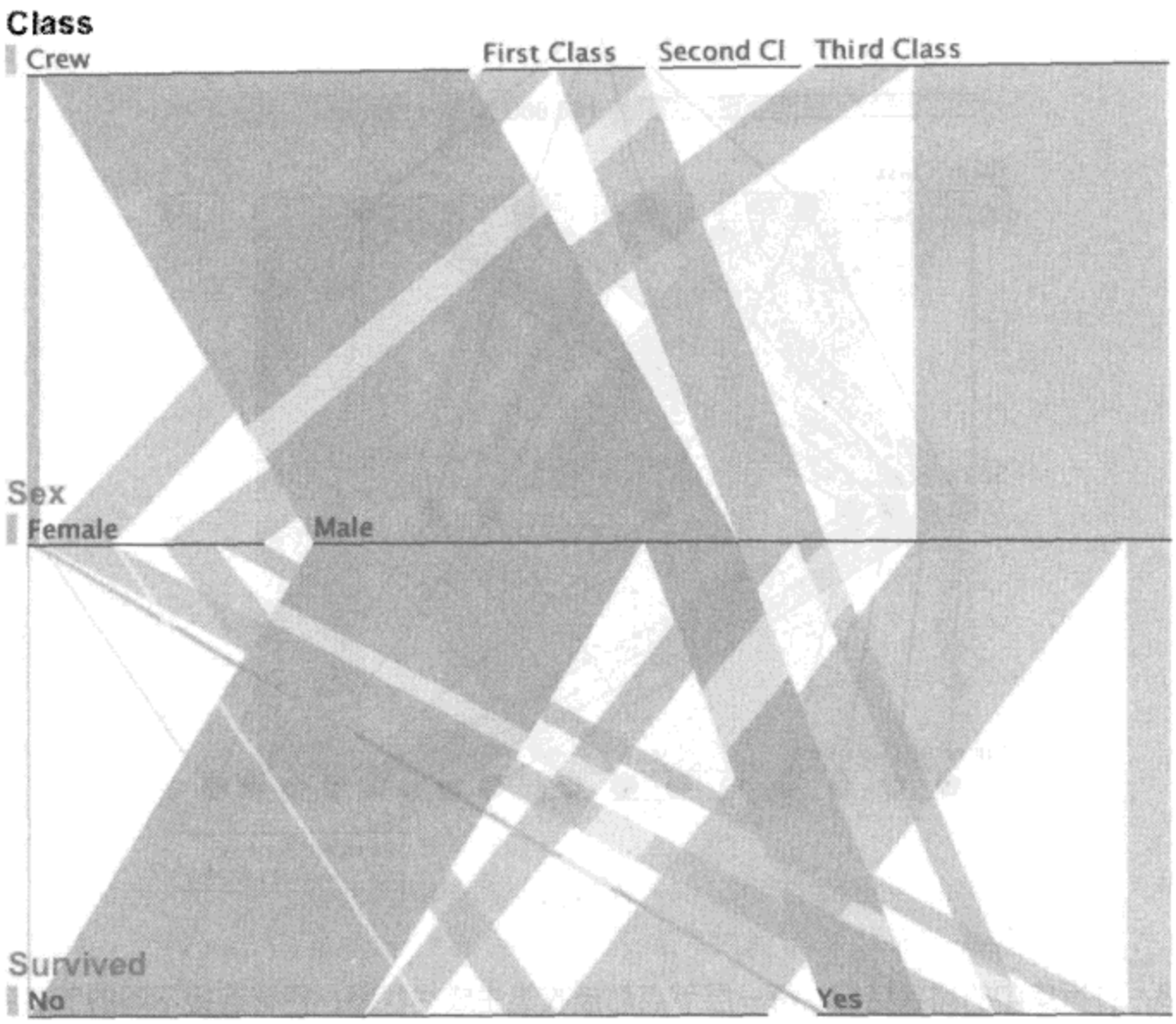


图12-4：新的并行集设计，更清晰地显示了树状结构（见彩图97）

根据聚类和集合来查看数据算不上新的想法。Polaris（Stolte、Tang和Hanrahan 2002）和Tableau[注1]是基于类似的思想构建的：对很多单个值进行聚集，并把聚集划分成不同的子集。对非层次数据的树形图的使用（这也是当前树形图被广泛使用之处）是基于相同的转换。根据数据创建一棵子集树可以促使用户使用任何层次级别的可视化来显示该数据。树形图主要专注于节点大小而不是树形结构，这是一个很自然的选择。

对最初设计的变更只需要对程序做很少的修改，但是从这点看（而且重新实现的性能也很一般），对可视化变化的认知上的需求仅仅是该程序数据模型的基本设计问题。

新的数据模型

在原始程序中，数据是以其初始的方式存储的：作为一张大表存储。后来，我们增加了给数据创建其他维度的功能，但并没有改变该原则。对于显示上的每个变化，程序处理

注1：　参见*http://www.tableausoftware.com*。

整个数据集并对分类组合进行计数。随着数据集的增大，该处理过程变得非常缓慢，需要消耗大量的内存。

根据集合来查看数据的一大优点是个别数据点确实没有什么意义，真正有意义的是数据子集。因此，下一步很自然地是要查看所有可能的数据聚集，这些聚集会被用于计算生成任何用户感兴趣的子集。

在统计学中，这种方式被称为交叉表（cross-tabulation）或透视表（pivot table）。在两个维度的情况下，其结果是生成一张结果表，其中一个维度的分类以列的形式显示，而另一个维度以行的形式显示，如图12-5所示。

船舱等级	性别				
	女		男		
一等舱	145 30.8%	44.6% 6.6%	180 10.4%	55.4% 8.2%	325 14.8%
二等舱	106 22.6%	37.2% 4.8%	179 10.4%	62.8% 8.1%	285 12.9%
三等舱	196 41.7%	27.8% 8.9%	510 29.5%	72.2% 23.2%	706 32.1%
工作人员	23 4.9%	2.6% 1.1%	862 49.8%	97.4% 39.1%	885 40.2%
	470 21.4%		1731 78.6%		2201 100%

图12-5：泰坦尼克号数据集中“船舱等级”和“性别”这两个维度的交叉表

这张表中存在两种类型的数字：计数值和百分比。在左上角，每个单元格包含不同标准组合的人数计数，右下角表示该数值所占总数的百分比。后者被称为优先级百分比（或概率）。然而，通常更有意义的是条件百分比（或称条件概率），它表示不同分类的组合。在每个单元格的右上角是给定某行，能够得出需要的列的概率（即在一等舱乘客中女性的人数）；在左下角是给定某列，能够得出需要的行的概率（即在一等舱中女性所占的百分比）。

由于数据是完全分类的，交叉表包含了所有相关的信息，而且是我们需要存储的所有信息。如果我们想要根据它重新创建原始数据，我们可以简单地通过生成尽可能多的行，每种分类组合都如给定单元格所示。唯一需要其他数据的情况是当数据集也包含数值列。

两种以上维度的交叉表有一些复杂，但是基本遵循相同的原则。需要构建和数据集一样多的维度的高维数组，数组中的每个单元格显示该值出现频率的计数值。

不幸的是，可能的组合数很快就变得非常庞大，而且实际上比在绝大多数数据集中的行数要大得多。举个例子，对于人口普查数据，只考虑（100多个维度当中的）有房或租房、建筑面积、建筑类型、建成年份、居住年份、房间数量、取暖材料、财产价值、家族/家庭类型和家族语言这几种维度就可以生成462 000 000种组合，而对于整个美国，1%的人口普查微观数据样本的数值仅为1 236 883。

这里的关键在于对于高维数据，绝大多数组合在实际数据中并不会出现。因此，只需要对那些真正存储信息的数据进行计数。这在我们当前的实现中是：简单地通过使用一组整数数组来保存每个行中的所有值，并使用该值作为散列表的键值。在绝大多数情况下，散列表所占用的空间要小于原始数据所占用的空间。

数据库模型

数据库本质上是包含每种分类组合的计数值的散列表的直接映射。每个数据集单独存储在一张表中，每个列表表示数据集中的一个维度。每行包含描述交叉表中的单元格的分类值，以及该组合出现的频率次数。还存在一个额外的域，称为关键字，该关键字对于每行都是唯一的，而且用于表连接时查看数值数据。

通过SQL查询完成对数据的聚集，该查询语句只是简单地选择用户感兴趣的维度和总的计数，对相同维度的结果进行分组（见表12-2）：

```
select class, sex, survived, sum(count) from titanic_dims
group by class, sex, survived;
```

因此，数据库对计数值进行聚集，返回只包含可视化需要的值的低维交叉表。

表12-2：查询泰坦尼克号数据集结果，只包含船舱等级、性别和是否幸存3个维度

船舱等级	性别	是否幸存	计数值
一等舱	男	是	62
一等舱	男	否	118
一等舱	女	是	141
一等舱	女	否	4
二等舱	男	是	25
二等舱	男	否	154
二等舱	女	是	93
二等舱	女	否	13
三等舱	男	是	88
三等舱	男	否	422
三等舱	女	是	90

表12-2：查询泰坦尼克号数据集结果，只包含船舱等级、性别和是否幸存3个维度（续）

船舱等级	性别	是否幸存	计数值
三等舱	女	否	106
工作人员	男	是	192
工作人员	男	否	670
工作人员	女	是	20
工作人员	女	否	3

该模型在原理上和数据仓库和联机分析处理（OLAP）非常相似。绝大多数数据库包含特定的“切片”（cube）或“上钻”（rollup）关键字，可以从一张普通的表中创建聚集。它的优点在于不需要前置特殊的处理，但是其缺点在于处理速度更慢，而且需要更多的磁盘空间来存储所有的原始值。为了加快读取速度和聚集性能，对数据进行特殊地结构化处理（如在数据仓库和数据库模式中的）可以显著提高普通操作的性能，其代价是当需要增加新的数据时需要做更多的处理。

虽然ParSets应用程序当前并不显示数值维度，它确实把维度信息存储在数据库中。这些维度信息是存储在一张单独的表中，该表包含这些值对应的行的键值，每个列显示一个数值维度。不是使用计数操作，而是使用简单的连接查询，对交叉表中的各个单元格中的任何数值维度进行聚合操作。任何标准的SQL聚合操作（sum、avg、min和max函数）可以用于这个目的。因此，该程序可以允许用户选择一个数值维度，用于对条状显示和色带进行扩展，而且可以选择已使用的聚合操作。

当前版本的并行集把数据存储在本地的SQLite数据库中。SQLite是非常有趣的开源数据库，它在一张表上执行操作。它可以应用于很多嵌入式应用中，而且对于数据损坏有很强的容错性（这些设备在任何时候都可能宕掉）。然而SQLite数据库不包含商业数据库的所有特征，它很小、很快而且不需要任何步骤。这使得SQLite数据库成为最佳的数据存储方案，其额外优势是查询语言规范。

树结构增长

然而，数据库存储以及可以被检索的交叉表只是其中一部分。为了向用户显示并行集展示，我们需要用一棵树来表示。每当用户改变维度或者重新对它们进行排序，应用程序就会查询数据库，检索到新的交叉表。然后，应用程序会遍历所有的结果数据来构建树。如果仔细查看，在表12-2中实际上已经可以看到这些信息。每当在同一列中多次出现相同的值，我们查看到的是这棵树中相同的节点，而只有树的右节点会变化，如表12-3所示。

表12-3：在表12-2的查询结果中内在的树结构

船舱等级	性别	是否幸存	计数值
一等舱	男	是	62
		否	118
	女	是	141
		否	4
二等舱	男	是	25
		否	154
	女	是	93
		否	13
三等舱	男	是	88
		否	422
	女	是	90
		否	106
工作人员	男	是	192
		否	670
	女	是	20
		否	3

程序所需要做的是一行一行遍历结果集，根据已有节点从左到右构建树，直到遇到不存在的节点。在树中增加该节点，并从数据库的记录中获取其计数值。

然而，数据库只包含树的叶子计数，而不是其内部的节点（其他数据库如Oracle，当执行切片查询（cube query）时，也返回内部节点）。但是，计算节点计数很简单，只需要从叶子节点到根节点，递归地对每个孩子节点的值进行求和。

计数值本身也只是原始分数值，一旦一个节点的所有计数值已知，就在同一个步骤中对所有分数值进行计数。为了准确地显示条状色带，我们使用百分比：每个分类的一个先验百分比（a priori percentage）是色带的中心，用它作为整个色带宽度的分数，而使用条件百分比（根据上一个分类在色带上显示下一个分类）来确定色带的宽度，作为分类条状宽度的分数。

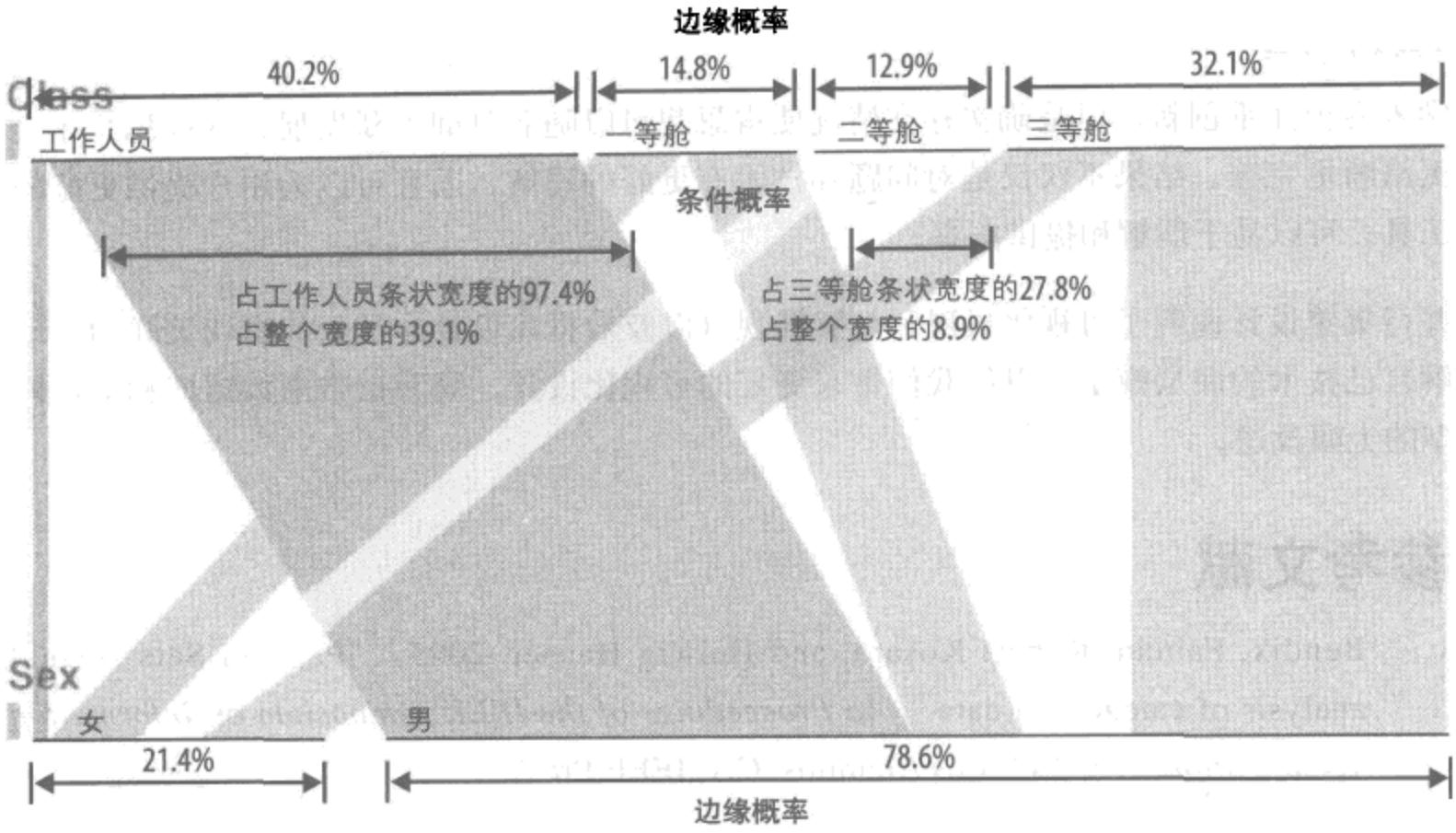


图12-6 每条色带的宽度表示其在所有数据集中的边缘概率（成比例分数），以及在每个分类的条件概率（见彩图98）

现实世界中的并行集

从2009年6月发布该应用程序后，它已经被下载了750多次（截止到2010年1月）。我们从很多用户那里收到来信，他们成功地把该应用程序用于自己的数据中。在2010年的VisWeek 2010发现展览会上，我们还因为对使用该程序做的3个案例研究报告而获得了一个奖章（*http://discoveryexhibition.org*）。3个案例是和另外3个人一起实现的：Joe Mako (Mako Metrics)，Jonathan Miles (Gloucestershire City Council, 英国)和Kam Tin Seong (新加坡管理大学)。

Joe Mako对该程序的应用方式尤其有意思，因为他使用该程序来显示看起来像多个处理阶段中的数据流。把最后一个阶段放在最上面意味着该色带是用最后一种结果进行着色，这使得他可以很容易查看哪里出现问题。实际上存在一种可视化技术，其在视觉上（虽然不是在概念上）和用于流的并行集相似，称为Sankey流程图。并行集可以模拟严格按照一个方向流动以及只有分割（没有合并）的流程图。Jonathan Miles和Kam Tin Seong对程序的使用和该程序本身的初始目的更接近，即提供有趣的洞察分别生成调查结果和支持客户。

结束语

学术界很注重创新，但是确实存在情况使得思想可以随着时间不断发展，这样思想才会更清晰更完善。结果不仅仅是对问题和技术有更好的理解，而且可以为用户提供更好的工具，可以易于理解和提供洞察。

并行集重设计说明了可视化展现和数据展现（以及数据库设计）是如何密切关联的。理解自己技术的底层模型可以给我们带来更好的可视化设计，同时也带来数据库和编程模型的大幅改进。

参考文献

1. Bendix, Fabian, Robert Kosara, and Helwig Hauser. 2005. "Parallel Sets: Visual analysis of categorical data." In *Proceedings of the IEEE Symposium on Information Visualization*, 133–140. Los Alamitos, CA: IEEE Press.

2. Inselberg, Alfred. 2009. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. New York: Springer.

3. Kosara, Robert, Fabian Bendix, and Helwig Hauser. 2006. "Parallel Sets: Interactive exploration and visual analysis of categorical data." *IEEE Transactions on Visualization and Computer Graphics* 12, no. 4: 558–568.

4. Shneiderman, Ben, and Martin Wattenberg. 2001. "Ordered treemap layouts." In *Proceedings of the IEEE Symposium on Information Visualization*, 73–78. Los Alamitos, CA: IEEE Press.

5. Stolte, Chris, Diane Tang, and Pat Hanrahan. 2002. "Polaris: A system for query, analysis, and visualization of multidimensional relational databases." *IEEE Transactions on Visualization and Computer Graphics 8*, no. 1: 52–65.

6. Theus, Martin. 2002. "Interactive data visualization using Mondrian." Journal of *Statistical Software* 7, no. 11: 1–9. *http://www.theusrus.de/Mondrian/*.

第13章

“X by Y”的设计：奥地利电子艺术节档案的信息美学探索

Moritz Stefaner

本章将介绍“X by Y”项目，它是一个囊括了奥地利电子艺术奖从1987年到2009年间所有获奖作品的可视化，奥地利电子艺术奖是一个著名的媒体艺术大奖。这个可视化的最终版由一系列大型印刷品组成，提交的作品根据不同的标准被划分为了多个类别。本章描述了完成这个最终作品的完整过程，并介绍了一些特定的设计决定的缘由。

简介和概念

Ludwig Boltzmann研究所的media.art.research网站和我在2009年春签约，工作内容是关于电子艺术大奖的参赛作品数据库。那一年是奥地利电子媒体艺术成立30周年，我们双方一起决定接受挑战，试着对过去22年以来提交给该奖项的所有作品进行可视化分析。在此之前，从未在整体上对包含这些提交信息的数据库进行分析。

在该项目的启动大会上，我们对目标进行了讨论。整个可视化项目的总负责人Dietmar Offenhuber富于创新，他提出需要开发不同的可视化，故而可以从3个不同的角度来审视艺术节。

定量分析

> 我们是否可以通过查看过去几年的提交作品来审视艺术节？不同的分类之间有何不同、这些作品来自哪里以及作品的价值随着时间的推移是如何变化的？

社交网络

在过去那些年，评委团成员都是谁？他们以及获奖人是如何互相联系的？

艺术历史背景

获奖项目有哪些影响？它们在哪里被引用以及它们在媒体艺术领域产生了什么样的影响？

下文将要介绍的我所做的项目属于第一类。具体地说，我将查看提交的数据来调查确定我们能够做出哪些假设、得出哪些见解，以及我们是否能够发现合适的可视化方法将“艺术世界”的特征展示给展览的参观者。

我和那些致力于分析电子档案的艺术历史学家们一起尝试定义了我们的首要兴趣，如图13-1所示。不需要详细查看数据库，假定我们能够处理一些基础维度，如作品的作者、作者的国籍、参赛年份、奖项类别、关键词以及是否获奖。该矩阵显示了这些因素的特定组合的先验兴趣，比如专家会预期有趣的发现将在哪里出现。举个例子，假定我们能够通过国籍对获奖者进行划分（并把结果数据和全局提交作品统计进行比较），然后就可以查看作者和分类之间的关联。

	作者	国籍	年份	奖项类别	关键字	获奖者?
作者			×	×	×	×
国籍			×	×	×	×
年份					×	
奖项类别					×	
关键字						×
获奖者?						

图13-1：初始兴趣在属性组合上的分布矩阵

了解数据形势

接下来，我开始和Sandor Herramhof一起寻找可用的数据。多年以来，人们使用了数据库模式，这些模式没有遵循统一的规范，对细节的描述相互之间也有很大的不同，这使得对已有的数据进行概览变得很困难。举个例子，有这样一个数据库，其特点是将备注信息以XML格式存储在一个文本域内，但只是部分提交的作品包含这种信息。为了简

化对数据总体状况的获取过程，我开发了数据可视化统计工具dbcounter[注1]，它很小、采用nodebox[译注1]的展现方式，能够帮助我们快速获取分类数据的大量集合的总体概况。*dbcounter*通过读取CSV文件，确定所有具有唯一值的属性，统计这些属性的出现频率，并把输出结果描绘成一张区域图。灰色区域（见图13-2）表示值被丢失或值为空。总体而言，实践证明该工具对于理解数据库内容是很有用的，尤其有助于发现缺失值和理解数据的多样性。

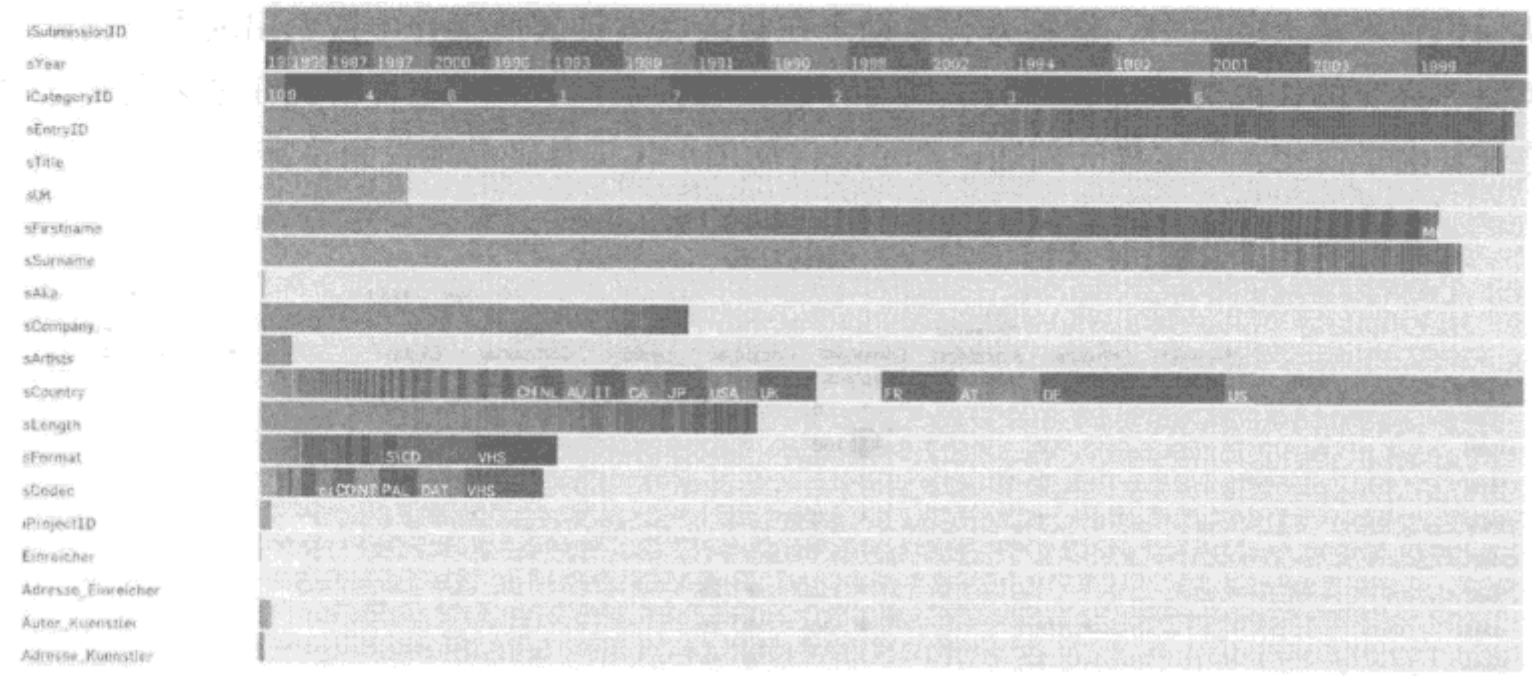


图13-2：通过dbcounter工具对数据库内容的首次概览，dbcounter是一款定制的nodebox脚本（见彩图99）

有了这些绘图，数据库中蕴含的一些事实信息很快就变得很清晰：

- 数据库中存在很多明显冗余的域，如“Land”（德语，表示“国家”）和“sYear”，这是由过去几年数据库模式的合并造成的；
- 作者名字、参赛年份和奖项类别信息很完整；
- 包含的国家、公司和Web地址信息远小于预期。

另一方面，这种快速地初步分析使我们能够了解哪些属性组合可能是有意义的，至少可以涵盖大部分有意义的属性组合数据。由于数据迁移是一个持续的过程，它还为我们提供在某些区域的有用的概览，通过探索这些区域，我们可以改进数据、寻找哪些域可以合并在一起或可以进一步补充得更完整。举个例子，我们这个团队在包含有国家域的数据库上所开展的工作实际上是尽可能地充实更多的信息（“看起来是真正有趣的信息，而我们已经与这些信息非常接近”）。

注1：*参考http://well-formed-data.net/archives/306/dbcounter-quick-visual-database-stats。*

译注1：NodeBox是用Python实现的开源的二维动画和图形应用，详见*http://nodebox.net/code/index.php/Home*。

探索数据

在对个体的各种属性的初步定量分析之后，下一步是对初始的数据集进行切片和切块，从而调研关联关系并为数据中出现的空白寻找一些线索。在这一步，我们使用商业软件Tableau[注2]，它允许我们在一个使用灵活且表达能力很强的工作区中使用可以交互的表格对导入的表格数据和数据库中的数据进行探索分析。举个例子，我们使用Tableau，对缺乏国籍信息的提交作品通过作品的参赛年份和奖项类别等属性进行区分（见图13-3），从而识别出最大的空白，这种方式有助于在目录文本等其他媒介中搜索出缺失的信息。类似“提交的作品的数量和分类之间有什么关联关系？”和“这种情况在过去几年之中是否发生了变化？”这样的问题，都可以借助图形化工具轻易地找到答案。

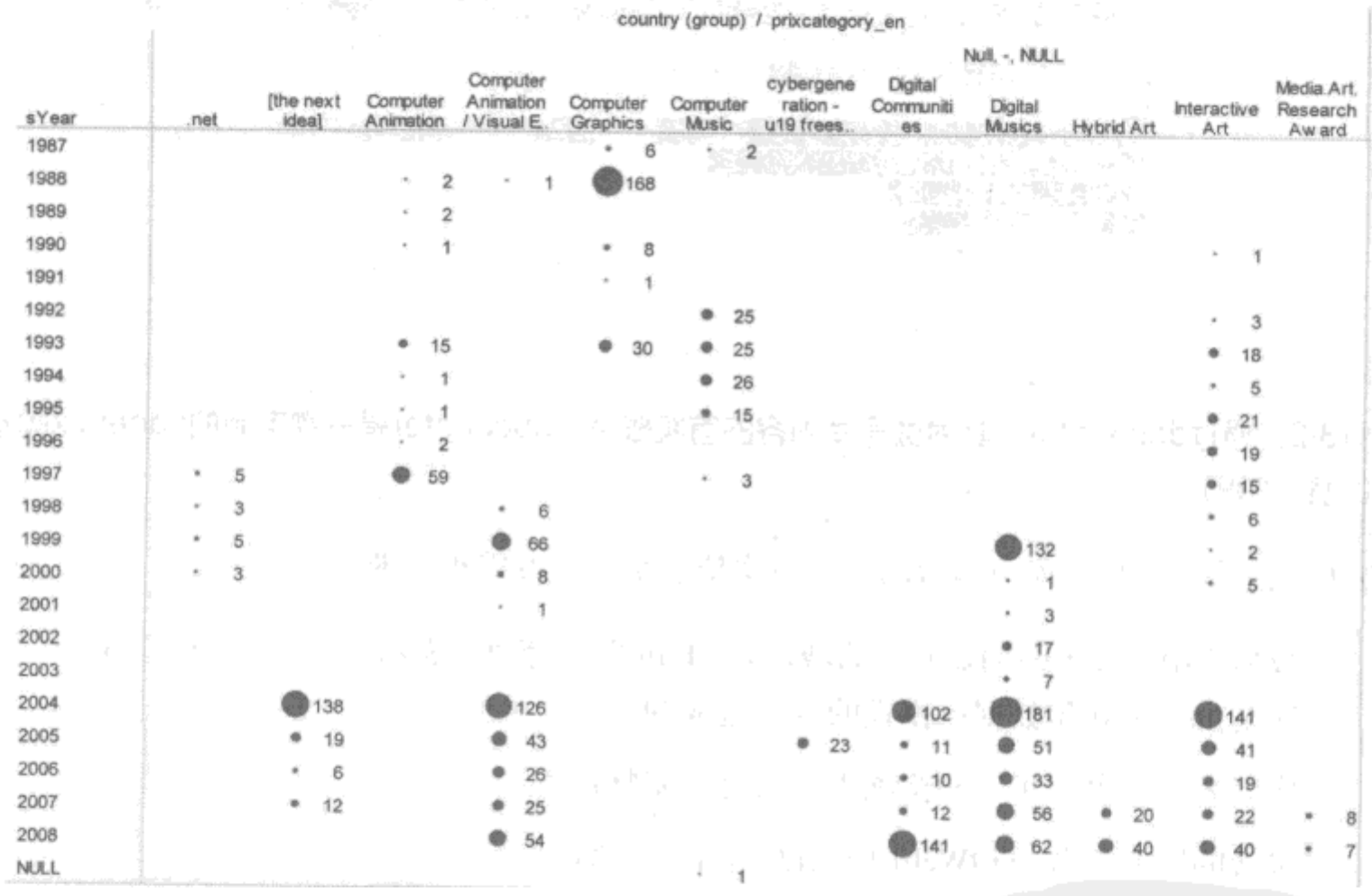


图13-3：对缺乏国籍信息的提交作品生成的绘图，通过作品年份和作品类别进行划分

其他探索包括根据提交作品的条目所属的类别对其公司进行特征化。例如，图13-4的图表揭示了一些潜藏的有趣的故事。然而，接下来很快就会发现，如果我们想要得出准确的结论，对不同数据库中公司名字的统一需要花费大量的人工操作。

注2：参考*http://www.tableausoftware.com*。

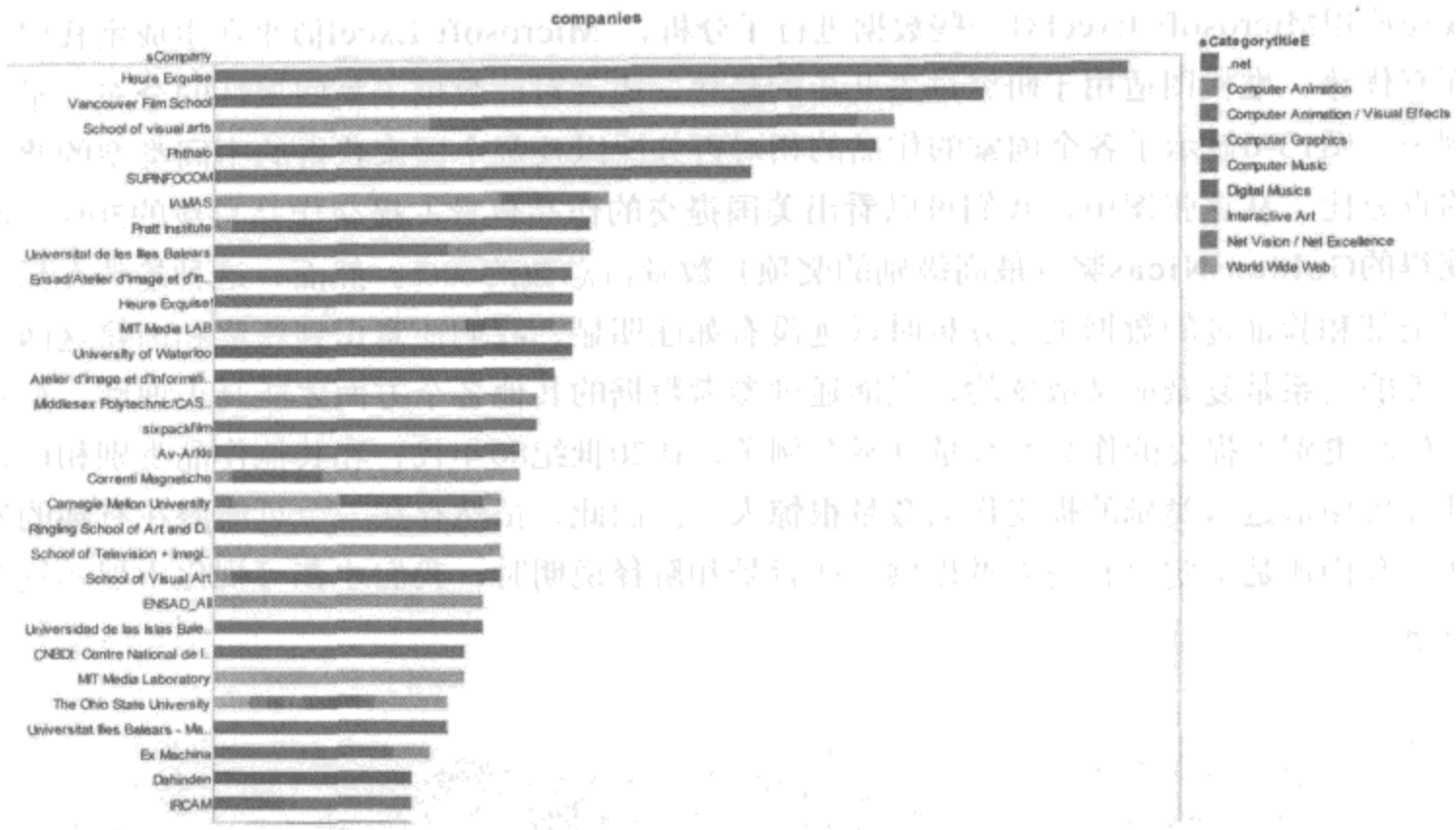


图13-4：按公司或研究所进行分类的提交作品，对不同分类进行着色（见彩图100）

我们还使用Tableau[译注2]工具生成了一张初始的关于提交作品的世界地图（见图13-5），每个国家通过一张饼图表示，它可以说明不同类型的作品的分布情况。这张世界地图展示了艺术节在本质上是以欧洲/美国为中心。我们很快意识到这种简单的生成统计图的方法对于这类分布不均匀的数据是很低效率的，因此，后面我们将会介绍更详细复杂的方法。

图13-5：包含每个国家的提交作品的世界地图，按作品类别进行划分（见彩图101）

译注2： Tableau是一款免费的数据可视化软件，详见*http://www.tableausoftware.com/*。

我还使用Microsoft Excel对一些数据进行了分析， Microsoft Excel似乎在生成堆栈图方面有优势，堆栈图适用于研究过去几年的趋势，或者对比数据子集间属性的分布。举个例子，图13-6显示了各个国家的作品的相对百分比以及每个国家获得的不同类型的奖项的百分比。从这张图中，我们可以看出美国提交的作品数量占提交作品总数的30%，而获得的Golden Nicas奖（最高级别的奖项）数量占总数的60%。然而，这种趋势在后期对全部和验证过的数据进行分析时远远没有如此明显。我们还意识到获奖和国籍这两个属性的关系是复杂而又敏感的，只能通过参考数据的其他各个方面才能真正理解，如每个作品类别下提交的作品的数量（举个例子，在20世纪80年代，和其他作品类别相比，计算机图形这一类别的提交作品数量很惊人）。因此，虽然存在一个可能潜在有趣的发现，我们还是决定只有当能够提供一些背景和解释说明时，我们才在可视化中展示这个故事。

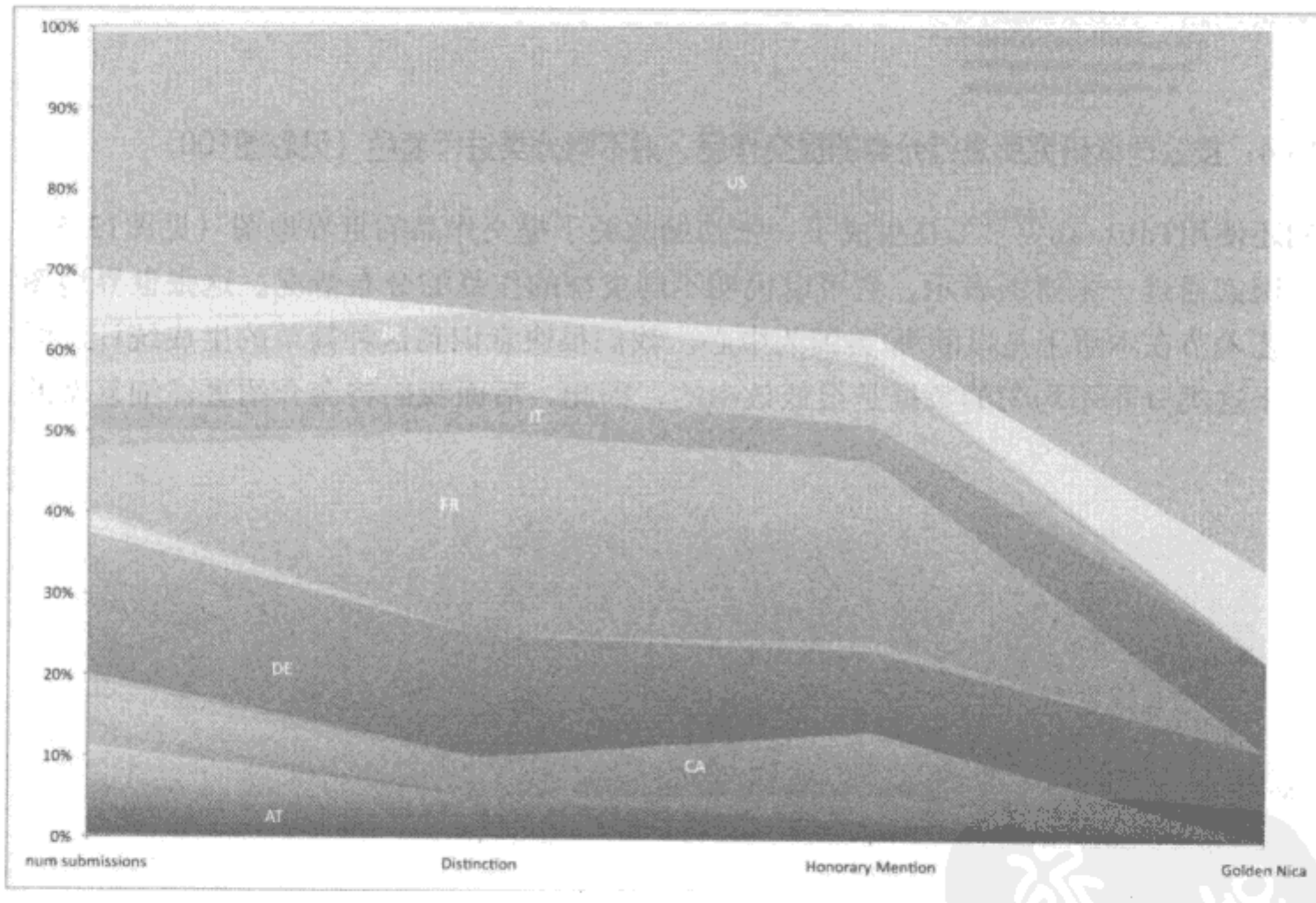


图13-6：不同国家的获奖情况（见彩图102）

初次可视化草图

分析过程中给数据增添了一些初始的思考，而且为我的合作者提供足够的机遇——可能超出他们的期望——对数据库的数据进行纠正、清洗和补充完备。在此基础上，借用Tom Armitage 的BERG博客上的帖子“在海量数据中埋头苦干：对数据探索的真正感

觉”[注3]，我对哪些数据是可用的、有意义的、有趣的，以及数据的规模有很好的把握。接下来要做的是致力于可视化原则。

为了对一些不同的可视化选择进行原型化，我改成使用Flash ActionScript 3中的flare库[注4]，它是适用于生成交互可视化的一个高级的通用框架。此外，我还使用Excel表格分析更多的堆栈图表选项。从这些图表中，我得到的其中一个收获是我们应该更多地强调独立的数据点（比如图13-7中的垂直轴上的各个年份），而不是生成连续的堆栈区域图。在电子艺术这个案例中，提交的作品仅仅是基于手工基础，因此不同年份间的可视化“插值”方式会造成对现实情况的误导和扭曲。

基于上述考虑，我们开发了看起来更“纤细”（fragile）的图表，通过降低插值区域来支持以下观点：

插值区域只是作为更“坚实”（solid）的每个年度事件之间的连接。

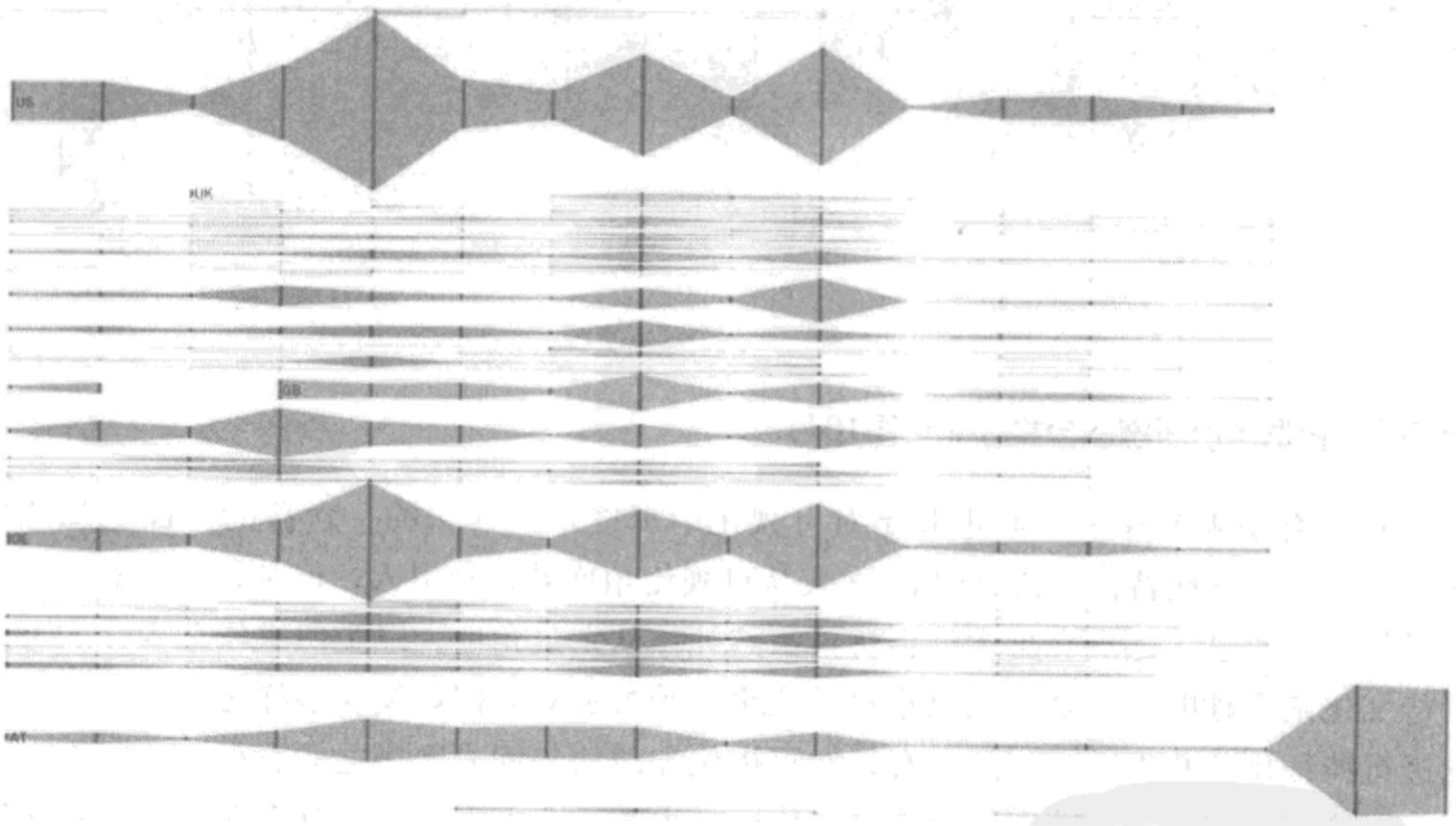


图13-7：按国籍来显示不同分类的初次尝试（见彩图103）

多年来对分类堆栈区域图的探索从概念上提出了需要解决的一些额外的问题。奥地利电子艺术节的分类结构在过去多年来不断地演变。举个例子，在1991年没有设置“计算机音乐”这个类别，而在其之前和之后都有这个类别。在1991年，删去了“计算机分类”

注3：参考*http://berglondon.com/blog/2009/10/23/toiling-in-the-data-mines-what-data-exploration-feels-like/*。

注4：参考*http://flare.prefuse.org*。

这个类别，增加了新的类别“数字音乐”。如何最佳地处理这些情况是一个很棘手的问题：一方面，存在明确相关的分类，而另一方面，对这些分类进行统一并通过不同的标签把它们作为相同的分类，这种方法可能过于简单。类似这种决定，需要考虑专家的意见和设计师的观点来制订准确、实用、易于理解的方法。我们讨论后，决定把这些类别作为独立的分类，但是在不同的可视化中对它们使用相同的绘图颜色的方法来解决这个问题（见图13-8）。

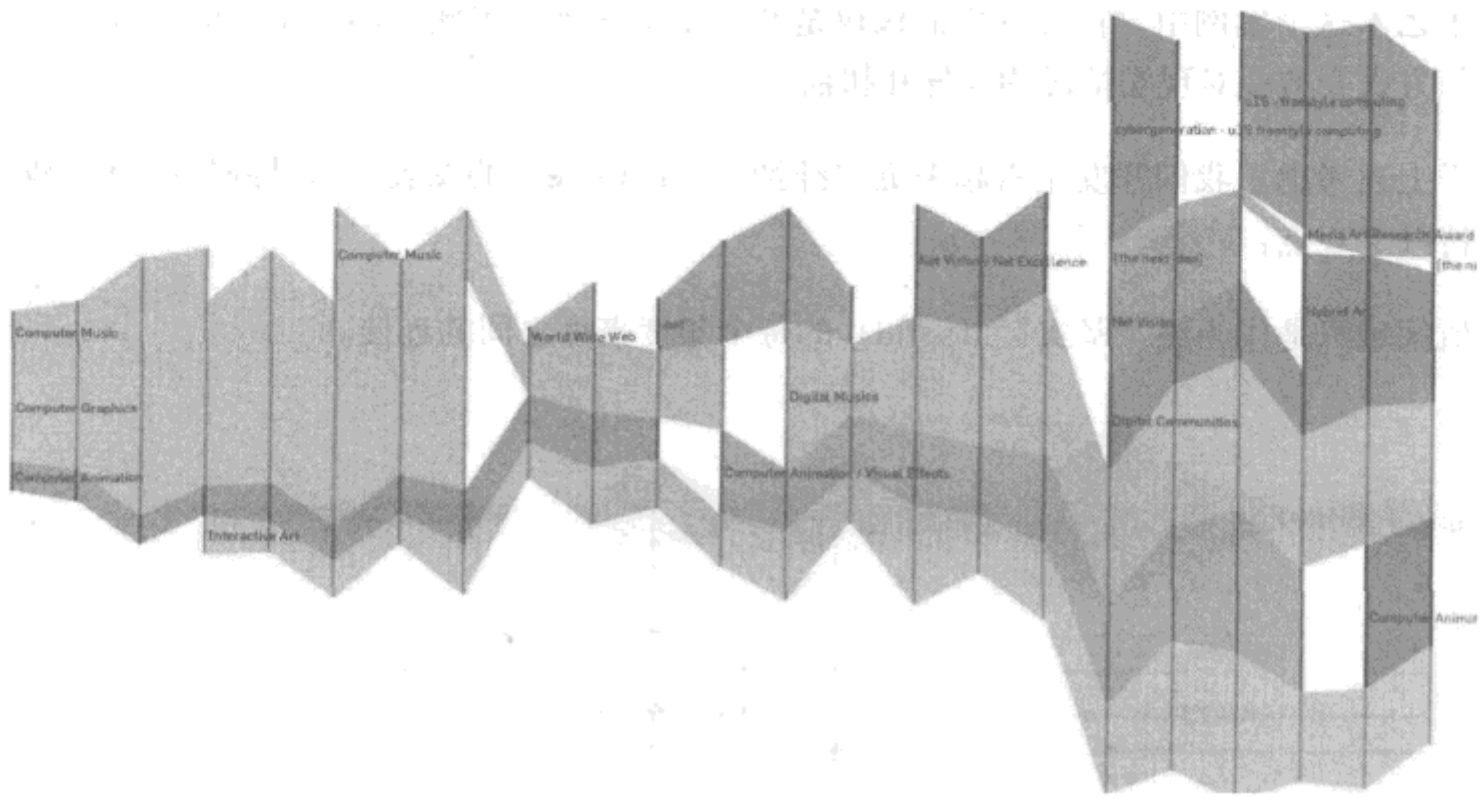


图13-8：根据年份所做的分类（见彩图104）

随着对已有图表的探索，我也开始对可视化中那些令人回味的、含蓄的方面更加感兴趣。我不喜欢某些特征，比如Flare图表从可视化角度看很吸引人，但是显示上有点过于“纤细”。不过，还存在更大的担忧：虽然纯粹从定量角度，以类似媒体艺术奖的方式看待文化是有趣的，但是我们感觉自己似乎开始失去对数据规模和多样化的认识，而对它进行特征化的角度过于宽泛。有效的可视化和总结与优先次序之间有很强的联系；然而，只是创建非常抽象的图表对于可视化这一主题没有太大意义。难点在于是否存在一种方式，可以显示作品总数、部分以及它们之间的相互关系，同时还可以不忽略甚至不隐瞒某些个别作品。

可视化原则

这种动机驱使我首先去探索密集像素的“马赛克”显示方式（Keim 2000），其想法是我希望看到每件作品的可视化标识。为了了解一个标准屏幕上适合多少数据点，我使用随机数据做了一些快速测试，如图13-9所示。

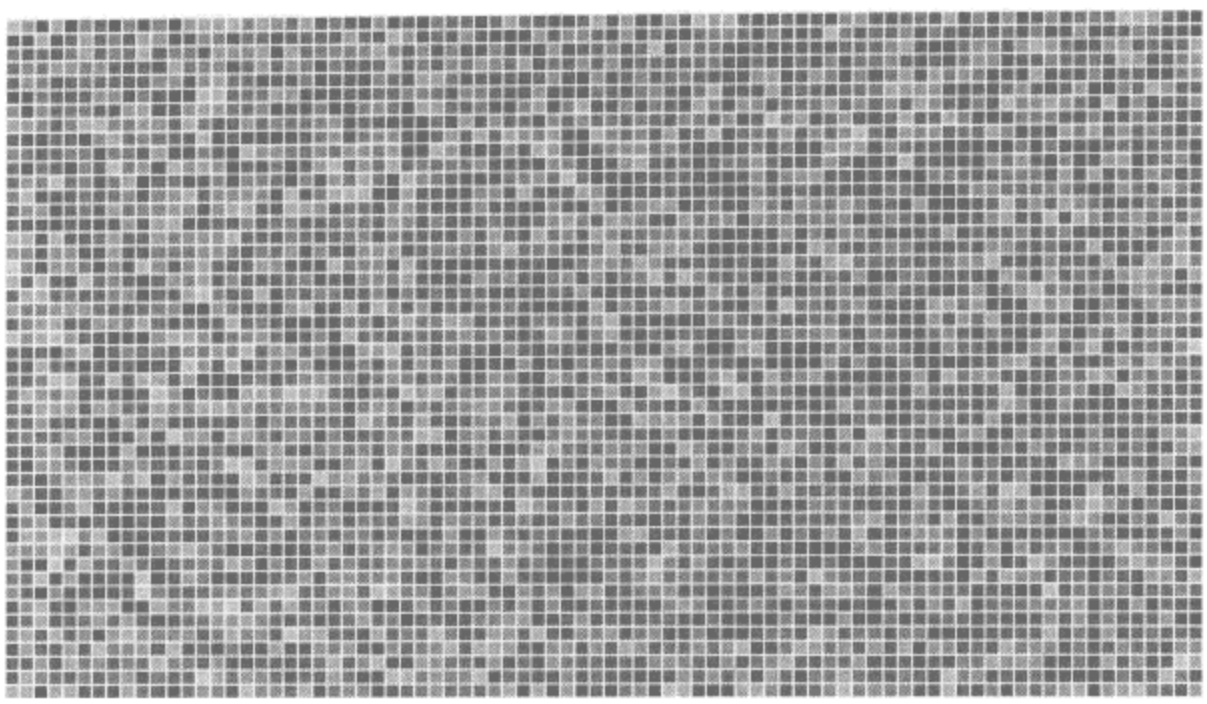


图13-9：对密集像素显示所做的实验（见彩图105）

我发现实验结果令人鼓舞，决定通过查看QR码[注5]来做进一步地调查。我们是否能够使用有意义的URL来构建真正的QR码，使得它在基于面积或像素的数据图上也能够正常显示？另一个想法是根据Wattenberg（2005年）的空间填充曲线的彩色分段来生成类似于树形图（也称“拼图”）的流行图。

然而，真正的重要时刻是当我想起在早期项目中使用过的布局算法。基于黄金角（一个完整的圆的“黄金分割”角度，即137.5°）的基础计算，它模仿了向日葵种子的排列方式，即把小的元素打包成大的圆圈的最高效、最优雅的方式。图13-10显示了我在几个小时内做出的第一个尝试，黑白交替变化表示年份（和树墩截面的年轮类似），省略点表示获奖的提交作品。

虽然可视化很复杂，创建这些类型排列的基本过程可以使用简单的规则来描述：对于第*n*个点的放置，选择*n*的平方根乘以某个常数比例系数作为半径。该点所在的角度即其前一个点的角度加上黄金夹角（2*pi/phi = 约137.5°）。

为了把这些点统一、均匀分布，准确使用以下数值是很重要的：假如我们使用137.4°，特征化的双螺旋将会被只有一个方向的单螺旋代替，点之间的距离将会开始变化。使用黄金夹角，我们可以无限制地增加点，而且每个点和其邻居节点的距离将会均匀分布。为什么会这样呢？我们选择的分割圆圈的每个有理数数值迟早会生成重复的角。在最简单的情况下，如果我们总是转半个圈，结果会是只有两个不同的角。对于任何有理分数，都存在重复，因此只能使用有限的角度集合。相应地，如果我们想对数据点的填充和分布进行优化，我们需要使用无理数——理想情况下是使用最大的无理数（即至少和一个分数接近）。该数值即phi，它表示黄金分割。

注5：　参考*http://en.wikipedia.org/wiki/QR_Code*。

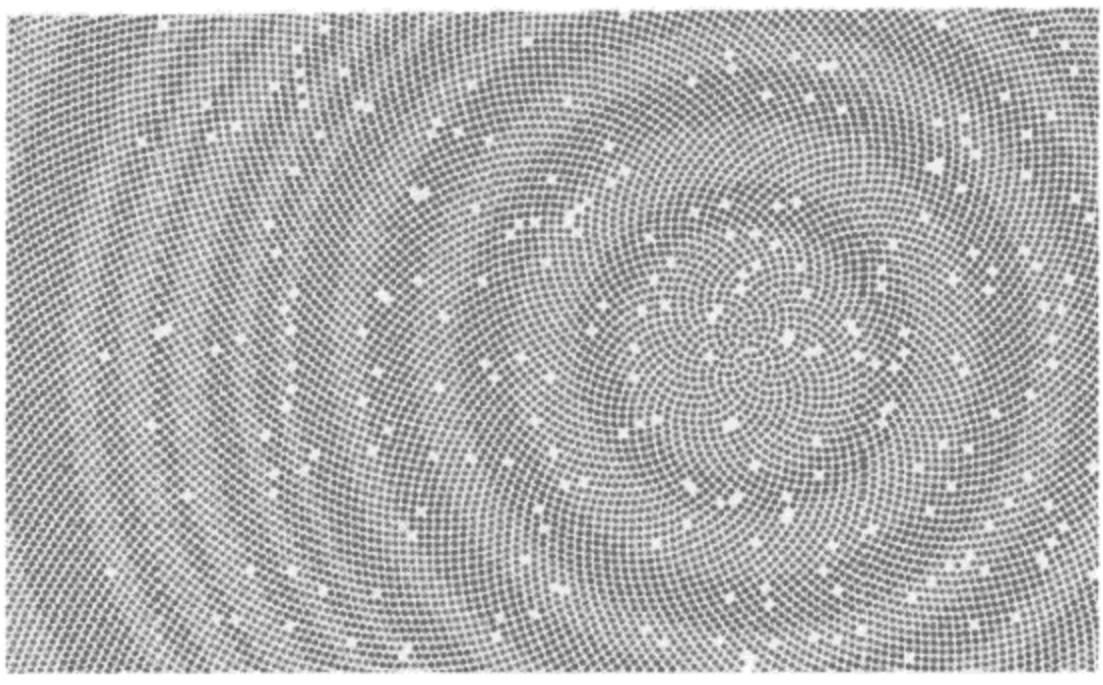


图13-10：每个提交作品对应一个点，像向日葵种子一样打包在一起（见彩图106）

最终产品

发现了可视化的指导原则后，很多开放问题和可能组合现在都自然地简化为可以在自我约束条件范围内正常工作。举个例子，该原则确定所有项分组的圆形形状。由于分类分布对于我们所讨论的所有方面都很重要，我们决定对显示的所有可视化的分类分布进行着色编码，对于可以合理作为同一族分类的所有分类用同一种颜色来表示（举个例子，在计算机动画和电影领域的分类都是显示成橘黄色。）此外，我还采用形状编码方式来表示某个提交作品是否获奖（圆圈表示没有获奖，钻石型表示获奖）。

正如前面所讨论的，在概念层次上，我开始对作品总数和某个人提交的作品数之间的关系感兴趣。因此，我需要找到一种方式，可以把该信息结合到最终的可视化中。我做了一些不成功的实验，在圆周围放置一些额外的标签来表示总数并把总的计数值放在圆上方，这种方式会导致显示上非常混乱。在这些尝试之后，我发现了一种更令人满意的解决方式：数字实际可以通过点模式本身来创建！对分类进行彩色编码的决定排除了其他所有对点本身的修改，我决定跳过序列中用于表示该数字的位置的点，如果它在圆上被其他数字的位置所覆盖（见图13-11）。该点将被置于下一个预计算的位置上，因此全部点的数目将还保持不变，但是圆的面积大小将会有些增加。显然，该原则只适用于那些包含足够的点来创建数值的情况，因此，该圆至少需要包含100项才能显示数值。

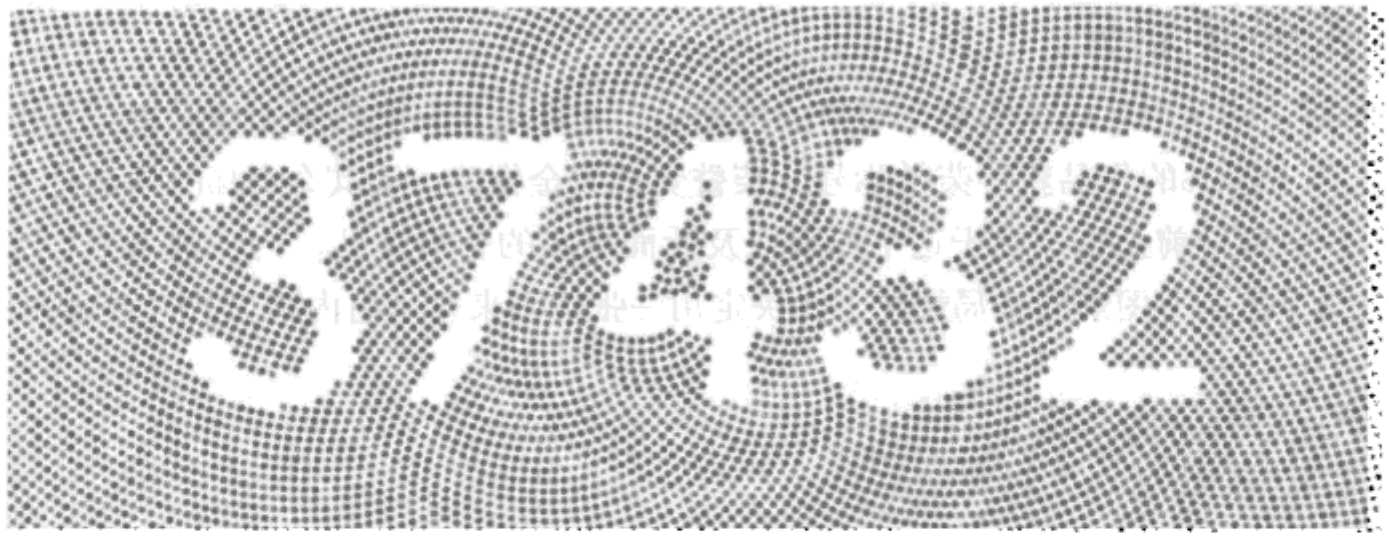


图13-11：通过在放置序列中跳过重复点的方式来生成数值（见彩图107）

所有的提交作品

图13-12显示了在过去22年所有提交到奥地利艺术节的作品。它看起来像一棵树的剖面，最早提交的作品被置于最中心，其他作品按时间先后顺序置于该作品周围形成圆。这种组成方式是生成所有其他图形的基础，每种图形都是该图形的一种划分，其包含的数据是根据不同的标准分析得到的。

图13-12：所有的37 432个提交作品，按分类进行着色，通过作品的提交年份从内（最老的）向外（最新的）排列（见彩图108）

按是否获奖划分

图13-13所示的图形足以作为整个项目的推动力：根据是否获奖对提交作品进行划分，结果说明了只有4%的作品获得荣誉称号、荣誉奖章或金像奖。而其余96%的作品是不对外公开的——到目前为止。由于这个原因以及后面更多的分析意见，为了避免中心圆圈在视觉感受上扭曲了图形的全局视图，我决定用一张饼图来显示组内数据的分类分布。

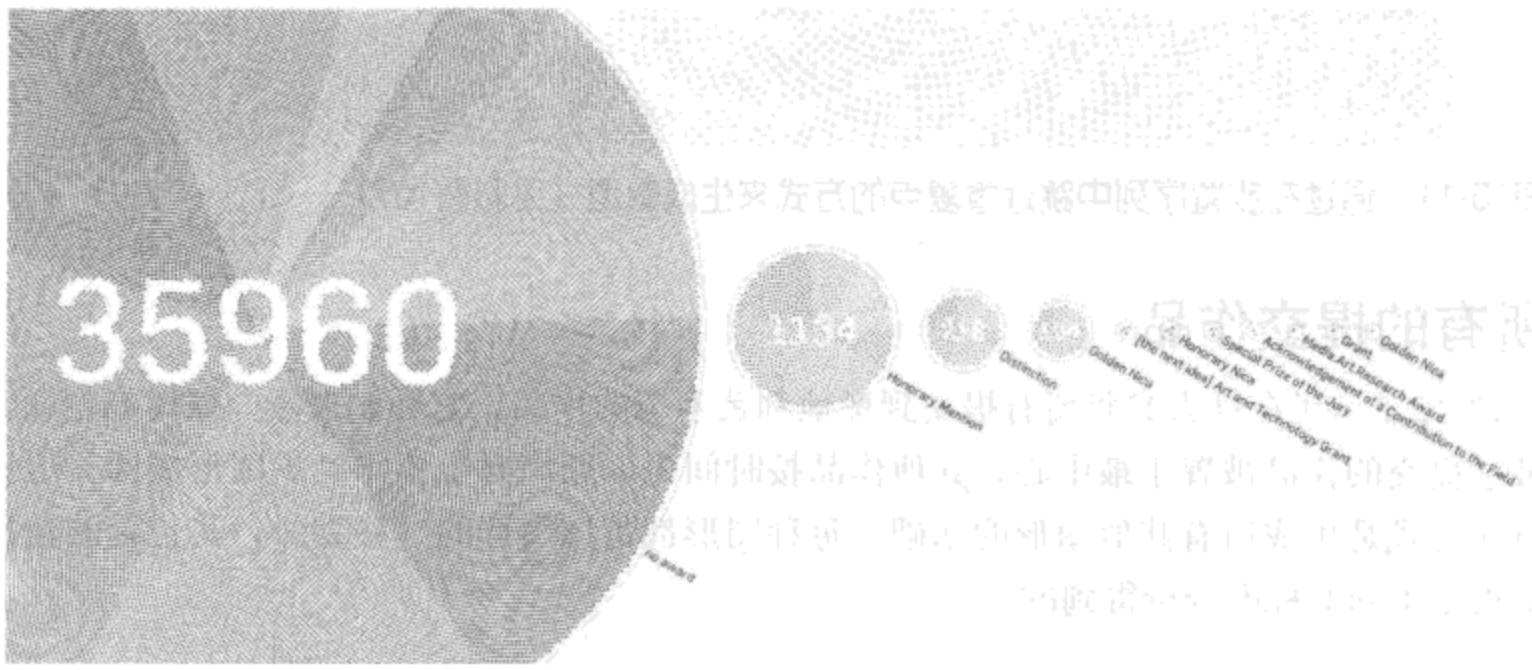


图13-13：按获奖分类的提交作品（见彩图109）

按作品类别划分

图13-14显示了按作品类别划分的对所有作品的定量分析。同时，它在饼图的浅色区域显示了每个分类的获奖项，在每个圆的右侧由钻石形状组成。比如，它说明了计算机图形分类的提交作品数量最高（按分类），而按提交作品数来看，其获奖数很低（由于该分类只有7年的历史）。根据Wang等（2006年）的论文，圆圈的布局是使用Flare工具的CirclePackingLayout算法来计算的。

按国籍划分

图13-15显示了提交作品的作者的国籍图。受到《纽约时报》的“奥运金牌图”的启发[注6]，该布局是采用物理实体模型和逼近精确位置来计算的，从而避免了圆的交叠（见图13-16所示的迭代优化过程的快照图）。

注6：参考*http://www.nytimes.com/interactive/2008/08/04/sports/olympics/20080804_MEDALCOUNT_MAP.html*。

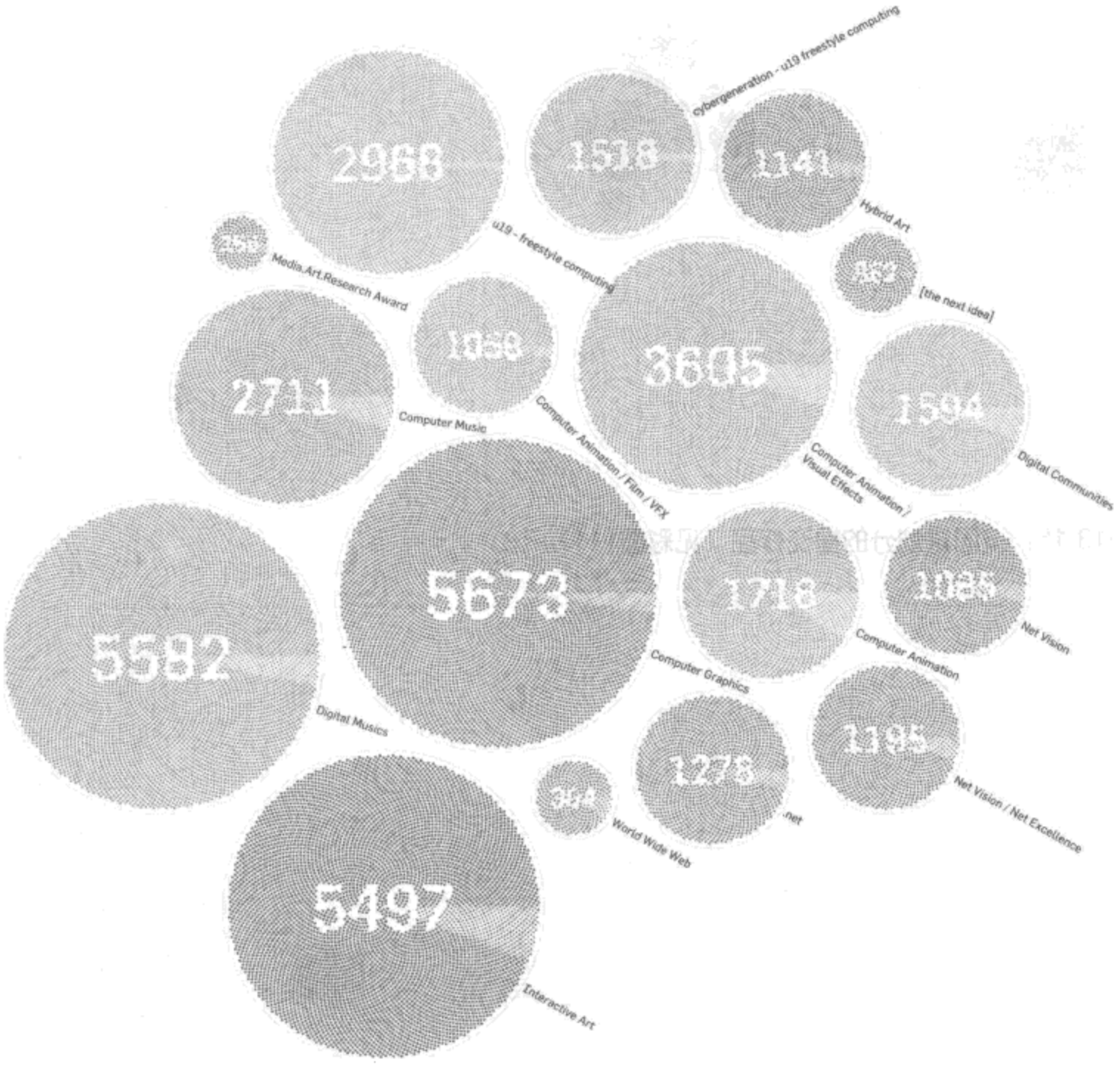


图13-14 按作品类别划分的作品（见彩图110）

为了得到国家名字的坐标，我使用了在线应用程序mapspread[注7]，它允许用户批量查询表格数据来获取地理坐标。然而，需要一些手工校正，因为一些国家名字无法确定（特别是东欧国家，东欧的政治格局在过去几十年有很大改变），而其他一些国家名字很含糊：实际上，甚至在最终版的地图中，格鲁吉亚国家还是被错误地放置在美国旁边，而实际上应该是坐落于俄罗斯和土耳其之间的东欧国家。

注7：　参考*http://mapspread.com*。

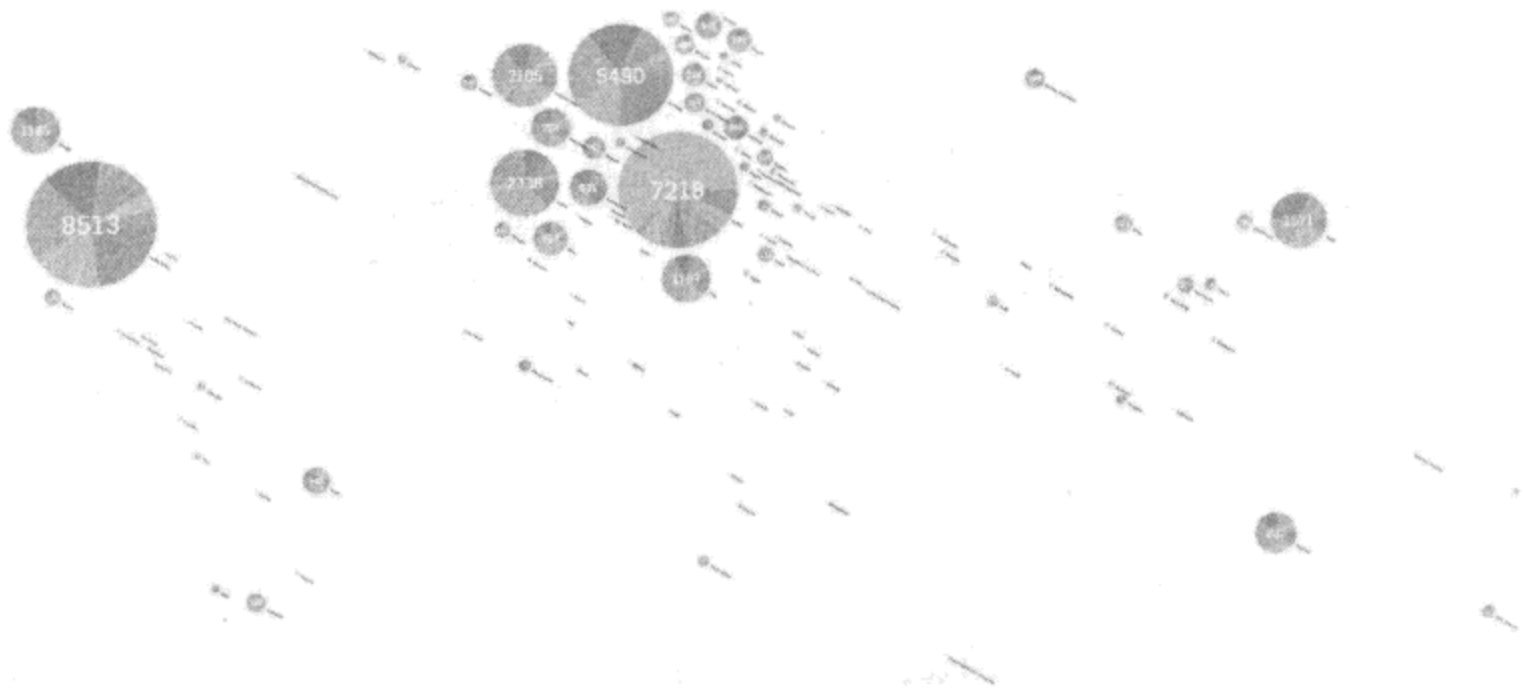


图13-15：按国籍划分的提交作品（见彩图111）

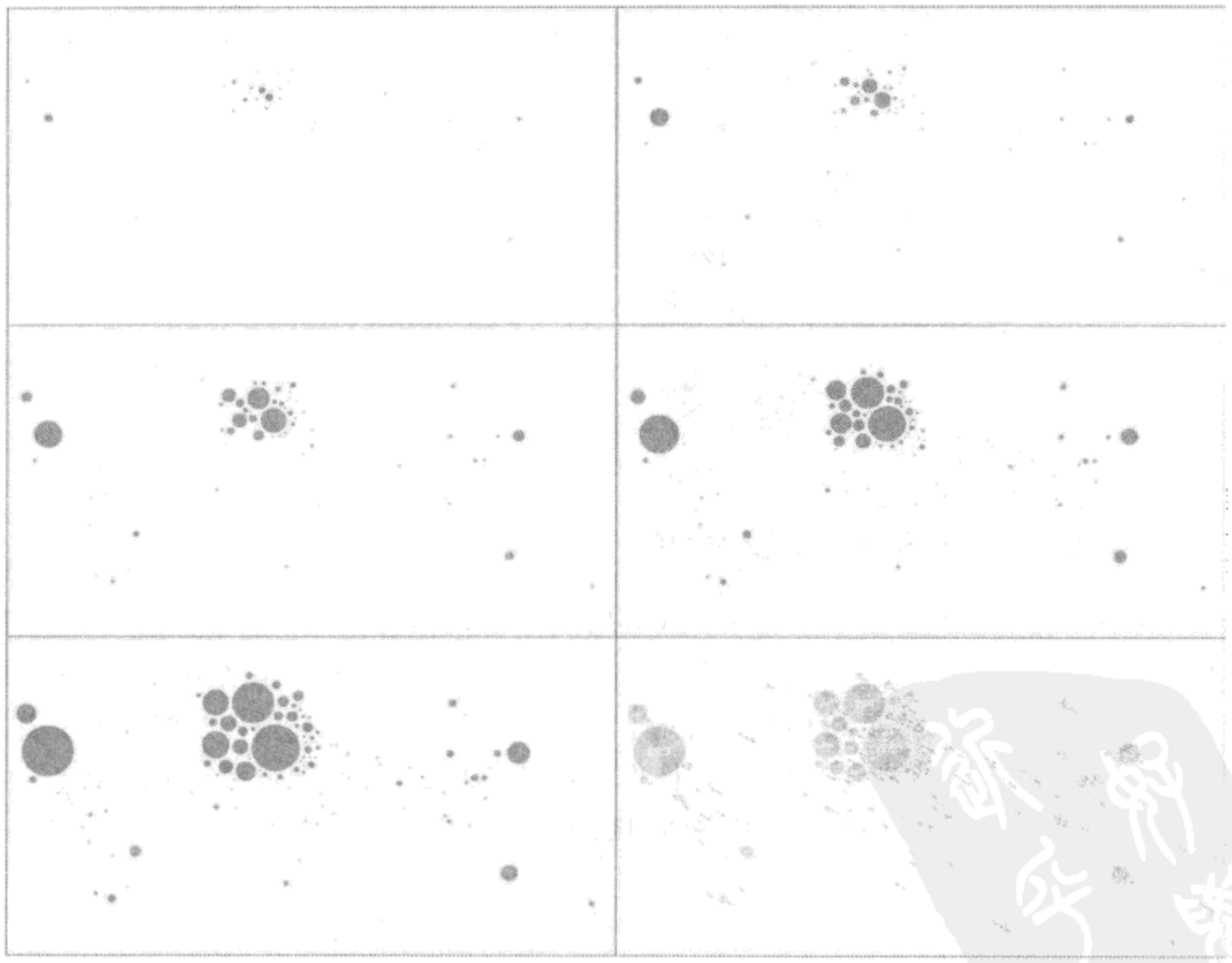


图13-16：迭代式图片优化快照（见彩图112）

仔细查看地图，可以发现媒体艺术的本质是以欧洲/美国为中心。南美洲、非洲、俄罗斯和亚洲（日本除外）的提交作品数很少。从历史上看，从法国和西班牙的大量的提交作品是关于计算机动画和电影（橙色显示）领域。从显示上看，意大利、瑞典和英国呈现出提交更多的音乐类别（紫色）的作品的趋势，而日本则似乎提交更多的是交互作品（蓝色）。相反地，德国和美国趋向于计算机图形（红色），至少在奥地利艺术节初期如此。几乎三分之二的奥地利提交作品（只有奥地利的）是属于U19类别。

按年份划分

图13-17所示的饼图序列显示了在3个时代的奖品历史的明确划分。在1995年，提交作品数量急剧减少，这和计算机图形领域的类别的终结和万维网类别的引入一致。这种下降的一个可能的解释是每年在计算机图形领域提交更多作品很平常。2004年后的年份显示出更强的种类多样化以及提交作品的陡增，主要是由于引入了19岁以下的奥地利艺术家的U19类别。

图13-17：按年份划分的提交作品（见彩图113）

按年份和类别划分

为了对个别类别团体的发展进行审查，图13-18显示了时间轴矩阵格式版本。对于颜色编码和行选择这两个方面，我们都决定对相应类别进行分组，即使它们的标题在过去几年有所变化。（反之，需要注意的是一些名字没有变化的分类在不同年份有不同的定位。）和单个年份图形相比，这个版本更易于观看动画/电影、音乐，而且后期的交互艺术称为Prix奥地利艺术奖的长期支柱。

展览

“测绘资料馆”（Mapping the Archive）是位于Brucknerhaus的历史展览，以由Dietmar Offenhuber、Evelyn Münster、Jaume Nualart、Gerhard Dirmoser和我一起创建的6种不同的数据可视化为特征（见图13-19）[注8]。

注8：　所有可视化都在网上有记录*http://vis.mediaartresearch.at*。

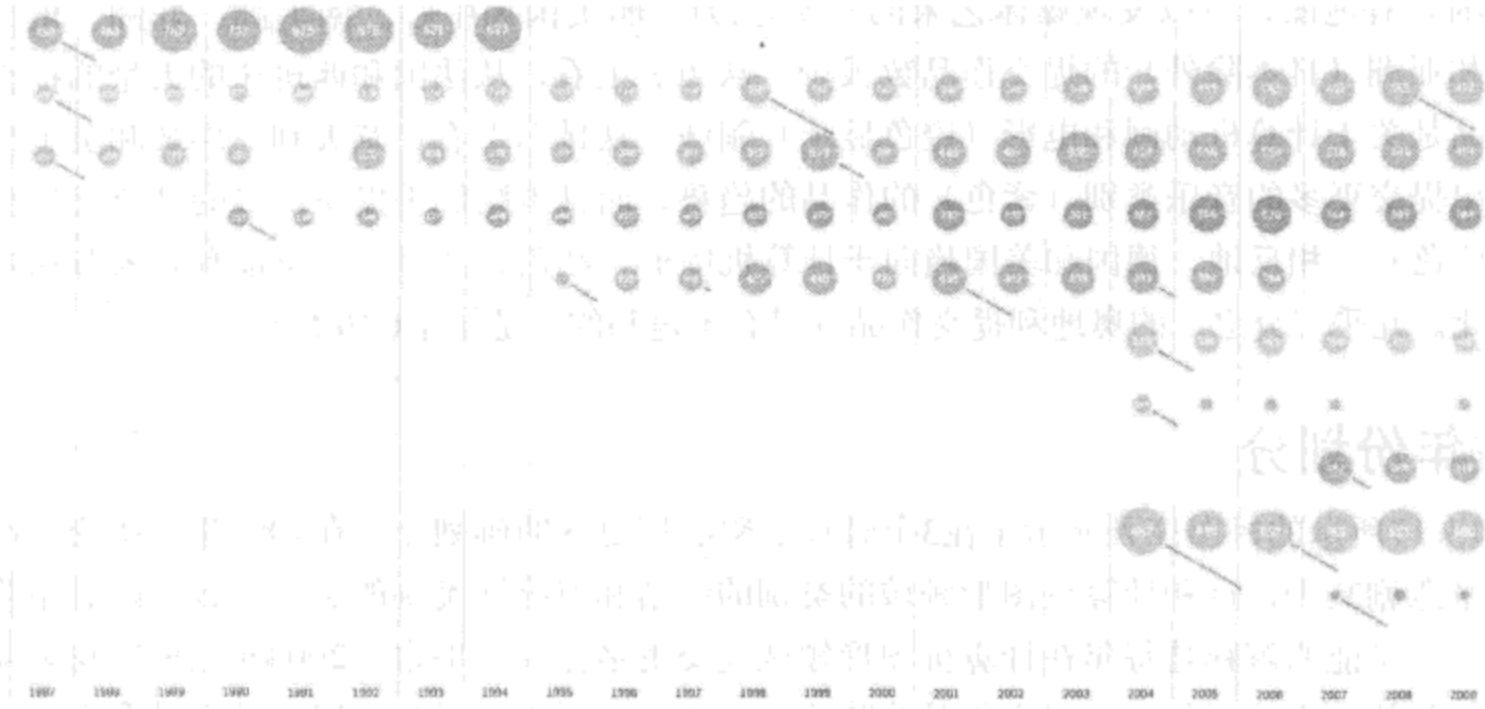


图13-18：按类别和年份划分的提交作品（见彩图114）

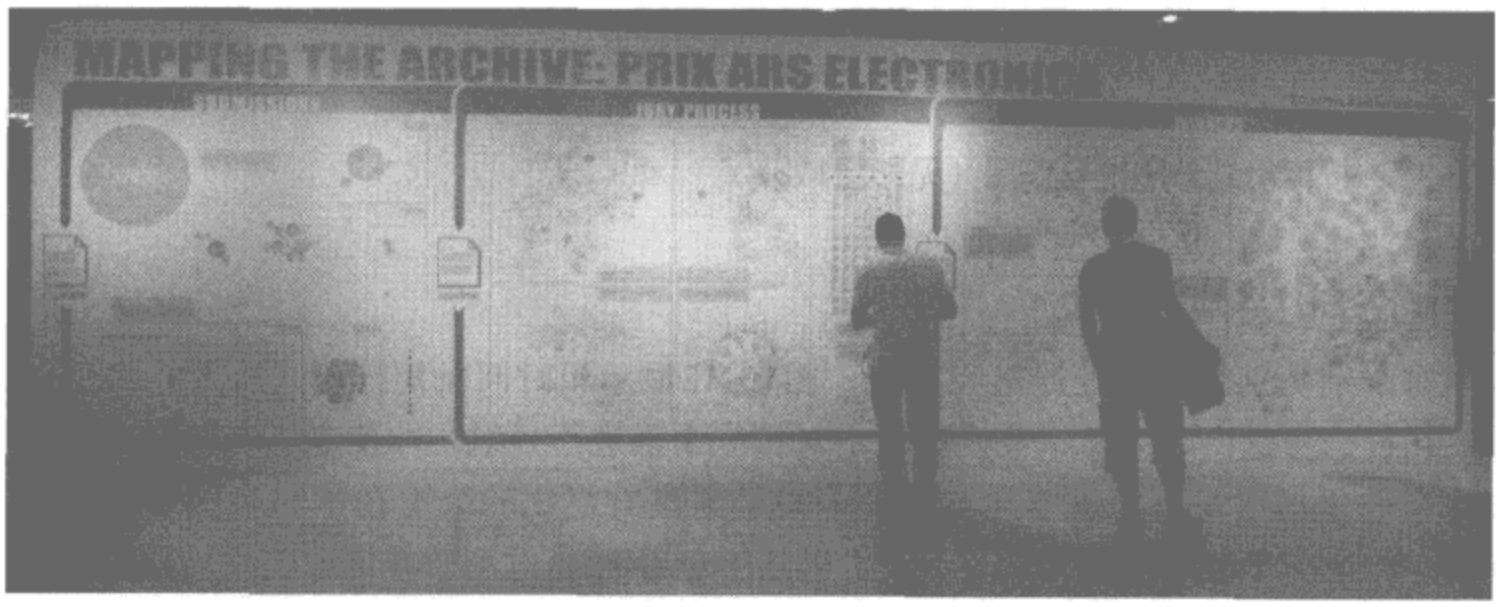


图13-19：展览中的作品

为了有助于发现数据中独特的故事，我们增加了少量的标注箭头来突出有趣的方面，如图13-20所示。我们还鼓励用户添加他们自己的注释，结果是生成了一些有趣的问题和标注。

结束语

本章介绍的可视化是在2009年夏开发的，不仅和负责资料库的技术人员不断交流思路和信息，而且也和对所展示信息的语义方面进行评论的媒体艺术专家进行了不断的交流。

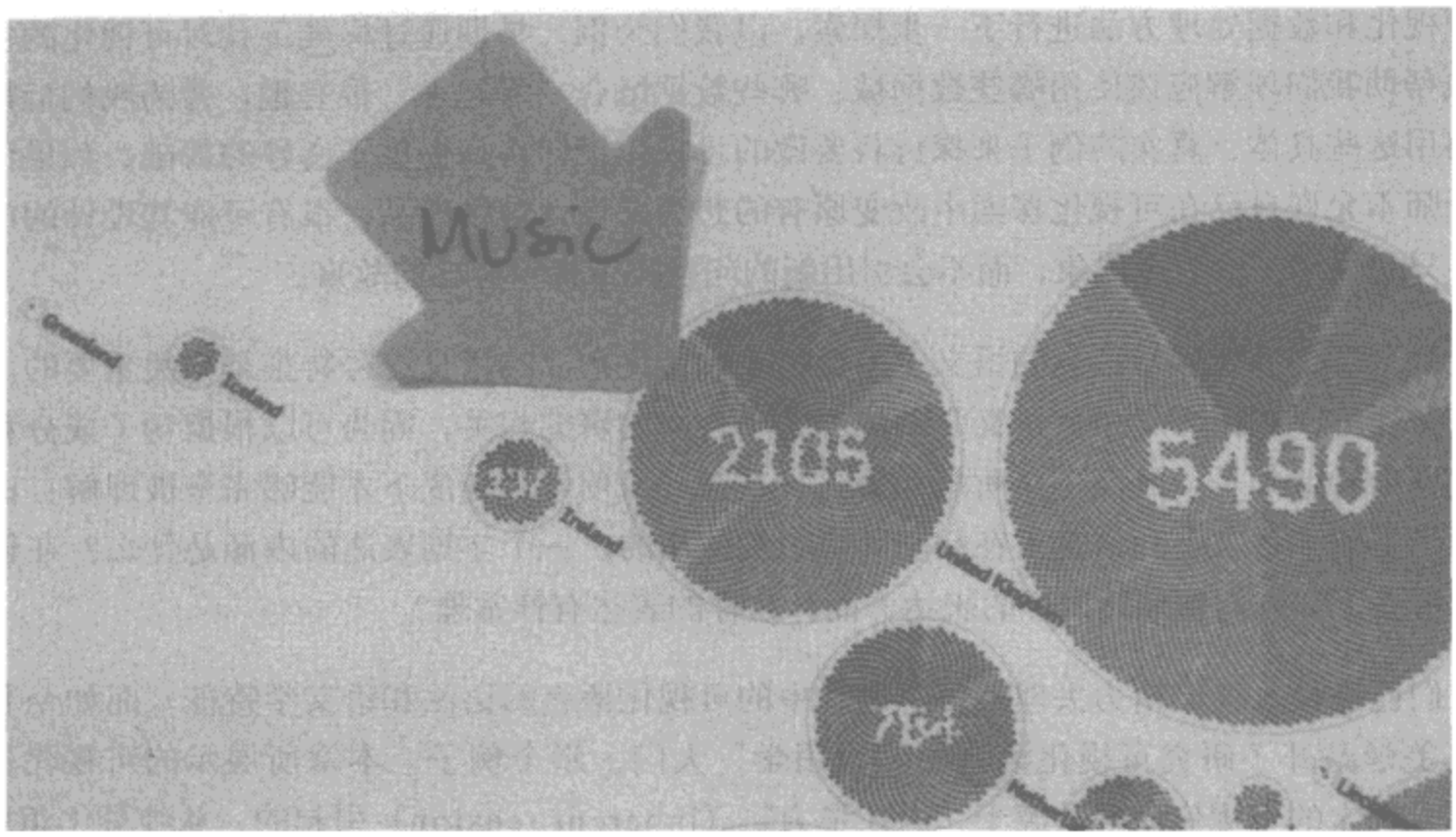


图13-20：包含手写注释的箭头形状的粘贴（见彩图115）

我认为该可视化工作是“信息美学”（information aesthetics）[注9]的新兴取向的一部分。信息可视化作为一门科学，通常涉及一般的视觉映射方法以及对生成的结果可视化的可读性和可理解性的优化。信息美学是基于该领域构建的；然而，作为一门设计学，信息可视化力求找到一种基于特定数据集的信息感性化的展现方式，这种展现方式不仅在显式数据展现层次上是可用和可读的，而且增加了设计的“命题密度”（propositional density）[译注3]——简而言之，它表示可视化中深层的形象特征，是可视化展现的“言外之意”。“信息美学”这门学科就是以这种方式介于传统的信息可视化、用户界面设计和美学学科之中。

我希望本章说明了“信息美学”这门学科的一些关键特征。首先，查看创建信息美学作品的过程是很重要的。根据我的经验，以真实数据工作是非常重要的，甚至是在早期的设计阶段。原则上，很多从理论中提炼的可视化想法在早期的数据结构中工作良好，但是它们是否传递有趣的信息以及是否有助于解决问题（或者提出新的问题），这些只能在处理实际数据时才能确定。可视化开发必然是一个不断自我引导的过程（bootstrapping process）：在早期你必须对这些方式进行实践，才能理解应该使用哪些

注9：“信息美学”这个术语是Lev Manovich创造的，在《Lau and Vande Moere》（2007年）中有详细说明。

译注3：“命题密度”这个术语是William Lidwell（2009年）定义的。想要了解该术语的确切含义，请参考*http://well-formed-data.net/archives/495/propositional-density-in-visualization*。

可视化和数据处理方法进行下一步探索。以我们为例，早期通过标准工具对可视化的实践帮助我们理解应该使用哪些数据域、哪些数据组合“看起来”很有趣，并为我们后期引用这些具体、真实的例子来探讨将要做的可视化设计特征提供了良好的基础。如果设计师不允许自己在可视化探索中改变原有的想法来设计最终产品，很有可能其设计的产品只会展示一些表面现象，而不会引出新的问题或者揭示有趣的故事。

此外，意识到所展示信息的语义上下文关系和最终产品的语义符号特征是至关重要的。打个比方，在语言学中，语义学领域和句子含义的研究相关，因为可以根据句子成分和组合来构造句子。然而，众所周知语言只有在“语用学”角度下才能够完全被理解：语用学是研究语言是如何真正在社交环境中被使用的。一个字词表达的内涵是什么？在特定情景下，人们预期什么样的表达，而什么样的表达有悖常理？

人们已经投入很大精力去理解信息展现中的可视化语言的语法和语义学特征，而如今信息美学敲开了研究可视化语言的“语用学”大门。举个例子，本章所展示的可视化作品，选定的可视化原则是源于“内在张力”（inherent tension）引起的、从纯量化角度来查看复杂的社会现象。在深入探索丰富多样的数据集中，只通过“几个数字”，从方方面面表示22年的媒体艺术历史，我们的可视化展现的是什么？可视化的展现方式是尝试捕捉上述内在张力，并解决其中一部分。

从以上的分析可以看出，可视化中的“美学“概念远远不只是“漂亮的照片”。当然，使用舒心是一项重要且一直被低估的因素——在很多情况下，关于用户体验的研究说明了在愉快舒适、令人兴奋的环境中互动的重要性。但是，正如史蒂夫·乔布斯的一句名言“设计不在于产品的外观和感觉，而是它如何工作。”一个真正的审美可视化，除了必须美丽外，而且必须能够表达现有的潜在隐含特征，并能够激励用户/读者去探索更丰富多彩的世界。

最后一点，查看可视化中展现的信息的含义和上下文，人们常常忽略了一点（甚至是在本章中所展示的可视化）：我们如何在更大规模上对信息进行特征化？我们是否能够通过连接到外部数据库，找到对观察到的模式的解释？以奥地利艺术节为例，比较每个国家的提交作品数统计比给每个国家提供更多的信息展示可能信息量更多。一个国家提交的作品数和其经济实力是否相关？或者是否和数字素养（digital literacy）相关？或者其他不太明显的因素？由于越来越多的开源数据源提供这些信息而且可以访问，为真正了解我们所分析和展示的数据库中的新兴的模式的重要性提供合适的背景和基线变得越来越重要。

致谢

感谢Ludwig Boltzmann研究所对媒体美学的研究（Linz），特别感谢Dieter Daniels和

Katja Kwastek为我们提供这个机会并提出专业意见，Sandor Herramhof收集和处理了大量数据，Dietmar Offenhuber对该可视化工作的富于创意的协调以及Ule Münster展示海报和用于分类的色彩调色板的设计。

参考文献

1. Keim, D.A. 2000. "Designing pixel-oriented visualization techniques: Theory and applications." *IEEE Transactions on Visualization and Computer Graphics* 6, no. 1: 59–78.

2. Lau, Andrea, and Andrew Vande Moere (2007). "Towards a model of information aesthetic visualization." In *Proceedings of the International Conference on Information Visualisation*. Washington, DC: IEEE Computer Society.

3. Lidwell, William. 2009. "More with less." *ACM interactions* 16, no. 6: 72–75.

4. Wang, Weixin, Hui Wang, Guozhong Dai, and Hongan Wang. 2006. "Visualization of large hierarchical data by circle packing." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press.

5. Wattenberg, Martin. 2005. "A note on space-filling visualizations and space-filling curves." In *Proceedings of the 2005 IEEE Symposium on Information Visualization*. Washington, DC：IEEE Computer Society.

第14章

矩阵探秘

Maximilian Schich

本章揭示了资料数据库中因为管理员的本地操作和数据源的异构而产生的一些非直观的结构。例子取自艺术史和考古学领域，之所以选择这两个领域是因为它们是我的专业研究领域。尽管如此，本章将要展示的成果——对数据库的复杂结构进行可视化呈现——同样适用于很多其他领域的结构化数据集，包括生物研究数据库和大众协作数据库，如DBpedia、Freebase或语义Web。所有这些数据集合都拥有很多共同的属性，这些属性往往不具备直接的应用价值，但是当我们想要充分挖掘已有数据的应用价值、或者确定应该从何处入手，以及如何花费精力和资金来提升这些价值时，这些属性将非常重要。

艺术史和考古学的资料库的数据来源有很多种，如图书馆目录和文献目录、图片归档库、博物馆目录以及一些通用的研究数据库。所有这些可能都是基于非常复杂的数据模型进行构建的，而且只要数据足够多，即便是最乏味的例子——不管表面上看起来有多么简单——其中的任意一种关联关系都会复杂得让人困惑。专题报道可能涉及所有的人造事物：比如美国国会图书馆分类系统会处理包括艺术家、食谱乃至物理学论文等所有东西。

我选择了一个数据集作为本章的例子，其规模足够大，结构足够复杂，但其数据规模尚处于可以有效地处理的范围之内。我们将对针对文艺复兴时期的仿古艺术品和建筑开展的普查统计进行可视化（*http://www.census.de*），该普查统计是由Richard Krautheimer、Fritz Saxl和 Karl Lehmann-Hartleben在1947年发起的。它收集了古代的历史遗迹，比如罗马雕塑和建筑，在西方文艺复兴时期的作品如写生、素描和旅游手册。用于存储这些

数据的数据库在2006年刚从基于图形的数据库系统（CENSUS 2005）转换成更传统的关系型数据库（CENSUS BBAW）。我们将分析数据库在转换之前那个时间点的状态。有了这份数据，我们将可以就历史状态和当前以及今后的成就进行比较。

越多越好吗

在艺术研究数据库领域工作的10余年之中，一直存在的最为耐人寻味的问题之一是如何衡量项目的质量。人文领域的数据库很少会像学术文章那样被引用，因此在出版发行行业中的常用评估标准并不适合。然而，大多数评估只是基于很多肤浅的标准，比如是否和制定的标准一致、用户接口质量、是否有很炫的项目名称以及在项目描述中是否使用了最近的流行语。而对于内容，评估者通常只是采用一些基本的衡量标准，如查看数据库中的记录条数、询问一些和很多特定条目的微妙之处相关的问题。

在数据标准的定义中，如数据模型中的CIDOC概念参考模型（CIDOC Conceptual Reference Model，CIDOC-CRM）或数据交换中的“获取元数据的开放信息仓库首创协议”（Open Archives Initiative Protocol for Metadata Harvesting，OAI-PMH）中存在的一个问题是，它们通常需要使用先验知识，同时在它们的框架中却没有提供与正在收集和处理的数据相关的任何信息。用户界面也存在同样的问题，其提供的关于内容质量的信息就好比只给一张打印纸提供了长宽比信息。此外，数据标准和用户界面都会随时间变化，这使得以其作为评估标准的合理程度的判断更为困难。正如每一个程序员所知，一个用老的Fortran语言实现的算法和用当前流行的Python脚本实现的可以一样优雅，而且速度甚至能够更快。因此，我们在项目评估中应该避免任何形式的系统主观偏见，也就是说，一个坚守某个标准的用户不应该畏惧其他标准的粉丝所做的评价。

即使我们一致认为应用标准是可取的，如“开放访问”（Open Access）标准（也称“开放存取”），但是其带来的影响也是值得商榷的：虽然“开放访问”给当前很多项目提供了积极的作用，但其在资料库领域的涵义并不完全清晰。我们是否真的应该满足于一个复杂但免费的用户界面（如图10所示，Bartsch 2008），或者我们是否应该更倾向于选择复杂的API以及周期性地对数据执行全库导出（如Freebase），后者是否会带来更严格的数据分析以及更高深的数据重用？如果都采用“开放访问”标准，还有谁会愿意给私有的企业数据资料库付费呢？

最后，我们必须查看任何给定项目的实际内容。正如本章中将会说明的，当对数据库进行评估时，只研究一些特定条目的微妙之处所带来的意义很有限，因为通常情况下不存在通用的信息来衡量任何特定的数据库条目。无处不在的“长尾”（long tails）问题（Anderson 2006，Newman 2005，Schich等 2009），我们在本章会遇到，这意味着外插一些富信息的数据条目到整个数据库中是不明智的——也就是说，在CENSUS，我们无法只基于“万神殿”推断所有其他的古代遗迹。

评估中常用的最公正的衡量标准是数据库的记录数。几乎所有的项目说明书中都包含该标准：百科全书列出了它们所包含的文章的数目（如维基百科）；生物医学数据库公布了化合物、基因或者其包含的蛋白质的数量（如Phosphosite 2003～2007或Flybase 2008）；甚至是传统的搜索引擎（但是数量越来越少）在它们的索引中提供了页面数量（Sullivan 2005）。因此，CENSUS项目也提供了一些数字说明是不足为怪的：

> 超过20万条目包含图像和文字文件、地点、人物、时代和风格、事件、研究文献和说明。登记的古迹约有6500个，古迹条目约12000，文献条目约28 000[注1]。

虽然从艺术史角度看，这些数字很让人震撼，因为一个大型展览目录通常仅仅包括几百个条目。但是在查看具体个例时，可以很容易地找到反例证明用记录数作为衡量数据库质量的关键指标是不合适的。由于搜索引擎处理邻近相关的副本（Chakrabarti 2003）、如CENSUS这样的研究数据库目标是对数据进行范化，其方式是通过消除原始数据中的不确定性和曾经的意见不一带来的明显的冗余。图14-1中的例子很让人吃惊。注意，连接总数在泛化前后保持不变，由此引出了一个更有意义的对质量进行初步近似评估的指标，使用连接数和节点数的比例：3/6和3/4（在本例中）。

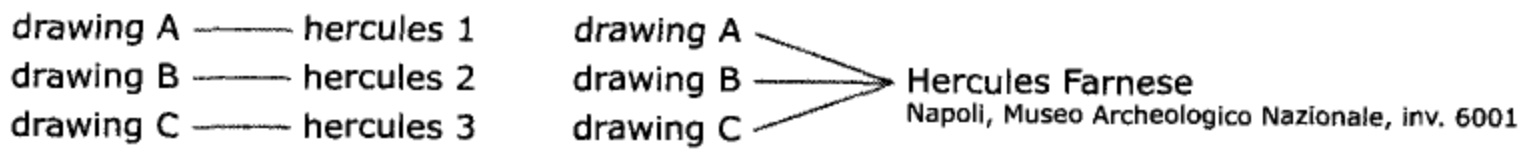


图14-1：缩小记录数，提高数据集质量

显然，为了评估给定数据库的质量，需要有更复杂的措施。如果我们真的想知道数据集的价值，我们需要查看生成的全局结构，常用指标无法显示这些。对于任何数据集，我们唯一可以预期的是全局结构可以特征化并生成一个复杂的系统。复杂性源于人们在本地所执行的操作（Chua 2005），也因为数据源的可用性和人们对它的关注度本质上是非常异构的。此外，每个资料库的管理员对于先验数据模型的定义都有不同的看法。由此导致的结构化复杂性难以预测，我们需要以有意义的方式对数据库进行衡量和可视化。

把数据库看做网络

艺术史和考古学领域的结构化数据，正如在任何其他领域一样，有很多形式，比如关系型或面向对象型的数据库、电子表格、XML文档和RDF图；wiki、PDF、HTML页面上的半结构化数据以及传统纸张上的（可能比其他领域包含的半结构化数据都要多）。不考虑这些表现形式的细节，基础的技术结构通常涉及3个领域：

注1：来源于*http://www.census.de*, retrieved 9/14/2009。

- 一个数据模型协定，包括从存放在木盒子里的简单的索引卡片的分隔板到你最喜爱的展示语言中的复杂的本体。
- 数据格式规则，包括显示模板如透镜（Pietriga等 2006）或者预定义的查询指令。
- 数据处理规则，根据数据格式化指令执行的处理规则。

在这里，我们最感兴趣的是选定的数据模型协定如何和已有的数据关联。

正如Toby Segaran在《数据之美》一书中所指出的，数据模型协定有两个不同的思路。其一，每当需要增加新的信息时，可以给数据库创建新表、给已有表增加新的列和索引，以及在不同表之间建立新的关联，这种方式导致数据库模型变得更加复杂。其二，可以创建一个非常基础的模式，如图14-2所示，该模式可以支持任何类型的数据，本质上是把数据表示成一张图而不是一组表。

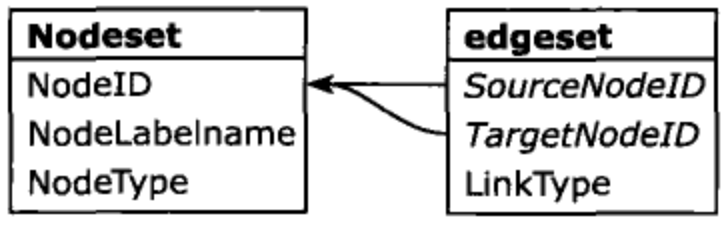


图14-2：数据库可以映射为基础的节点和边模式

如上表所示，可以认为每个数据库都是一个网络。数据库条目代表网络的节点，而节点间的关联关系代表网络的边（即所谓的边或连接）。如果我们把艺术研究数据库看作网络，就会产生很多可能的节点类型：节点可以是各种实体对象如古迹、文献、人物、地点、时间或事件的条目（Saxl 1974）。两个节点之间的任何关系（如“图片A是由B创建的”）都可以是一条连接或边。因此，基于不同的节点之间的关系，存在大量可能的连接类型。

网络中的节点和边的类型的先验定义和传统的数据模型一致，能够描述由很多管理员生成的大量数据的集合。此外，采用网络描述使得在复杂网络科学中的计算分析方法可以直接应用，获取所有可用数据的大范围的全局概览成为可能。因此，对于隐藏于当前的知识范围以外的通过对数据库概念化和普通的本地查询等方式无法发现的结构，我们现在拥有发现能力了。反过来，这种方式也促使我们超越通 常的质量评估衡量标准：可以先检查数据和数据模型的适合度，采用的标准是否是恰当的，以及将数据库与其他数据源关联起来是否合理等。

可见的数据模型定义

为了对基础的结构有一个了解，我们在数据库评估中首先希望看到的是数据模型——可能的话，它应包含描述数据在模型内的分布情况的一些指标。如果是从数据库的图形表

示出发，如图14-2所示，这是一个简单的任务。我们所需要的就是一个节点集合和一个边集合，这两个集合可以很容易地通过一组关系表生成；如果数据库可以导出为RDF格式（Freebase 2009）或者作为连接数据（Bizer、Heath和Berners-Lee 2009），甚至还可以免费获取。有了这两个集合的数据之后，使用制图应用程序如Cytoscape（Shannon等2003）——一个起源于生物网络科学社区的开源应用程序，可以很容易地生成节点-连接图。最终的图表如图14-3所示，使用类似于普通的实体-关系（ER）图（Chen 1976）的方式描述给定的数据模型，并在图中包含了一些实际数据的量化信息。

图14-3中的CENSUS数据模型是从图14-2中描述的数据库模式中抽取出一个“元数据网络”（metanetwork）：每种节点类型都是一个“元节点”（metanode），每种连接类型是一个“元连接”（metalink），它连接两个元节点。元节点的大小反映了节点的实际数目，元连接线的长度反映了连接的实际数量，这种方式为我们提供了一种数据库模型内的有效的数据分布的直观表述形式。注意节点大小和连接线长度在不同类型之间是高度异构的，在我们的例子中包含了4～5个不同维度。常见的节点和连接类型在实际中出现的次数要远远多于绝大多数不常见的节点类型——传统的ER数据结构图通常并没有反映出这一点，这往往导致在一些特定数据模型中人们对一些几乎不相关的领域进行了冗长的讨论。

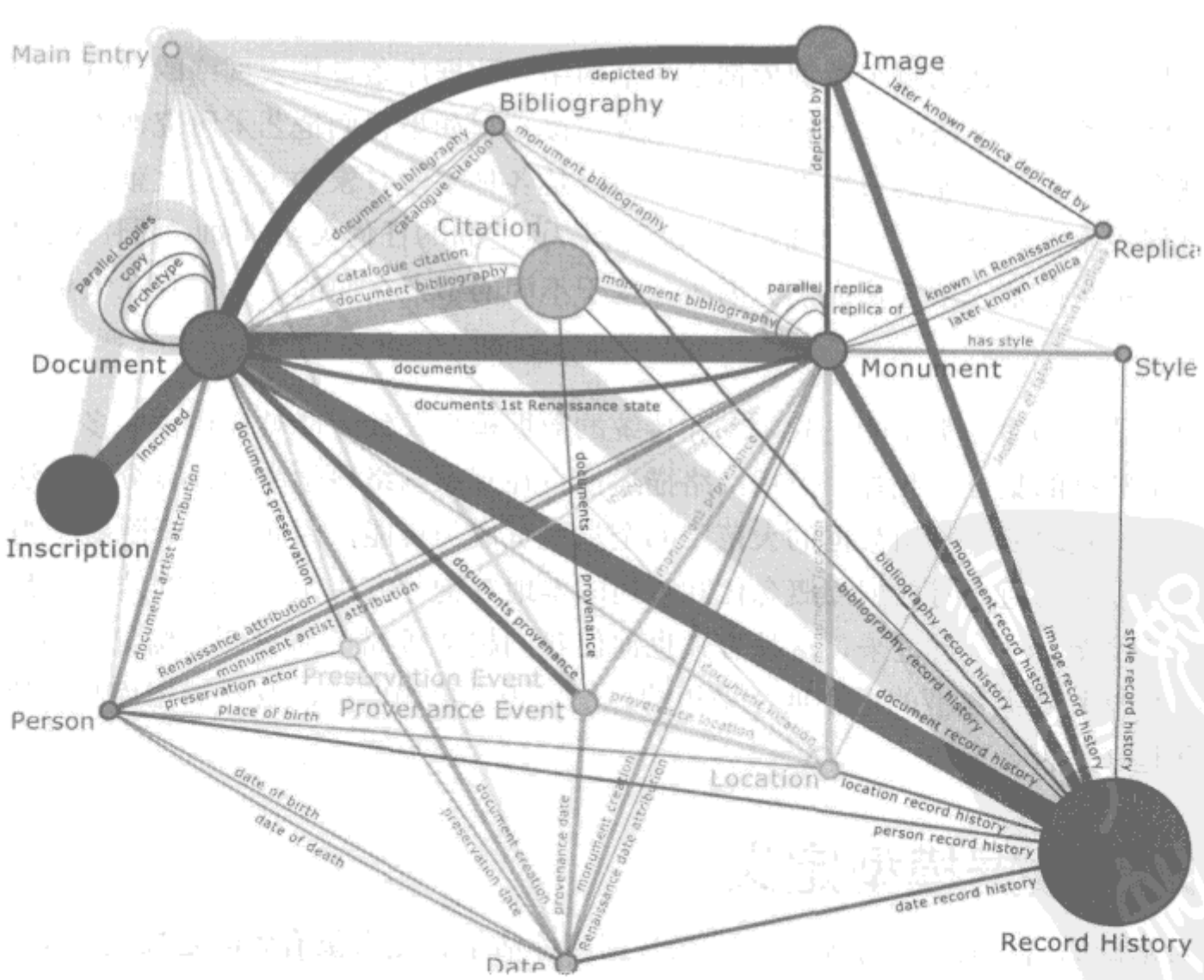


图14-3：CENSUS数据模型和加权的节点-连接图（见彩图116）

节点和连接类型频度的异构性并非仅仅存在于图14-3所给出的例子中。在很多数据集中都可以观察到这一现象，而不管其类型数目是预定义的还是随着管理员的人数而变动的，比如研究数据库（Schich和Ebert-Schifferer 2009）、大量的文献目录（Schich等2009），Freebase和连接数据云。据我所见，每种节点类型的节点数目和每种连接类型的连接数目都呈现出右偏衰减分布，即众所周知的“长尾”（Anderson 2006，Newman 2005）现象，并且在分布中并不具备正态高斯分布中均值相同的特征。Web页面中超链接的“长尾”结构——也就是说，一种特定的连接类型只存在于一种节点类型之上——在过去10年中一直是众所周知的（Science 2009）。图14-3清晰地证明了在节点和连接类型中所观察到的异构性，在更加结构化的数据图形中，不同层次的节点和连接类型中也存在异构性。

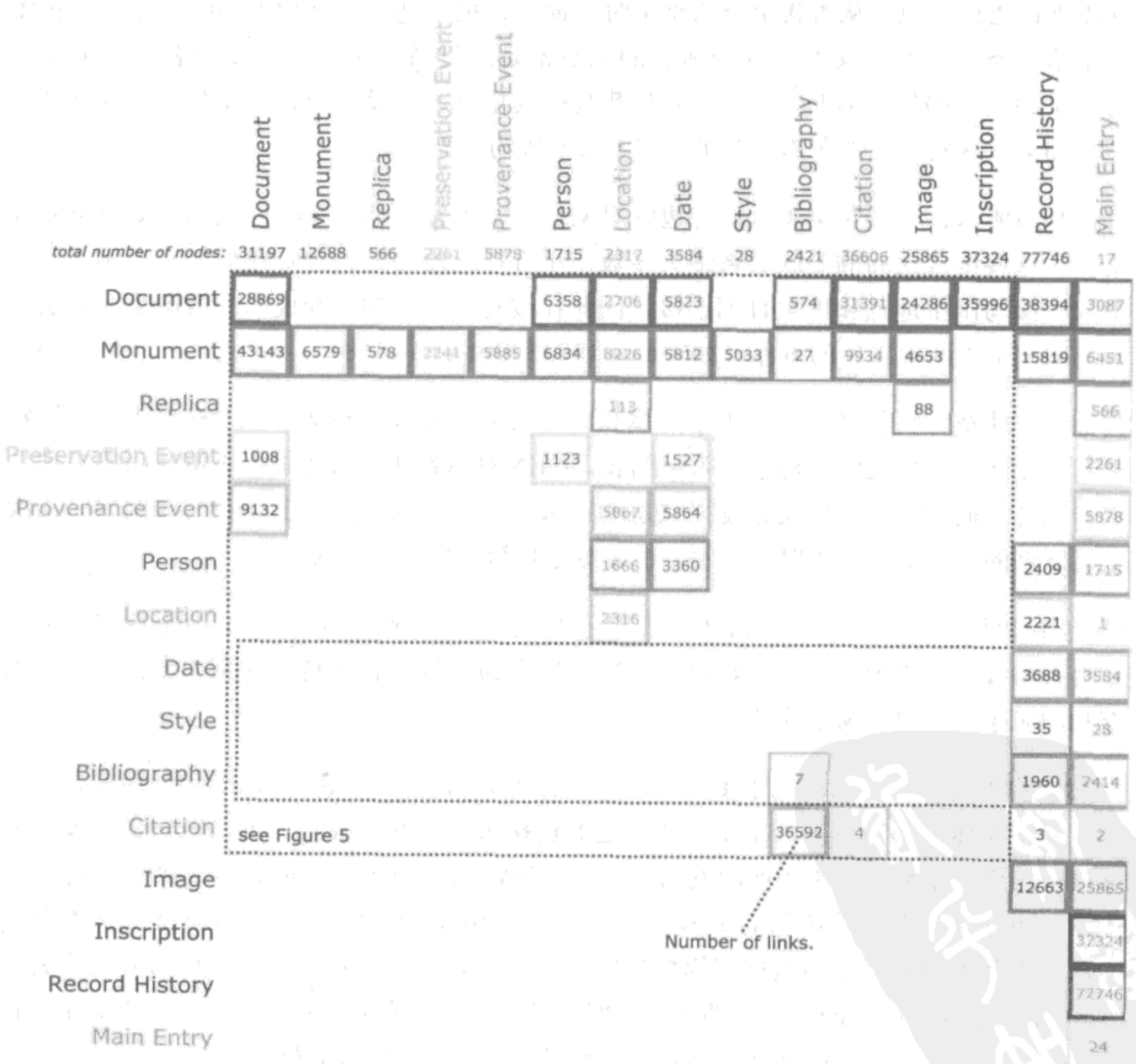


图14-4：以加权邻接矩阵形式表示的CENSUS数据模型（见彩图117）

网络维度

进一步观察图14-3，我们可以发现CENSUS数据库的核心维度——古迹和文献——为一些额外信息所包围。古迹和文献都是实体对象，但是到目前为止，它们之间的区别在于前者是中心文献连接的目标，而后者是中心文献连接的源头。虽然通常来说任何实体对象都可以作为古迹或文献，但是CENSUS把它们划分成了离散的节点类型，因为这两种类型属于不同的时期（古典和西方文艺复兴）：文艺复兴时期的绘画、素描、文本等记录了古代罗马的雕刻和建筑构造。

除了上述几个重要的维度，还有另外一种被称之为“副本”的节点类型代表实体对象，用于代表后来的副本古迹，它只存在于文艺复兴的特定时期之后。如果要对CENSUS数据库进行泛化，包含从古代至今的整个时间窗口，把古迹、文献和副本结合成一个实体对象节点类型是有意义的，因为所有的函数都是通过特定节点的入度或者出度来定义的。在20世纪80年代早期，当刚开始构想数据模型时，其设计受到关系数据库的某些功能的约束。这些约束现在不存在了，因此改变成为可能。

如图14-3所示，分布在实体对象旁边的对象包括：人物、地点和时间范围（如日期和风格）。这些维度之间的关联关系绝大多数使用直接连接的方式进行建模。举个例子，每个人直接与出生地点和出生日期连接，因此在没有进一步的注释说明的情况下，无法识别出同一个人两次出生的事件（如Venice 1573和Bologna 1568）。

其他示例快捷方式包括文献艺术家归属和第一次文艺复兴状态文献。同样，不增加注释说明是无法消除歧义的。对于艺术家归属，CENSUS管理员需要做出决策，而不是记录多个意见。而对于第一次文艺复兴状态的记录，定义上只存在一个唯一的实例。第二次文艺复兴的状态被记录成保存事件，很显然这是简化数据模型的一个机会。

保存和起源事件是前文中提到的捷径的一个值得注意的例外情况。它们指出特定的古迹是由人物改变或者展现在一个特定的位置、特定的日期，正如特定文献中所记录的。保存和起源事件都很容易消除歧义。

对文献的不同意见可以反映于多个事件中，把不同的古迹、人物、地点和日期粘合在一起。对于实体对象，事件的本质是由特定的连接来定义的。因此，可以进一步对数据模型进行泛化，正如CENSUS所激发的某些项目如Winckelmann Corpus（2000）。通常情况下，事件可以归结为所谓的星形模式（参考Milo等 2002），包含特定的连接类型。当前，事件类结构是很多数据库模型的标准特征，如Freebase，它们被称为复合值类型（compound value type）。原则上，我们还可以使用其他类型的网络查看这些事件，它们并非非常明显，而是内在地作为新型星形模式存在（如连接数据图）。

通过提供很多“元维度”（metadimensions）的信息源，如（现代）文献目录，

CENSUS数据库变成了权威，即被引用。文献目录又进一步被划分成引用，它是由单独的节点类型来表示的。另一个来源维度是图像节点类型，它包含从主要的图像库中拍摄的照片。同样的，文献目录和图像都有表示实体对象功能，它是通过一些相邻连接定义的。

其他节点类型包括：记录历史，管理员把他们的操作日志记录到其他节点中；主条目维度，在把CENSUS转换成关系数据库后可能不会再存在。前者是基于图形的系统，由于缺乏数据表，需要通过主条目把数据库分成不同部分，把任务、地点等结合起来促进导航。

矩阵“缩小镜”

图14-3的节点连接图是描述CENSUS数据模型的众多可能方式中的一种。正如由节点和边构成的任何网络一样，我们也可以使用所谓的邻接矩阵（参考Garner 1963；Bertin 1981；Bertin 2001；Henry 2008）来表示这个数据模型，如图14-4所示。在这种描述形式里，节点的类型使用表的垂直列和水平行表示，在单元格中显示节点信息。比如出生地信息，你可以假定存在一条连接，从“人物”（Person）所在的行穿过不同单元格指向了“位置（Location）所在的列。

类似于节点连接图，邻接矩阵还可以描述出两种类型的节点之间的连接数；数字显式地出现在相应单元格中，而不再通过如图14-3中线条的宽度来表示。这是节点邻接矩阵不同于节点连接图的重大之处：我们现在关注的主要是连接而非节点了。引人注目的是，图14-4中的矩阵不仅显示了不同类型节点之间的连接，而且非常清晰地说明了哪些节点类型间没有直接关联。换句话说，邻接矩阵可以同时表示正关联和负关联关系。其中的一个例子是不存在从作者、出版地点、出版日期到文献目录的连接；虽然CENSUS提供了这些信息，但它只存在于节点描述文本和节点标签缩写中（如Nesselrath 1993）。当然，我们从节点连接图中也能发现这种信息缺失，但是在邻接矩阵中这一点更为明显。

除了两种节点类型之间的连接总数，在邻接矩阵单元中还可以放置很多其他有用的信息。举个例子，如图14-5所示，我们可以看到一个包含了所有节点的节点连接图以及位于一个单元格内的表示两种节点间关系的连接。这个图是我们使用一种布局算法（比如Cytoscape应用中的yFiles有机布局算法）生成的，这是一种运算成本相对较低的方法。因此，数据库中的所有显式的节点和连接数据都在这个数据模型矩阵中得到了展示。

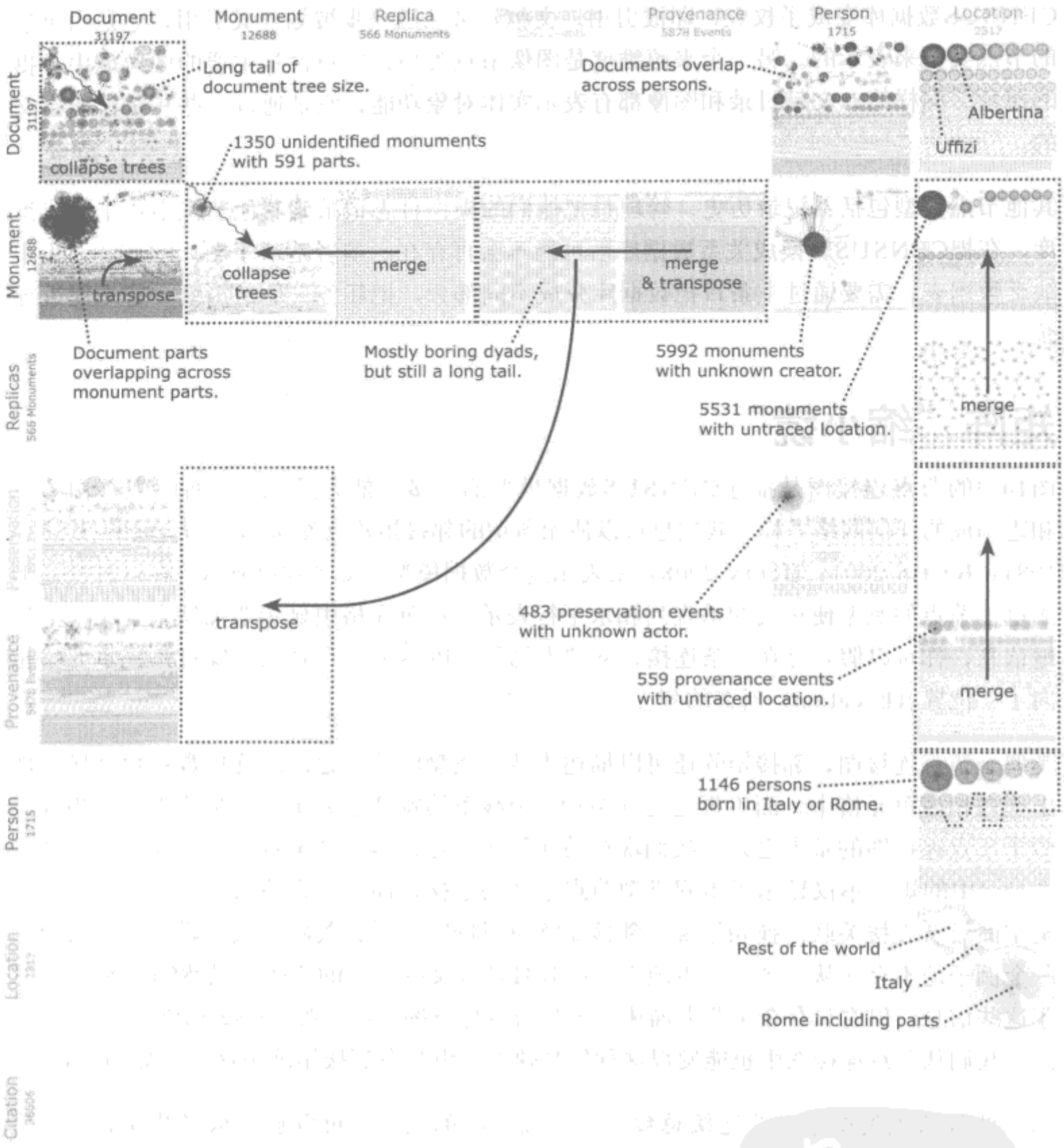
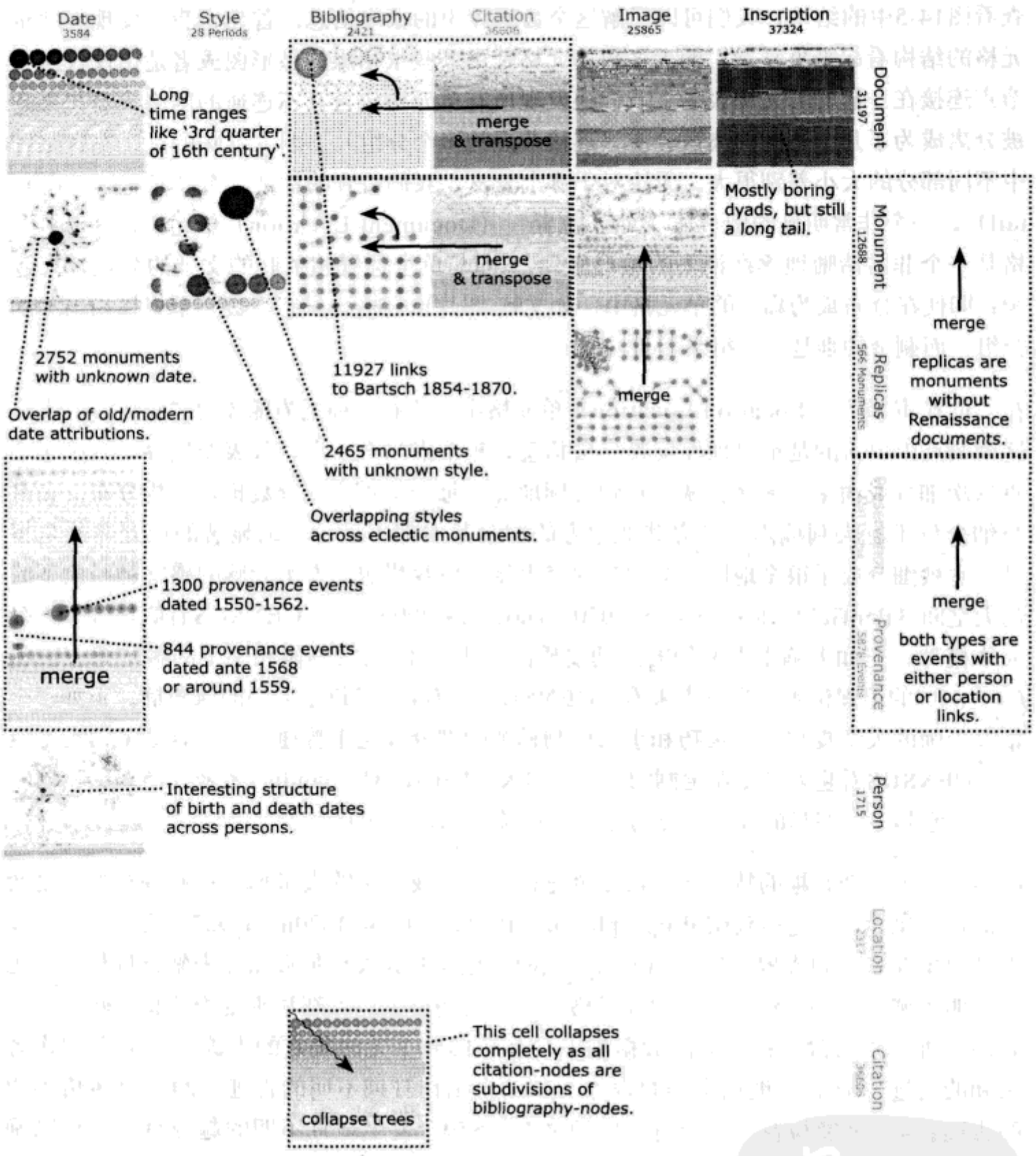


图14-5：CENSUS数据模型的邻接矩阵表示，包含节点连接图，即实际数据（见彩图118）


Date
3584
Style
28 Periods
Bibliography
2421
Citation
36606
Image
25865
Inscription
37324
Document
31197
Monument
12688
Replicas
566 Monuments
Provenance
Person
1715
Location
Citation
Long time ranges like '3rd quarter of 16th century'.
merge & transpose
merge & transpose
Mostly boring dyads, but still a long tail.
merge
replicas are monuments without Renaissance documents.
2752 monuments with unknown date.
11927 links to Bartsch 1854-1870.
merge
Overlap of old/modern date attributions.
2465 monuments with unknown style.
Overlapping styles across eclectic monuments.
1300 provenance events dated 1550-1562.
844 provenance events dated ante 1568 or around 1559.
merge
merge
both types are events with either person or location links.
Interesting structure of birth and death dates across persons.
collapse trees
This cell collapses completely as all citation-nodes are subdivisions of bibliography-nodes.

查看图14-5中的结果，我们可以了解这个数据库中的很多信息。首先，我们发现有些单元格的结构看起来更复杂，而大多数单元格是由一些很枯燥的星形图或者是由仅仅两个节点连接在一起的二元图表示。我们还发现所有单元格包含互不连通的网络，看起来是被分离成为了几个不同的离散分支（连接节点的多个分组）。耐人寻味的是，在这张图中不同部分的大小差别很大。无论对于哪个分支，我们看到的都是一个“长尾”（long tail）。一个非常明显的例子是“文献-位置”（Document-Location）单元格，这个单元格是一个非常清晰地逐渐消失的星形序列，即与单个位置相关联的文献的数量越来越少；即使在分布最为扁平的单元格中，如文献-图片单元格，只有少数几个规模相当大的分组，而剩余的则是二元组。

在“位置-位置”（Location-Location）单元格中，还有一种更为稀疏的“长尾”形式。这个节点中包含的是世界地理位置分层信息，根节点只有一个，代表全世界，各级子节点依次细分成国家、地区、城镇直到个别地点。每个位置的划分数也是异构分布。大多数细分位于意大利境内，世界其他地方的信息基本都不显示。最显著的位置显然是罗马，它被细分成了很多地区。对罗马的突出显示使我想起了人类大脑的感官神经模型的超大空间（Penfield和Rasmussen 1950；Dawkins 2005）——CENSUS看似包含了一个人体模型。正如大脑中大面积的运动皮质区是用于手-眼的协作和手上的触觉感应。从CENSUS的地理位置分层特点来看，CENSUS重点收集了位于罗马的雕塑群。正如一个钢琴大师的大脑皮层中与灵巧和手工控制相关的部分较之于普通人会占有更多的皮层空间，CENSUS看起来是专业定制的——如引入了Ulisse Aldroandi的名著（1556年和1562年），它列出了罗马的成千上万的雕塑（参考Schich 2009）。

图14-5的另一个有趣的特征在于很多单元格中的不成比例的大星形图。有些星形图是数据的自然属性，如连接到Bibliographic节点Bartsch 1854–1870的11 927个文献节点，或者是出生在意大利或罗马的1146个人。然而，绝大多数大型星形图和未知条目相关，比如不明古迹、人物、位置、日期或风格；所有这些单个节点都和确定的信息关联，这样有助于进一步收藏。在我们的数据集中，存在1350个无法确定的古迹、5992个创作者未知的古迹、5531个地点未知的古迹、2752个创作日期不明的古迹、2465个风格不明的古迹；483个参与者未知的遗迹保护事件、559个发生地点不明的起源事件。可以确定的是，允许所有这些含有未知属性的条目存在并非是一个错误；比如未知日期属性可以驳倒一个错误的文艺复兴时期的日期属性。此外，这些数字还说明了我们的知识的局限性。另一个考虑是如果我们想要分析每个单元的网络结构，我们需要绕开（或者具体化）未知节点；否则，以地理节点为例，位置未知的节点会将很多位于不同地方的未被关联的节点连接起来。

减少复杂性

如果我们回过头再去查看图14-3，我们会发现CENSUS数据库中共有31 197条文献记录，其中只有3087个节点连接到了主条目下的文献管理处。这说明了一个重要事实：数据库中的大量文献是以节点树的形式组织的。实际上只有3087个文献，包括28 110个子节点，这些子节点被用来表示页数、图形和那些图形或文本段落内的各个部分——一个直到现在仍然很少为人们所探讨的数据库的事实。古迹也存在同样的现象：只有少量的记录（特别是结构分类）可以划分成包括建筑部件、房间甚至是很小的建筑装饰上的特征。第三个例子是文献目录，它被进一步划分成了多种引用，比如在现代学术著作中的文本段落。

如图14-5中所示，引入这些子分类的结果是特定连接指向或者源自特定子节点：从部分古迹指向部分文献，而不是整个古迹指向整个文献，或者从表示装饰特征的某个列指向特定的速写图中的一部分。这些划分使得无重大信息损失的数据存储成为可能。然而，在这个配置中我们可以解决的问题通常过于具体。为了揭示更为有趣的全局性属性并回答诸如一组古迹中有多少手抄本出现（而不是总共有多少图形），或者它们在书籍中被引用的频率（而不是总共有多少引用）之类的问题，我们需要改善邻接矩阵图。该问题的一个解决方案是折叠如图14-6中的各个子分类下的文献、古迹和文献目录引用节点，图14-7a所示的是据此重新绘制而得到新的邻接矩阵图。

把文献、古迹和文献目录引用树折叠成单个节点的方法如下（参考Schich 2009）。在图14-6a中，我们首先找到原始文献树：一本包含很多页的书，被划分成多个子图形。单个连接指向多个古迹或者古迹的一部分。为了对树进行折叠，我们把书表示成单个节点，并把所有和子划分相邻的连接组合起来，如图14-6a'所示。为了保存尽可能多的信息，我们给新的节点分配权重，用来表示被折叠起来的子分类数，给连接分配另一个权值，用于表示在书中出现的连接的次数。从图形上看，权值对应节点大小和线条宽度：书的节点越大，在它的折叠树中包含的子节点数越多；线条越粗，连接越多。以实际数据为例，原始矩阵的“文献-文献”（Document-Document）单元格中的每个文献树都将会被归约成单个节点，如图14-6b/b'所示。在原始状态中看起来很繁琐或简单的矩阵单元在折叠后变得复杂而有趣，如图14-6c/c'中所示的“文献-古迹”（Document-Monument）单元格。

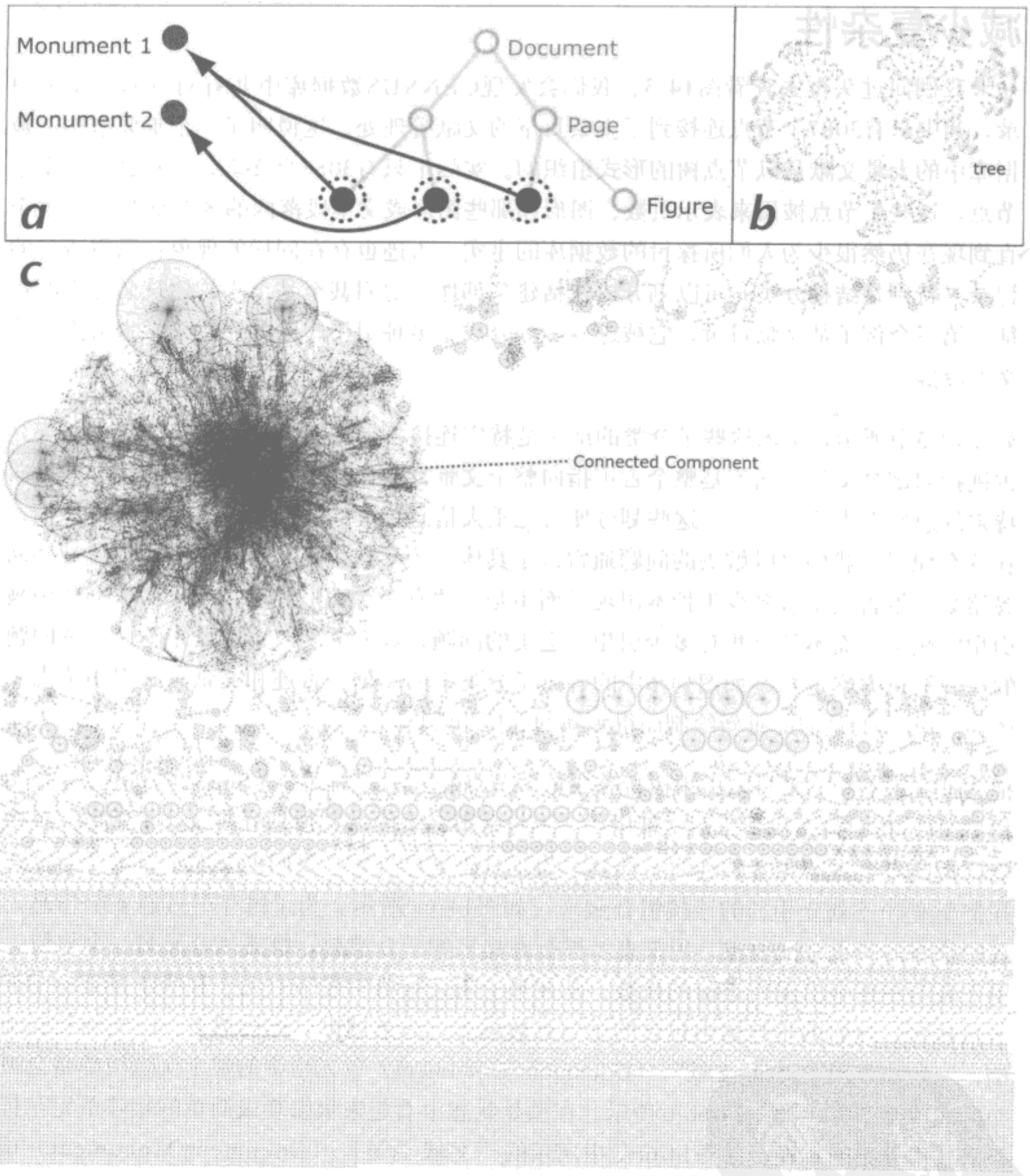
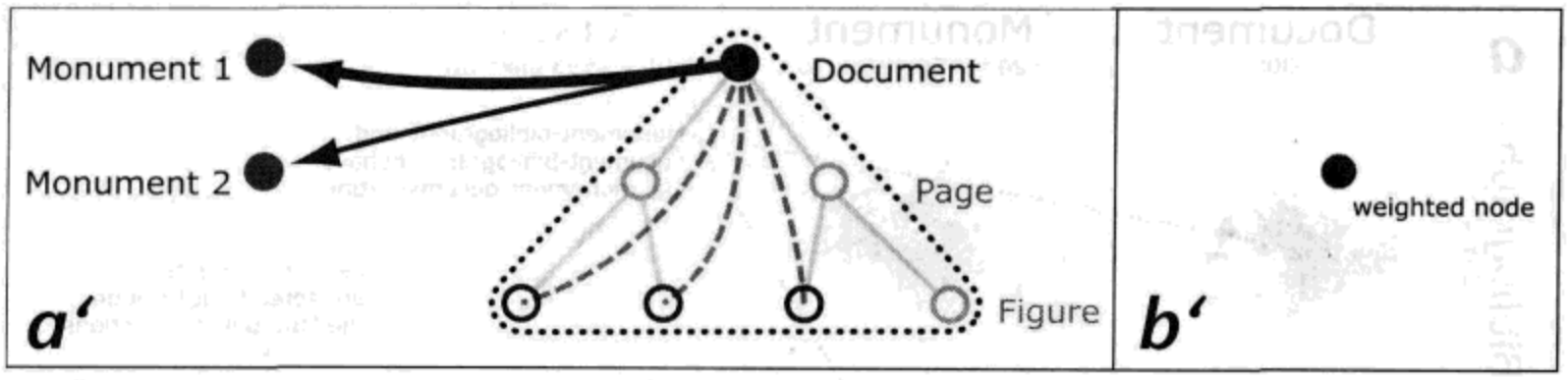


图14-6：在原始数据中对子分类目录进行折叠，出现了一些有趣的复杂的特征（见彩图119）



c'

cluster of architecture

brush-like structures

Giant Connected Component (GCC)

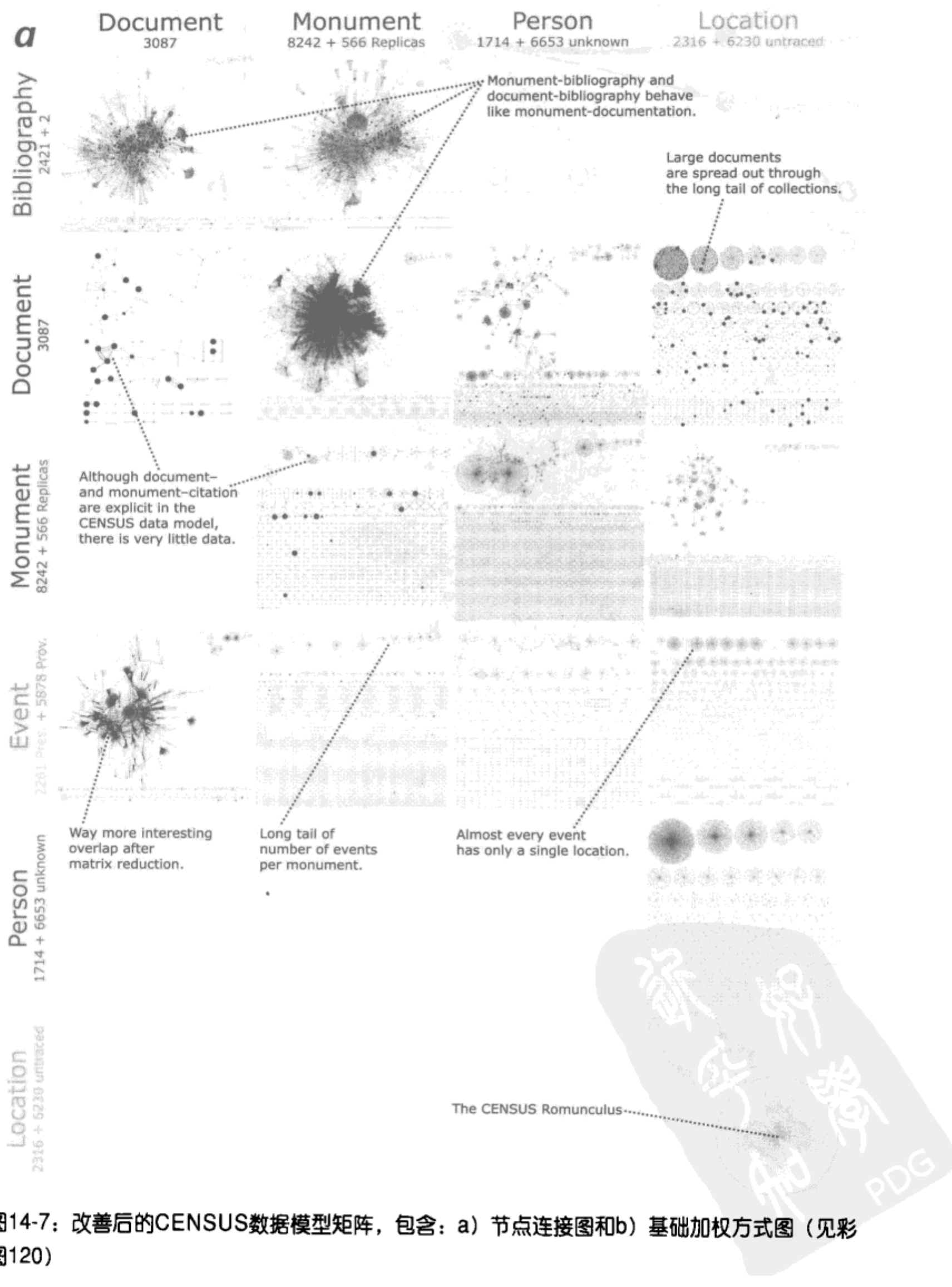
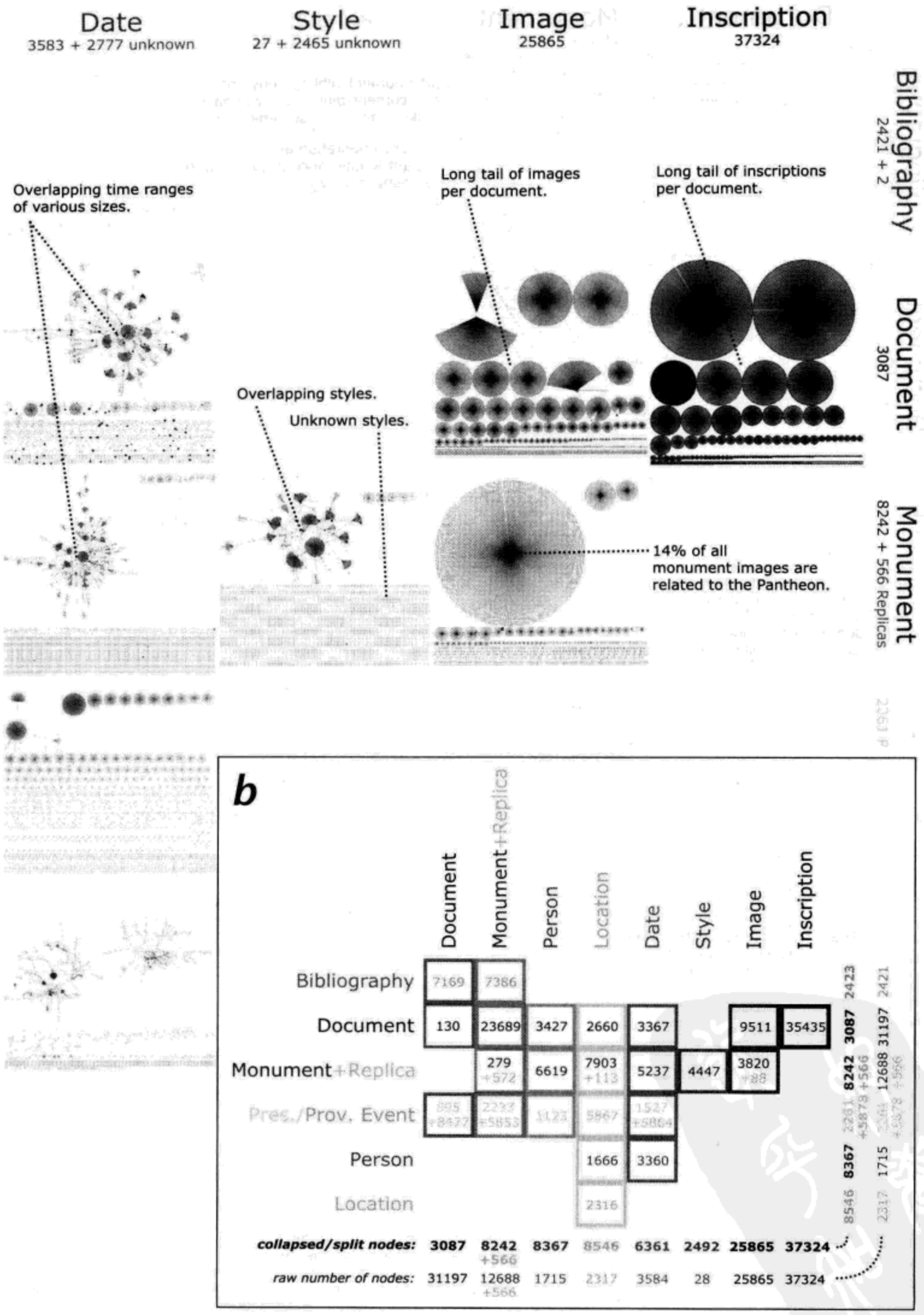


图14-7：改善后的CENSUS数据模型矩阵，包含：a）节点连接图和b）基础加权方式图（见彩图120）


Date
3583 + 2777 unknown
Style
27 + 2465 unknown
Image
25865
Inscription
37324
Bibliography
2421 + 2
Document
3087
Monument
8242 + 566 Replicas
Overlapping time ranges of various sizes.
Long tail of images per document.
Long tail of inscriptions per document.
Overlapping styles.
Unknown styles.
14% of all monument images are related to the Pantheon.
b
Document
Monument+Replica
Person
Location
Date
Style
Image
Inscription
Bibliography
Document
Monument+Replica
Pres./Prov. Event
Person
Location
collapsed/split nodes:
raw number of nodes:

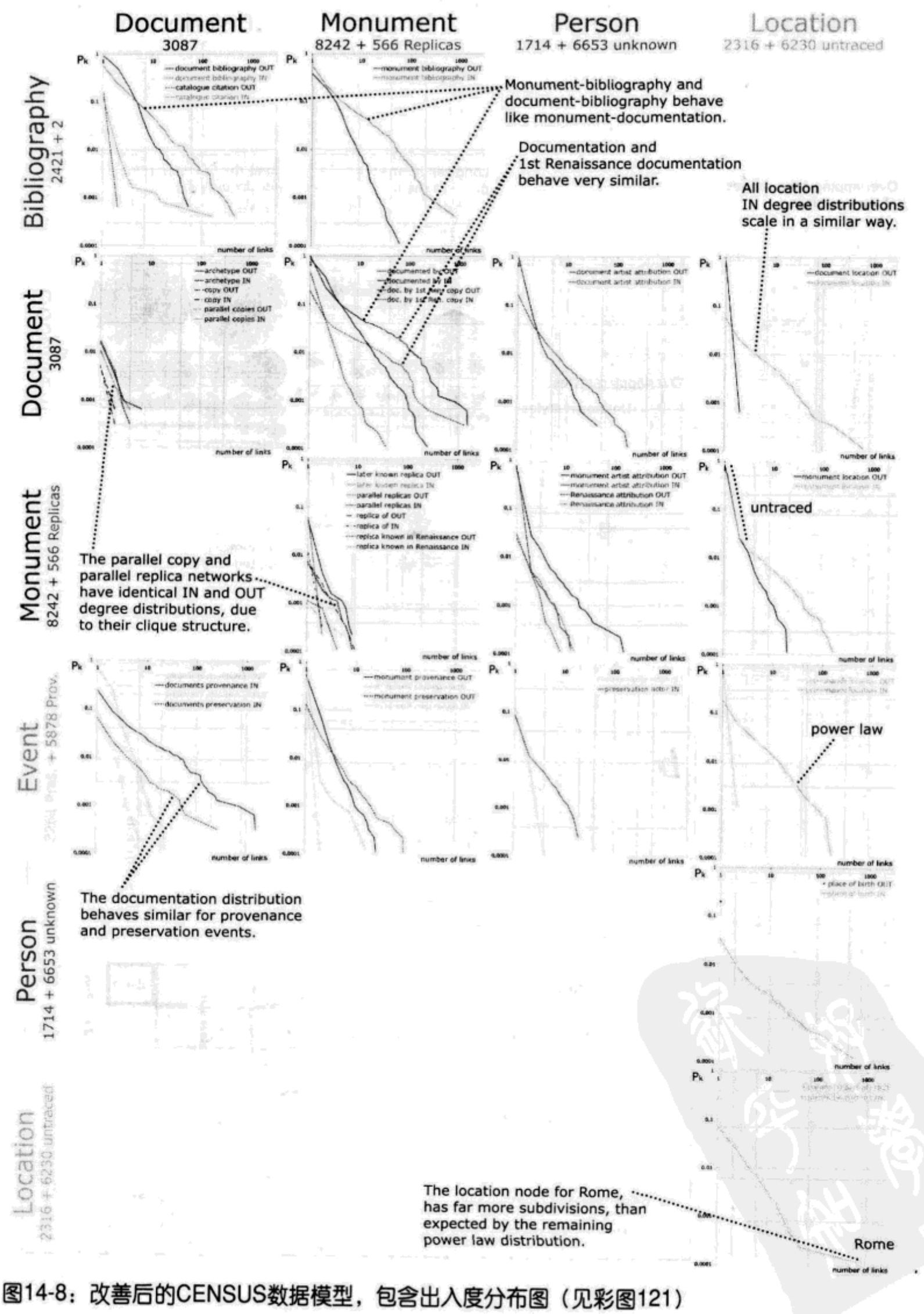


图14-8：改善后的CENSUS数据模型，包含出入度分布图（见彩图121）

Date
3583 + 2777 unknown

Style
27 + 2465 unknown

Image
25865

Inscription
37324

Bibliography
2421 + 2

Document
3087

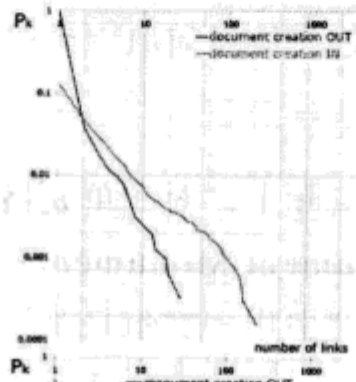
Monument
8242 + 566 Replicas

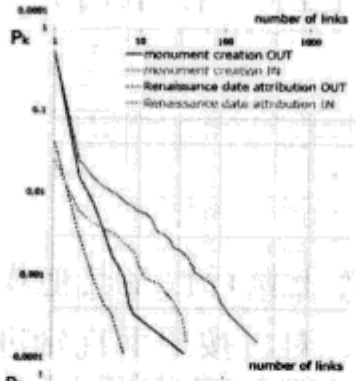
Event
+ 5678 Prov.

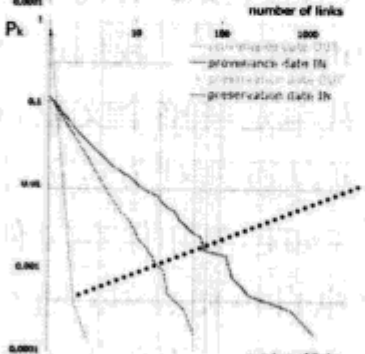
Person
1714 + 6653 unknown

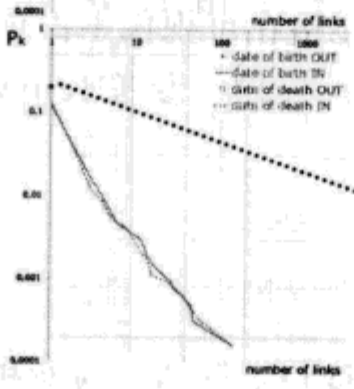
Location
2316 + 6230 untraced

Only 15% of all images
are linked to monuments;
40% are linked to documents.
45% of the images scanned in 1994
are not linked at all.

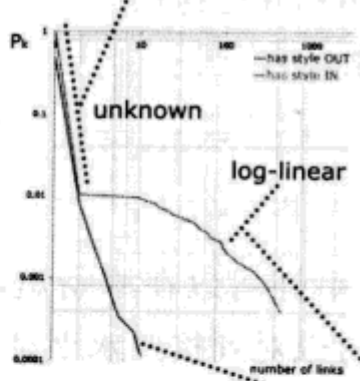


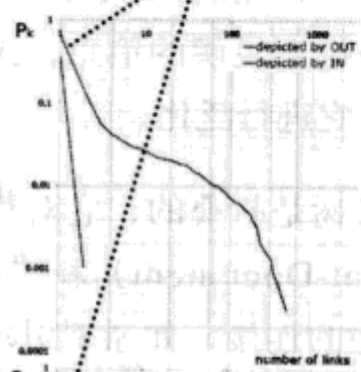


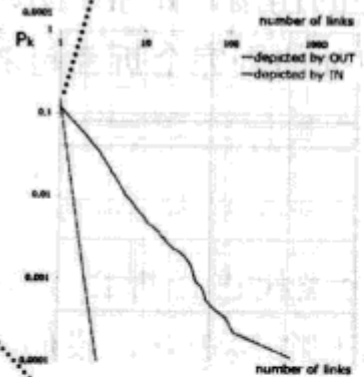


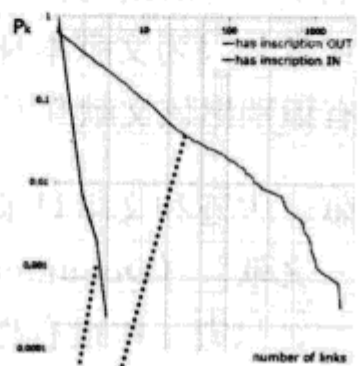


Steep probability drops
from 1 to 2 links are often
due to the many unknown
or untraced nodes, such as
unknown styles.









The tails of many IN and OUT degree
distribution pairs are asymmetric:
On one side we find something between
a power law and a log-linear distribution
vs. a steep drop on the other.

Quasi 1:n relation:
less than 0.1% of all
provenance events have
more than one date.

1:n relation indicates star
or tree network structure.
Here: only one birth date
per person, i.e. stars of
persons around single
dates.

如图14-6c'所示，改善后的单元格的最显著特征是出现了所谓的最大连通分支（Giant Connected Component，GCC），它连接了CENSUS数据库中接近90%的古迹和文献——即存在于很多复杂网络中的相变（phase transition ）现象，并诞生了很多关于信息传播的重要理论（Newman、Barabási和Watts 2006；Schich 2009）。在最大连通分支的中心，我们发现一个庞大的建筑古迹群，它连接到了很大的概览文献节点，如指南、手册和城市地图。在最大连通图周边的一个令人惊讶的特征是存在大量的连接到大文献节点的像刷子一样的结构：显然，在CENSUS中有很大一部分的古迹都连接到了同一个文献，这或者是因为文献本身缺乏足够的信息，或者是因为（也可能是其他任何原因）管理员没有识别出该文献并对它进行泛化。

因为文献、古迹和文献目录树是折叠的，它对整个矩阵都有影响。实际上，对角单元格“文献－文献”（Document-Document）和“古迹－古迹”（Monument-Monument）很少存在，只有一些很有趣的连接，如原型引用和并行拷贝关系。“引用－文献目录”（Citation-Bibliography）单元格则完全折叠起来。

矩阵操作进阶

除了绕过未知节点和对子分类树进行折叠，我们还可以在原始矩阵上执行很多其他操作，如图14-5所示。对于任何邻接矩阵，我们可以对列按照水平轴、对行按照垂直轴进行排序（或预计算），而且不会丢失任何信息（Bertin 1981；Bertin 2001）。我们还可以对单元格进行变换，如将古迹-事件单元转换到事件-古迹单元，甚至是将整个文献目录列转移到文献目录行，从而有效地翻转连接方向。最后，我们可以通过对节点创建超级类型（如事件、古迹和文献目录）的方式对相同的节点类型（如起源和保存事件（Provenance and Preservation Events）、古迹和副本、或者是文献目录和引用）进行归并。这种归并方式可以减少矩阵中列和行的数目，允许每个单元格在可视化中占据更大的空间。此外，矩阵可视化方面的资料还介绍了很多其他可能的操作（参考Henry 2008）。

改善后的矩阵

图14-7a和b显示了目前讨论的优化操作的最终结果。整个邻接矩阵变得更简洁、清晰和信息丰富。我们可以很容易地看到CENSUS数据在数据模型内是如何分布的：古迹－和文献目录－显然类似于古迹－文献，展现的数据信息量都很大。另一方面，对于文献－文献和古迹－古迹依赖关系（如引用），甚至是在数据模型中所显示的各个连接，则几乎没有任何数据。显然，数据收集工作流没有以正确的方式来系统性地收集这类信息。

与在原始矩阵中状况一样，我们在每个改善后的单元格中都发现了分支大小的“长尾”

现象。有些单元还是主要包含星形图，这对于每个古迹涉及的事件数、每个文献/古迹的图像，每个文献的雕刻或者每个位置发生的事件，都是适用的。对于文献-位置单元的一个有趣之处是我们发现了大的文献节点跨越了所有集合，从佛罗伦萨的Uffizi到每个包含单个手册的个人收集。其他单元表现出更重叠的结构，如在文献和古迹中的重叠日期（或者时间范围）、或者是从风格到电子古迹如《the Arch of Constantine》，它们一起显示了罗马帝国时期不同的浮雕。当然，古迹－文献和相关的文献目录包含最复杂的重叠，因为该单元是CENSUS项目的重心。

数据规模扩大

网络领域的读者可能会指出在矩阵中使用节点连接图，如图14-7a所示，对于比CENSUS数据库大一个量级的数据集是不可行的，更不用说庞大的语义Web。确实，这是一个问题，因此问题是如何对以上方法进行扩展，使它可以适用于真正的大型数据库。一个解决方案是使用维度分布图，甚至更复杂的数值网络衡量方式，在数据模型中获取关于实际数据的想法。

如图14-8所示，我们为矩阵单元中的每个连接类型描绘了一张累积入度和出度分布曲线图（Broder等 2000；Newman 2005）。由于每条连接相当于源节点类型的出度、目的节点类型的入度，对于单元中的每条连接都有两个分部。每条曲线的*x*轴表示连接数*k*；*y*轴表示累积概率P(*k*)，每个节点至少包含*k*条连接。注意分布曲线是以双重对数尺度描绘的，这意味着每个刻度是表示在*y*轴上从100%到0.01%快速衰减，而在*x*轴上是从1到3000的快速增长。（在规则线性投影中，每个分部的倾斜度很高，我们无法找出任何有趣的内容。）令人吃惊的是，在这些曲线图中都不存在我们所期望的高斯钟形曲线，如人的平均高度。相反，我们发现其分布呈“长尾”分布，从最初美丽的幂等律曲线一直到对数-线性曲线，中间是一些较混杂的分布曲线。

几乎所有的入度和出度对看起来都是不对称的。举个例子，“出生日期”和“人物”是以1：*n*的方式连接的，而n值的变化很大。这也不奇怪，因为该信息区并不受大众不同的观点所影响，正如人脸图像数据库那样，它的重点在于人物而不是事物。其他区域如保存事件发生的位置，呈现出接近精确的1:*n*约束关系，因为一个事件很少但不是不可能出现在多个地理位置。在真正的*n*:*n*关系中发现了最有趣的不对称性，如中心古迹－文献连接，我们在连接两边都发现了不同的倾斜分布。目前，应该如何充分解释该不对称性还不完全明确；但是，通过比较很多数据源，显然这些不同的分布是由很多因素导致的，如对源数据的物理限制和可访问性，以及管理员的关注和其他认知限制。

CENSUS中所发现的唯一对称关系是 “文献－文献”和“古迹－古迹”单元间的多份拷贝和多份副本连接。理想情况下，入度和出度分布应该是完全一致的，因为相关节点会

全部连接到所谓的“簇集”（clique）中。实际上，入度和出度这两种连接类型都随着进一步取向分布的“尾巴”而变得更加不对称，因为很难维护大的簇集。正如我在2003年对CENSUS项目提出的建议，连接到包含*n*条连接的未知“文献”比*n*份拷贝之间通过手工生成*n* (n−1)*个连接要更有意义。

同样，我们在图14-7中观察到的一些关系，如“古迹－文献目录”和“古迹－文献”之间呈现等价关系，在图14-8中得到进一步确证（Schich 和 Barabási 2009）。这些单元之间不仅呈现出很明显的相似关系，而且在单个单元格的不同连接类型中也发现了同样的功能对等关系。一个很有说服力的例子是在“文献－古迹”单元，一般的文献和文艺复兴时期的第一个文献之间的分布曲线几乎是水平的；而对于“事件-文献”单元，起源和保存文献也呈现类似的曲线。在“位置”这一列，其入度在所有相关的单元中都呈现非常相似的分布曲线。观察到的两个例外是在每个位置包含一个或两个古迹的概率曲线急剧下降（因为有很多位置不明的古迹），而“位置－位置”单元的“长尾”分布的“尾巴”则不断上升（由于人造物现象引起）。

最后一点，我们可以观察到所有的曲线都是包含所有节点类型的一部分节点，这是所有以单个连接类型组成的单个网络的内在特征。查看每条曲线穿过*y*轴的值可以说明很多信息，如少于15%的图像是连接到古迹，而少于40%的是连接到文献。反之，CENSUS项目的出版合作伙伴在1994年扫描的共24 000张图片中，我们可以确定至少有45%的图片在2005年还没有添加到数据库中。

深层次应用

本章介绍的可视化可以作为各种活动的起点。除了资助人和项目负责人所做的具体项目目标的评估，进一步的研究领域包括识别有趣的研究课题：矩阵中的每个单元都可以进行广泛研究，正如我的博士论文主要研究古迹文献和可视化文献引用（Schich 2009）。很多单元都展示出有趣的交互，可以结合在这种研究中。举个例子，为了对在时间和空间上涉及很多事件的物体和人物建立轨迹（González、Hidalgo和Barabási 2008），或者为了研究网络交互效果（Leicht和D'Souza 2009）。最后，可以使用很多等价可视化来比较已经使用了相似的数据模型的整个数据库，如Winckelmann语料库和CENSUS数据库，或者可以映射到相同标准如CIDOC CRM的数据库。

此外，如果不按照本章所述的方法对数据库进行分割，在类似的可视化中结合不同的网络也是很有趣的。在可能的网络多元世界中也可以很容易找到这些结合（举个例子，引用、多名作者共有著作权、社会科学中的图像标记数据库或生物学中的基因转录、蛋白质相互作用和基因疾病数据库）。

通过对文献、古迹和文献目录树进行折叠，粗粒度显示也可以通过很多其他方式实现；

举个例子，基于特定子树折叠或者是更复杂的方法如“区块建模”（blockmodelling）（Wassermann和Faust 1999）或者社区发现（Lancichinetti和Fortunato 2009；Ahn、Bagrow和Lehmann 2009），切实解决如何真正定义网络中的节点和连接（Butts 2009）。

最后，本章给出的矩阵和节点－连接图组合可以进一步扩展；比如在数据模型的相关单元中替换节点－连接/矩阵组合（Henry、Fekete和McGuffin 2007）或者可扩展的图像矩阵（Schich、Lehmann和Park 2008）。

结束语

正如本章所述，丰富完善后的数据模型矩阵对于数据库项目的评估是非常有用的，它揭秘了很多非直观的数据属性，这些属性难以简单地通过数据库或者常用的质量指标来捕捉。由于数据以关联数据（Linked Data）、RDF图和关系表导出的形式变得更易于访问，项目资助人或负责人可以应用以上提出的方法，以几乎自动化的过程在很短的时间内实现。

本章所示的可视化是第一个呈现了整个CENSUS数据库的大图，我们可以从中看到最初的数据模型定义和在收集到的数据中的新兴的复杂的数据结构。通过查看这些可视化，我们发现项目描述中给出的很多数字都是不完整甚至是误导人的。有些新的数据可能比最初给出的值小，我们从分析中汲取的一点教训是：有时少即是多——多了就不同了（Anderson 1972）。

致谢

感谢在威尼斯的NetSci09会议和Mountain View的SciFoo09会议的听众，以及波士顿东北大学BarabásiLab实验室的同事提出的有见地的反馈意见。感谢慕尼黑Stiftung Archäologie的Ralf Biering和Vinzenz Brinkmann提供数据和德国研究基金（DFG）对我的研究的资助。要查看关于CENSUS数据库的完整的文献目录，请查看Schich 2009，第13页，注解20-25。

参考文献

1. The presented visualizations are available online in large resolution at *http://revealingmatrices.schich.info*.

2. Ahn, Yong-Yeol, James P. Bagrow, and Sune Lehmann. 2009. “Link communities reveal multi-scale complexity in networks.” *http://arxiv.org/abs/0903.3178v2*.

3. Aldroandi, Ulisse. 1556/1562. "Appresso tutte le statue antiche, che in Roma in diversi luoghi, e case particolari si veggono, raccolte e descritte (...) in questa quarta impressione ricorretta." *Le antichità della città di Roma*. Ed. Lucio Mauro. Venice.

4. Anderson, Chris. 2006. *The Long Tail*. New York: Hyperion. *http://www.thelongtail.com*.

5. Anderson, P.W. 1972. "More is different." *Science* 177, no. 4047: 393–396.

6. Bartsch, Adam. 1854–1870. *Le Peintre-Graveur, nouvelle edition*. v. 1–21. Leipzig: Barth.

7. Bartsch, Tatjana. 2008. "Distinctae per locos schedulae non agglutinatae" –Das Census-Datenmodell und seine Vorgänger. *Pegasus* 10: 223–260.

8. Bertin, Jaques. 1981. *Graphics and Graphic Information Processing*. Berlin: de Gruyter.

9. Bertin, Jacques. 2001. "Matrix theory of graphics." *Information Design Journal* 10, no. 1:5–19. doi: 10.1075/idj.10.1.04ber.

10. Bizer, Christian, Tom Heath, and Tim Berners-Lee. 2009. "Linked data—The story so Far." *International Journal on Semantic Web & Information Systems* 5, no. 3: 1–22.

11. Broder, Andrei, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. 2000. "Graph structure in the Web." *Computer Networks* 33, no. 1–6: 309–319. doi:10.1016/j.physletb.2003.10.071.

12. Butts, Carter. 2009. "Revisiting the foundations of network analysis." *Science* 325, no.5939: 414–416. doi: 10.1126/science.1171022.

13. CENSUS. 1997–2005. *Census of Antique Works of Art and Architecture Known in the Renaissance*. Ed. A. Nesselrath. Munich: Verlag Biering & Brinkmann/Stiftung Archäologie. *http://www.dyabola.de*.

14. CENSUS BBAW. 2006. *Census of Antique Works of Art and Architecture Known in the Renaissance*. Ed. Berlin-Brandenburgische Akademie der Wissenschaften and Humboldt-Universität zu Berlin. *http://www.census.de*.

15. Chakrabarti, Suomen. 2003. *Mining the Web: Discovering Knowledge from Hypertext Data*. San Francisco, CA: Morgan Kaufmann.

16. Chen, Peter P.S. 1976. "The entity-relationship model—Toward a unified view of data." *ACM Transactions on Database Systems* 1, no.1: 1–36. doi: 10.1145/320434.320440.

17. Chua, Leon O. 2005. "Local activity is the origin of complexity." *International Journal of Bifurcation and Chaos* 15: 3435–3456. doi: 10.1142/S0218127405014337.

18. Crofts, Nick, Martin Doerr, Tony Gill, Stephen Stead, and Matthew Stiff, eds. 2006. *Definition of the CIDOC Conceptual Reference Model (CIDOC-CRM), Version 4.2.1. http://www.cidoc-crm.org/docs/cidoc_crm_version_4.2.1.pdf.*

19. Dawkins, Richard. 2005. *The Ancestor's Tale. A Pilgrimage to the Dawn of Life.* London: Phoenix.

20. DBpedia. 2009. *DBpedia*. Sören Auer, Christian Bizer, and Kingsley Idehen, admins. Leipzig: Universität Leipizig; Berlin: Freie Universität Berlin; Burlington, MA: OpenLink Software. *http://dbpedia.org*.

21. Doreian, P., V. Batagelj, and A. Ferligoj. 2005. *Generalized Blockmodeling (Structural Analysis in the Social Sciences)*. Cambridge: Cambridge University Press.

22. Flybase. 2008. Rachel Drysdale and the FlyBase Consortium. FlyBase. *Drosophila*: 45–59. doi: 10.1007/978-1-59745-583-1_3. See also *http://flybase.org/static_pages/docs/release_notes.html*.

23. Freebase. 2009. *Freebase*. San Francisco, CA: Metaweb Technologies. *http://www.freebase.com*. For data dumps, see *http://download.freebase.com/datadumps/*.

24. Garner, Ralph. 1963. "A computer-oriented graph theoretic analysis of citation index structures." In *Three Drexel Information Science Research Studies*, ed. Barbara Flood. Philadelphia, PA: Drexel Press.

25. González, Marta C., César A. Hidalgo, and Albert-László Barabási. 2008. "Understanding individual human mobility patterns." *Nature* 453: 779–782. doi: 10.1038/nature06958.

26. Henry, Nathalie, J-D. Fekete, and M. McGuffin. 2007. "NodeTrix: A hybrid visualization of social networks." *IEEE Transactions on Visualization and Computer Graphics* 13, no. 6: 1302–1309.

27. Henry, Nathalie. 2008. "Exploring large social networks with matrix-based representations." PhD diss., Cotutelle Université Paris-Sud and University of Sydney. *http://research.microsoft.com/en-us/um/people/nath/docs/Henry_thesis_oct08.pdf*.

28. Lagoze, Carl, Herbert Van de Sompel, Michael Nelson, and Siemeon Warner. 2008.The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 2002-06-14. *http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm.*

29. Lancichinetti, A., and S. Fortunato. 2009. "Community detection algorithms: A comparative analysis." *Physical Review E* 80, no. 5, id. 056117. doi: 10.1103/PhysRevE.80.056117.

30. Leicht, E.A., and Raissa M. D' Souza. 2009. "Percolation on interacting networks." *arXiv* 0907.0894v1, *http://arxiv.org/abs/0907.0894v1*.

31. Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. "Network motifs: Simple building blocks of complex networks." *Science* 298, no. 5594: 824–827.

32. Nesselrath, Arnold. 1993. "Die Erstellung einer wissenschaftlichen Datenbank zum Nachleben der Antike: Der Census of Ancient Works of Art Known to the Renaissance." Habilitation thesis, Universität Mainz. Available at the CENSUS office at HU-Berlin.

33. Newman, Mark E.J. 2005. "Power laws, Pareto distributions and Zipf's law." *Contemporary Physics* 46: 323–351. doi:10.1080/00107510500052444.

34. Newman, Mark E.J., Albert-László Barabási, and Duncan J. Watts, eds. 2006. *The Structure and Dynamics of Networks*. Princeton, NJ: Princeton University Press.

35. Penfield, W., and T. Rasmussen. 1950. *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*. New York: Macmillan.

36. Phosphosite. 2003–2007. *PhosphoSitePlus*™, A Protein Modification Resource. Danvers, MA: Cell Signaling Technology. *http://www.phosphosite.org*.

37. Pietriga, Emmanuel, Christian Bizer, David Karger, and Ryan Lee. 2006. "Fresnel—A browser-independent presentation vocabulary for RDF." In *The Semantic Web—ISWC 2006*, vol. 4273, Chapter 12. Eds. I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. M. Aroyo. Berlin, Heidelberg: Springer Berlin Heidelberg.

38. Saxl, Fritz. 1957. "Continuity and variation in the meaning of images." Lecture at Reading University, October 1947. In Lectures. London: Warburg Institute.

39. Schich, Maximilian. 2009. "Rezeption und Tradierung als komplexes Netzwerk. Der CENSUS und visuelle Dokumente zu den Thermen in Rom." Ph.D. diss., Humboldt-

Universität zu Berlin. Munich: Verlag Biering & Brinkmann. urn:nbn:de:bsz:16-artdok-7002.

40. Schich, Maximilian, and Albert-László Barabási. 2009. "Human activity—from the Renaissance to the 21st century." In *Cultures of Change. Social Atoms and Electronic Lives. Exhibition Catalogue: Arts Santa Mònica, Barcelona, 11 December 2009 to 28 February 2010*. Gennaro Ascione, Cinta Massip, and Josep Perelló eds. Barcelona: Arts Santa Monica. urn:nbn:de:bsz:16-artdok-9582.

41. Schich, Maximilian, and Sybille Ebert-Schifferer. 2009. "Bildkonstruktionen bei Annibale Carracci und Caravaggio: Analyse von kunstwissenschaftlichen Datenbanken mit Hilfe skalierbarer Bildmatrizen." Project report. Rome: Bibliotheca Hertziana (Max-Planck-Institute for Art History). urn:nbn:de:bsz:16-artdok-7121.

42. Schich, Maximilian, César Hidalgo, Sune Lehmann, and Juyong Park. 2009. "The network of subject co-popularity in classical archaeology." urn:nbn:de:bsz:16-artdok-7151.

43. Schich, Maximilian, Sune Lehmann, and Juyong Park. 2008. "Dissecting the canon: Visual subject co-popularity networks in art research." 5th European Conference on Complex Systems, Jerusalem (online material). urn:nbn:de:bsz:16-artdok-7111. *Science*. 2009. Special Issue on Complex Systems and Networks. *Science* 325, no. 5939: 357–504. *http://www.sciencemag.org/content/vol325/issue5939/#special-issue*.

44. Segaran, Toby. 2009. "Connecting data." In *Beautiful Data*. Sebastopol, CA: O' Reilly Media.

45. Shannon, Paul, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. "Cytoscape: A software environment for integrated models of biomolecular interaction networks." *Genome Research* 13, no. 11: 2498–2504. doi: 10.1101/gr.1239303. See also *http://www.cytoscape.org*.

46. Sullivan, Danny. 2005. "Search engine sizes." *Search Engine Watch. http://searchenginewatch.com/2156481*.

47. Wassermann, Stanley, and Katherine Faust. 1999. *Social Network Analysis: Methods and Applications, Fourth Edition*. Cambridge: Cambridge University Press.

48. Wikipedia. "Wikipedia: Size comparisons." *http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons*.

49. Winckelmann Corpus. 2000. *Corpus der antiken Denkmäler, die J.J. Winckelmann und seine Zeit kannten*. Winckelmann-Gesellschaft Stendal, ed. DVD and online database. Munich: Verlag Biering & Brinkmann/Stiftung Archäologie.

第15章

1994年：基于《纽约时报》上的文章搜索API的数据探索

Jer Thorp

2009年2月份，《纽约时报》宣布将它28年的数据向公众开放——新闻故事、电影评论、讣告和政治统计，全部都可以免费访问。面对如此庞大的信息量，即约260万篇文章，我们需要面临着3个重要问题。如何获取我们需要的数据？如何处理这些数据？以及可能是最重要的，为什么要做这件事？本章将尝试回答以上这些问题。我们将了解如何使用《纽约时报》文章搜索API（NYTimes Article Search API）（*http://developer.nytimes.com/docs/article_search_api*）来访问信息，查看一些实际的可视化例子，探讨在数据开放时代那扇面向艺术家、企业家、设计师和社会科学家的探索之门是如何开启的。

获取数据：文章搜索API

“API”是众多3个字母缩写词之一，3字母缩写词只包含3个字母，直接包含的含义很少，即使知道API的全称：应用程序编程接口（application programming interface），仍然难以确定这个缩写的含义。这个缩写相当通用，在软件开发领域中被广泛应用，通常是为了使一个软件和另一个软件可以通信。如果我们把数据库想象成存储信息的实体仓库，那么API就是运输和接收部门，而且对外开放。

总之，通过API进行交互非常简单。向API发送一条请求（该请求可以非常简单，也可以非常复杂），该API会给我们发回一串格式化的信息。不同API之间通信的语法以及向我

们发回的响应信息的格式有很大区别。有些API的功能非常少，而有些API功能则很强大，包含很多有用的功能。幸运的是，在我们看来，《纽约时报》的文章搜索API是功能最强大、结构最良好的API之一。

那么，我们能够向API发送什么请求呢？通过一些简单的请求，API可以回答以下任何一个问题，而且数量上几乎没有限制：

- 1982年发表了多少文章？
- 关于欺诈的文章中，哪个企业组织被提及最多？
- 在1991年关于时尚的文章中，“超色”（hypercolor）被提及多少次？

我们先来尝试一个简单的问题：在1994年，有多少文章提到O.J. Simpson[译注1]？可以通过几种不同的方法向API发送这个问题，它们都需要发送一个指向特定URL的HTTP请求，在该请求中可以加上一些可选的参数。以下是最简单的请求：

http://api.nytimes.com/svc/search/v1/article?query=O.J.+Simpson

该请求会给我们返回数据库中包含字符串“O.J. Simpson”的所有的文章（数据库中存储了从1981年至今的所有文章）。为了限制为1994年的文章，我们给该查询增加了一些额外参数：

http://api.nytimes.com/svc/search/v1/article?query=O.J.+Simpson&begin_date=19940101&end_date=19950101

最后，该API会记录访问者的信息并确保没有用户超过发布的最大限制值。因此，我们每次调用API时，都必须在查询请求中加上一串API密钥，该密钥是《纽约时报》系统为每个用户生成的一串唯一的字符串[注1]。

http://api.nytimes.com/svc/search/v1/article?query=O.J.+Simpson&begin_date=19940101&end_date=19950101&api-key=1af81d##########################:##:########

如果你继续往下操作，把该请求粘贴到浏览器地址栏（用你自己的API密钥取代#内容），你将会得到一些请求结果；查看数据源，得到API返回的真正数据。返回给我们的数据是以JSON格式封装，我们将在本章的后面详细介绍该格式。在返回的数据块的下方，我们能够找到以上问题的答案：2218。

译注1：OJ Simpson是橄榄球兼电影明星，因谋杀妻子案审判，在美国引起轰动。后面会介绍更多。

注1：在nytimes.com上登录你的账户，访问*http://developer.nytimes.com*，点击“Getting Started”标题下的“Request an API key”。

我们将把这些请求封装成一个多功能的包，这些请求是本章的基础。对文章搜索API的任何请求都是通过这种通用的方式进行构建的，如图15-1所示：

基础URL + 查询 + 维度 + 额外参数 + API密钥

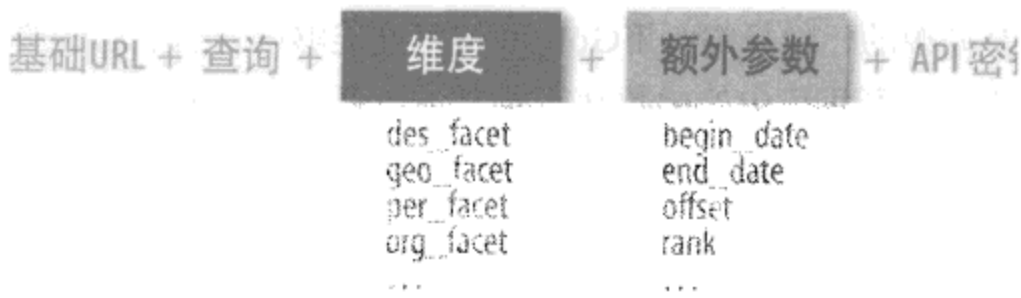


图15-1：《纽约时报》文章搜索API请求总是通过几个相同的关键项来构建的

其中有些项（查询，API密钥）是必需的，而其他一些项则是可选的（额外参数，维度）。然而，基础结构一直没有改变，基本方法也一样保持不变：向API发送一个请求，得到一个请求结果。但是，我们真正希望实现的是可以向API发送很多请求，得到很多请求结果。为了实现这一点，我们需要一个更好的系统，而不是简单地在Web浏览器地址栏中执行拷贝和粘贴。

管理数据：使用Processing编程语言

在20世纪90年代，美国艺术家Mark Lombardi创作了一系列非常复杂的绘画作品（他称之为“叙事式结构”（narrative structures）），这些作品揭露了涉及政治和金融诈骗的人们和企业组织之间的关系。Lombardi认真地梳理报纸文章和杂志，手工记录他的发现。他既没有一个可以发送请求的API，也没有任何数据库或软件来存储其结果。相反地，Lombardi积累了14 000多张索引卡片，把所有的问题和答案都记录到这些卡片上，并根据这些卡片描绘其历史图表（见图10-1）。

除非你碰巧有几千张索引卡片和几周的业余时间来做这件事，否则我们就需要找出一种更快捷的方式来管理所有的问题和答案。借助一台计算机，解决这个问题的方式会有很多种，有很多不同的软件工具和编程语言都可以实现该任务。我使用了一种称为Processing的编程语言来处理数据，在本章的例子中也将使用该语言。Processing可以免费下载，而且使用相对简单。本章将假定你已经下载并安装了Processing（如果你需要帮助，请访问Processing的官方网站：*http://www.processing.org*）。

在最后一节，我将演示如何使用《纽约时报》的文章搜索API发送请求并得到JSON格式的响应结果。我们将使用Processing来管理请求，解析并存储响应结果，然后把结果显示在屏幕上。这个过程最复杂的部分是处理返回的JSON格式的请求结果。我将使用以前写过的一些简单的Processing代码，而不是长篇大论地告诉你应该如何构建自己的引

擎，这样可以使这个说明过程变得更简单。我把用于处理文章搜索API的很多关键函数功能封装成了一个库，你可以从*http://www.blprnt.com/libraries/nytimes*下载。

安装Processing库很简单，只需要简单地把解压后的文件夹拖到绘图本的库所在的目录（同样，如果需要帮助，请访问*http://www.processing.org*）。如果你想了解这些库的内部实现，该项目是开源的，你只需要用Google搜索就能够得到需要的链接。但是，目前你需要了解的是你可以利用库中的函数功能来做一些有意义的事情。首先，我们一起来看看如何使用这个库向API发送一个前面提到的关于O. J.问题的请求。

首先，我们通过导航条中“绘图导入（Sketch Import）库”的下拉菜单导入“《纽约时报》的文章搜索（NYTArticleSearch）库”。然后，设置画布大小，并把背景设置成鲜亮的白色：

```
import blprnt.nytimes.*;
size(800,350);
background(255);
```

接着，我们开始通过API密钥对库进行初始化：

```
TimesEngine.init(this, "YOUR-API-KEY-GOES-HERE");
```

下一步，我们将创建TimesArticleSearch对象来管理请求（查询）和结果（回复）：

```
TimesArticleSearch mySearch = new TimesArticleSearch();
```

这个简单的对象可以帮助我们发出任何需要使用文章搜索API的请求。我们先来发送一个类似于之前的关于1994年的问题的查询，这次把结果限制在1994年和1995年：

```
mySearch.addQueries("O.J.+Simpson");
mySearch.addExtra("begin_date","19940101");
mySearch.addExtra("end_date","19960101");
TimesArticleSearchResult r = mySearch.doSearch();
println("RESULTS ABOUT O.J.:" + r.total);
```

这看起来似乎比我们的第一个例子稍复杂些，在第一个例子中，我们发送的只是一个http请求，但是在这个例子中，我们不需要处理JSON格式的数据，而且有充分的自由来定制搜索。文章搜索API为我们提供很多对搜索请求进行结构化的选项，允许我们实现非常具体或者非常通用的请求。

先考虑一下搜索。我们向API发送请求，查找在1994年或1995年发表的、包含字符串“O. J. Simpson”的所有文章。那么，对于包含Orenthal James Simpson的文章是否会被包含在结果之中呢？或者只包含O.J.呢？或者包含“The Juice”的呢？文章搜索API的一个强大之处在于它和《纽约时报》的编辑机构关联在一起。当《纽约时报》发表了一篇文章，该文章会通过一组编辑信息来索引。该信息是由人们手工添加和规范化

的，API可以访问该信息并使搜索更有效。对于该例子，我们不需要查看短语“O.J. Simpson”，而是可以通过合适的维度标签来找到和“O.J. Simpson”匹配的结果（即“SIMPSON, O J”）。

编辑人员会把该维度添加到任何提到或引用过O.J.的文章中，不论文章正文使用了什么名字。因此，搜索如下：

```
import blprnt.nytimes.*;
size(800,350);
background(255);

TimesEngine.init(this, "YOUR-API-KEY-GOES-HERE");

TimesArticleSearch mySearch = new TimesArticleSearch("YOUR-API-KEY-GOES-HERE");

mySearch.addFacetQueries("per_facet","SIMPSON, O J");
mySearch.addExtra("begin_date","19940101");
mySearch.addExtra("end_date","19960101");

TimesArticleSearchResult r = mySearch.doSearch();
println("RESULTS ABOUT O.J.:" + r.total);
```

使用维度的唯一棘手之处在于如何找出可用的维度以及他们的标准名字是什么。访问该信息的一个简单的方法是使用《纽约时报》的API请求工具，在*http://prototype.nytimes.com/gst/apitool/index.html*可以获取。该工具可以帮助你测试所有的搜索查询并查看相关结果，这些都不需要编写任何繁琐的代码或者获取API密钥。为了获得关于“O.J.”的合适的维度，我们可以在搜索查询（Search Query）域中输入“O.J Simpson”，在维度查询(Facet Query)域中输入“per_facet”，结果如图15-2所示。

当然，在1994~1995年发生的事情远远不止“白色吉普车和带有血迹的手套”一案[译注2]。使用API工具，我们可以收集在那个时期的一些其他事件的准确信息，比如南非种族隔离政策的结束（geo_facet=SOUTH AFRICA），以及卢旺达的种族屠杀（geo_facet=RWANDA）。我们可以为每个搜索构建新的“《纽约时报》文章搜索”（TimesArticleSearch）对象，或者每次清空维度查询，重新使用相同的对象。第二种方式更合理，因此我们可以尝试一下。

```
import blprnt.nytimes.*;
size(800,350);
background(255);

TimesEngine.init(this, "YOUR-API-KEY-GOES-HERE");
TimesArticleSearch mySearch = new TimesArticleSearch();
```

译注2：这里指的是橄榄球兼电影明星O.J Simpson涉嫌谋杀前妻及其男友的案件，该案件在美国家喻户晓，引起空前的轰动。如想要了解更多，可以访问*http://en.wikipedia.org/wiki/O._J._Simpson_murder_case*。

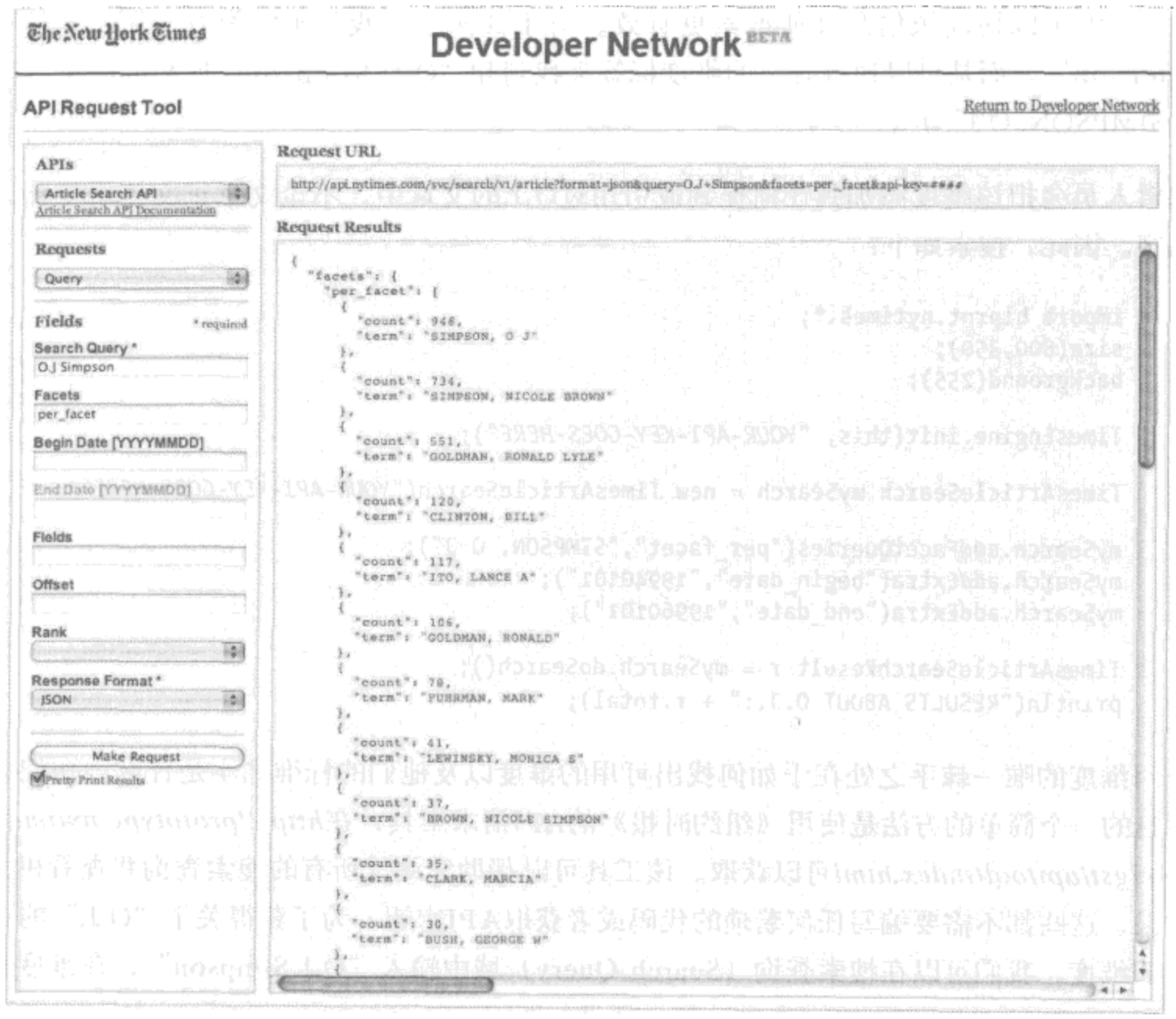


图15-2：API请求工具可以用于查找关于人物、话题和位置等《纽约时报》的官方维度

```
// OJ search
mySearch.addFacetQuery("per_facet","SIMPSON, O J");
mySearch.addExtra("begin_date","19940101");
mySearch.addExtra("end_date","19960101");
TimesArticleSearchResult r1 = mySearch.doSearch();
println("OJ:" + r1.total

// South Africa search
mySearch.clearFacetQueries();
mySearch.addFacetQuery("geo_facet","SOUTH AFRICA");
TimesArticleSearchResult r2 = mySearch.doSearch();

println("South Africa:" + r2.total);

// Rwanda search
mySearch.clearFacetQueries();
mySearch.addFacetQuery("geo_facet","RWANDA");
TimesArticleSearchResult r3 = mySearch.doSearch();
println("Rwanda:" + r3.total);
```

这种方式可以得到3个“《纽约时报》文章搜索结果（TimesArticleSearchResult）对象”，这些对象包含每个结果的文章总数（我们后面可以看到这些对象也可以保存其他有用的信息）。看起来现在正适合对这些数据执行一些（非常）简单的可视化。条形图，或者其他？如图15-3所示。

图15-3：对O. J.（红色显示）、南非（绿色显示）和卢旺达（蓝色显示）所提及次数的简单的图形比较（见彩图14-8）

```
// O.J. bar
fill(255,0,0);
rect(0,50,r1.total,50);

// South Africa bar
fill(0,255,0);
rect(0,150,r2.total,50);

// Rwanda bar
fill(0,0,255);
rect(0,250,r3.total,50);
```

我承认图15-3中的可视化永远都算不上是有趣的。然而，它囊括了在海量的、信息丰富的《纽约时报》文章数据库中探索时所需的几乎所有概念。它还引出了一个非常非常简单的三步模式，即使是在制作最为复杂的数据可视化时，该模式依然有效。

三个简单的步骤

我们先停下来考虑一下在可视化项目中的最基础的处理过程：

1. 获取数据。
2. 把数据转换成有意义的结构。
3. 对数据进行可视化。

通常，这个简单的过程在项目中会重复两次：一次是在发现过程，另一次是在生产过

程。在研究阶段，挑战是挖掘到一组数据，发现一些有意义或有趣的方面，“获取数据”阶段可能会重复很多次，而可视化阶段可能会尽可能地简单。相反地，生产周期通常是在识别完数据后出现的。这意味着我们花了很少的时间来获取数据（因为我们已经有这些数据），而在可视化阶段花了更多的时间。

第二步是研究和生产周期中都存在的：把数据转化成有意义的结构。这些是什么样的结构？是什么使得这些结构有意义？对我而言，这个过程通常意味着把数据分片封装成对象（相关信息能够存储在一起的编程结构）。它通常还涉及把这些对象填充成一些集合，即一个分组列表使得易于对数据进行排序和检索。

在我们的O.J.例子中，这个过程几乎都是由《纽约时报》的Processing库处理的，我们在刚开始可视化时就导入了这些库。我们发现每次执行搜索时都会创建对象。我们创建了一个对象TimesArticleSearch来对所有的API请求进行管理：

```
TimesArticleSearch mySearch = new TimesArticleSearch();
```

以及一个TimesArticleSearchResult对象来存储该API返回的所有请求结果：

```
TimesArticleSearchResult r1 = mySearch.doSearch();
```

这些普通的TimesArticleSearchResult（TASR）对象存储了每个搜索结果的相关信息。到目前为止，我们所做的就是获取接收到的结果的总数，该总数指的是在每个结果对象中以整数形式存储的一个属性：

```
println("RESULTS ABOUT O.J.:" + r.total);
```

但是，TASR对象存储的信息远远不止于这些。实际上，对所有在1994/1995年由《纽约时报》发表的标有O.J.维度的文章，我们可以获取文章的标题、作者、URL、摘要等更多信息——这些信息全部都存储在TASR对象中。每块数据都是以TimesArticleObjects的形式存储在每个TASR对象中，很整齐地排列在文章数组中。默认情况下，TASR保存前10条搜索结果。如果我们想要获取列表中第一篇文章的作者，我们可以这么做：

```
println("FIRST HEADLINE:" + r.articles[0].title);
```

或者，为了获取第10篇文章的Web URL:

```
println("100th ARTICLE URL:" + r.articles[9].url);
```

或者是获取每篇文章的标题列表：

```
for (int i = 0; i < r.articles.length; i++) {
    println("AUTHOR #" + i + ": " + r.articles[i].author);
};
```

在这里，我们开始看到《纽约时报》文章搜索API带给我们的数据的冰山一角。到目前

为止，我们完成了3个相当基础简陋的搜索，结果是约2000条通过一些TASR对象进行封装的文章搜索结果。我们已经知道了如何访问（至少部分的）搜索结果，我们现在来查看一些使搜索和结果更智能的方式。

维度搜索

在前面的例子中，已经介绍了我们是如何通过维度（facets）搜索来确保得到我们需要的结果的。但是尚未提到的是在我们的搜索结果中也可以包含维度。通过结果中的维度信息，我们可以从各个搜索中找到更多的信息，而且可以发现在文章数据库内不同维度（人物、国家、主题）之间的关系。

让我们以一个简单但非常有用的例子来说明我们如何使用维度结果来优化搜索。在前一个例子中我们发现在1994年和1995年，有488篇结果文章的维度标签是“Rwanda geo_facet”（卢旺达地理维度）。如果我们进一步分解，找出在1994年每个月发表了多少篇文章？可以使用我们之前证实的方法，做12次搜索：每个月搜索一次。每次搜索，我们可以使用不同的额外参数“开始日期”（begin_date）和“结束日期”（end_date）来确保每个月份返回相应的结果，但是这看起来需要很大的工作量，不是吗？

可能你已经想到了，这种搜索的一种更好的方法时使用维度结果。实际上，只需要执行一次搜索，就能得到我们想要的结果。搜索的方法和之前的例子相同：

```
TimesArticleSearch mySearch = new TimesArticleSearch();
mySearch.addFacetQuery("geo_facet","RWANDA");
```

但是，不是使用额外参数“开始/结束日期”来约束时间为1994年，这次我们使用的是“发表年份”（publication_year）这个维度：

```
mySearch.addFacetQuery("publication_year","1994");
```

以下介绍一些较神奇的。除了返回通常的搜索结果（一个文章列表），我们将要求API返回一些维度，比如在这个例子中是“发表年份（publication_year）维度”：

```
mySearch.addFacets("publication_month");
```

当我们运行搜索时，维度结果会和所有其他数据一起封装在TASR对象中：

```
TimesArticleSearchResult r = mySearch.doSearch();
```

为了从TASR中获取publication_month结果，我们需要访问感兴趣的特定维度的TimesFacetObjects数组（TASR可以包含任何数量维度的结果）：

```
TimesFacetObject[] months = r.getFacetList("publication_month");
```

现在我们可以知道在1994年1月有多少结果：

```
println("January results: " + months[0].count);
```

我们还可以对整年的结果进行绘图（见图15-4）：

```
for (int i = 0; i < 12; i++) {
    fill(random(150,255),0,0);
    float w = width/12;
    rect(i * w, height, w, -months[i].count * 3);
  };
```

图15-4：1994年《纽约时报》提到“卢旺达”的每月频度

对于该可视化，我们实现了一个非常简单的程序来发现一年内所有提到卢旺达的文章。但是这个小小的可视化实际上可以有很多扩展。它可以对从1981年至今任何一年的维度项的描述进行可视化。虽然我很愿意详述这个过程，但为了节省时间和纸张，还是不这么做了。你可以在*http://www.blprnt.com/examples/nytimes*下载NYTimesGraphMaker。虽然这种探索非常有用，但目前我们还只是局限于在文章数据库中的离散搜索。当我们开始使用API来探索人物、地点和主题之间的联系时，一切变得更加有趣。

连接

当我们向文章搜索API发送任何请求时，可以要求API返回在已经找到的包含了搜索项的文章中包含的维度的一个列表。举个例子，我们可以发现在提到卢旺达的文章中还包含哪些国家，或者在关于O.J.的文章中还提到哪些人，或者哪些主题和南非的种族隔离制度结局最相关。我们还可以做出更通用的请求。完全忽略一个搜索项但是指定一个时间段，我们可以请求这段时间内的所有文章。如果我们想要这些文章的维度列表，我们可以发现一个给定月份、年份或10年期间的最好的维度。举个例子，我们找出谁是1994年最有名的人物。首先，我们创建了一个搜索对象，并给它赋一个空查询（使用加号+来替代空格）：

```
TimesArticleSearch mySearch = new TimesArticleSearch();
mySearch.addQueries("+");
```

现在，我们把搜索局限于1994年，在搜索对象的结果中包含维度per_facet：

```
mySearch.addFacetQuery("publication_year", "1994");
mySearch.addFacets("per_facet");
```

并执行如下查询：

```
TimesArticleSearchResult r = mySearch.doSearch();
```

如果我们想要列出在1994年中提到的最著名的人物，我们可以这么做：

```
TimesFacetObject[] stars = r.getFacetList("per_facet");
for (int i = 0; i < stars.length; i++) {
    println(stars[i].term);
};
```

它会输出非常复杂的一组名字：

```
CLINTON, BILL
GIULIANI, RUDOLPH W
CUOMO, MARIO M
CLINTON, HILLARY RODHAM
PATAKI, GEORGE E
SIMPSON, O J
SIMPSON, NICOLE BROWN
KERRIGAN, NANCY
GINGRICH, NEWT
RABIN, YITZHAK
CORTINES, RAMON C
ARAFAT, YASIR
RENO, JANET
WHITMAN, CHRISTINE TODD
BERLUSCONI, SILVIO
```

这个列表使我们回想起一些关于《纽约时报》的事情：它同时还是一份城市报纸、国内报纸和国际报纸。想到这一点，当我们发现当时——以色列总理Yitzhak Rabin（他在1994年赢得了诺贝尔奖）被提及的次数仅比纽约市教育部长Ramon Cortines多一些——就不会感到太奇怪了。虽然我们对于该搜索涉及的范围之广可能很满意，我们可能还想把搜索限制在某个“版本”。我们可以使用维度完成。这次我们将通过使用desk_facet维度，选择只在Foreign Desk上发表的文章。

```
mySearch.addQueries("+");
mySearch.addFacetQuery("publication_year", "1994");
mySearch.addFacetQuery("desk_facet","Foreign Desk");
mySearch.addFacets("per_facet");
TimesArticleSearchResult r = mySearch.doSearch();

TimesFacetObject[] stars = r.getFacetList("per_facet");

for (int i = 0; i < stars.length; i++) {
    println(stars[i].term);
};
```

这个查询结果生成了更普通的一组结果：

```
CLINTON, BILL
ARISTIDE, JEAN-BERTRAND
YELTSIN, BORIS N
ARAFAT, YASIR
```

```
RABIN, YITZHAK
CHRISTOPHER, WARREN M
BERLUSCONI, SILVIO
MANDELA, NELSON
GOLDSTEIN, BARUCH
BOUTROS-GHALI, BOUTROS
CEDRAS, RAOUL
CARTER, JIMMY
POPE
KIM IL SUNG
MAJOR, JOHN
```

这个列表是由不包括关键字的查询或维度搜索生成的；我们可以选择任何一个或者所有这些名字，查询和这个人物相关的最有名的人物列表。这里，我们将搜索在1994年和Yitzhak Rabin相关的人物列表：

```
mySearch.addQueries("+");

mySearch.addFacetQuery("per_facet","RABIN, YITZHAK");
mySearch.addFacetQuery("publication_year", "1994");
mySearch.addFacetQuery("desk_facet","Foreign Desk");
mySearch.addFacets("per_facet");
TimesArticleSearchResult r = mySearch.doSearch();

TimesFacetObject[] stars = r.getFacetList("per_facet");

for (int i = 0; i < stars.length; i++) {
    println(stars[i].term);
};
```

这个查询的输出结果列表如下：

```
ARAFAT, YASIR
HUSSEIN I
CLINTON, BILL
PERES, SHIMON
GOLDSTEIN, BARUCH
ASSAD, HAFEZ AL-
CHRISTOPHER, WARREN M
CHRISTOPHER, WARREN
WAXMAN, NAHSHON
MUBARAK, HOSNI
SHARON, ARIEL
ABDELSHAFI, HAIDAR
BHUTTO, BENAZIR
BOUTROS-GHALI, BOUTROS
```

我们现在开始不仅仅是简单地获取我们搜索的结果，而且还包含了这些结果之间的关联。如果要使用第一个列表中其他人物来重复Rabin的过程，我们将在“超级列表”中包含225个人物。不过，这个超级列表是包含重复项的：正如我们在Rabin列表中所看到的，有些人物已经在我们的第一个列表中出现了（Arafat、Clinton、Goldstein和Boutros-Ghali）。

这些关系是《纽约时报》提供给我们的很有意思的数据的一部分。通过检视这些关系，我们可以发现人物、地点、主题之间明显的和隐藏的关系。如图15-5所示，我们之前提到的相同列表的255个名字以网络图表方式说明，其中的连线表示提到的人物之间的关联关系。

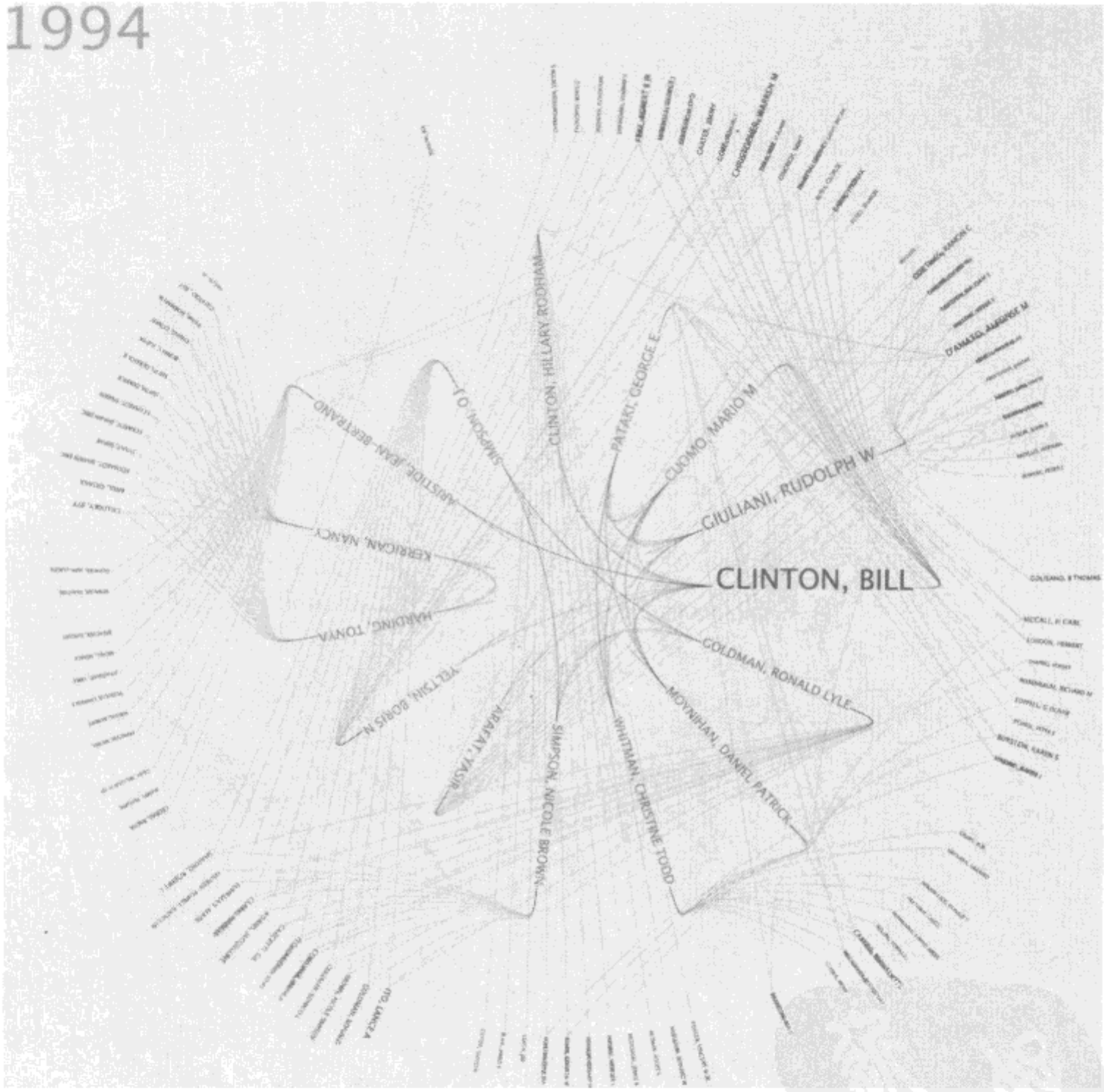


图15-5：说明了1994年新闻价值最高的人物的网络图

该图像把海量的新闻信息浓缩成一张图形。对于典型的数据检索系统，这种图形需要花费非常多的时间才能生成。正如我们所看到的，《纽约时报》文章搜索API使得这一过程容易了很多。

以前面提到的例子为例，再结合人物和机构组织会使它变得稍微更有趣一些。该API只包含了31个查询，我们可以创建一张图片，能够显示在1994年的新闻中，成百上千的人物、企业和国家是如何相互关联在一起的（该例子的全部源代码可以在*http://www.blprnt.com/examples/nytimes*获取）。其结果如图15-6所示。

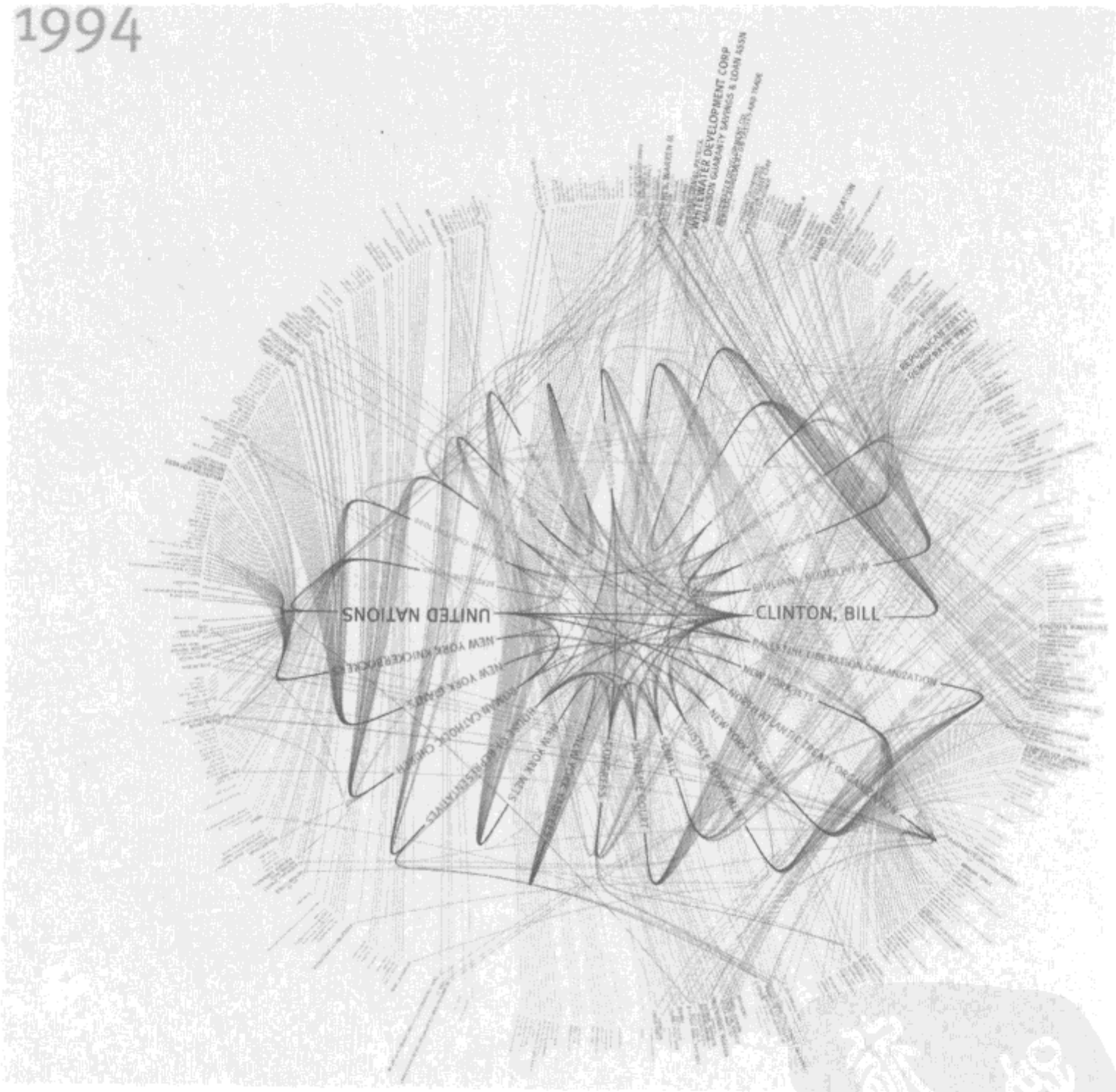


图15-6：该图显示了在1994年《纽约时报》中最频繁提及的人物和组织

结束语

《纽约时报》的API为所有领域的研究人员提供了丰富的信息。其数据库既包含历史记录也包含实时报道——每时每刻都在创建着新的内容。当然，无论《纽约时报》的资料

库信息多么丰富，它只是海量的开放数据中很小的一部分——每周跨越式不断增长的条目。实际上，似乎已经解决了过去关于开放数据的首要问题，即如何开放数据，而当下面临的是第二个更大的问题是，我们如何尽可能地利用如此大量的信息？

我认为该问题的部分解决方法在于促使尽可能多的人们访问和探索可用的数据。很多大规模的开放数据的目标在于服务于已有的数据人员：软件开发人员、计算机科学家和培训过的信息专业人员。大部分的重点在于使数据在整个企业范围内可用。然而，正如我们在本章所见的，至少我们可以使用一些简单的工具来发送一些简单的请求，以获取其中一些数据。这个技能对于记者、社会学家、历史学家、艺术学家和科学家都是必需的，如果我们真的想要在这个新的开放数据领域找到有真正的价值的发现。

下一步要做的就是去探索。深入文章搜索数据库，问一些你自己的问题，然后共享结果。而这只是个开始。可以使用你在本章学到的技术来探索很多其他API，可以在其中发现成千上万的答案。祝你好运！

第16章

《纽约时报》的一天

Michael Young 和 *Nick Bilton*

你是否曾经想过《纽约时报》网站的读者会涵盖什么类型的人？我们想过。我们还在想他们倾向于在一天之中的什么时候来访问网站，使用什么工具访问以及他们都来自哪里，纽约、巴黎或博伊西？从他们是谁到在什么时候、以什么方式以及为什么等，所有这些问题都在我们的思考范围之内。

本章将要介绍的这个可视化项目的开发源于在《纽约时报》研发试验室的一次午餐时就这个话题开展的一次简单讨论。正如你将看到的，从非常简单的基于地理的数据集合开始，很快就深入到海量数据和潜在可视化。最终，我们创建了一个可视化用于显示每天《纽约时报》网站*nytimes.com*和手机移动网站*mobile.nytimes.com*在世界和美国的流量。

我们这个可视化探索的第一阶段是数据收集。《纽约时报》网站每个月的页面浏览量可以达到几亿，其中独立访问量在1700万～2100万之间波动。此外，还有很多网关可以访问该网站，包括移动网站、时报阅读器航空应用（Times Reader AIR application）、iPhone应用、API等很多方式。

具体到这次实验，我们仅选择了标准的《纽约时报》网站*nytimes.com*和其移动版本（*mobile.nytimes.com*）。虽然为了简化实验而仅选择了上述两种资源，但是即使是在仅仅使用这两个数据集的情况下，需要筛选和可视化的信息仍然是巨量的。

我们的探索的第二个阶段是创建一个基于地图的可视化。该可视化显示了流量模式以及在过去24小时内Web站点和移动站点的读者数量的波动。

随着可视化的不同阶段的逐步完成，我们不仅为该网站的读者数的庞大程度而深感吃惊，同时也对读者们访问该网站的时间特征感到惊奇。从视频*http://bit.ly/nytdayinlife*中你可以看出，*nytimes.com*网站在晚上是相对活跃的，而午夜到早上5点其用户访问量却是几乎保持不变的。当住在美国东海岸的《纽约时报》网站的读者陆续醒来时，流量就开始暴涨，可视化开始膨胀；当人们中午吃饭休息期间开始查看每天的消息时，也会出现类似的流量暴涨。访问手机网站（*mobile.nytimes.com*）和Web站点（*nytimes.com*）的用户数的变动特征也是很有意思的；如后面的可视化所示，在每天的某些时刻，会出现手机网站的访问流量比标准网站的访问流量还要大的情况，在另外一些时刻也有与此相反的现象存在。

随着数据处理的愈加方便，接下来可做的一些有趣的处理方式逐渐明朗了起来。只要时间允许，我们希望每天甚至是在流量高峰时刻，比如在一些重大新闻事件发生的时刻，能够自动渲染视频。数据收集和可视化编码方面还有很大的优化空间（总是如此）。最终，我们讨论了如何对更为具体的数据进行可视化。举个例子，显示每天来自特定设备（如iPhone）的流量，或者抽取出位于加州的用户，对他们阅读的新闻进行地理编码，进而就可以分析他们是更倾向于查看关于纽约的新闻还是倾向于查看和他们自己的位置更为接近的新闻。其他的可能方案包括在重大日子或者有重大新闻时可视化读者的阅读模式，从而了解新闻是如何在Web、社交网络和特定地理位置之间传播的。

分析的方法是无穷无尽的。我们相信一张图所蕴含的信息量与上千个词的信息量相仿，但是一个数据集所能表达的却可以达到上千篇文章的效果。

收集一些数据

在深入介绍可视化本身的内容之前，我们首先对隐藏于其后的数据进行一次讨论。为了对*nytimes.com*和*mobile.nytimes.com*的24小时的流量进行可视化，我们需要创建一个可以从《纽约时报》的访问日志中抽取和清洗数据的程序。考虑到我们想要创建一个可以显示在一天内网站的访问次数的可视化并且是一个基于地理信息进行展示的可视化，我们需要的数据包括：

- 在24小时内，用户每次访问Web站点或手机网站的时间戳。
- 每个用户每次访问时所处位置的经度和纬度。

原始的访问日志包含了人们访问Web站点和手机站点的很多有用的信息（比如每个访问者使用什么浏览器）；但是，其中有很多信息对我们而言是没有用的，因此需要把它们从日志信息中过滤掉。此外，日志中并不包括每个用户每次访问时的经纬度信息，因此这是我们在日志“清洗”过程中需要添加的信息。

《纽约时报》的Web站点，是一个流量在新闻类网站中排名前五的站点（根据Nielsen[注1]排名），其月独立访问读者约2000万。这意味着，在任何一天Web站点和手机站点上都有几百万次的页面浏览（或点击）；这是我们准备为可视化收集的基础数据。

数据清洗

处理原始的访问日志的第一步是“清洗”数据。对于处理任何类型的Web日志的人来说，这通常都是一个常见的步骤。对于可视化以及其他日志数据的分析，我们只对来自人们的在Web页面和手机网站的点击数感兴趣——而不是来自网络爬虫（spider）、机器人（bot）或抓取程序(scraper)。为了过滤这些不必要的数据，我们实现了一段Java代码用于识别出非人工的访问日志并将其从日志中删除。每天Web站点原始的日志数据访问量大约有500MB～700MB（压缩格式的），手机网站的访问量约80MB～100MB（压缩格式的）。在对数据进行清洗过程中，我们还执行了IP到经纬度的转换，从而可以得到每个访问用户的精确位置。原始访问日志中已经包含了用户的IP地址，然后我们使用商业数据库把IP转换成地理位置信息。有很多公司提供GeoIP（地理位置IP）数据库，可以用于实现该转换。举个例子，MaxMind公司提供了一个商业数据库以及一个包含了各种客户端库的免费版本，借助这些客户端库就可以访问该公司的数据库了。

一旦数据被清洗完毕并准确地进行了地理位置编码，只需要对数据再做最后一轮处理。由于原始的访问日志的收集、存储和清理方式，新清洗完的数据是存放在多个文件中的，需要对它们排序之后合并到一个结果文件中去，该文件将包含可视化所需的数据，即一天访问数据。

每天“清洗”后的《纽约时报》网站*nytimes.com*的日志数据被存储到360个文件中，每个文件大小约30MB～40MB（压缩格式的）。由于每行中增加了一些额外的字段，如GeoIP 信息，“清洗”后的日志文件要大于原始文件。对于手机网站，因为数据集小得多，清洗后的数据存储在一个文件中，大约有70MB（压缩格式的）。我们每天需要整理当天的每个清洗后的日志文件，并创建按照对Web站点和手机网站的访问时间戳以及访问者所在的经纬度排序的单个文件（Web站点和手机网站分别生成一个文件）。排序结果看起来如下（每行一条数据记录）：

```
00:00:00,-18.006,-070.248
00:00:00,-22.917,-047.080
00:00:00,-33.983,0151.100
00:00:00,014.567,0121.033
...
```

注1：*参考http://blog.nielsen.com/nielsenwire/online_mobile/msnbc-and-cnn-top-global-news-sites-in-march/*。

Python、Map/Reduce和Hadoop

数据处理的最后一步，我们用Python语言创建了一个简单的map/reduce脚本，它可以从清洗后的日志文件中过滤掉所有不需要的数据，并输出以逗号作为分隔符的数据，最后还会对数据进行排序。（在研发组，我们通常使用Python来收集、处理和解析数据。当对大的数据集进行可视化时，我们用Python来处理所有繁重的数据处理，创建在可视化应用程序中易于读取和解析的文件。）我们使用Amazon的弹性MapReduce Web服务，它允许我们在很多基于Hadoop的EC2的运行实例中运行Python实现的map/reduce。Amazon的EC2运行实例的“配置”不同（低配、中配和高配），不同的配置会提供不同的RAM、CPU核数和内存，因此我们在很多EC2实例中试验运行map/reduce代码，从而找到性价比最好的配置。数据处理需要约10～20分钟（价值几美元），具体所耗时间会依赖于机器的数量（我们从4～10台都尝试了一遍）和EC2实例的配置（我们尝试了低配和中配）。

map/reduce（Hadoop）Job的输出结果是很多有序的文件，这些文件保存在Amazon的S3桶（buckets）中。为了在可视化中把数据放到一个文件中（与前述方式相同，Web站点和手机网站分别存储，各自有一个独立文件），我们从S3下载结果文件到本地，然后按照传统的方法进行排序和归并。现在，数据已经按照期望的方式保存在一个文件中了，可视化的准备工作已经完成。

可视化的第一步

重申一下，该项目的目标是对《纽约时报》Web站点*nytimes.com*和手机网站*mobile.nytimes.com*一天的访问量进行可视化，并查看在一天之中用户对这两个站点的访问是如何变化的。我们想查看在特定地理区域甚至是全球范围内，是否出现了某些有趣的模式。在美国的哪个地区、什么时间手机网站流量达到峰值？我们是否会看到在手机普及率比美国高的国家，如中国和印度，其对手机网站的访问量是否更高？Web站点和手机网站在一天的某些时间段，如凌晨、上班时间、午餐时间和下班时间的访问量如何？有些问题通过最基础的流量报告就可以回答，但是我们希望给这种普通的报告增加一种新的视觉维度，使人们可以看到在一天的不同时段上访问量按照地理维度的分布情况。

我们在可视化上做的第一个尝试是创建了一个简单的世界地图，将一天之中对*nytimes.com*的每次访问用一个小的黄色圆圈表示，对*mobile.nytimes.com*的每次访问用一个小的蓝色圆圈表示。除了全球范围的视图，我们还希望创建一个聚焦（或缩放）于美国的视图。

对于我们所创建的第一个可视化在后面将会详细介绍，在此不做赘述。对我们而言，这次尝试主要是一个学习积累的过程——对如此庞大的数据集进行合理可视化会面临很多

挑战，而且我们马上就意识到了这一点。在当前版本之前，我们对代码进行了多次修改，而且只要有时间，我们仍然会不断对数据处理和可视化处理的模块进行优化。

Processing

Processing（面向设计的开源编程语言和集成开发环境）被选作我们的可视化工具，有几个原因。首先，在《纽约时报》研发小组中的成员当中有些人已经有使用Processing完成小的数据可视化的项目经验，他们还拥有使用传感器作为数据收集设备进行探索的经验。此外，我们都是Ben Fry、 Casey Reas（Processing创始人）和Aaron Koblin使用该工具所创造的作品的超级粉丝，我们认为Processing将会成为对海量数据进行可视化的理想工具。

对于该可视化，我们需要做的第一件事是将网站的访问用户的经纬度信息映射到Processing中的二维可视化图形中。Aaron Koblin友情提供了一些他在前一个项目中实现该功能的代码——很不错的、紧凑的Java类，可以把经纬度组转换成*x*、*y*坐标。我们需要做的就是向Java库传递数据文件中的经纬度元组，Java库就会返回*x*、*y*坐标。然后，我们把这些坐标值传给Processing的绘图API来定位《纽约时报》Web站点*nytimes.com*和手机网站*mobile.nytimes.com*的每个用户的位置。

基础层地图

创建基础层地图——如刚刚绘制的世界地图——所需的时间会远远超过你的想象。首先，我们需要对美国和世界做出准确的表示。经过大量的数据探索后，我们最终使用加州大学洛杉矶分校的CENS组数据集，它描绘了世界上每座城市的经度/纬度坐标。

在使用该数据集的初始阶段，每当程序启动时，直接在Processing集成环境中进行渲染，但是这个渲染花费的时间比我们期望的要多很多；因为知道该数据不会变，最后，我们创建了一个JPEG地图，向背景地图中加载一个非常小的文件（见图16-1和图16-2）。这种方式给我们节省了好几分钟的渲染时间（当解析大数据集时，这部分工作所需的时间会更长）和处理能力，并且成为所有后续的数据输出和视频的背景。

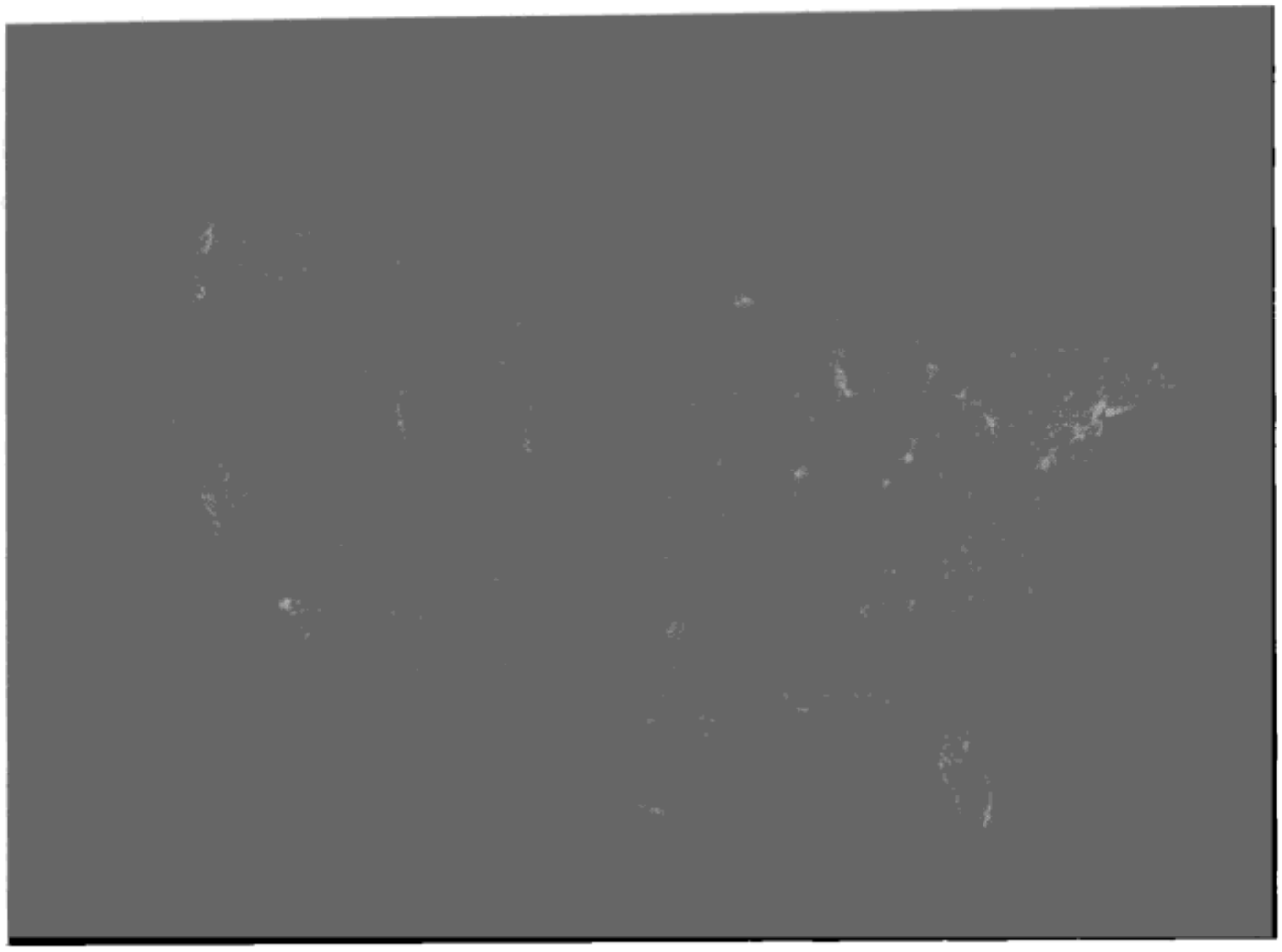


图16-1：美国人口地图（见彩图123）

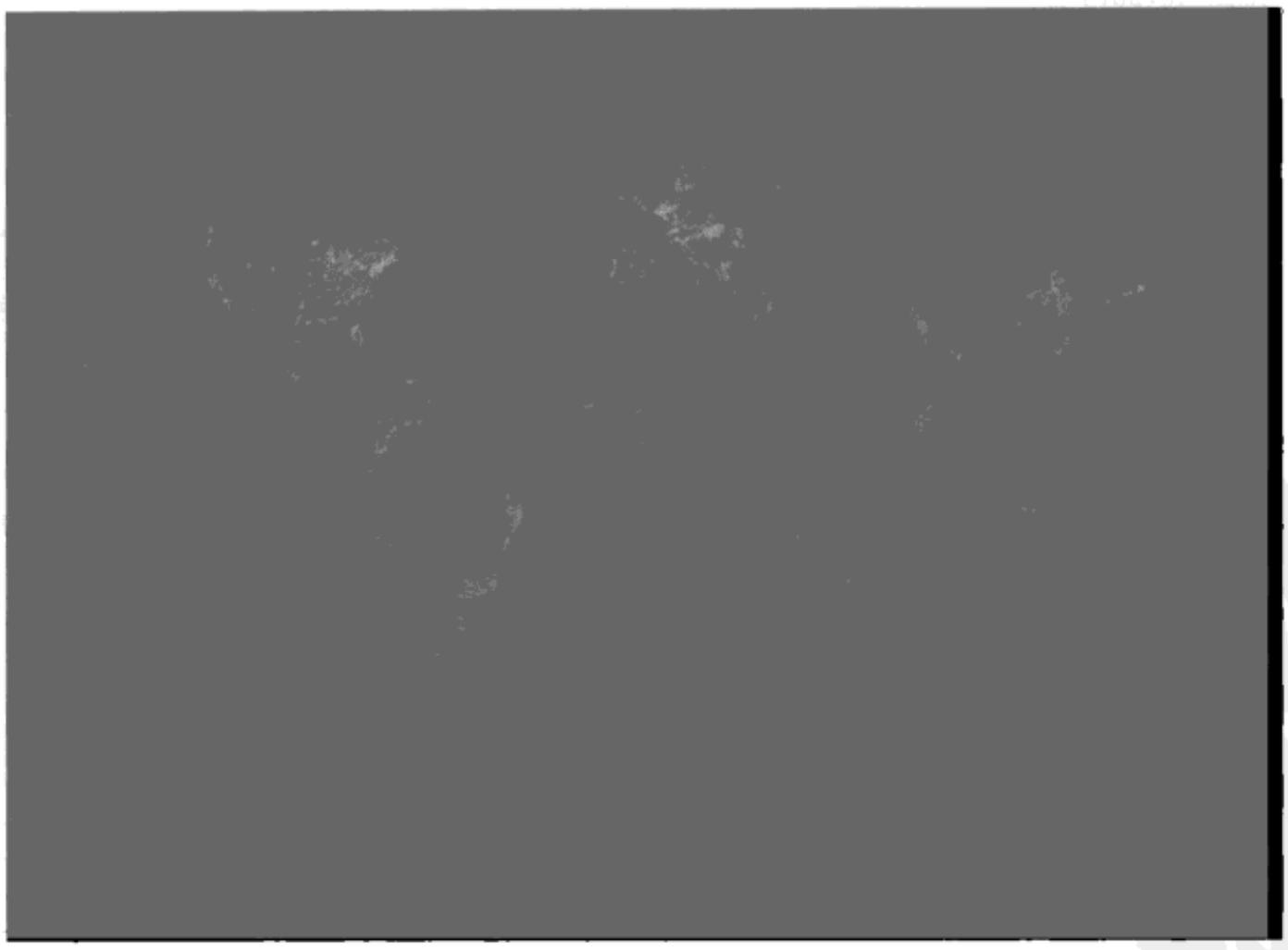


图16-2：世界人口地图（见彩图124）

刚刚处理的数据哪去了

有了纬度/经度投影代码和地图轮廓，我们开始在地图上描绘交通数据图。在可视化初期，我们使用不包含重大新闻的任意一天的数据（2009年2月15日）。这一天的Web站点和手机网站的流量/访问次数和平均值一致。

我们之前已经对数据进行过清洗、排序和添加地理位置编码，它包含了时间戳、Web站点和手机网站上给定一天的用户每次查看/点击时所处的纬度/经度值。现在到了创建一个Processing应用程序的时刻了，它可以扫描Web站点和手机网站的日志文件，对于用户的每次查看/点击，会在地图上描绘一个基于用户点击时所在位置而生成的点。

场景1，步骤1

Processing应用在绝大多数情况下由两部分组成的：启动（setup）和循环绘制（draw）。在Processing 应用的setup()函数中，你可以执行应用需要的任何工作，比如变量初始化、打开输入文件、字体加载等。循环绘制是Processing代码的根本。Processing应用中的draw()函数通常每秒钟会被调用30～60次（这是时间帧速率）。

我们的第一次尝试的内容大体如下（简单的伪代码描述）：

```
void setup()
      - open up both the mobile and web log files
      - load the data for the world map
void draw()
      - draw the world map
      - read a second's worth of log data from the web and mobile log files
      - draw a yellow point for each visit/hit to nytimes mobile site (during that
second in the log file)
      - draw a blue point for each visit/hit to nytimes.com website (during that
second in the log file)
```

这段代码尽管存在一些问题，但是能够生成一些可以在屏幕上观看的画面。可以多次运行该应用程序，查看图片中描绘的点，这些点表示《纽约时报》Web站点和手机网站一天的流量。随时间变化的流量的模式让人难以置信——画面看起来似乎是活生生的，闪烁的灯光散布在整个地球上，如图16-3所示。

这是伟大的第一步，但是我们的代码和方法都需要做些修改。以下部分将介绍需要改进的3个方面。

图16-3：原始可视化显示了《纽约时报》Web站点nytimes.com和手机网站mobile.nytimes.com在全世界的流量——黄色圆圈表示Web站点的流量，蓝色圆圈表示手机网站的流量（见彩图125）

没有具体比例

首先，该可视化没有显示来自每个用户位置的Web站点和手机网站的流量的比例。比如，在一天的某个时刻，可能有很多Web站点和手机网站的用户是来自相同的地方，比如纽约，可以看到有非常高的流量）。有时，可能有成千上万用户来自同一个地理位置。同样，假如是纽约！

在该应用程序的最初版本中，日志文件中出现的每个地理位置（一组经纬度值）在我们的地图上都是使用相同大小的点表示的。为了能够表示比例，需要基于与某个位置关联的用户量来调整每个位置的可视化表示（地图上的蓝色和黄色点）。

其次，因为黄色（表示Web站点流量）和蓝色（表示手机网站流量）点大小相同，而我们（在绘制循环中）先画表示Web站点的点，再画表示手机网站的点，当两种点击类型位于同一个地理位置时，蓝色点会覆盖黄色点。这对可视化而言不是一个好的选择。

没有考虑时间

在可视化的第一阶段，我们没有考虑人们在Web站点或手机网站上每次访问或页面查看所花费的时间，只是简单地在地图上为每次访问画了一个点，在可视化的整个过程中都

不再管它了。这样，就没有人会注意到在某些大城市《纽约时报》有持续较大的流量，而在一些小的偏远地区我们可能一天只有几次查看，这种表示方式会使我们错误地认为这些地区整天都有流量。

我们需要解决这个问题，并结合比例表示问题，也就是说，我们需要提出一种新的方法，可以精确地表示从任何一个位置有多少人访问该网站，以及他们在某篇文章上停留了多长时间，或者在整个网站上停留的时间。

最重要的是，我们在一天的每一秒内都必须做这件事！

定时拍摄

最后，我们选择将整天的数据流量创建成为一个定时拍摄视频，从而使得我们能够在整个《纽约时报》公司内共享该可视化。为了解决这个问题，我们决定使用Processing的一个内置的视频库，它能够将循环绘制生成的时间帧保存到视频文件中，进而创建出很清晰的电影形式的输出。

场景1，步骤2

在项目的第一个版本代码基础之上，我们增加了通过Processing的MovieMaker库将可视化捕获下来并保存到一个文件中的功能。我们还增加了应用支持，能够使一对Web站点或手机网站的每次点击的可视化都能够体现该次访问的生命周期。平均来说，Web站点和手机网站这两个站点的一次访问时间是历时3～4分钟。因此，在迭代过程中，不再是在地图上画一个点并在后面整整24小时都不管它，我们尝试慢慢地每3分钟淡出消减一个点。当然，一个独立用户不是每3分钟对Web站点或手机网站执行一次点击——日志文件中显示的很多点击都是来自同一批用户，或者是用了更长的时间浏览了网站的很多页面的用户。但是为了避免可视化的最初版本过于复杂，我们就笼统地认为每次对网站的访问都是“3分钟访问”。

对于这种简化的表示，我们需要保存一天内的每次查看/点击淡出3分钟以上的点。这意味着需要在内存中存储很多对象。对于每秒钟内Web站点和手机网站上的每次点击，我们都会在Processing应用程序中创建一个对象，它的任务是保存该点击的“生命周期”，也就是说，这个点需要在屏幕上停留多长时间（3分钟），使用这些对象来帮助我们在可视化的整个周期内对点实现淡出效果。

因此，我们再回过来看Processing的绘制循环。我们还是每秒钟从Web和手机网站的日志文件中读取数据，但是对于每次单击，我们创建一个Hit（单击）对象，其初始生命周期设置为3分钟，初始不透明度是100%（这些值在迭代循环的每次绘制中不断减少）。读完日志数据后，我们遍历内存中Hit对象集合。对于每个Hit对象，我们重新描绘表示该

单击的点，其透明度是基于该单击剩余的生命周期，在3分钟时间内把它淡出。当每个Hit对象达到生命周期时，把它从内存中删除，并从地图上删除相应点（即不再重新描绘它）。

因为每秒钟大约需要对400～500次点击进行可视化，这种方法意味着任何时刻都需要在内存中存储很多对象，来保存所有点击（或用户）轨迹。我们已经意识到这个问题，并想到了一些优化方案，但是还是想尝试这些简单的步骤并确定这种方法是否可行。

让我们运行一下，看看会发生什么

增加支持每次点击在3分钟后淡出的功能，使我们更接近于对该网站流量的可视化，但是还需要做更多的工作。一方面，我们还没有把每个地理位置的流量比例显示添加到可视化中。另一方面是速度问题——运行这个版本，我们在25分钟内只能生成历时45秒钟的视频。内存和处理器处理都很慢，可视化的运行和渲染更慢。我们试着在实验室几台不同的机器上运行（Mac Minis，1GB RAM；MacBook Pros，4 GB RAM和Mac Pro），但是该应用程序在每台机器上渲染都很慢。虽然该可视化与我们期望的结果进一步接近了，但是需要对它做一轮新的优化——我们需要生成历时1天的可视化视频，而目前我们最多能够生成历时1个小时的视频。

第一版的可视化可以通过如下链接查看：*http://nytlabs.com/dataviz*。

可视化的第二步

既然我们已经清楚想要什么样的可视化，我们需要实现它。除了增加支持能够显示每个地理位置的流量比例，我们需要对应用程序进行优化，它需要我们重新思考如何收集数据。

重新回到比例问题

每秒钟显示每次点击并不能显示任何比例。在第一版的应用程序中，来自加拿大农村地区的少量的点击和来自纽约的成千上万的点击，其可视化权重是一样的。此外，从内存和处理器对可视化进行渲染的处理能力而言，每秒钟显示所有的点击代价太高。

想清楚后，我们认为答案是对每分钟每个地理位置的点击次数进行可视化，而不是每秒钟进行可视化。对于访问日志文件中的每分钟的数据，我们会累加每个地理位置的点击总数。这种方式使得可视化结果可以显示每个地理位置的流量比例，而且会极大地减少Processing应用程序的原始数据输入。但是，这种方式意味着我们需要改变数据处理和map/reduce作业。

进一步处理数据

之前用Python实现的map/reduce脚本，其目的是从原始访问日志中解析出我们需要的数据，并基于时间对数据进行排序，因此，需要做些修改。现在，该脚本需要对每分钟、每个地理位置（一组纬度/经度值）的所有点击进行计数，输出结果数据并根据访问时间进行排序。

如果你对map/reduce是如何工作的还不熟悉，我们建议你从网上获取一些基本教程进行阅读。从根本上说，map/reduce是一个编程模型，支持海量数据处理。其处理过程分成两个任务：mapping（映射）和reducing（规约）。Mapper通常是接收一些输入（在我们的例子中是日志文件），对数据做一些较小的处理，然后以键/值（key/value）对的方式输出数据。Reducer的任务是接收Mapper的输出结果数据，对数据进行归并或规约，通常生成较小的数据集。

在我们的应用程序中，Mapper脚本读入原始的访问日志文件，对于每一行，以如下格式输出键/值对：

```
Timestamp of the access (in HH:MM format),latitude,longitude 1
```

在这个例子中，key（键）是以逗号作为分隔符，包含了日志文件中每次点击的时间戳、纬度、经度，而value（值）是1（表示一次点击计数值）。

然后，Reducer逐行读取Mapper的输出，保存每分钟每个地理位置的点击计数值。因此，它把Mapper输出的每个“key”存储到一个Python字典中，每次遇到Mapper的输出有相同的“key”，就把其在字典中的计数值增加1。Python字典看起来大概如下：

```
{
     "12:00,40.7308,-73.9970": 128,
     "12:00,37.7791,-122.4200": 33,
     "12:00,32.7781, -96.7954": 17,
     # cut off for brevity...
     "12:01,40.7308,-73.9970": 119,
     "12:01,37.7791,-122.4200": 45,
     "12:01,32.7781, -96.7954": 27,
     # ...
}
```

一旦Reducer读取了Mapper的所有的数据输入，它对数据进行排序（基于key），然后输出排序的结果：

最初版本中Mapper和Reducer的代码如下：

```
Mapper
#!/usr/bin/env python
```

```
import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
      # remove leading and trailing whitespace
      line = line.strip()
      # split the line into words
      words = line.split('\t')

      try:
            # output the following:
            # time(HH:MM),latitude,longitude 1
            time = words[1]
            hours,mins,secs = time.split(":")
            t = hours+":"+mins

            print '%s,%s,%s\t%s' % (t, words[44], words[45], 1)
       except Exception:
            pass
```

Reducer

```
#!/usr/bin/env python

from operator import itemgetter
import sys

locations = {}

# input comes from STDIN
for line in sys.stdin:
      # remove leading and trailing whitespace
      line = line.strip()

      # parse the input we got from mapper.py
      key, count = line.split('\t')

      try:
            # update the count for each location (lat/lng pair)
            # per minute of the day
            count = int(count)
            locations[key] = locations.get(key, 0) + count

      except Exception:
            # count was not a number or some other error,
            # so silently ignore/discard this line
            pass

# sort the data and then output
sorted_locations = sorted(locations.items(), key=itemgetter(0))
for key, count in sorted_locations:
      try:
            time,lat,lng = key.split(',')
            print '%s,%s,%s,%s'% (time, lat, lng, count)
      except Exception:
            pass
```

新的数据格式

在原始访问数据上运行完新的map/reduce脚本后，我们得到了一组更准确的数据集。这个过程不仅减少了总的数据量（Web站点的访问数据，从3000万行左右减少到300万行），而且为我们生成了每个地理位置的计数值。现在，我们需要确定比例因子。以下是新的结果数据的样本——注意时间戳、纬度、经度和（每分钟的）点击计数值。

```
12:00,039.948,-074.905,128
12:00,039.949,-082.057,1
12:00,039.951,-105.045,3
12:00,039.952,-074.995,1
12:00,039.952,-075.164,398
12:00,039.960,-075.270,1
12:00,039.963,-076.728,4
12:00,039.970,-075.832,2
12:00,039.970,-086.160,4
12:00,039.975,-075.048,23
```

可视化比例和其他可视化优化

有了新形式的数据，我们不再是每秒钟为每次点击画一个点，而是可以每分钟为每个地理位置的点击数值画一个圆圈，并根据点击数计算圆圈大小。这种方式可以生成期望的比例显示，使得可视化的读者可以轻松地区分来自加拿大农村和纽约市的不同的流量差别。

这种方式也极大地减少了应用程序需要的内存量。我们还是需要在内存中保存Web站点和手机网站的所有点击（这样我们才能消隐去时间超过3分钟的点击），但是因为我们现在保存的是每分钟每个地理位置的点击数，极大地减少了需要的Hit对象数量。对于任一分钟，来自全世界的流量通常包含2000～3500个不同的地理位置。每个位置的Hit对象必须存储在内存中；每个Hit对象生命期是3分钟，因此对于任一时刻，内存中可能有6000～12 000个对象——数量还是很多，但是已经远远小于前一版本的对象数量。

现在，需要更新Processing应用程序，从而可以实时保存每个位置在任一时刻的点击数，而且圆圈大小比例可以根据点击数调整。我们一起看个简单的例子。

假定数据是来自于纽约是某个特定的纬度/经度的对Web站点的访问日志（数据集中有非常多这样的数据）。只查看一天中很短的一段时间，假定在某个时刻，点击数如下：

```
12:00 - 100 hits
12:01 - 110 hits
12:02 - 90 hits
12:03 - 80 hits
12:04 - 100 hits
```

当在地图上为这个地理位置的点击数画圈时，我们希望圆圈大小能够反映点击数，这样

可以显示比例。然而，我们不能简单地基于当前一分钟周期内的初始点击/查看计数值来计算圆圈大小。为什么呢？记住通常对一个站点的访问能够持续3分钟，因此我们决定为每个地理位置的点击数保留3分钟，只有当超过3分钟后才把这些地理位置的计数值删除。使用以上的点击计数，每分钟总的点击计数值将会如下：

```
12:00 - 100 hits (assuming no previous hits)
12:01 - 210 hits (100 + 110)
12:02 - 300 hits (100 + 110 + 90)
12:03 - 280 hits (110 + 90 + 80)
12:04 - 270 hits (90 + 80 + 100)
```

注意，对于任意某一分钟，我们都保存了该时刻的新的计数值以及其之前两分钟的点击计数值。

更新Processing应用程序代码，保存每分钟每个地理位置的总的点击数，生成的结果如图16-4所示。该新版本允许我们查看任何时刻地图上不同地理位置的点击比例显示，而且也说明了该比例如何基于每个地理位置的流量的增长而扩大，或减少而收缩。

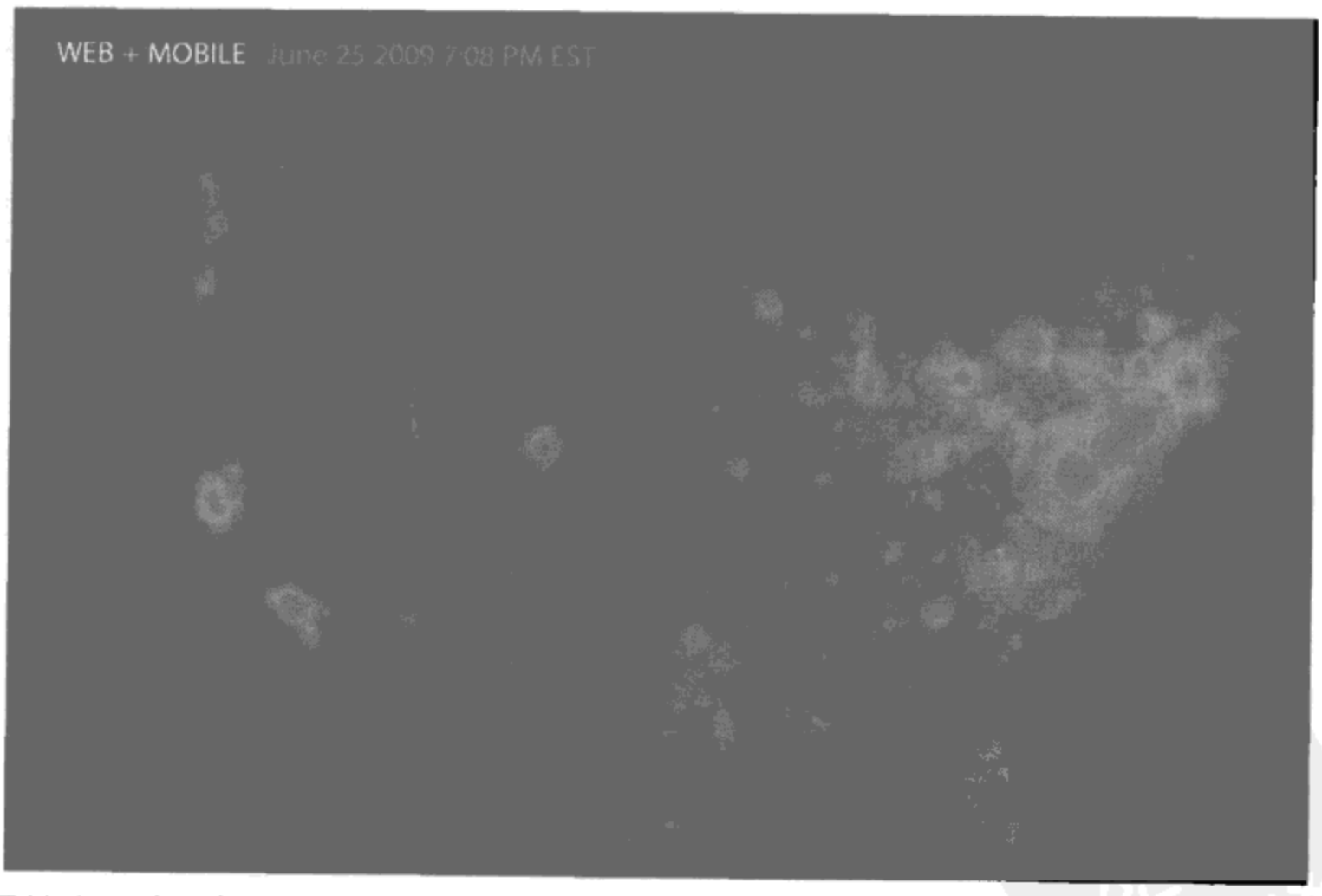


图16-4：更新后的可视化显示了在2009年7月25日《纽约时报》Web站点nytimes.com和手机网站mobile.nytimes.com来自美国的流量——黄色圆圈表示到Web站点的流量，而红色圆圈表示到手机网站的流量（见彩图126）

使定时拍摄能够正常工作

对Processing应用程序进行升级使其能够处理新的数据格式和方法，在此之后，我们创建了一个完整的历时24小时的定时拍摄视频。我们新的代码每次能够正常运行几个小时，不存在之前遇到的内存和整体机器延时，现在是生成完整的定时拍摄视频的时候了。不再像第一次那样尝试在地图上为历时24小时定时拍摄渲染Web站点和手机网站数据，我们只使用手机网站的数据（其数据量大约是Web站点数据量的10%）；这样，我们就可以比同时渲染Web站点和手机网站数据更快地查看到结果或者发现可能存在的问题。

由于不确定应该对24小时的定时拍摄进行多大程度的收缩（视频应该在1分钟、10分钟还是中间某个值的时间内，展示完整的24小时拍摄？），我们决定测试一下，采用10分钟。该项目最激动人心的时刻之一是当我们首次尝试渲染24小时的手机网站数据时，点击Processing的运行（Run）按钮那一刻。把数据在一台MacBook Pro机上渲染成10分钟的定时拍摄视频花了约2个小时。结果生成了！

大家互相击拳祝贺后，开始观看视频。看了大约两分钟，我们意识到视频时间太长了——感觉视频太慢了！开始重新装载数据，创建一个历时接近1.5分钟的视频。经过几次尝试以及对代码和帧速率的调整，我们生成了新的视频。对较小规模的手机网站数据集进行渲染可以正常工作后，我们开始在Web站点和手机网站的混合数据集上尝试。由于数据量比之前大得多，渲染花费的时间也长很多——之前是2个小时，这次渲染花了24～36小时，这取决于其所用的机器的性能。

半自动化

最后，我们希望能够对整个过程实现自动化，这样程序接收到输入命令后，可以执行任何一天的定时拍摄渲染。该过程现在是半自动化的，我们可以很容易为同一天渲染多个定时拍摄的视频。举个例子，我们可以针对以下任何一种情况进行渲染：

- 世界地图的Web站点和手机网站的数据。
- 美国地图的Web站点和手机网站的数据。
- 世界地图和美国地图的Web站点的数据。
- 世界地图和美国地图的手机网站数据。

每种类型的数据需要花多长时间渲染？这取决于日期以及那一天是否是重大新闻日（即是否有很大流量）。平均来说，以下是该可视化大约的输入数据量和渲染花费的时间：

手机网站数据

数据文件约7MB、30万行

渲染时间约2个小时

Web站点数据

数据文件约70MB、300万行

渲染时间约1～2天

Web站点 + 手机网站数据

数据文件约77MB、330万行

渲染时间约1～2天

渲染定时拍摄视频的数据计算

在Processing应用程序内，我们每秒钟捕获15帧的视频。对于每一帧，在屏幕上绘制了1分钟的日志量。对于24小时的数据量，需要捕获1440分钟的数据。把每15分钟的数据渲染成时间长度为一秒的视频，则1440分钟的数据会生成96秒钟的视频（约1.5分钟）。

生成的视频有什么用

在本书要付印时，我们刚刚完成对数天的数据进行渲染。在纽约时报大厦28层的走廊上挂着10台监视器，播放着我们所做的一些可视化视频，包括这些流量图。其中有6台监视器自动播放本章介绍的定时拍摄视频；其他4台屏幕上显示的是《纽约时报》Web站点和手机网站当天全部流量的快照（美国和全球）。我们开始在公司内分享这些视频，并且探索更多的可视化来查看一天内可以发现哪些模式。我们还观察“重大新闻日”和“平常日”中，用户使用模式的差异。

结束语

我们从目前创建的可视化中观察到了一些有趣的模式，绝大多数如图16-5到图16-8所示。

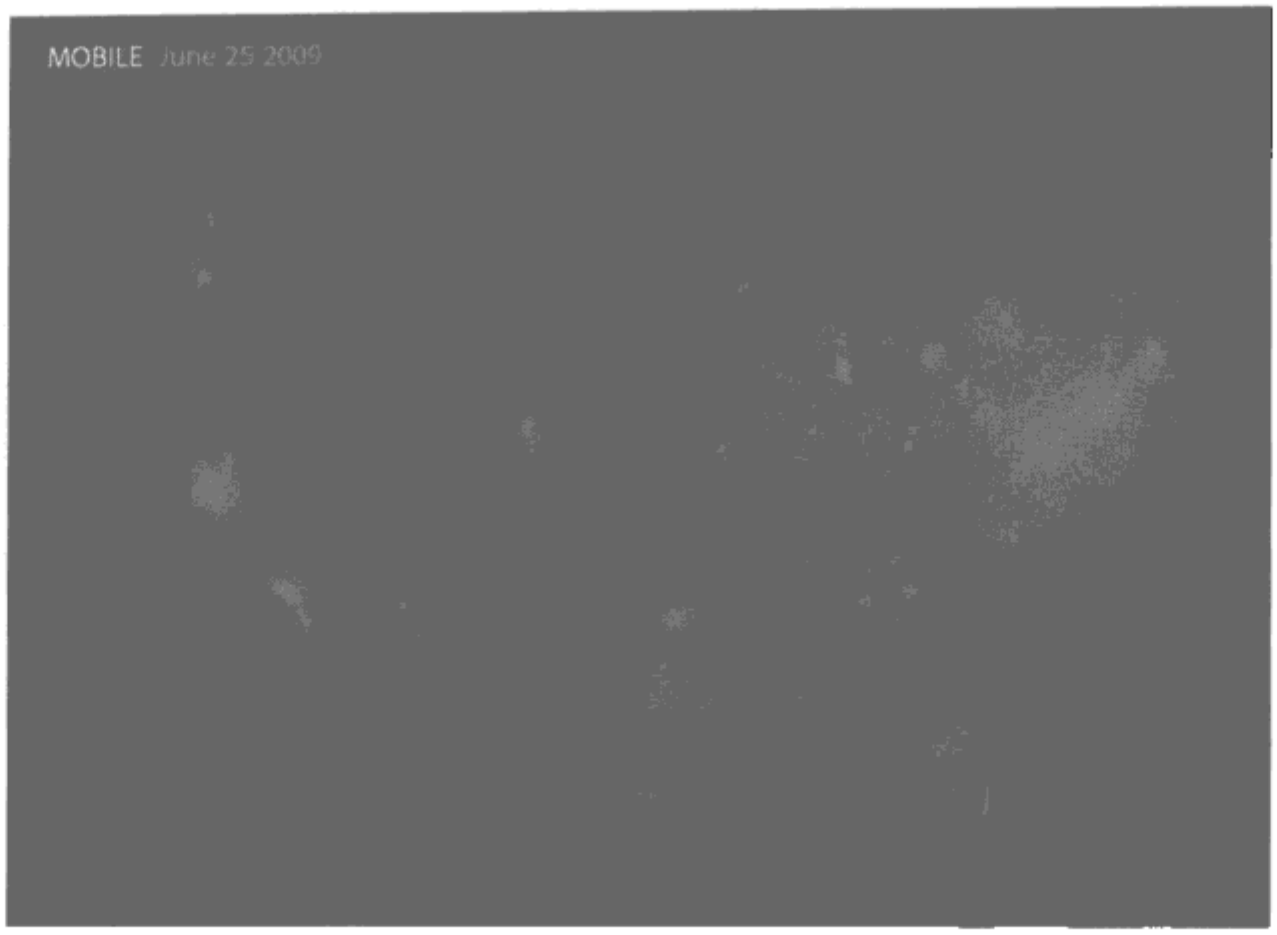


图16-5：手机网站mobile.nytimes.com在2009年6月25日这一天在美国的流量（见彩图127）[注2]

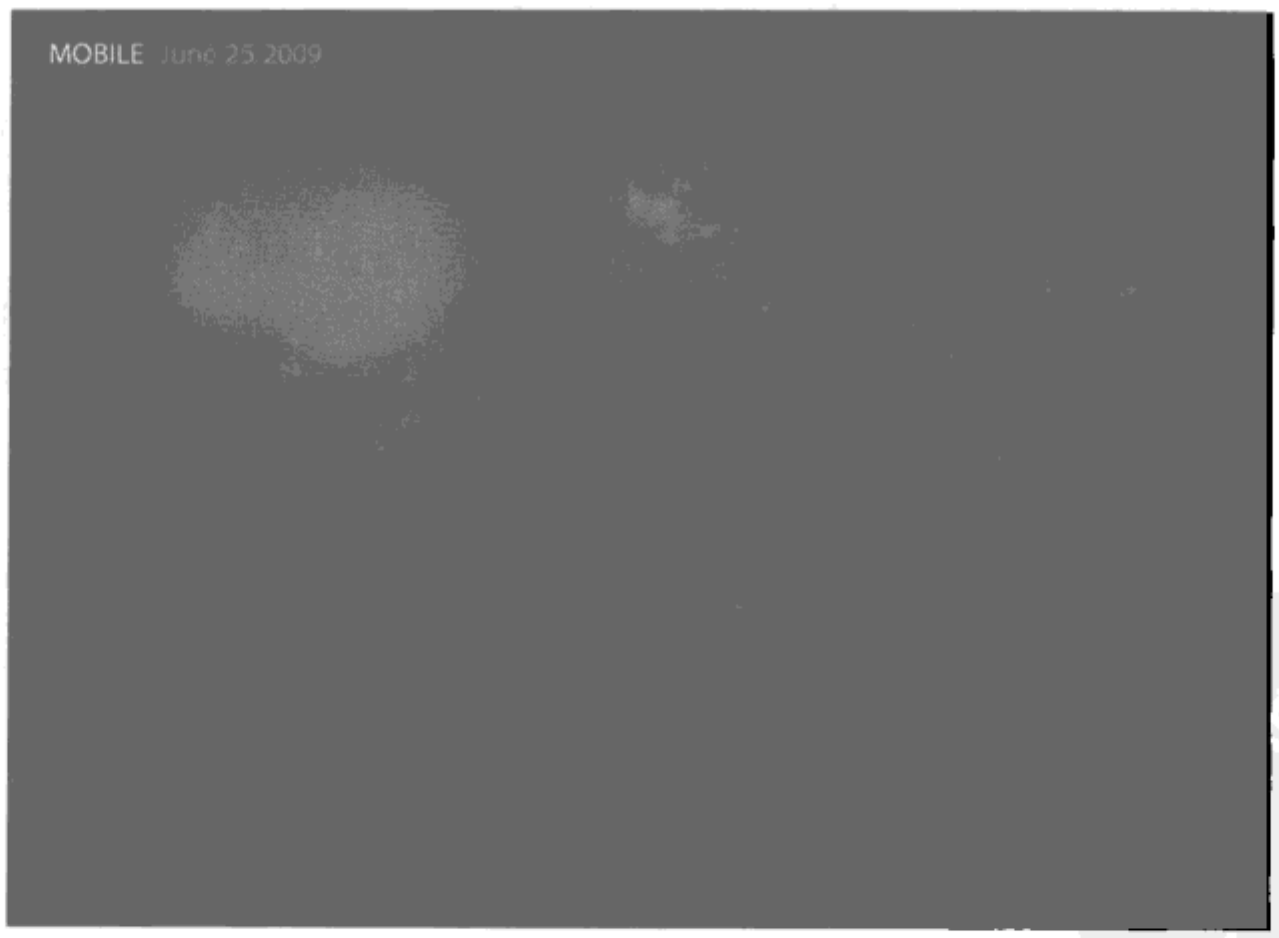


图16-6：手机网站mobile.nytimes.com在2009年6月25日这一天在全球的流量（见彩图128）

注2：两个大圆圈在Dallas、Texas、Waterloo 和 Ontario附近。这些城市都是手机网站的中枢城市（如Waterloo是黑莓/RIM的总部），大量的手机流量在到达我们的服务器前是先通过Dallas和Waterloo的代理服务器中转的。

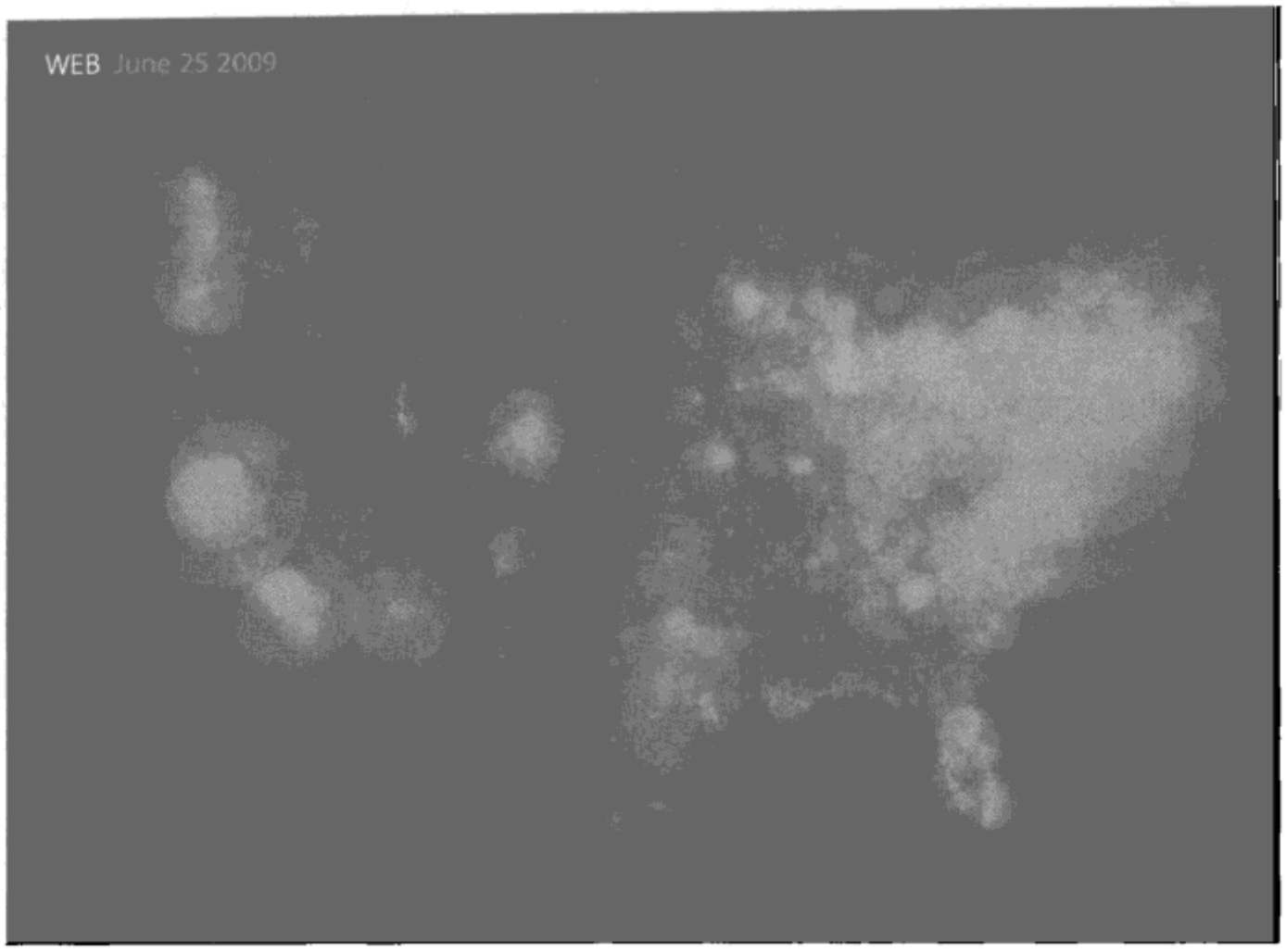


图16-7 Web站点nytimes.com在2009年6月25日这一天在美国的流量（见彩图129）

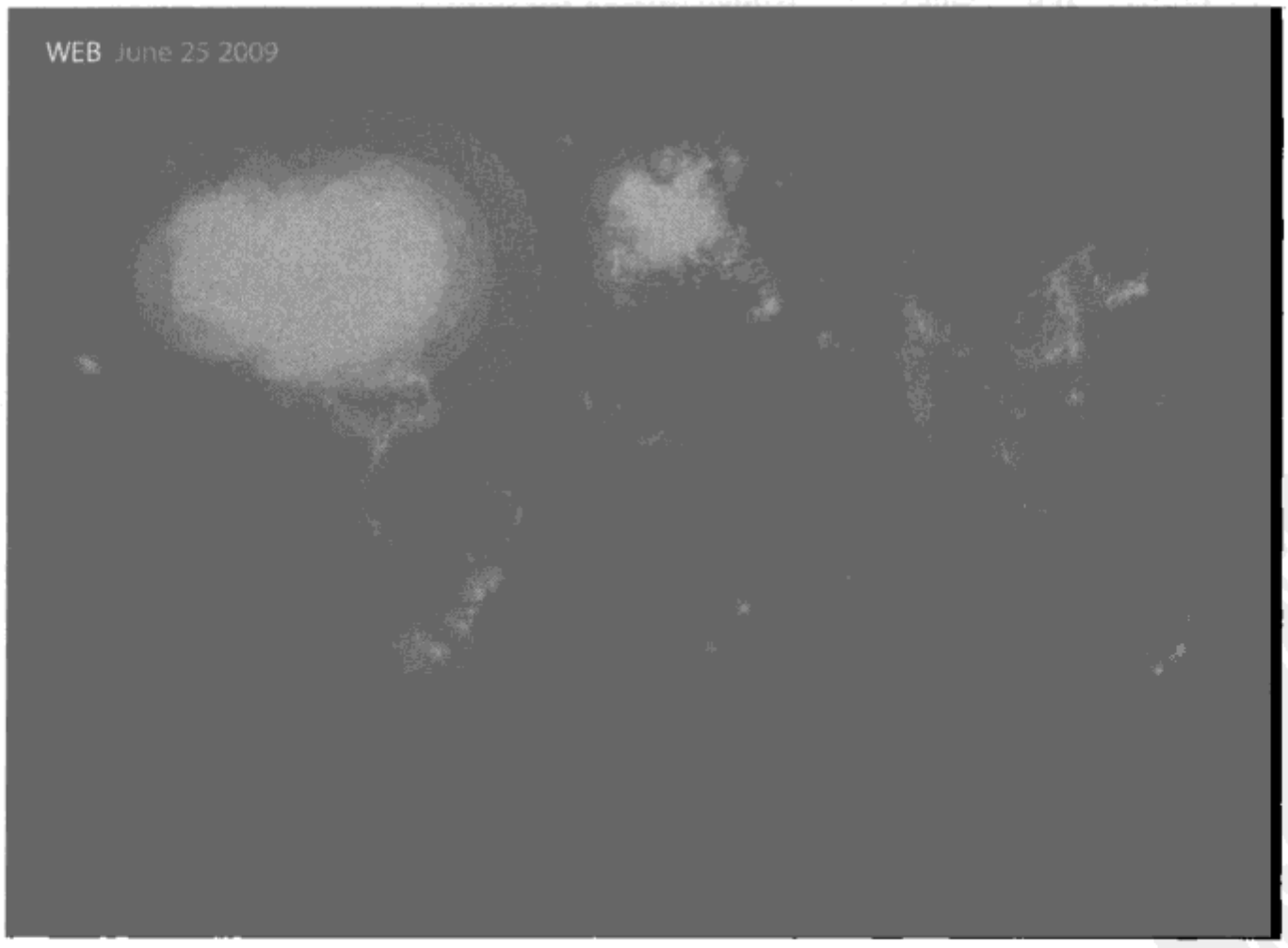


图16-8 Web站点nytimes.com在2009年6月25日这一天在全球的流量（见彩图130）

第一个模式是手机网站的流量在美国约早上5点或6点开始暴涨，该时段人们醒来开始去上班（尤其是在东海岸）。在约8点半或9点人们到达办公室前，手机网站流量一直很大；而当人们到达办公室时，Web站点流量开始第一次大增。Web站点的流量在一整天都很大（尤其是午饭时间），下午稍有点下降，很可能是人们在下班路上，而这时手机网站的流量又开始增加。这个观察和我们开始研究前的预期相同，但是该可视化进一步证实了我们的猜想。

另一个有趣的模式是Web和手机网站的国际流量都很大，非洲、中国、印度和日本某些地区的手机网站流量也很大。

我们相信从国际性和美国内的流量上可以观察到更多有趣的模式，由于可以从流量数据中渲染更多的视频，我们将会探索这些模式。我们邀请你也一起来观察，并告诉我们你所观察到的模式！你可以从下面的链接中查看一些可视化例子：*http://nytlabs.com/dataviz/*。

致谢

Noriaki Okada（《纽约时报》研究实验室的一个实习生）做了本周的可视化代码的实现和研究的大部分工作。可以在*http://okada.imrf.or.jp*上查看他的可视化作品。同时感谢Michael Kramer、Ted “Chevy's” Roden和Dick Lipton对该项目的大力支持。

第17章

深入揭秘复杂系统

Lance Putnam、*Graham Wakefield*、*Haru Ji*、*Basak Alper*、
Dennis Adderton 和 *JoAnn Kuchera-Morin*

媒体艺术和技术，

加州大学圣巴巴拉分校

多模式“竞技场”

走进现实版的“全息甲板（Holodeck）”[译注1]或“大脑”，进入一个从未见过的、震撼人心的新世界，这会是一种什么样的感觉？除此之外，大自然中迄今为止仍然未知的方方面面，如果我们突然能够亲身体验一下，那又会是一种什么样的感觉？实际上，这些问题也正是位于美国加州的加州大学圣巴巴拉分校纳米技术研究所AlloSphere项目[译注2]的科学家和艺术家们正在探索的。我们拥有一台设备，这台设备使得我们有能力对复杂、高维的数据和系统进行探索并与之交互——无论是亚原子粒子、移动接入网络（UMAN）装置抑或是一个完整的综合生态系统——在这台设备的帮助之下都可以成为能够让人亲身体验的世界。

AlloSphere是世界上最大的兼具科学性和艺术性的设备，也可以称作实验室。其功能涵盖“沉浸式可视化”（immersive visualization）[译注3]、“可听化”（sonification）和多模式数据管理。AlloSphere是一个三层楼高的球体，为改善其感知体验而进行了良好的

译注1：“Holodeck”，全息甲板，指的是《星际迷航》电影中的一种高科技设备。如想要了解更多，可以参考*http://memory-alpha.org/wiki/Holodeck*。

译注2：AlloSphere是加州大学圣巴巴分校的一个雄心勃勃的项目，试图以全新的视角去观察和诠释科学数据。后面会介绍更多。

译注3：“沉浸式可视化”即多维的可视化，用户可以融入其中去体验和感受。

调校，拥有一个360°视角的、超黑、非反射的大屏幕，屏幕四周环绕布置了一套多路扬声器阵列，整个系统位于一个无回声的工作室中，如图17-1所示。站在中央桥（见图17-2）上的多个用户在体验着立体图投影和空间声音的同时，还可以通过无数的多模式设备进行交互。

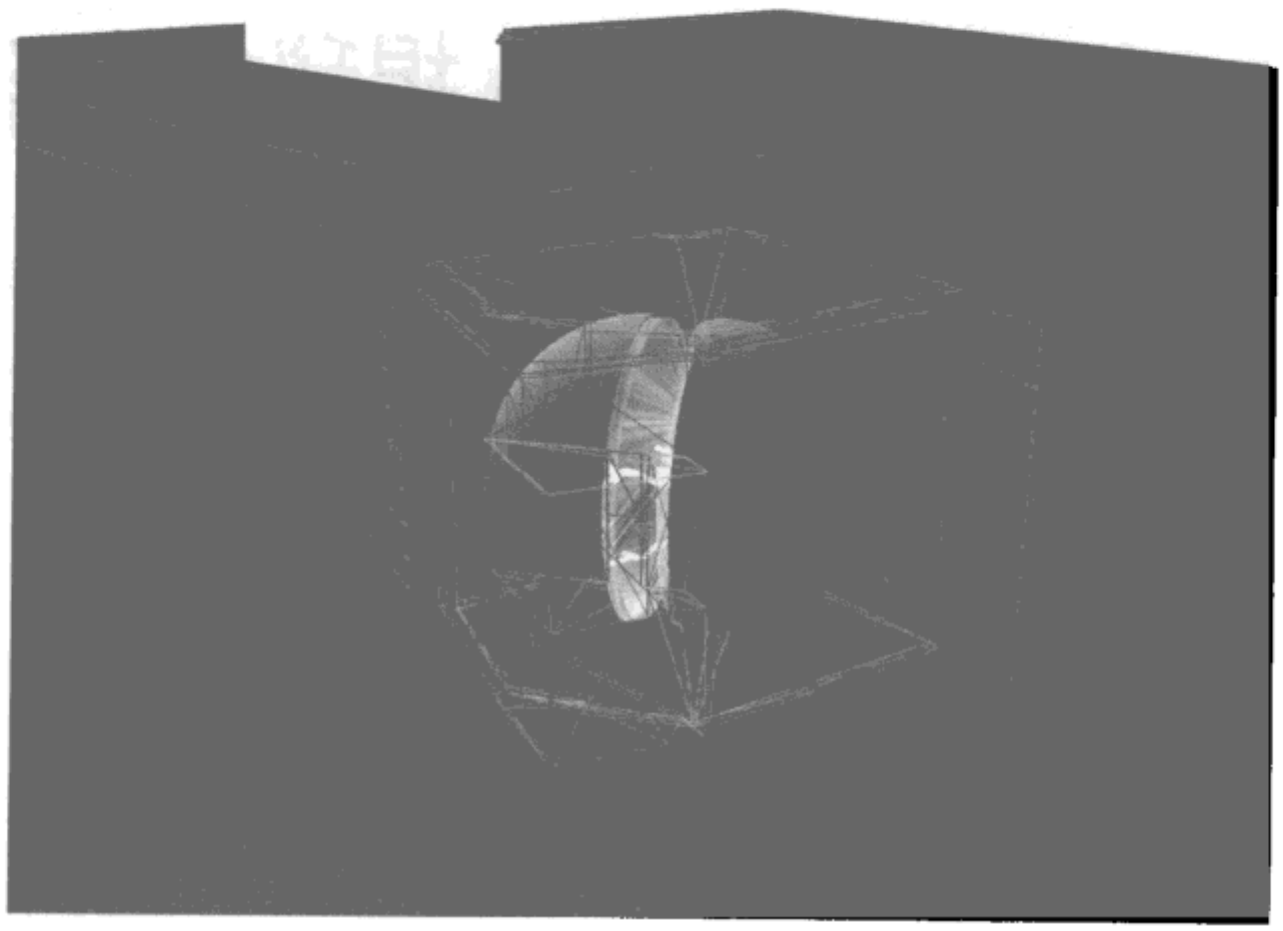


图17-1：真实比例的AlloSphere虚拟模型（见彩图131）

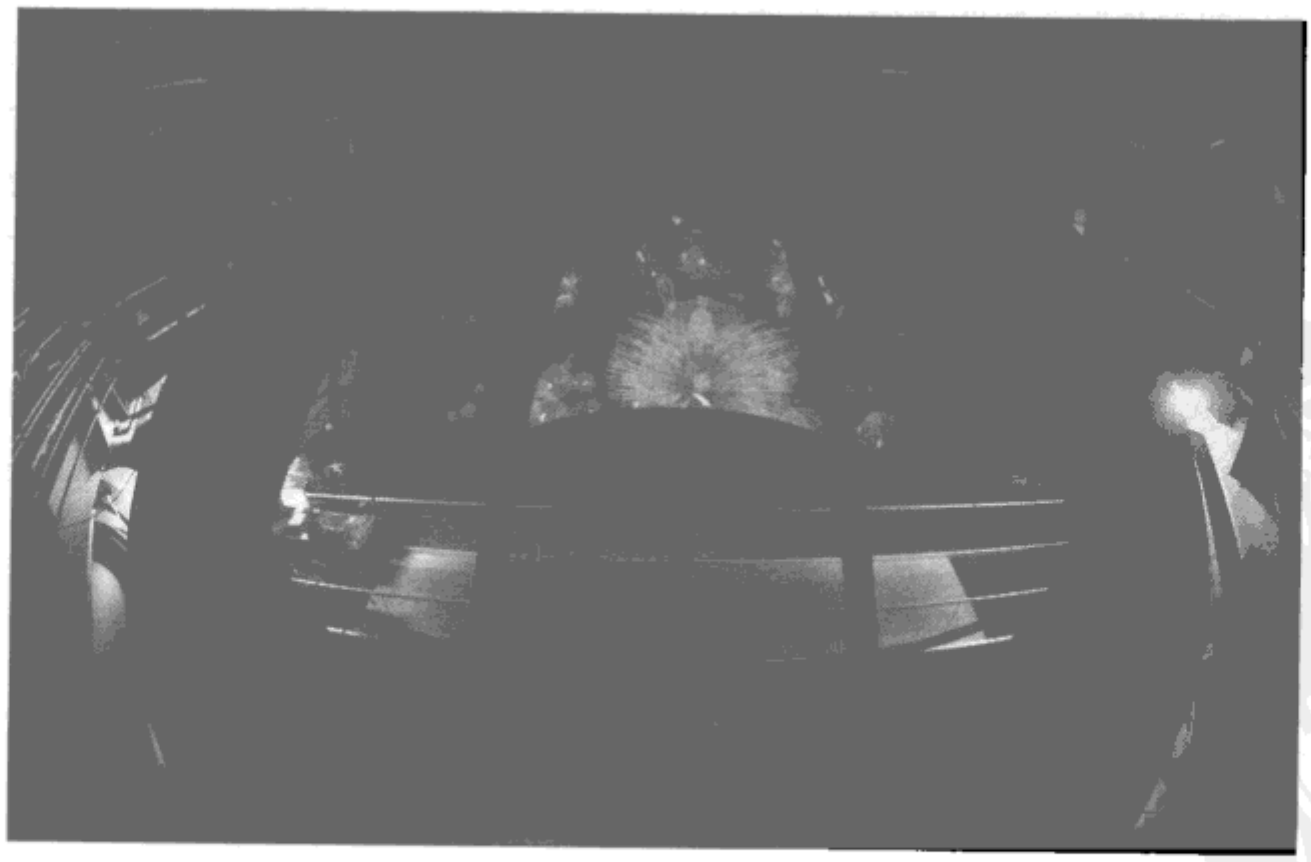


图17-2：AlloSphere的全景图（见彩图132）

AlloSphere的构想源于作曲家JoAnn Kuchera-Morin，希望能够找到一种通用的可以挑战视觉和听觉极限的多媒体设备，从而为艺术表现和科学探索找到新的模式。其目的是为各个领域的研究人员提供一个共同的场所来分享见解并共同探索类似于对称性、美丽、模式形成和出现等类型的基础问题。面对这样一个独一无二的机遇，我们期望能够建立起一种同时以艺术和科学这两门学科为基础而不局限于其中任何一门的前沿研究。这就需要对我们的创造性方法的基础因素进行全局性的反思：计算、数据、处理、感知、交互、融入和评估。

在AlloSphere项目中，艺术家、科学家和工程师一起工作，通过独特而且有趣的模拟和可视化方式来揭开新的世界的面纱，我们正在实现我们的“美即真”的理念。我们通过对有趣的方程进行可视化和可听化的方式帮助研究人员发现了这个真理。这些可视化为展开等式方程提供了优雅的解决方案。随着这些方程的展开，我们既能够发现其中的对称性也能找到残缺的对称性。

创造性思维的路线图

AlloSphere确实为新型跨学科研究提供了有趣的、互动的和多模式的环境。从一开始，它就采用了定量和定性相结合的方式来解决和发现问题。AlloSphere还提供了独特的体验方式——“开启”用户的所有感官——亲身体验复杂的系统如何随着时间展开。在确定如何以计算机语言进行描述以及如何以富于美感且对称的方式来展示系统的过程中，我们发现美丽和对称之间存在一些共同的主题。因此，构建美丽的可视化的挑战和机遇在于，在数学真理和感性表达中找到一种平衡，从而引出了一种认识论的新型的艺术和研究。

美丽和对称

毫无疑问，美丽在我们的感知中起着至关重要的作用，它和对称性密切相关。实际上，从古代Pythagoreans时期[译注4]开始，美丽和对称之间的关系就已经非常密切，Pythagoreans认为美丽的核心在于各个组成部分的比例以及它们之间的相互关系，而对称与和谐分别是视觉和听觉领域的相互关系（Tatarkiewicz 1972）。纵观我们的整个文明史，这个理论经久不衰。

事实上，对称性——其更正式的定义是“变换不变性”（invariance to transformation）（Weyl 1952）——是一些最深远的科学理论的本质基础，包括狭义相对论、守恒定律和旋理论。对称性在计算模拟上也起到了相当鲜为人知但却至关重要的作用。在古代，我

译注4：Pythagoreans是公元前6世纪希腊哲学家、数学家。

们只能观察到周围的自然形态；今天，通过计算能够支持的比例控制，我们能够精确地自主构建生成出具有复杂的自然模式的系统。在这些复杂的模式的核心中，我们确实发现了对称性。实际上，对称性经常能够指导我们在数据中搜寻有意义模式的研究。

计算方法

计算和数学为科学模型和艺术实践提供了很好的共同语言。计算是科学模拟的重要工具，而且是艺术的开放性素材。通过设计和实例化复杂的自治系统，我们敞开了基于部件人工合成的新的知识领域的大门[注1]。

不管我们想要问的是什么问题，计算要求我们必须对数据的基本组件有正式、确定的描述，并对实时处理中的局限性有充分的考虑。我们发现，特别是基于物理的模型，需要处理的数据主要包含与空间和/或时间关联的值。这些值表示特定的内部强度，比如速度、流量、频率或复杂阶段，而且通常与空间的位置和/或时间关联。我们采用的很多可视化技术需要筛选出某个特定位置（如交叉位置）的值或某个特定值的位置。

程序执行时如何对这些值和位置进行初始化是不一样的。值可以是显式的（比如定期采样点或位置/值组合对）或隐式的（使用公式或算法实时计算）。同样，位置可以是显式的（作为位置/值组合对）或隐式的（根据规则网格维度确定）。

在各种不同的计算模型中，我们观察到了数据存储和处理的3种通用模式：

- 作为样本值的规则网格。
- 作为位置/值组合对的集合。
- 作为位置函数。

前两种模式之间的区别与计算机上图像的两种通用的展现方式间的区别相同：基于光栅（作为像素矩阵）或基于矢量（作为用曲线连接的一组点的集合）。第三种模式看起来更像一个黑盒子，输入是一个位置，输出是一个对应的值。

每种模式都有自己特定的优缺点。网格允许模型中包含未知信号量和局部交互，但是它需要容易导致频谱混叠的取样，同时当要以合适的分辨率进行系统建模时可能需要消耗大量的内存。相反地，位置/值组合对模式和函数模式支持高分辨率的、任意的空间分辨率，但是实体之间的交互建模计算会很复杂。

一个很自然地遵循这些模式的概念划分是介于“时空领域”（spatiotemporal field）和

注1：举个例子，在人造生物领域，为了更好地理解生物，人们试图通过digito软件重构该过程，但是它引起了很多关于人工创造的讨论。

“自由媒介”（free agent）之间的。域（field）是一种空间维度上的规则网格（时间维度可能是变化的），是复杂系统的底层。它们定义了整个结构的底层架构和系统的动力学。域表示如密度分布、流体和波之类的事物。很多学科存在域的概念：发育生物学包含形态域和遗传观，进化生物学包含适应观（fitness landscape），而物理学包含量子学领域和波函数。媒介（agent）是位置/值组合对的集合，是复杂系统的上层。媒介代表实际的离散实体，在连续空间维度上则可能是移动的。媒介使我们能够更为细致地观察整个系统的部件并过滤查看其不变模式，进而能够更清晰地观察域。此外，媒介往往通过对一个域的值的读写来相互交互。

作为过滤器来解释

我们的工作不仅涉及复杂系统的设计和实例化，而且包括过滤器的组建。过滤器与系统的设计和实例化同等重要，其定位是将庞大的计算/数学空间简化为可以从中观察并提取涵义的形式。换句话说，可视化和可听化都涉及材料的组织（组成）和作为我们的研究目标的模式的展示（解释）。

我们经常问自己这样的问题“在数据或系统中我们要寻找的是什么？”对于这个问题，我们可以回答说是正在探索一些有趣的模式，这些模式能够揭示系统展开过程中的一些本质特性。此外，我们发现利用对称性有助于引导我们找到重要的模式。我们经常应用的可视化技术如等值面、等高线、流线和粒子流，显示了值（或派生值）等价或不变的系统的方方面面。这些“袖珍对称”（pockets of symmetry）说明了系统的相似性，也为对自己的行为和模式的更深入理解建立了一个良好的起点。我们知道，对称性太多会降低其重要性，而太少又会使其显得过分重要；过滤一定是落于有序和无序这两者之间。这一原则也适用于时间：兴趣模式必须使其特点保持足够长的时间以确保能够被分辨出来，但是其变化也必须足够频繁以吸引眼球。

创建过滤器是一个自适应的过程，它可以出现在一个模式中，也可以跨越多个模式。我们发现多模式展现对于揭示隐藏于数据中的或者不明显的对称和不对称性是很重要的。有时，数据集或处理过程的最自然的感官模式无法充分表达其结构的重要特性。举个例子，我们发现波形的对称性更容易被观察到，而空间数据中被略微破坏的对称性更容易听出。我们使用计算转换能力对不同模型进行映射，寻找一种平衡使得可以给出对当前现象进行更完整地描述的大脑图。实际上，有证据表明，大脑记忆系统包含“情节缓存”（episodic buffer），它可以把视觉和听觉感官信息集成到和长期记忆交互的多维代码中，因而后续可以影响长期的学习过程（Baddeley 2000）。

基于媒介的模式在我们的数据和系统的过滤和展示中扮演了一个至关重要的角色。媒介在视觉和听觉上都很有吸引力，因为它们可以更流畅、更连续地运动，其运动也不会局

限于离散网格中。因此，媒介允许我们在一致的结构中观察系统中的主导模式，从而降低噪音。使用媒介的一个例子是使用连续平滑的曲线显示粗糙的采样域。

项目探讨

在本章，我们将讨论6个研究项目，涵盖从艺术/科学数学抽象到基于实际的科学数据和理论的精确的计算模式的多模式表现。我们讨论的范围非常广泛，从真正的生物数据到仿生进化演化算法以及原子世界；然后又从原子层探讨到单一氢原子的电子层，我们最后将探讨展示电子自旋连贯运动的一个项目。

Allobrain

Graham Wakefield，John Thompson，Lance Putnam，Wesley Smith和Charlie Roberts（媒体艺术和技术）

学科主任：JoAnn Kuchera-Morin教授和Marcos Novak教授（媒体艺术和技术）

在Allobrain，我们穿越了人类大脑皮层（见图17-3）。使用功能性磁共振成像（fMRI）的结构化组件数据创建了一个“太空”，通过它遍历探索“世界”。原始数据将大脑的代谢活动密度值映射到了大脑空间的各个网格内；可视化包含数据集的两个“等值面”（isosurface），该等值面是根据fMRI扫描得到的大脑组织的密度来选择的。（等值面是由在某一个维度取值相同的点构成的三维等高线。）在Allobrain这个“世界”里，“搜索媒介”（search agent）通过自动导航的方式挖掘出数据，在空间上和视觉上展示出来，然后对兴趣区域进行聚类，并通过音乐通知我们。“漫步者媒介”（Wanderer agent），对特定大脑区域颜色编码，随机访问数据，查找高浓度的血液密度。“漫步者媒介”还可以接收命令，发送结果到屏幕中心，而且通过音乐表示血液密度等级，音调越高血液密度越高。

想象那些不仅适合于医疗诊断而且适合于认知和感知的心理研究的应用：Allobrain在单个视图中融合很多维度的信息的方式，有助于尽早发现细胞紊乱，也有益于理解大脑是如何工作的。实际上，视觉艺术家兼跨领域建筑师Marcos Novak——Allobrain世界及大脑之父——构想出该项目正是为了研究审美的神经学基础。他对于自己的工作有如下描述：

> 当我们说某些事物是“美丽的”时，大脑的哪些部分参与了该评估，它们是如何参与的？因为在艺术审美上人们的观点千差万别，研究“美”的更好的方法可能是专门研究仅有一个或者几个实例组成的封闭系统，尽可能深入地了解这些实例，然后确定在该实例上的特征是否可以泛化到其他事物。
>
> 特别地，这项工作旨在构建一种情景，在该情景中，绝大多数使事物“美丽”的元素都可以调查。具体如下：

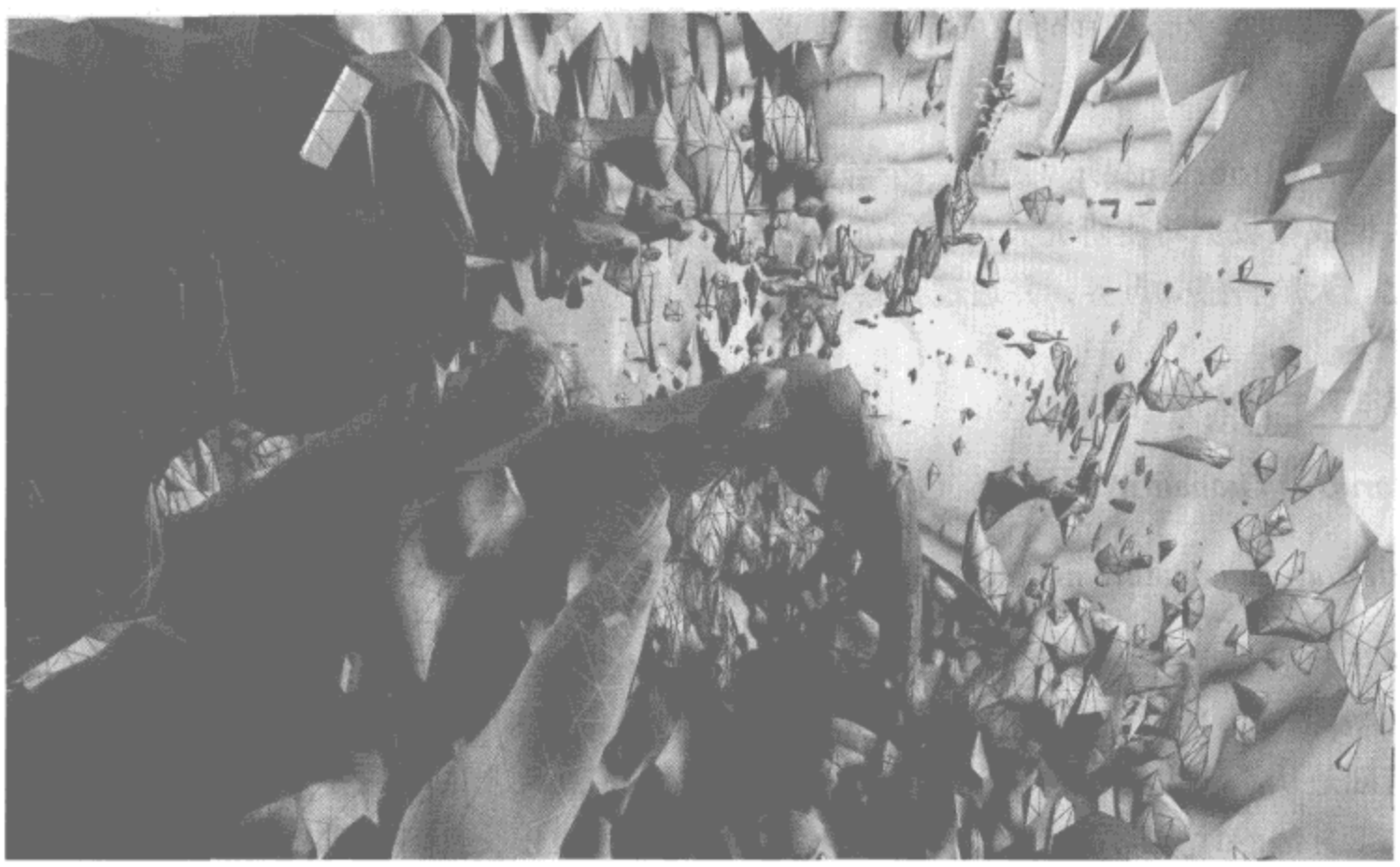


图17-3：Allobrain的内部图（见彩图133）

- 这项工作是否被评为“美丽”。
- 其生成方法和机制。
- 工作的创作者、鉴定员和调查员。

此外，我们的目标（科学上和艺术上）是创建一条反馈回路，在该回路中，艺术影响大脑，而大脑生成新的数据，这些新数据创建新艺术，而艺术又反过来影响大脑，而大脑又生成新数据，如此反复循环。

为了创建该过程，我实现了一个生成算法，它可以生成我本身无法具体给出的激发因子（stimuli），而这些因子来源于我对“美”的反应（视觉上和空间构成上）。激发因子包含以下任意一种：1）交互式的/生成的移动/变化的图像；2)该图像的记录视频，可以使用fMRI成像机重放。fMRI成像机给我播放了这个视频（我之前从未见过）。观看视频的过程中，每当遇到在我看来非常美丽的场景时，我就点击一下按钮。对按钮的点击动作会被计时，因此，借助时间就可以将点击动作和那一瞬间的大脑活动关联起来。fMRI成像机的数据被转换成一种沉浸式的环境或者说“世界”。这一过程可能引发两种可能：从科学角度看，这种转换使得结构化的和功能上的数据能够以常规情况下不可能的可视化方式来查看。从艺术角度看，它提出了崭新的艺术形式，在该形式中，大脑（以及思想）生成世界，而世界改变思想，思想又生成新的世界，如此反复。在这两种情况中，都可以构建反馈回路，在

该回路中，用户的反应有助于生成激发因子，该因子又会激发反应，从而放大了效果影响。

目前，Allobrain揭示思想的一个静态快照。随着项目的推进，具有实时交互功能的核磁共振成像数据将使研究人员能够沉浸在自己的想法中，观察如Novak所描述的转换和变化。大脑将感知世界，并通过其感知改造世界。

人工自然

Haru Ji，Graham Wakefield （媒体艺术和技术）

http://artificialnature.mat.ucsb.edu

现在我们将话题从原始的生物数据切换到作为生命之本的过程和系统。“人工自然”（Artificial Nature）是一门跨学科的研究项目，是一种受生物启发的虚拟艺术装置，其基础是从系统生物学、人工生命、复杂性科学中演化而出的生成模型而不是经验数据。人工自然的计算世界是一个生态系统，由在动态环境中交互的有机体组成，观众可以和这些有机体进行交互。

环境是基于流体动力学的空间域。流淌于其中的简单粒子流具有不同的营养类型（色调）和能量水平（亮度），彼此互动交互。这些粒子为生物体提供代谢燃料，可以作为独立的媒介。这两种营养物质的摄入和代谢废物的处理都是生存和繁殖的必要条件。

生物体的外观和自主活动是由对其本身所处的位置（空间上和历史上）的遗传描述的解释决定的。举个例子，积累足够的能量可以触发一些生物体通过无性繁殖生成下一代，只存在很小突变概率。这些生物体的形状是基于Boy 曲面方程（Boy 1901），并随着生命周期不断变化来表示逐步的增长和发展，而健康还是使用不透明度来表示。

摄食、繁殖和探测邻居等活动都伴随着各种不同的啁啾般的歌曲，它在AlloSphere是完全空间化的。这些声音音质明朗、短暂而富含信息、紧密聚合，使得可以很容易彼此区分、定位并连接到视觉活动中。

观众可以使用“六度自由”（six-degrees-of-freedom）导航设备自由、无止境地探索世界并间接地影响世界，正如他们儿时在溪流或沙坑玩时，不时地“激起千层浪”。通过摄像头、麦克风以及时不时的触摸收集到的感官数据开始成为生物体必须适应的环境条件。流体的湍流也反过来影响观众的探索。整个生态系统，包括观众本身，生成了连续模式的“自然美”（见图17-4和图17-5）。

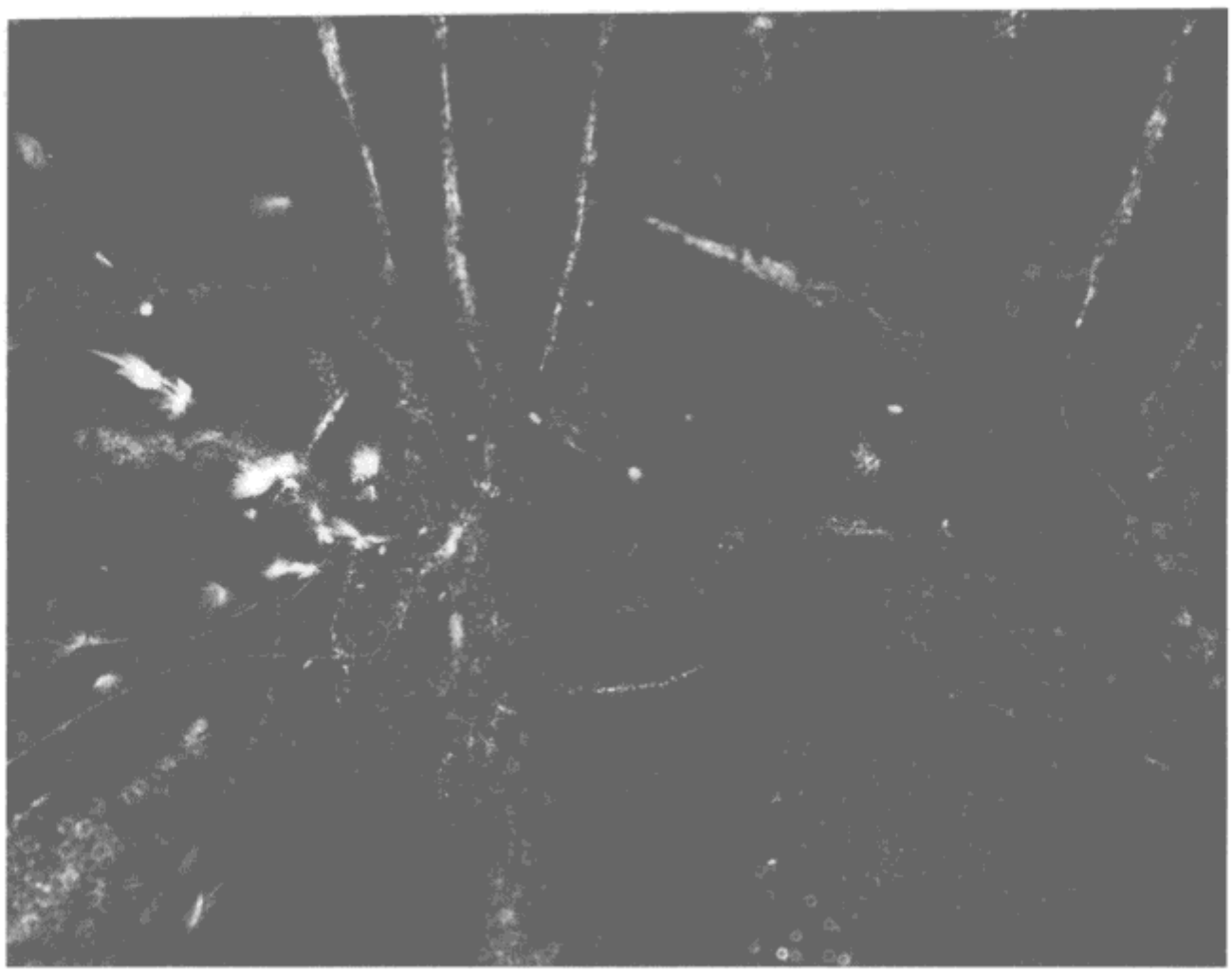


图17-4：在人工自然流体领域生成和分散的人工营养成分（第1版本：“无限博弈”，见彩图134）

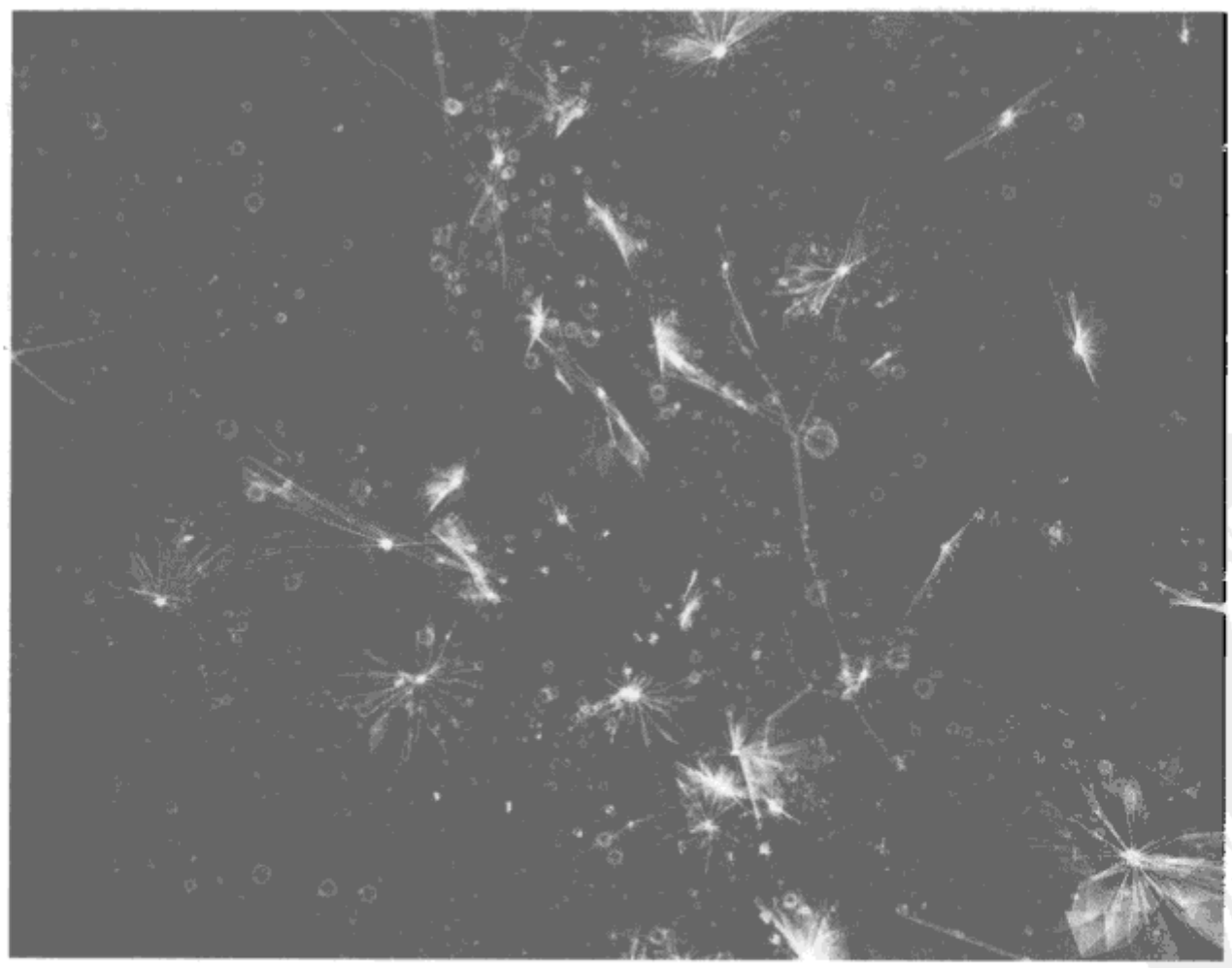


图17-5：在人工自然领域生长和交互的人造生物体（第2版本：“流体空间”，见彩图135）

我们在想，什么样的艺术形式可以在AlloSphere空间中自然进化。人工自然作为一种虚拟化艺术作品，很自然地回答了这个问题。人工自然是在一个替代性环境中的全新的体验——一个展现无限可能的世界。人工自然的开放性本质正是基于人类的复杂的自适应系统。这些基于媒介的技术给自己带来了真实模拟，而多模式交互使观众也融入了生态系统网络中。

人工自然本身是一个有较大发展的项目。随着我们在其中嵌入更多的维度和关系，新的模式潜能、结构、涵义和美丽开始出现。

氢键

Basak Alper，Wesley Smith，Lance Putnam 和 Charlie Roberts（媒体艺术和技术），Anderson Janotti（材料研究实验室）

学科主任：JoAnn Kuchera-Morin教授（媒体艺术和技术），Chris G. Van de Walle教授（材料研究实验室）

谈完生物和宏观世界，我们现在开始探讨原子世界以及无污染科技的新材料化合物——多中心氢键。它是制造透明太阳能电池和低成本显示设备的非常关键的一环。通常情况下，氢和其他元素一起形成共价键（指的是氢和其他元素共享一对电子——因为氢只有一个电子，它每次只能形成一对共价键），但是在氧化锌晶体中，它和4个锌原子形成共价键，生成一个四面体键结构。

加州大学圣巴巴拉分校（UCSB）固态照明和能源中心材料科学研究所的同事发现了这种独特的氢键结构，希望由我们以他们现有的工具所无法做到的方式来从视觉上和听觉上展示他们的模拟数据。我们拿到的数据是氢键晶体的三维晶格的静电电荷密度。对这类“体数据（volumetric data）”[译注5]进行可视化具有很大挑战，因为无法通过自然途径看到坚实的固体内部。

可视化体数据的一种通用的方法是绘制等值面来显示内部曲率。对电荷密度应用等值面，键结构形状更加清晰易见，这种方式和在地图上使用等高线来表示不同高度变化类似。在局部数据域中查找最大值/最小值对于科学家也是一个非常重要的功能，它能够帮助人们识别出键中的临界区。我们通过使用梯度场描述体数据域的方式解决了这个问题。刚开始，我们没有得到任何结果，因为数据抽样时所用的采样间隔远大于查找区域。我们解释了可视化算法的工作方式，从而说服了科学家们生成分辨率更高的数据。得到高分辨率的数据之后，在梯度场绘制零值等值面成功地说明了局部极大值/极小值域。

译注5：在医药学应用中，通过MRT或CT得到的数据称为体数据。

为了找出更多的局部极大值/极小值域形状，我们使用了称为“流线”（streamline）的可视化技术，它生成沿着向量场流动的曲线。我们将流线的起点定在氢原子中心附近，允许它顺着梯度场递减的方向向外流出，使用色调表示运动速度，红色代表快，绿色代表慢。虽然我们的科学家伙伴最初觉得流线很怪异，但是最终流线证明了其有效性，它们能够在键结构的临界区融合在一起。

我们对标准可视化工具进行了扩展，增加了可视化模式的选择功能和在单张视图内叠加选定的多种可视化的功能（见图17-6）。在一张视图中包含不同层次的信息需要绘制一张图，能够最大限度地降低混乱和模糊。为此，我们使用了一种自定义的照明算法，它减少照明扩散从而突出等值面的曲率。我们对透明和线框渲染进行了混合，减少存在多个透明区域的错觉。我们发现流线和等值面是自然的视觉补充，因为它们能够在垂直方向上显示信息。同时，在视觉上看，显示流线和等值面要比显示多层等值面的效果更好，因为流线和等值面在视觉上很容易区分。

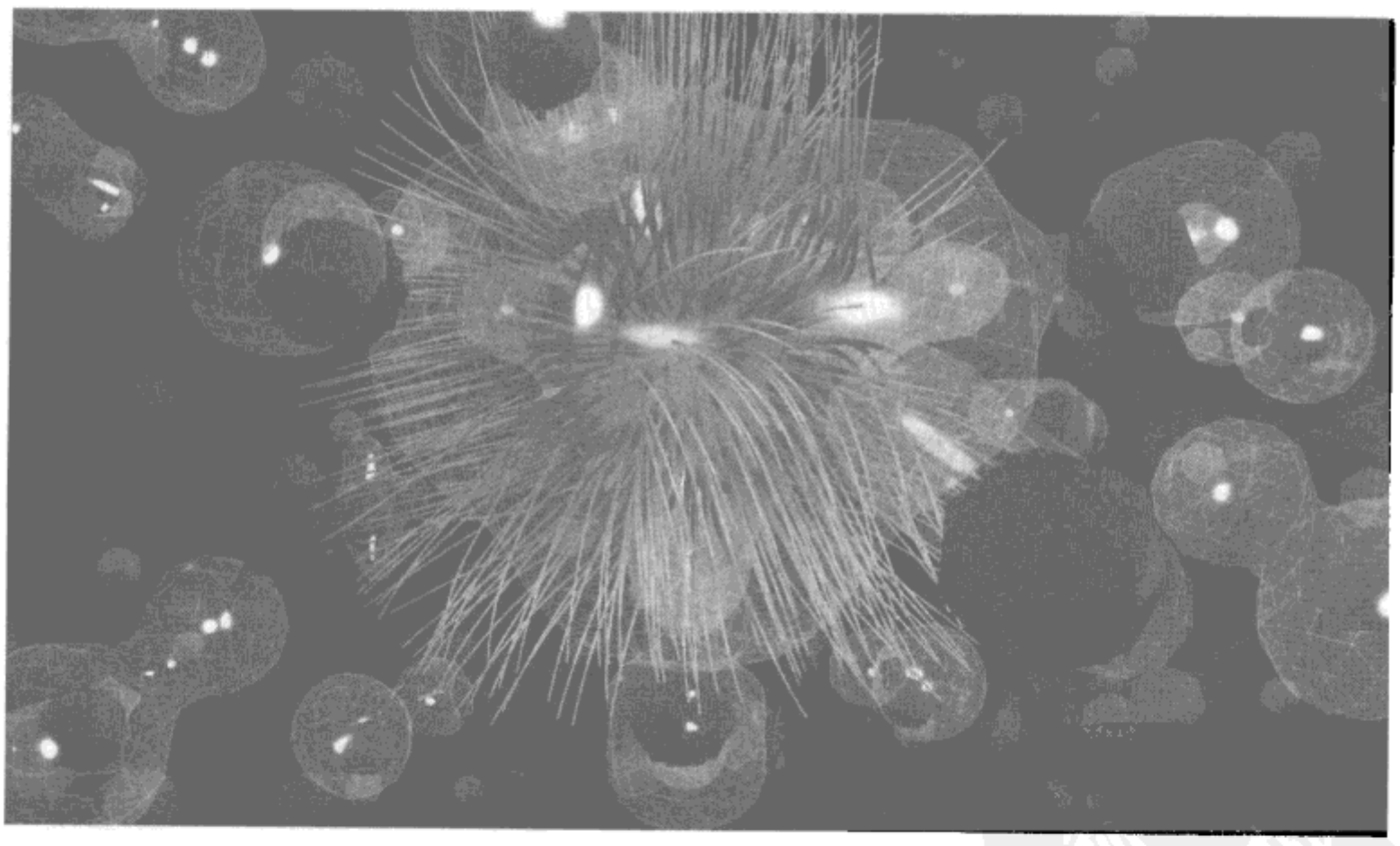


图17-6：包含4个锌原子的四面体氢键的特写图（蓝色，见彩图136）

除了可视化，我们使用空间音频来定位晶体中键的位置和用户的位置（见图17-7）。为了给原子添加音调特征，我们按照10个八度音阶，根据氢、锌和氧的放射频率来调节其音调，生成氢、锌和氧的放射光谱（相对电磁辐射）。

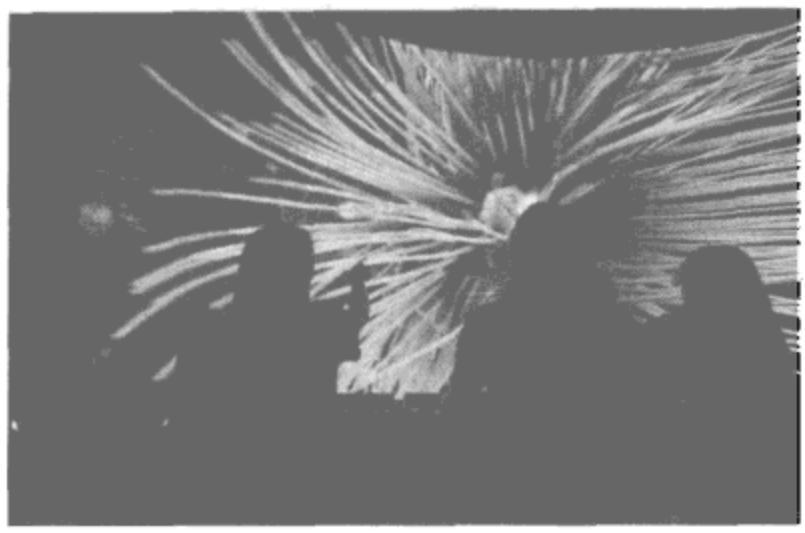


图17-7：沉浸于氢键中的研究人员（见彩图137）

由于数据具有时间不变性和三维特征，因此如何为它配音是一个很大的挑战。我们提出的一个解决方案是扫描参数曲线的密度场。我们使用Lissajous曲线[译注6]，因为它展现出高维度的空间对称性和平滑性，最大限度地减少音色失真。虽然该技术不具备视觉补充，但它生成的特征化音调有助于定位氢键，从而产生更完整的多模式体验。

氢原子

Lance Putnam 和 Charlie Roberts（媒体艺术和技术）

学科主任：Luca Peliti教授（Kavli理论物理研究所）和JoAnn Kuchera-Morin教授（媒体艺术和技术）

现在，我们的话题从原子晶体切换到更小的空间单个氢原子电子云。人们对氢原子轨道的形状有很多了解，物理学家可以轻而易举地在大脑中描绘出它们。然而，当两个或者更多随时间变化的轨道叠加后产生的电子云将很复杂而且很难通过个别公式分析。此外，数学公式和静态图片无法捕捉复杂的、随时空演化的动态特征。

我们的这项工作旨在通过电子波函数的交互式可视化和可听化，创建“类氢”原子的多模式体验。我们把原子轨道模拟成随时间变化的Schrödinger方程的解，包含Coulomb的静电力法所描述的球状静态势。在这个模型中，原子核和电子之间的关系类似于装满液体（电子）的碗（原子核），其区别在于液体可以包含很多不同的静止形状，而且可以延伸到碗外面。为了计算，单轨道的时间不变结构预先计算好并存储在三维晶体中；然后，在模拟过程中，它们分别独自进化，而且空间上混合在一起。我们对一些预设置的轨道叠加进行编程，观察如光子放射和吸收的动态行为特征。

我们尝试的第一种可视化技术是把电子云渲染成3D立体声。这样可以更易于查看波函数

译注6：Lissajous曲线，其数学定义是指两条沿着互相垂直方向的正弦振动所合成的轨迹。

的全局外形，但是难以查看其内部、更局部的结构。为了解决这个问题，我们在立体渲染时把媒介集合叠加起来，这些媒介在波函数中沿着不同的流运动。通过这种方式，我们可以同时理解云的全局和局部结构。我们发现彩色线条在映射维度数量、可视化复杂性、计算高效性之间提供了一种合理的平衡（见图17-8）。彩色线条媒介给我们提供了3种色彩的内部维度、4种方位的空间维度以及可以用于映射的长度。我们使用色彩来区分不同的流和方位类型来表示方向。此外，线条的亮度和长度不同，这样可以平滑地把媒介从展现中淡入或淡出。

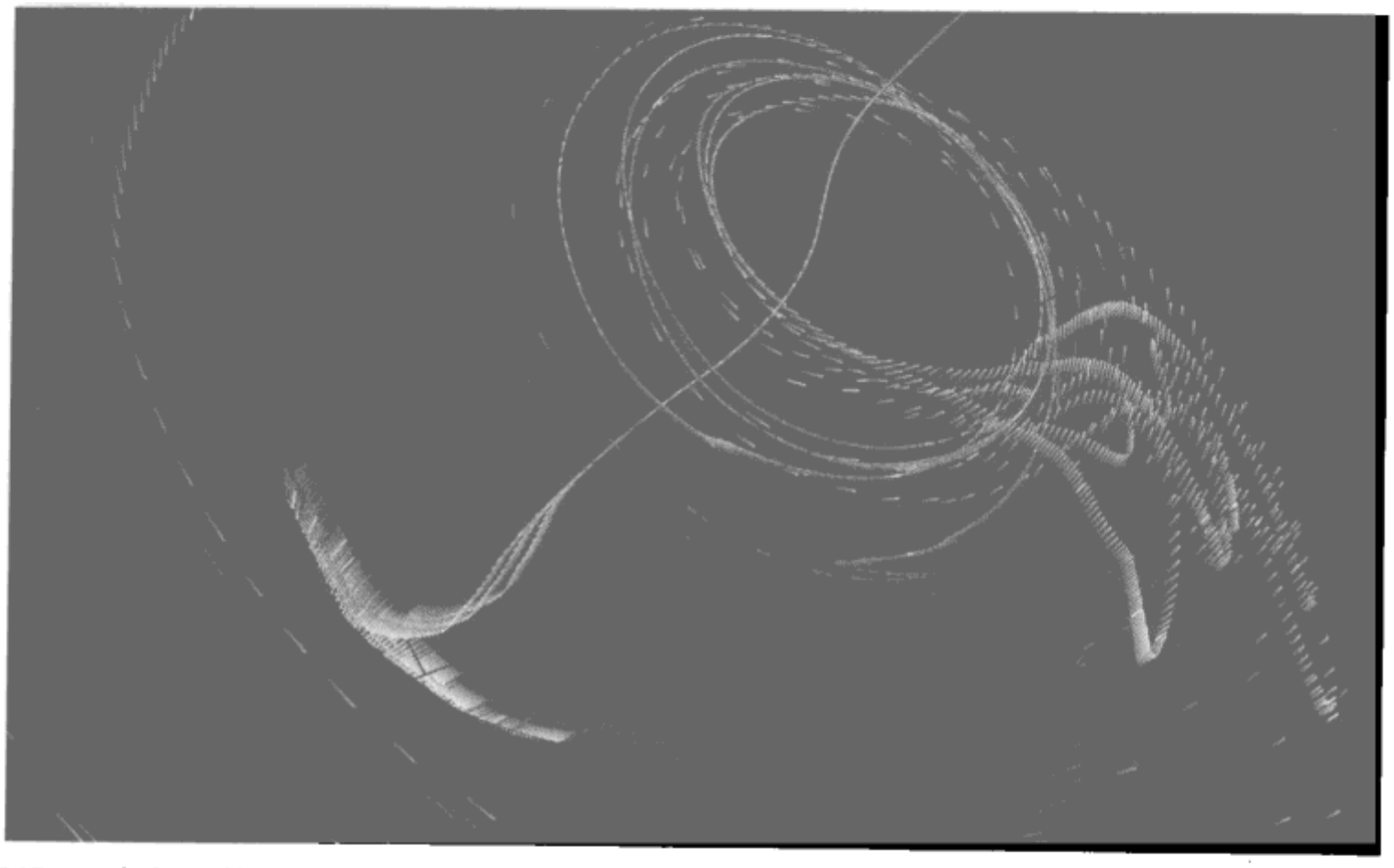


图17-8：氢原子的光放射配置（见彩图138）

我们还想使用声音来通知特定事件类型——比如某些形状类型的出现和消散——只在云内出现。为了做到这点，我们使用了一种称为扫描合成（scanned synthesis）的合成技术的变体。我们以类似于录音带的读取磁头的方式扫描媒介，然后聆听其所在位置的波函数的振幅。通过改变扫描速率，我们可以改变声音的音调。音调低的用于显示局部形状变化效果最好。而音调高的用于表示全局特征效果最好。我们还发现给不同类型的媒介分配不同的音调类型（对八度音调进行分割）很有效，这样可以在听力上互相区分开。这种扫描方法可以成功地提醒我们媒介聚类何时以及在何处形成奇异点或吸引域，但是关于特定形状的形成的通知的效果不太好。更全面系统地表示系统的方法不是增加单一方式，而是采取多模式方法，使得视觉上可以显示整体形状，听觉上可以感知局部结构随着时间的变化。

实现该展现的一个意想不到的结果是，波函数模式从单一轨道模式转变成到混合模式

中时展现出来的复杂性和丰富性，如图17-9所示。出现的组成模式和部分没有明显的关系，从数学方程上看一点都不明显。我们发现作为简单且众所周知的物理机制的波的干扰，在思考创建复杂模式和新兴行为时，可以作为强大的概念。

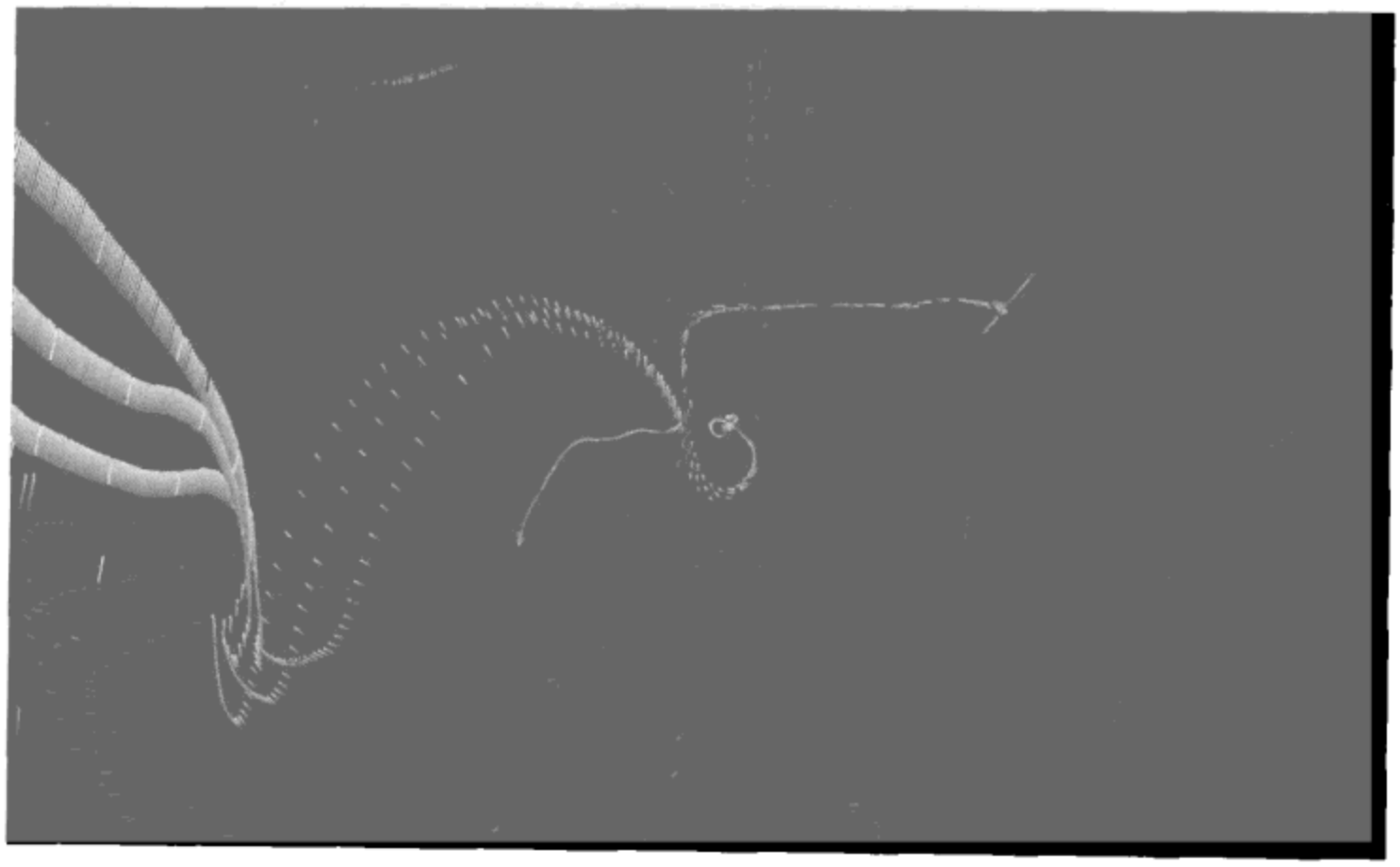


图17-9：氢原子的高阶轨道混合（见彩图139）

纺织氢原子

Lance Putnam（媒体艺术和技术）

学科主任：Luca Peliti教授（Kavli理论物理研究所）和 JoAnn Kuchera-Morin教授（媒体艺术和技术）

在这个项目中，我们期望使用更完整的包含自旋因子的物理模型对之前的氢原子项目进行扩展。我们还希望从原始的对波函数空间的抽样提升到更高的空间分辨率。我们决定不再预先计算和存储轨道，而是实时计算一切，这样我们将能够得到空间中所有点的波函数的准确值。从这个意义上说，波函数的计算表示形式从晶体值变成了位置函数。这种新的方法也使我们有机会以新的视角来观察媒介作为通用目的的可视化和可听化工具时的效果。这些媒介不仅能够显示波函数通过个体运动产生的流，而且能够表示其他一些状态，如其振荡阶段。此外，媒介上的软件程序可以以类似合奏的方式来创建更平滑、更紧密连接的形状。

我们开始通过对网格线条上的媒介进行定位，然后基于底层的波函数振幅来修改方向

和长度。虽然这种方式使我们能够很好地理解全局特征，但是我们发现由于空间造型（Moiré模式）在空间上的规则定位，导致在视觉上看起来相当令人困扰且具有误导性。为了避免这些不好的效果，我们尝试在一个立方体中对媒介进行随机定位。这种方法可以很好地消除之前的干扰性，但是它又引出了更严重和基础的问题。首先，我们发现难以将所有媒介从原来各自的线条形状融合为一个连贯的线条。其次，我们发现把媒介均匀分布在三维空间中并不能生成自然的发声方法。虽然我们在之前的项目中（即关于氢键项目）已经发现可视化和可听化可以独立使用而互不影响，但是听觉的可听化和视觉的可视化表现的基础连通性对于理解场景是非常重要的。

我们解决这些连通性问题的方法是把这些线性媒介组成环状，通过弹簧使这些媒介相互连接。这种方式可以生成一条弹性带子，它保持媒介之间的平滑连接，而仍然能够在空间中自由运动，并显示被衡量的域的本地属性。把环的宽度映射为概率密度，宽带的大幅的上升代表在该位置发现电子的概率很高（见图17-10）。此外，环在用于显示波函数的状态时也能工作良好，波函数在整个空间上的分布更为广泛（见图17-11）。

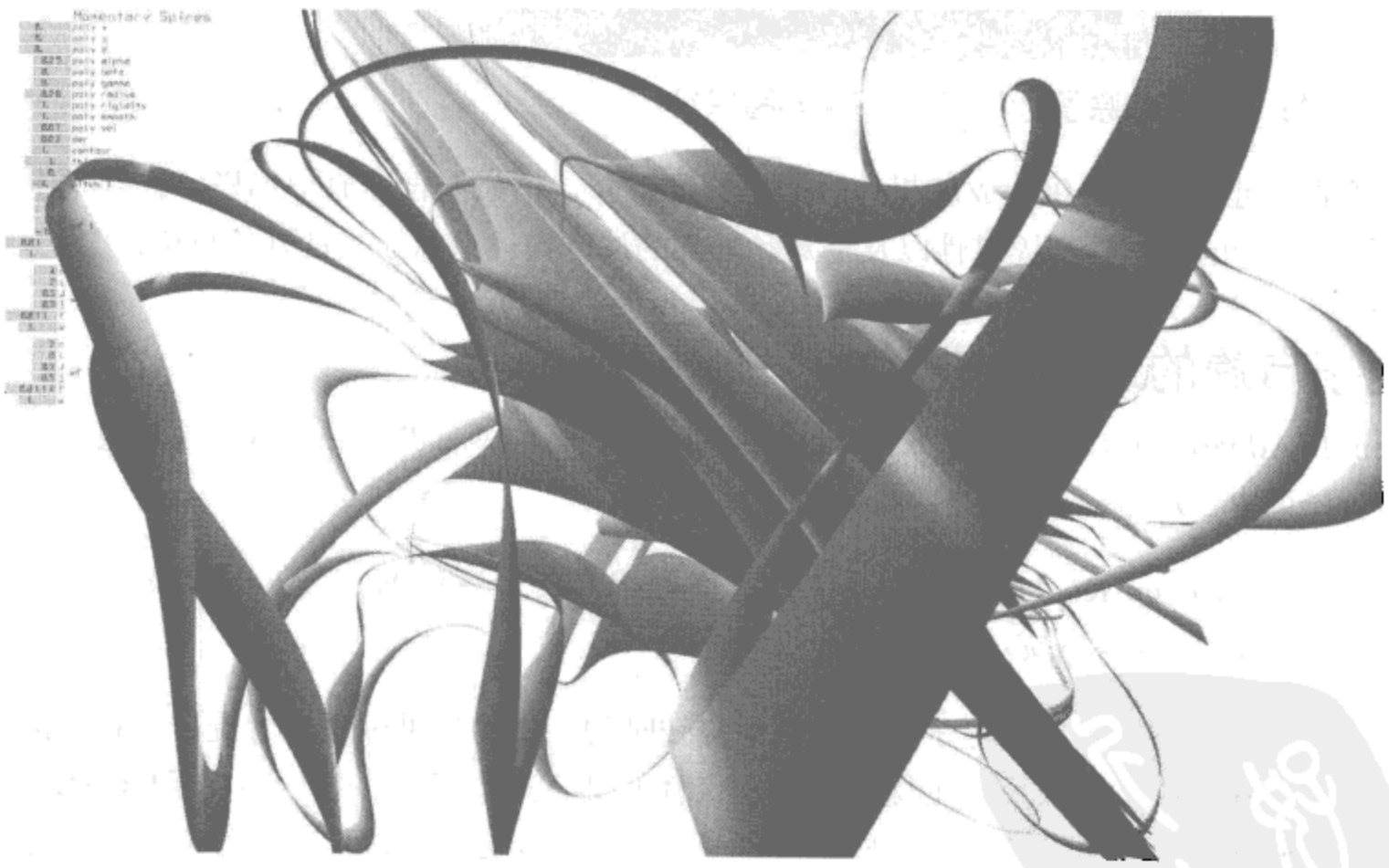


图17-10：自旋的氢原子的轨道之间的相位干扰（见彩图140）

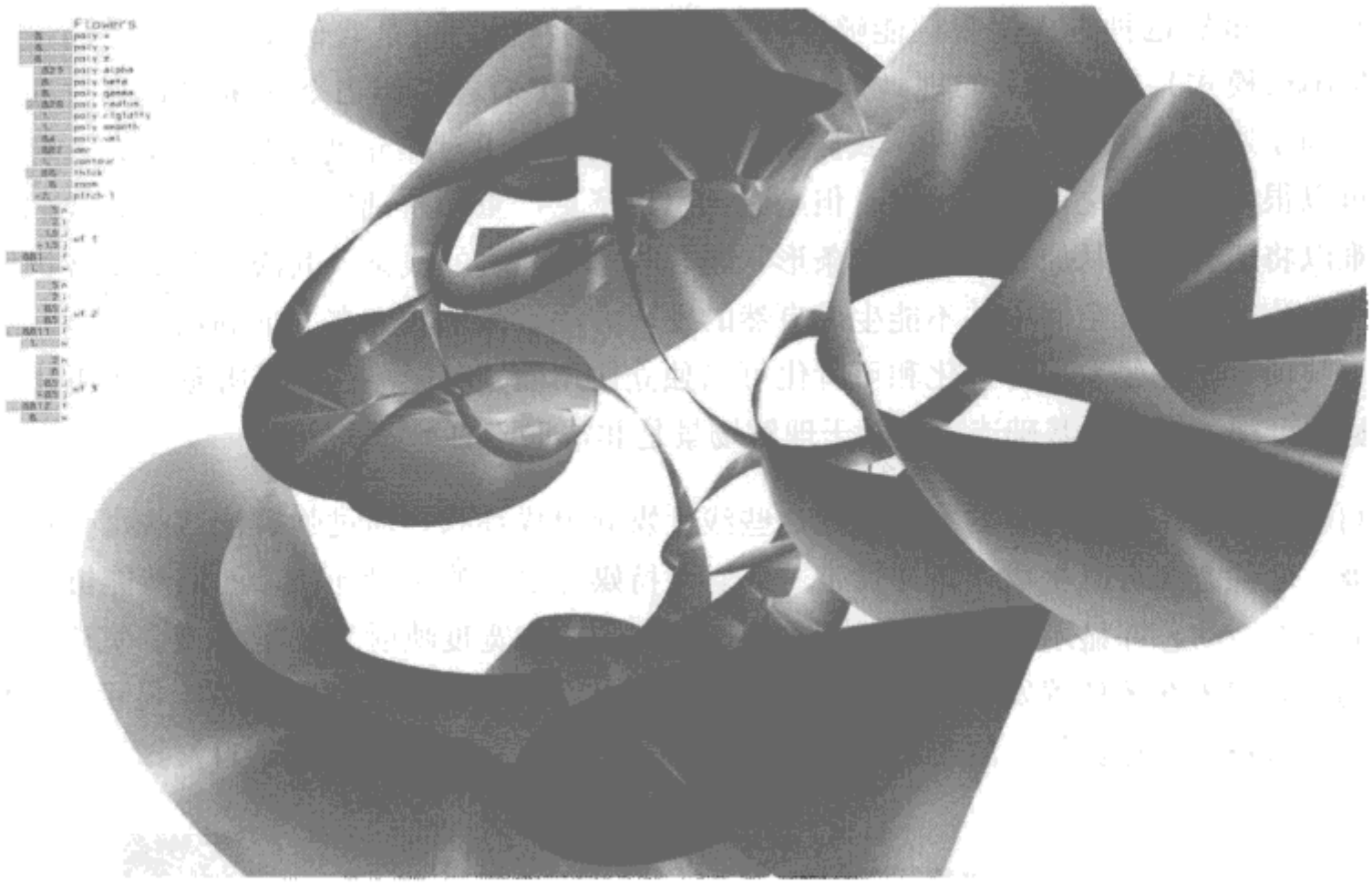


图17-11：自旋的氢原子的外壳混合（见彩图141）

平滑的环可以为可听化生成理想的外形来扫描媒介，正如无自旋原子那样。在视觉上，环状为形状的透明性和连贯性以及全局和局部的属性描述之间提供良好的权衡。

电子自旋的连贯旋进

Dennis Adderton和Lance Putnam（媒体艺术和技术），Jesse Berezovsky（自旋电子学和量子计算中心）

学科主任：JoAnn Kuchera-Morin教授（媒体艺术和技术）和David Awschalom教授（自旋电子学和量子计算中心）

本项目的目标是展示一个电子在量子点中旋转时的连贯的旋进过程或者旋转过程中的变化。为了找出在纳米级设备上衡量量子连贯性的最合适的机制，我们参观了UCSB物理系的自旋学实验室，以便了解自旋显微镜学。这是一个光学实验室，这里有一个速度非常快的激光脉冲射向半导体量子设备。脉冲的偏振作用能够诱导单个电子在量子点中自旋生成连贯的旋进。后续的脉冲可以衡量电子在量子点内的旋转偏振，从而捕获旋进过程图。通过这种测量方式，可以量化设备的量子连贯性特征衰减时间。量子态的非连贯性标志了从量子态到经典物理态的跃迁。

为了通过可听化来展示实验现象，我们把速度降低了100万倍。这使我们能够听到电子的发声以及脉冲激光的嗡嗡声。为了对自旋进现象进行可视化，我们在Bloch球体上绘制

了相位角，这对于物理学家来说是标准的图形化工具。我们根据一篇论文的实验中的一个简单公式（Berezovsky 2008）生成了三维动态图（见图17-12）。

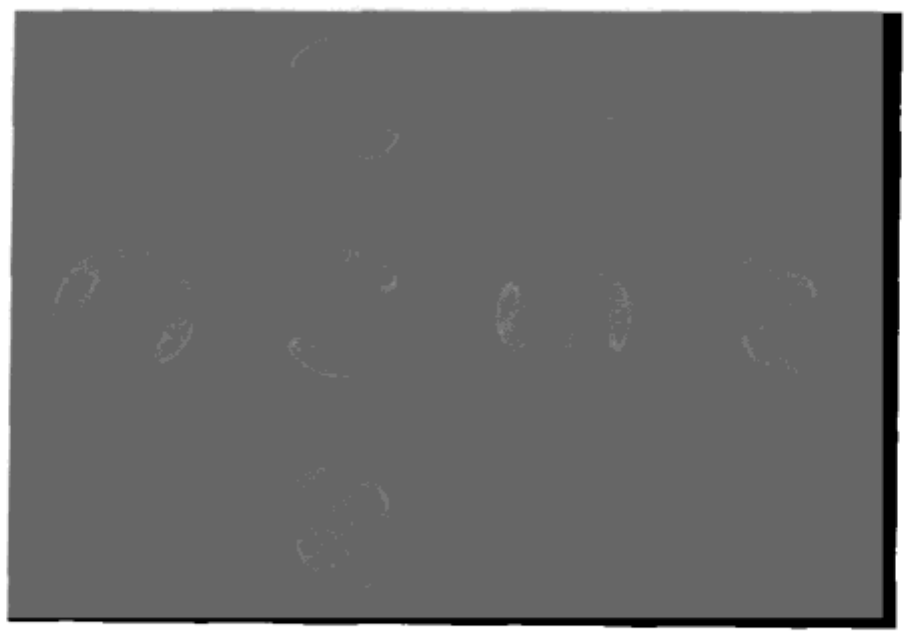


图17-12：多角度显示自旋进的Bloch球体（见彩图142）

虽然初步的测试激发了我们的激情，但是马上就发现了该模型的一个过于简单的方面，在开始的数据集中这一点还不明显。虽然在视觉上生成的是有趣的球形模式，但是其时间组件呈显著的正弦振动，因此产生的声音很快就开始让人厌烦。显然，要想融入到量子世界中，我们需要一个更复杂的系统。

为了发挥感官作用，我们需要一个更完整的自然量子力学模型，而不是实验的简化模型。表示理论模型需要进行翻译解释，使用听觉和视觉进行类比。作为一名艺术家，需要先构建一个艺术品，这样可以讨论一些有形的东西。艺术品在揭示真理上成为一门哲学“武器”——直接和数学关联的真理，并进行了可视化和可听化。这些作品可以作为哲学假设的基础，美丽的可视化是连接到可以创造和打破对称的复杂的数学系统的可视化和可听化。

结束语

在AlloSphere空间中，可视化转化成美丽的多模式虚拟展现、转换和创造，最终生成一个独特领域的演化过程。这个新的领域融合了艺术和科学的不同的标准和指标——艺术负责推测、生成和转换，科学负责模型/理论的构建和验证。随着我们的研究的进一步推进，产生了一种新的、“经典”的思考方式，它能够把科学和艺术结合到新的环境中：在这个新环境中，新艺术和新技术的产生是相辅相成的。随着这个新兴领域和计算驱动的媒介的发展，艺术家、科学家和工程师之间的差别开始消失，我们意识到我们都是工程师、科学家和艺术家——一起设计、分析和创造。

参考文献

1. Baddeley, Alan. 2000. "A new component of working memory?" *Trends in Cognitive Sciences* 4, no. 11: 417–423.

2. Berezovsky, J., M.H. Mikkelsen, N.G. Stoltz, L.A. Coldren, and D.D. Awschalom. 2008. "Picosecond coherent optical manipulation of a single electron spin in a quantum dot." *Science* 320, no. 5874: 349–352.

3. Boy, Werner. 1901. "Über die curvatura integra und die topologie der geschlossener flachen." PhD diss., Universität Göttingen, Göttingen.

4. Tatarkiewicz, Wladyslaw. 1972. "The great theory of beauty and its decline." *Journal of Aesthetics and Art Criticism* 31, no. 2: 165–180.

5. Weyl, Hermann. 1952. *Symmetry*. Princeton, NJ: Princeton University Press.

第18章

解剖可视化：真正的黄金标准

Anders Persson

本章的主题对于致力于医学信息可视化领域的人们而言格外重要。新兴的技术正在使得可视化表现和交互技术成为可能。可视化技术充分利用了人类视觉到心灵间的高带宽，使用户可以同时观察、探索、了解并验证大量的复杂信息。

今天，临床诊断和医学研究的一个显著特征是信息量变得无比庞大，特别是图像形式的信息。需要医生处理的图片越来越多（数百或上千而不是几十个），而且是越来越复杂、维度越来越高的信息（向量或张量值，而不再是标量值，是直接和解剖面对应的立体图像，而不是平面图像）。然而，目前通常还只是使用简单的二维设备如传统的显示器来一张一张地检查图像流。当前的瓶颈已经不再是数据采集，未来的发展将是开发合适的方法来处理和分析信息，并且使用户可以理解这些信息。其中最重要的一个问题是工作流。从数据采集到临床医生收到诊断信息这一过程必须优化，而且新的方法的效果必须是可验证的。

通常情况下，依据病人进行效果验证存在局限性。在某些情况下，只要病人还活着，就无法知道收集到的信息是否准确；缺失了真正的黄金标准。解剖成像有可能可以解决这个问题。

从19世纪中叶引入尸检的方法以来，迄今为止一直没有发生过重大的技术变革。然而，新的放射成像方法，如多层电脑断层扫描（MDCT）和核磁共振成像（MRI），今后有

可能成为临床和法医病理学的主要诊断工具。鉴于对新的成像技术和保健措施的校验能力，解剖可视化可能会成为未来改进人体健康的重要途径。

背景

尸检过程的重要性在于其死亡原因可以被人们所了解。对于法医，尸检可以提供至关重要的信息，而且可以为刑事调查提供指导。在过去几年，尸检的频率不断下降，这成为一个非常严重的问题。

尸检工作流中一个最近新增的功能使对尸体解剖成像成为可能——以3D形式显示，也称为虚拟尸检（virtual autopsy，VA）——使用从尸体扫描的MDCT或MRI数据，而且采用的是直接立体渲染（DVR）的三维技术。虚拟尸检的发展基础在于现代影像学可以生成大的、可精确到毫米以下的高质量的数据集。这些三维数据集的交互可视化可以促进有价值的认知，而且促进无损伤性的诊断过程。但是，对数据集进行高效的处理和分析也会带来很多问题。举个例子，在解剖尸体的CT成像中，由于不局限于每个病人所能承受的辐射，数据集可以生成非常高清的图像，当前的资料检索和交互可视化系统难以处理这些图像，尤其对于全身扫面生成的图像。

一些研究证明了虚拟解剖在法医调查中的巨大潜力。本章将探讨虚拟解剖作用不断增加的一些原因。

对法医工作的影响

在检查尸体时需要评估的主要问题是死亡的原因和方式、遭受的伤害的严重程度以及基于这些实现法医重建的可能性。法医尸检的结果文件主要是基于几个世纪以来一直使用的尸检技术和协议。尸检的主要工具是手术刀，语言描述和照片。这种方法的主要缺点在于文档记录过于随意、主观和对观察者过于依赖。没有记录的任何发现将随着尸体被送到火葬场而被无可挽回地销毁。当代层析（cross-sectional）成像技术可以克服这些缺点，因为它们提供了真实维度的发现结果的数据集，而且可以长期存储（见图18-1和图18-2）。数字化采集的数据可以在任何时候使用，也可以发送给其他专家咨询意见。

图18-1：通过计算断层扫描，很容易查出身体中的金属物体。在这起谋杀案中，有刀穿过脸，但是断层扫描证实这并不是死亡原因（见彩图143）

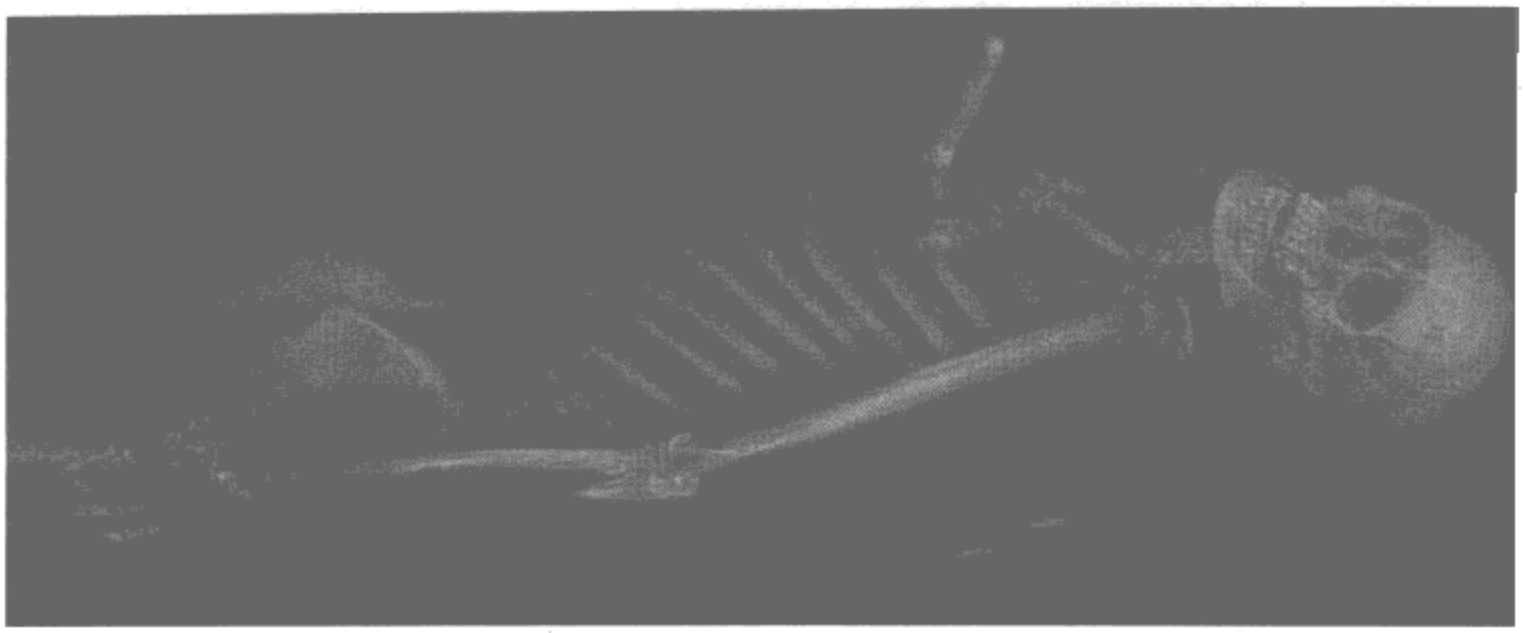


图18-2：这个图像说明了另一个案件中的死亡原因，受害人被菜刀刺穿心脏（见彩图144）

有些传统的尸检方式难以发现的信息，通过全身计算断层扫描可以很容易发现，如体内的空气分布——例如气胸、心包积气、血流（空气栓塞）以及伤口通道，如图18-3所示。计算断层扫描对于查找异物如金属碎片和子弹是非常有用的，这对于法医病理学家也是至关重要的（见图18-4）。

图18-3：获取到的计算断层扫描数据可以通过不同的参数设置进行交互可视化：在这个例子中，软组织在体内左侧，空气分布在体内右侧（见彩图145）

图18-4：通过尸检计算断层扫描可以很容易对短枪中的小碎片进行可视化。在传统的尸检中，这些碎片很难甚至不可能被发现（见彩图146）

虚拟尸检流程

瑞典Linkoing大学医学图像科学与可视化中心（CMIV）和瑞典国家法医学委员会协作开发了虚拟尸检流程，它已是用于法医工作的常规程序。从2003年以来就一直使用该方法，而且到目前为止已经被用于300个案例中（主要是谋杀）。虚拟尸检的使用经验表明全方位、高清地数字视频录像机等新技术在刑事调查和对病人的诊断中有着非常重要的作用。我们的工作重点是尸检多探头计算断层扫描（MDCT）的全部工作流，而且关注于开发新的可以对全身数据集进行可视化的软件，而之前只能通过一些独立的模块查看并且只有很有限的交互性（见图18-5到图18-7）。

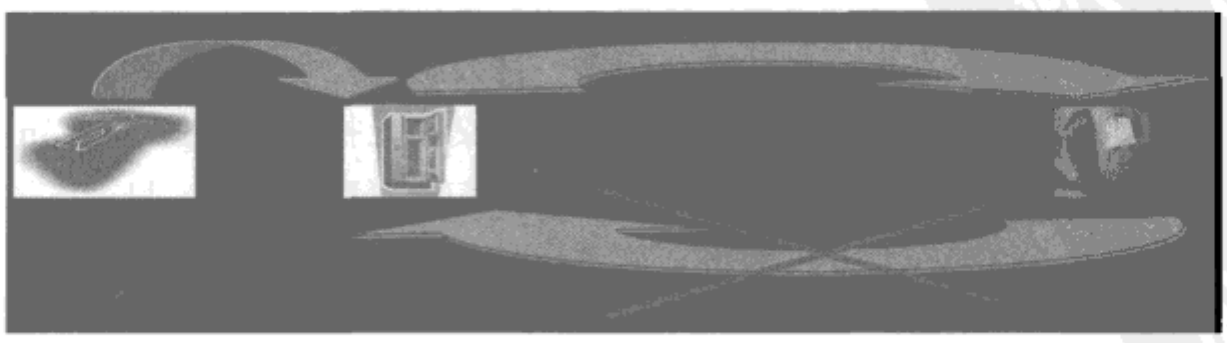


图18-5：在传统的尸检完成后，就不可能重新检查了。当尸体被送到火葬场后，没有记录的结果就无可挽回地被销毁了（见彩图147）

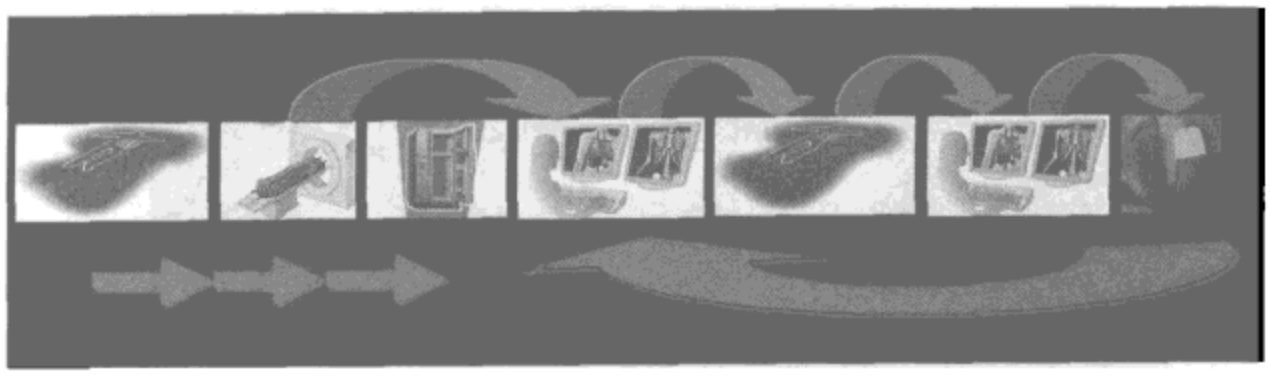


图18-6：把计算断层扫描或/和核磁共振添加到管道（pipeline）中，可以重做虚拟尸检。任何时候有新的疑问时，都可以参考数字化存储的数据，而且可以把这些数据发送给专家咨询意见（见彩图148）

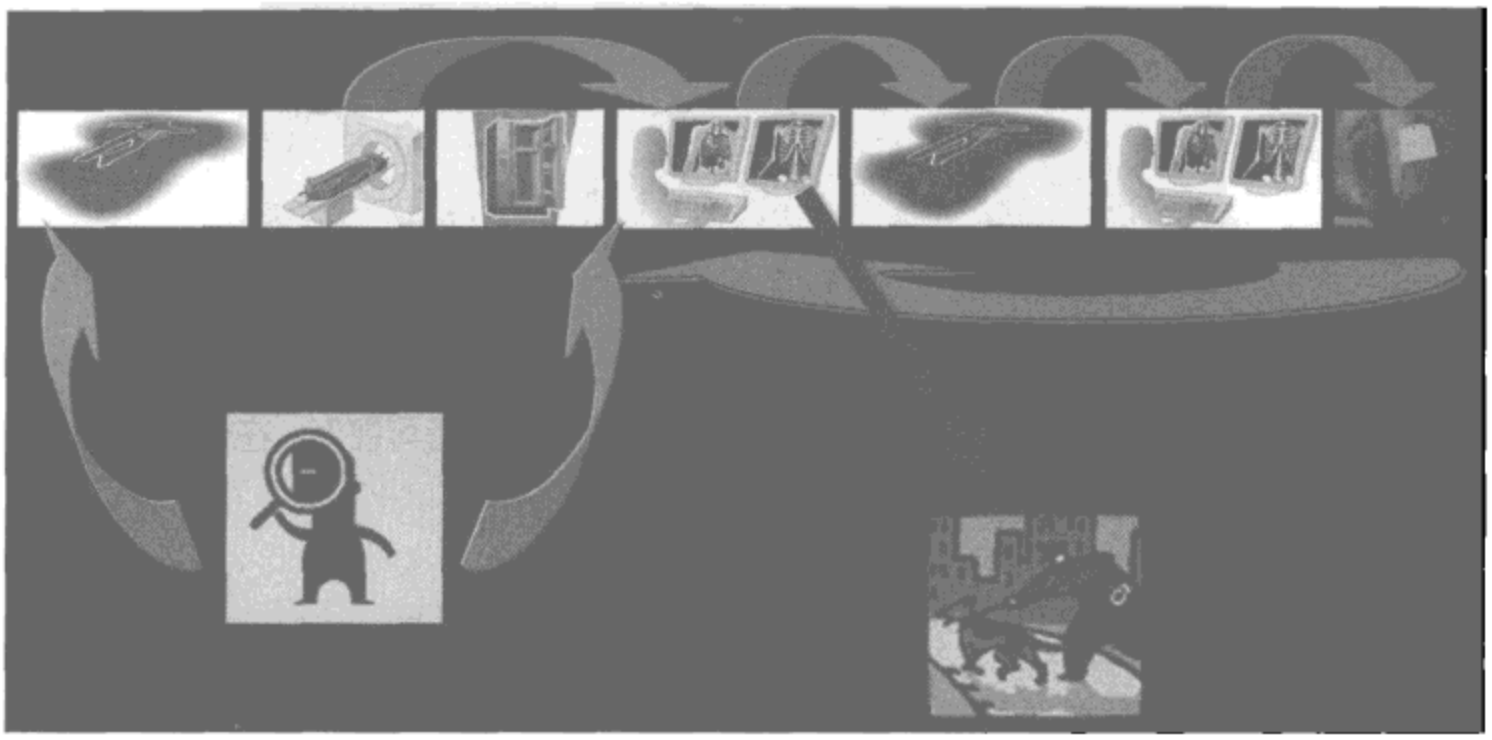


图18-7：犯罪现场调查人员和警察在把尸体保存在冷藏室方面存在矛盾。警察希望尽可能快地完成尸检。而犯罪调查现场人员希望在尸检完成之前结束犯罪现场调查。尸体成像解决了这个问题。对尸体计算断层扫描检查的初级报告使得有可能把尸体保存在冷藏室中（见彩图149）

数据采集

在瑞典Linkoing大学医学图像科学与可视化中心对传统的物理尸检进行扩展，为虚拟尸检添加了计算断层扫描和磁共振成像。在绝大多数情况下，法医来到案发现场，监督对受害人尸体的处理，尸体在运送到法医部门前，被放置到一个密封的尸体袋子中并做入库处理。第二天早晨，通过前沿技术SOMATOM定义闪光扫描器，在瑞典Linkoing大学医学图像科学与可视化中心执行全身双源计算断层扫描（DSCT）。目前，同时使用单能模式和双能模式进行虚拟尸检的案例，如图18-8a和b所示。在选择的案例中，执行的是磁共振成像检查（使用荷兰飞利浦医疗系统的Achieva 1.5T扫描器）。所有孩子都例行执行磁共振成像检查，因为比起DSCT，如图18-9所示，它提供超强的大脑可视化。

在整个虚拟尸检过程中，尸体一直是在密封的尸体袋子中，这样可以确保司法鉴定有价值的技术证据的安全性，如纤维和体液，并避免污染。

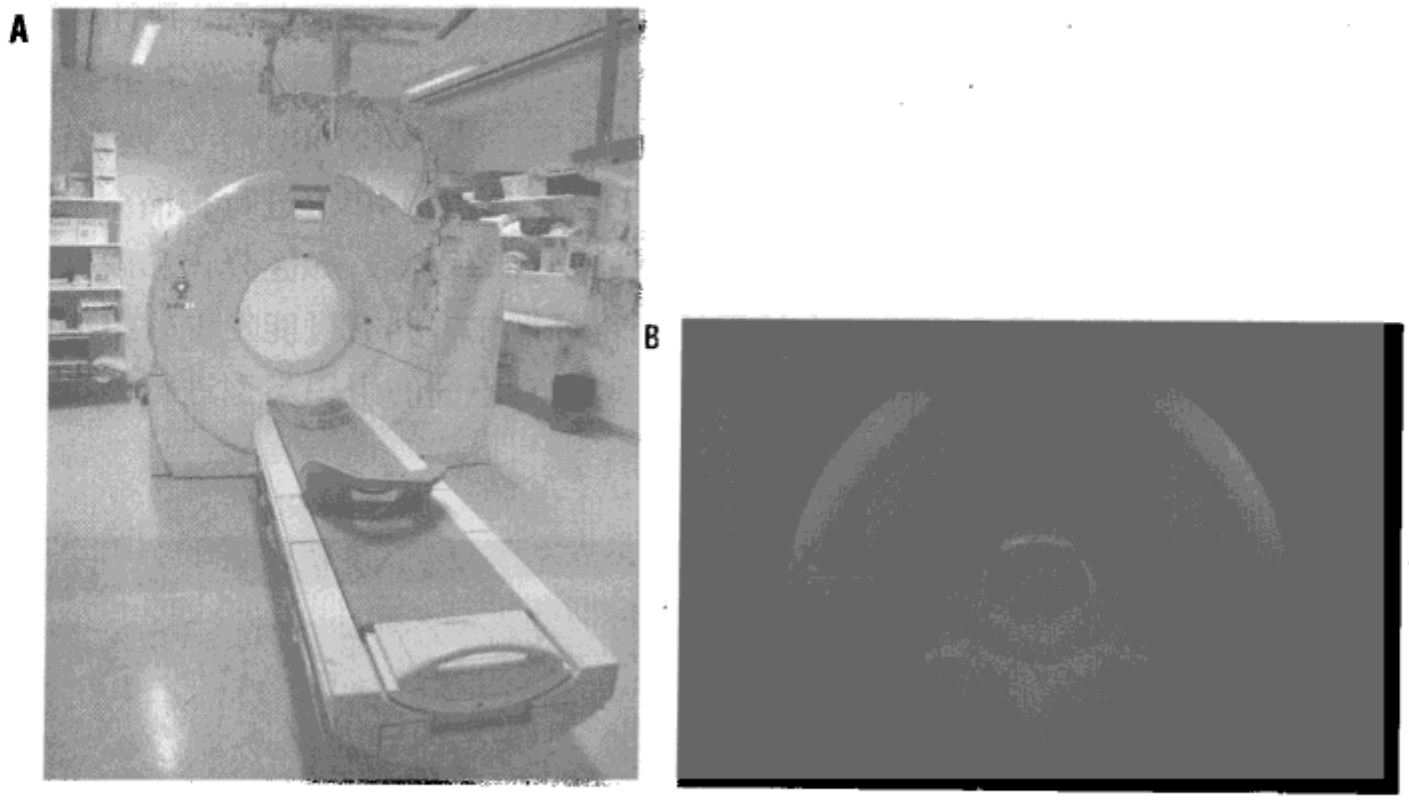


图18-8：a）前沿领先的包含双能可能性的双能计算断层扫描器，b）核磁共振扫描器。在瑞典Linkoing大学医学图像科学与可视化中心，这两个扫描器都是用于虚拟尸检（见彩图150）

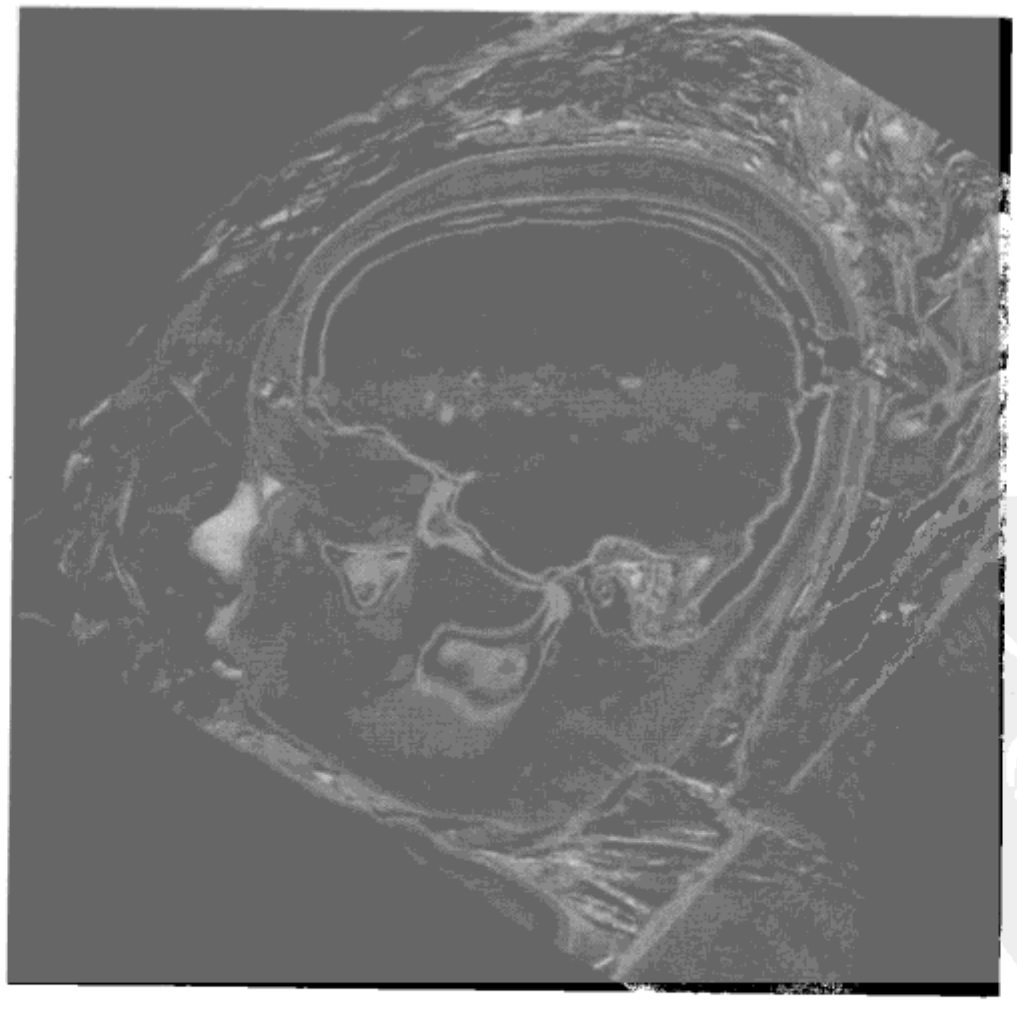


图18-9：被子弹打中的小孩的双能计算断层扫描。注意对子弹和子弹轨迹的出色的可视化。易于在法庭上展示（见彩图151）

计算断层扫描：使用双能计算断层扫描

拥有以不同能量同时运行两束x射线的双能计算断层扫描（Dual energy CT，DECT）可以获取两个数据集，显示不同的衰减层次。双能计算断层扫描可以得到计算断层扫描中的关于基础化学成分的额外信息。使用两种不同的平均照片能确定康普顿散射（Compton scattering）[译注1]，它分别对应两种管电压（80kV和140kV）。换句话说，x射线吸收依赖能量。例如，使用80kV对物理进行扫描与使用140kV进行扫描会得到不同的衰减结果。该物理现象可以用于区分包含相似原子数的物体，如区分钙和碘。还可以使用该技术来更好地对尸体血管中的血液凝块进行可视化，并有可能发现软组织出血。在衰减中，如结果图所示的对特定材料的区别有助于对不同的组织类型进行分类，如血液、软组织肌腱和软骨（见图18-10）。

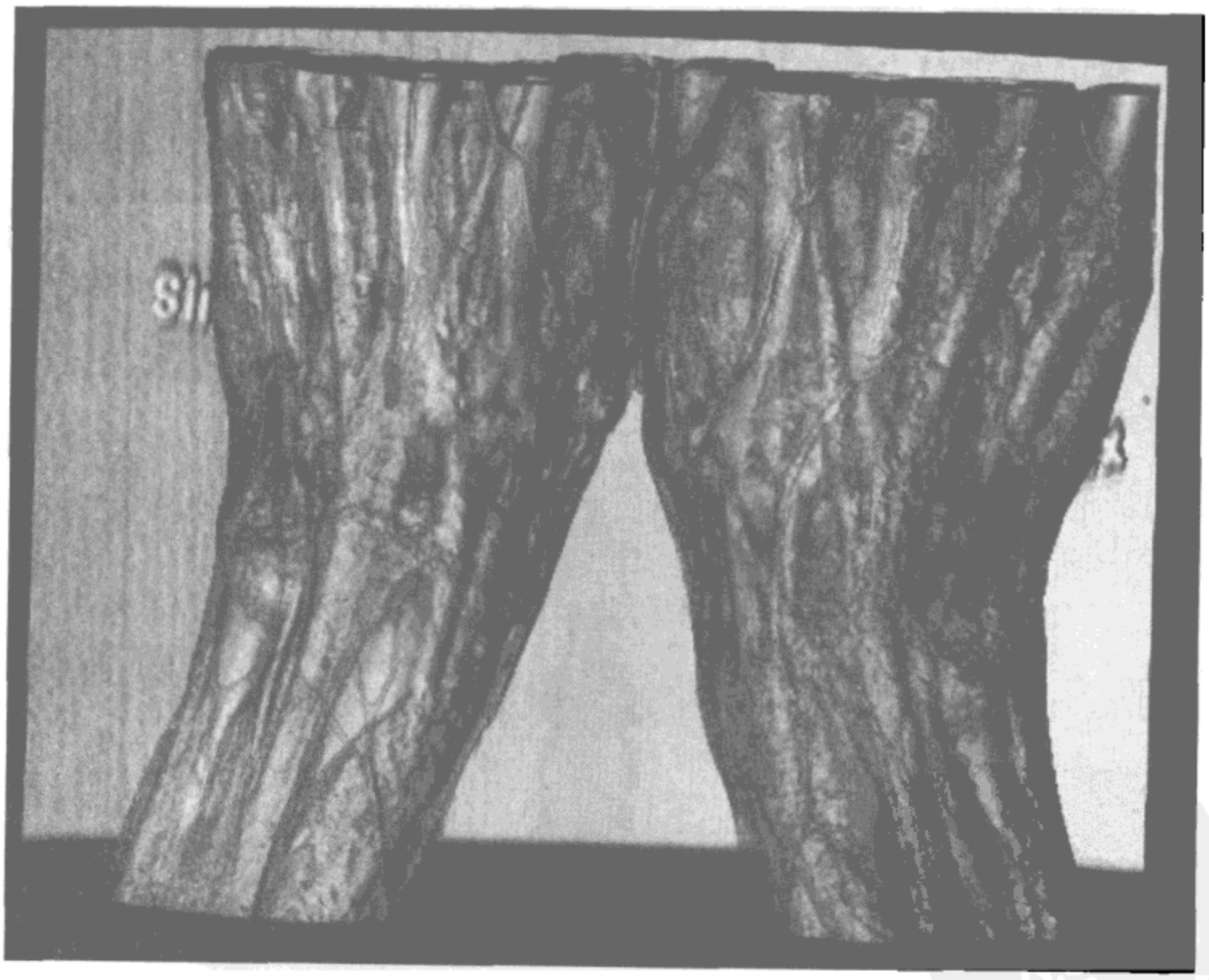


图18-10：通过双能计算断层扫描做的肌腱检查。肌腱和小的血管可以不使用静脉注射造影剂进行可视化。对腕骨之间的韧带进行可视化（见彩图152）

译注1：康普顿散射，也称康普顿效应，在物理学上，它是指当x射线或伽马射线的光子跟物质相互作用，因失去能量而导致波长变长的现象。由于它是高能量x射线与生物中的原子核间最有可能发生的相互作用，因此亦被应用于放射疗法。

双能计算断层扫描（DECT）有成为未来重要医疗诊断工具的潜力。但是，需要进行进一步的深入研究来探索这门新技术。虚拟尸检有助于这项研究。

核磁共振成像：使用合成核磁共振成像

在冷却的尸体上生成高对比度的核磁共振成像很难——体温会影响所有有机组织的核磁共振松弛次数，因此在临床医学上制定的协议需要调整为在任何给定温度下都能生成最佳的图像。这个问题可以通过计量组织特有的绝对磁共振参数T1、T2和质子密度等解决。

由于临床上应用的核磁共振成像扫描仪难以解决以上问题，瑞典Linkoing大学医学影像科学与可视化中心发明了一种新的方法，即核磁共振成像（synthetic MRI）。在这种方法中，3个绝对参数被翻译成了普通的核磁共振对比图片（见图18-11和图18-12）。借助一种色标，这样每个组织可以获取依赖于核磁共振组织参数并且不依赖于体温的颜色成分。因为核磁共振参数是绝对的，所以一种颜色转换将与一个颜色-组织之间的映射关系相对应。这种方式对于解剖成像格外有意义，因为图片对比度可能会随着温度的变化而产生非常大区别，如图18-12所示。

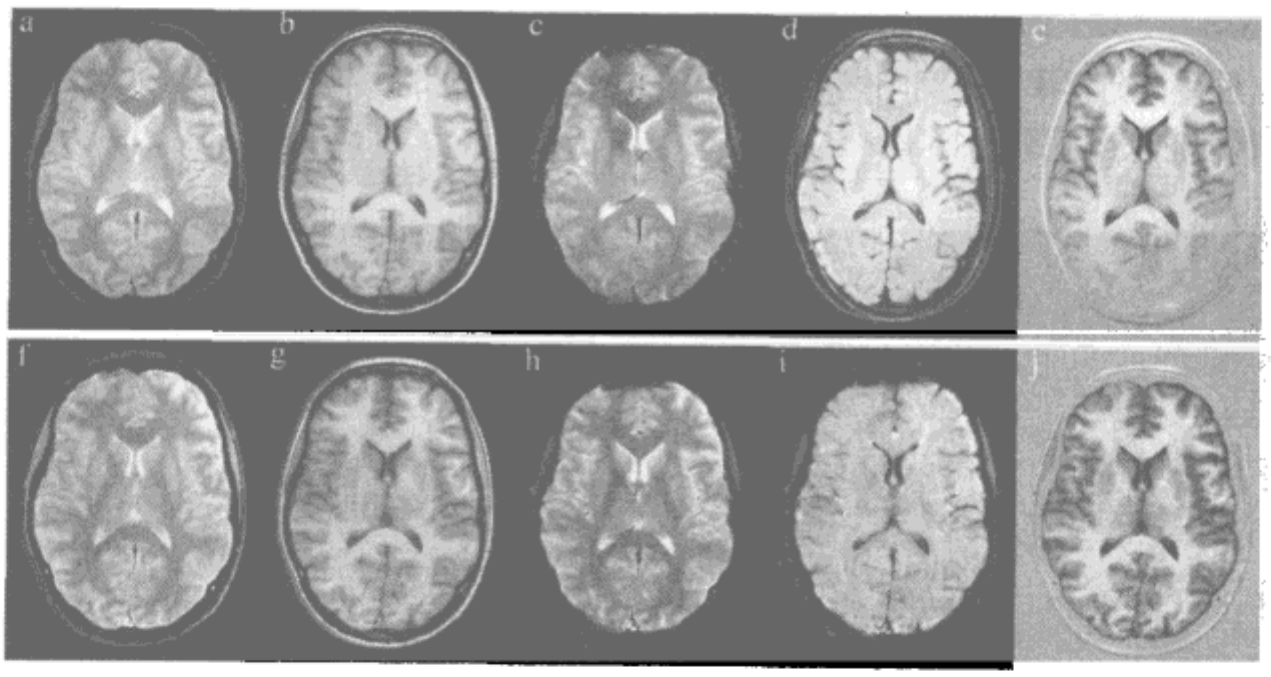


图18-11：一个活着的病人的合成核磁共振成像例子：第一行是传统的图像，第二行是基于同一个数据集生成的合成图像

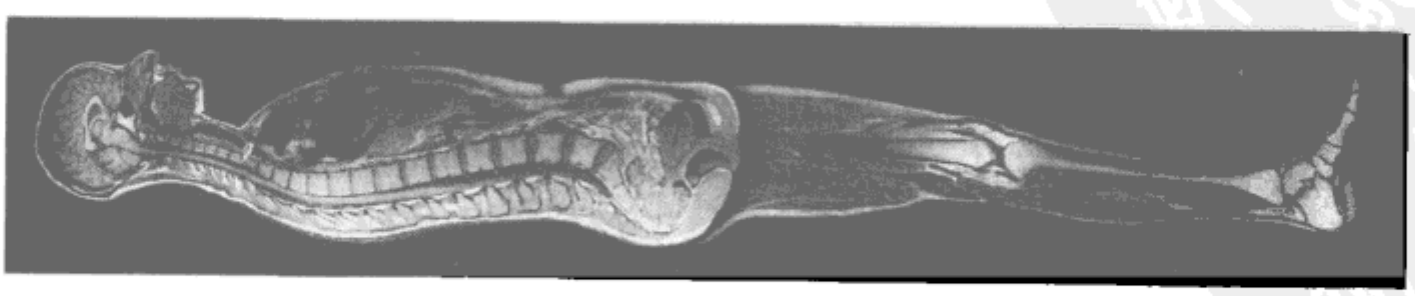


图18-12：全身合成核磁共振扫描。对比度可以人工合成，软组织可以进行分割，甚至温度也可以基于核磁共振参数确定

尸体检查并不需要考虑运动因素，可以通过长时间的扫描来获取高清晰度图像。比如，图18-13显示了1.2mm同性分辨率的头部中弹伤口。因为磁共振成像基于绝对值，因此可以在计算机断层扫描后借助处理软件渲染三维图像，最终生成了如图18-13和图18-14所示的立体渲染。

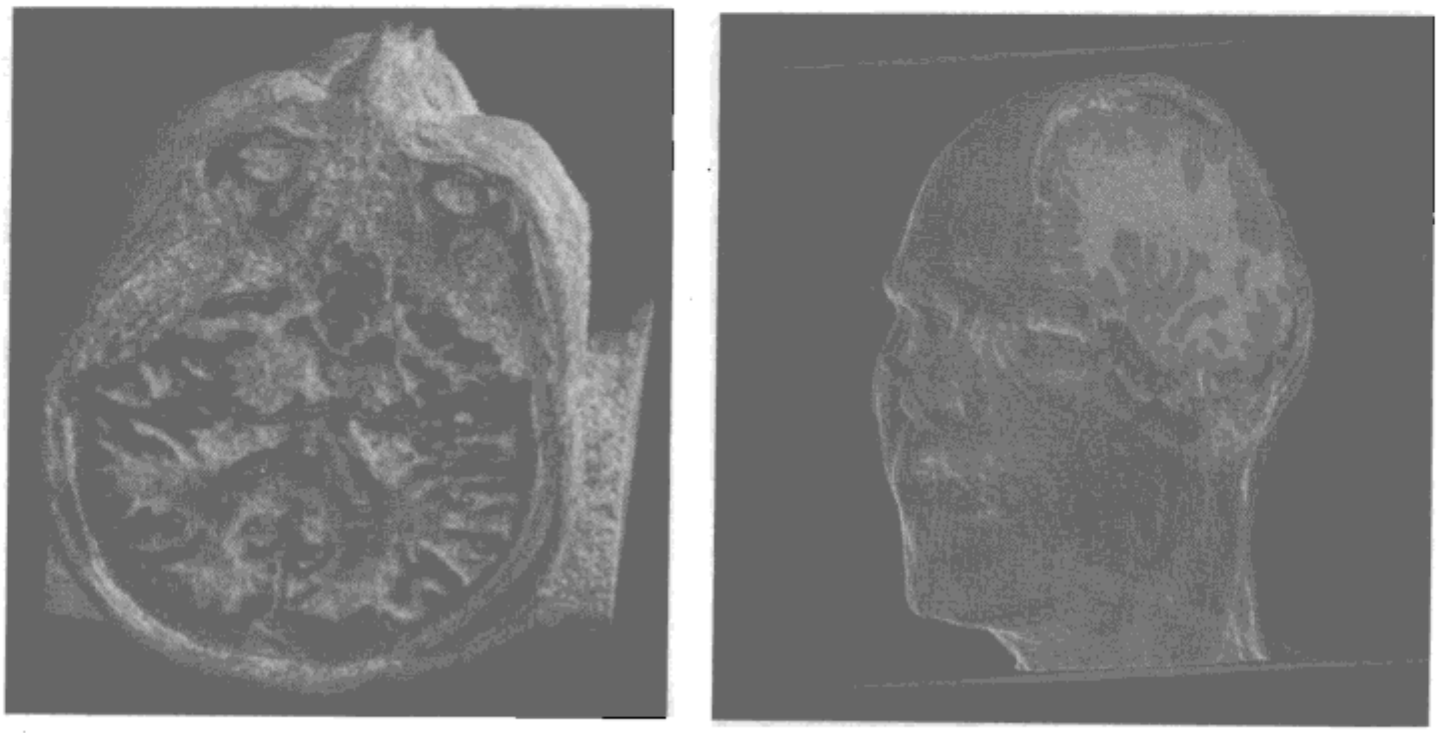


图18-13：使用高分辨率的各向同性方案为一个子弹伤口生成的尸检合成磁共振成像。左边图像中的红色代表血液（见彩图153）

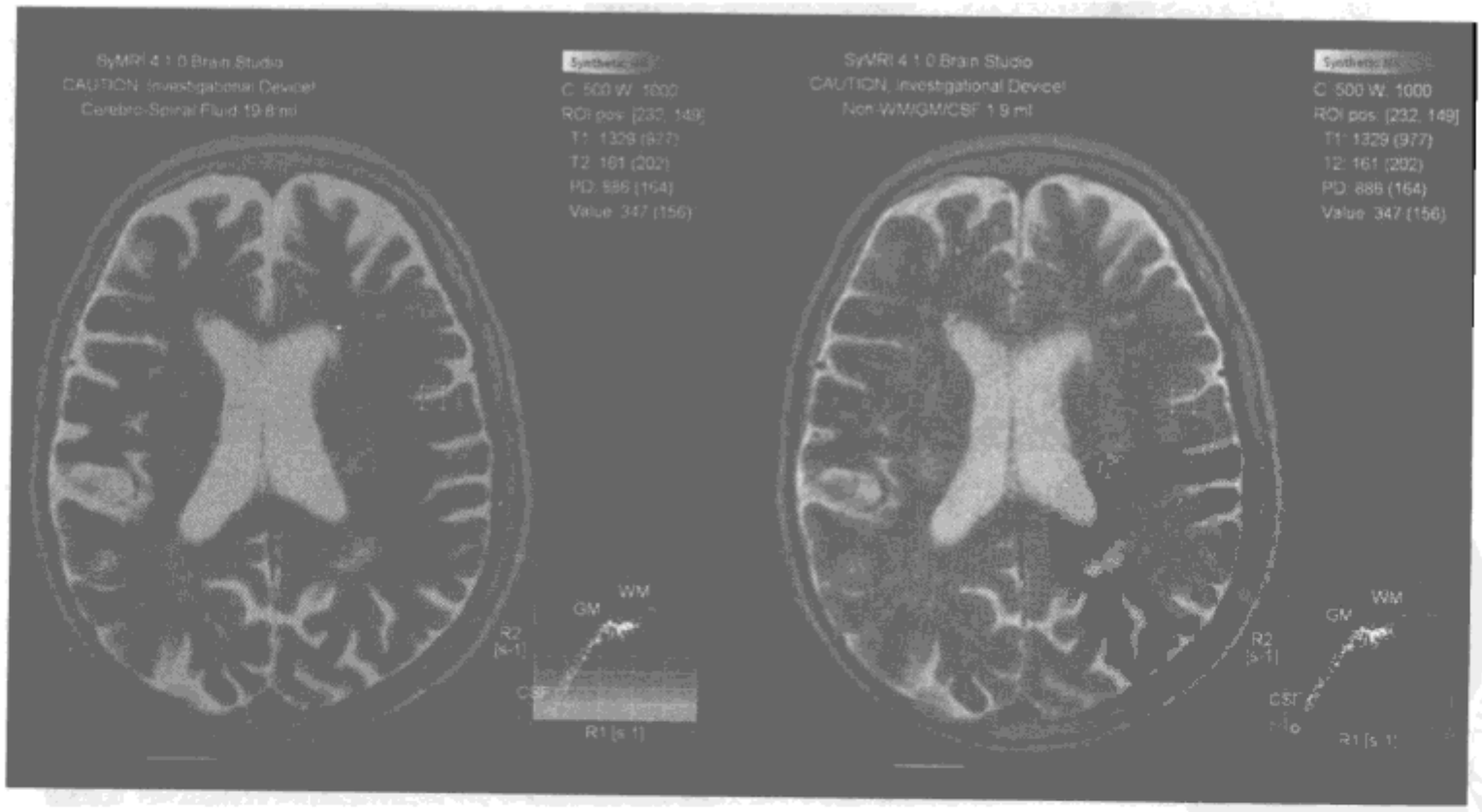


图18-14：通过合成核磁共振成像，对脑脊髓液（该切割图中是19.8ml）和病灶（该切割图中是1.9ml）的自动分割（见彩图154）

可视化：图像分析

在物理尸体解剖的准备过程中，病理学家和放射学家举行了同步进行的一场协作式的数字视频会议。他们可以快速地对整个尸体进行清晰的调查，定位骨折和气泡。尸体全身处理的整个过程支持对异物如金属碎片或子弹的快速定位。另一个重要的方面是数据分辨率很高，在无缝可视化中可以抽取细节信息（如牙科）详情用于鉴别（见图18-15）。这种方式可以为警方的初期调查提供必要信息。在完成扫描后，法医离开瑞典Linkoing大学医学影像科学与可视化中心，开始传统的尸检。协作的数字视频录像会议中获取的数据被转移到法医研究所供他们使用，在后期如果需要更多的信息，可以再联系放射科医生。

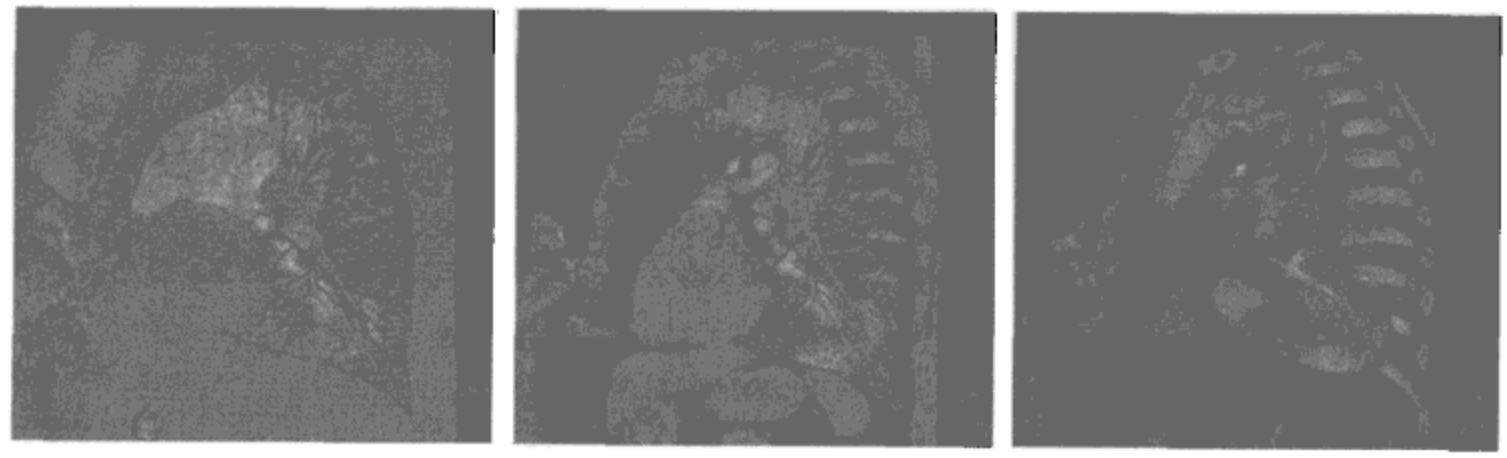


图18-15：有了三维立体渲染，可以交互式地改变背景，这样就可以对尸体从皮肤到骨骼进行无缝可视化（见彩图155）

客观记录

虚拟尸检为尸检过程增加的一个重要的价值是存储了捕获到的双源计算断层扫描数据，这使得可以对尸检过程进行迭代。通常情况下，在物理尸检期间的发现可能会引出新的问题，而虚拟尸检可以回答这个问题。病理学家和犯罪调查人员还可以在调查期间的任意时刻对尸体进行重新检查以查找其他信息，如图18-16所示。此外，在犯罪现场调查中，新的发现可能依赖一些其他假设，这些假设可以通过尸体成像进行确认。

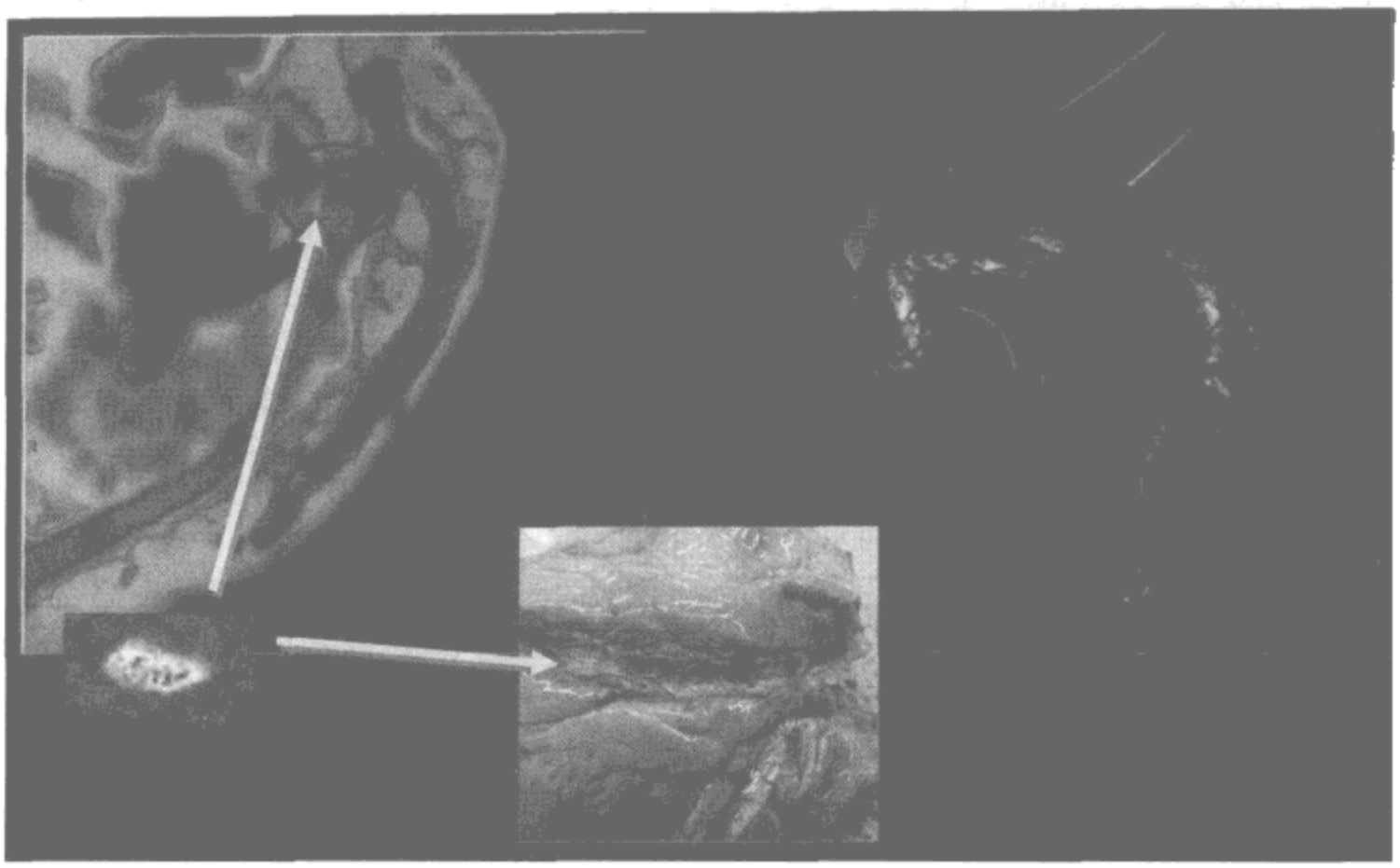


图18-16：对心脏和动脉的双源计算断层扫描。比起传统的单源成像（红色圆圈所示），双源计算断层扫描可以对更模糊的组成部分进行可视化（见彩图156）

目前，虚拟尸检是对尸检过程的补充。然而，应该注意的是，将其引入工作流的代价是最小的，因为和物理尸检相比，双源计算断层扫描和可视化需要的时间是短暂的，而且它使得尸检更高效。病理学家在开始尸检前，可以提前对案件的背景知识有所了解。在整个虚拟尸检过程中，尸体一直在密封的尸袋中，这样可以确保司法鉴定得到的技术证据的安全性，如纤维和体液，这对于法院的案件的判决非常重要。

虚拟尸检的优势和不足

首先，我们来了解一下和传统的尸检技术相比，虚拟尸检的优势。

- 节约时间。虚拟尸检作为标准尸检的补充，能够对整个尸体进行全方位、广泛、系统的研究，而传统尸检要做到这一点通常很难而且时间代价很高；比如，对整体骨骼结构的检查或者查找体内存在的气泡（见图18-3和图18-4）。
- 没有创伤。传统的尸检一旦完成，整个尸体就无法重新组合成原始状态，这导致其他法医病理学家无法对该尸体重新进行分析（见图18-5、图18-6和图18-7）。
- 家庭成员可能会出于宗教信仰如禁止亵渎尸体而拒绝传统的尸检。
- 在刑事案件中，尸检协议和照片作为证据，经常会让陪审员感觉难以理解。虚拟尸检会清晰得多（见图18-4和图18-9）。

- 对虚拟尸检的数据保存基本不存在问题，而传统的尸检记录如组织切片通常难以长期贮存（见图18-16）。
- 对于可能对人们构成越来越大威胁的全球性流感如禽流感（禽流感A）和H1N1病毒，取出受害者的内脏会让验尸官、病理学家、医学家冒着非常大的健康风险。有了虚拟尸检，这些风险都可以降至最低。

然而，虚拟解剖也包含一些缺点：

- 对于多探头计算断层扫描，软组织区分度很低。能量分辨的计算断层扫描（DECT）有可能解决这个问题（见图18-10）。
- 对生成的大量数据进行分析是个问题，但是更好、更快地后处理程序应该能够解决这个问题。
- 核磁共振成像是很费时的调查方式，而且对于冷却的尸体不是最佳方式。合成核磁共振成像是一个很有前景的备选方式（见图18-14）。
- 使用多探头计算断层扫描成像和核磁共振成像进行尸检的方式无法为尸体记录任何颜色信息。可以通过新的三维立体渲染和尸体表面扫描技术来解决这个问题（见图18-15）。
- 不存在宏观形态（没有组织学和化学）。使用多探头计算断层扫描的活组织检查或核磁共振光谱，可以在一定程度上解决这个问题（见图18-16）。
- 难以对循环和可能的流血点进行可视化，虽然通过对尸体血管摄影得到了可喜的成果。众所周知，对尸体的计算断层扫描摄影是从虚拟尸检中获取更多信息的一种可行方式，如图18-17所示。
- 尸体散发的气体和其他气体（小肠气、伤口渠道积气）难以区分开。因此，在死后尽快进行尸体成像检查是很重要的（见图18-18）。

虚拟尸检的未来

多探头计算断层扫描和磁共振成像都可以用于尸体成像。原则上，很容易通过多探头计算断层扫描对骨骼、气体和金属进行可视化。但是，重要的是，不仅要注意这些技术的能力，还应该注意它们的局限性。

将来的可视化研究必须包含实现虚拟解剖工作站的整体目标，它包括了前沿的虚拟尸检技术需要的所有方面。需要开发提高虚拟尸检过程质量和效率的可视化工具。需要专注于新的渲染与分类技术的研究和开发工作，以提高这些技术的可用性，并专门解决法医问题。另一个重要的目标是为主要的法医案件分类建立专门的备忘录。

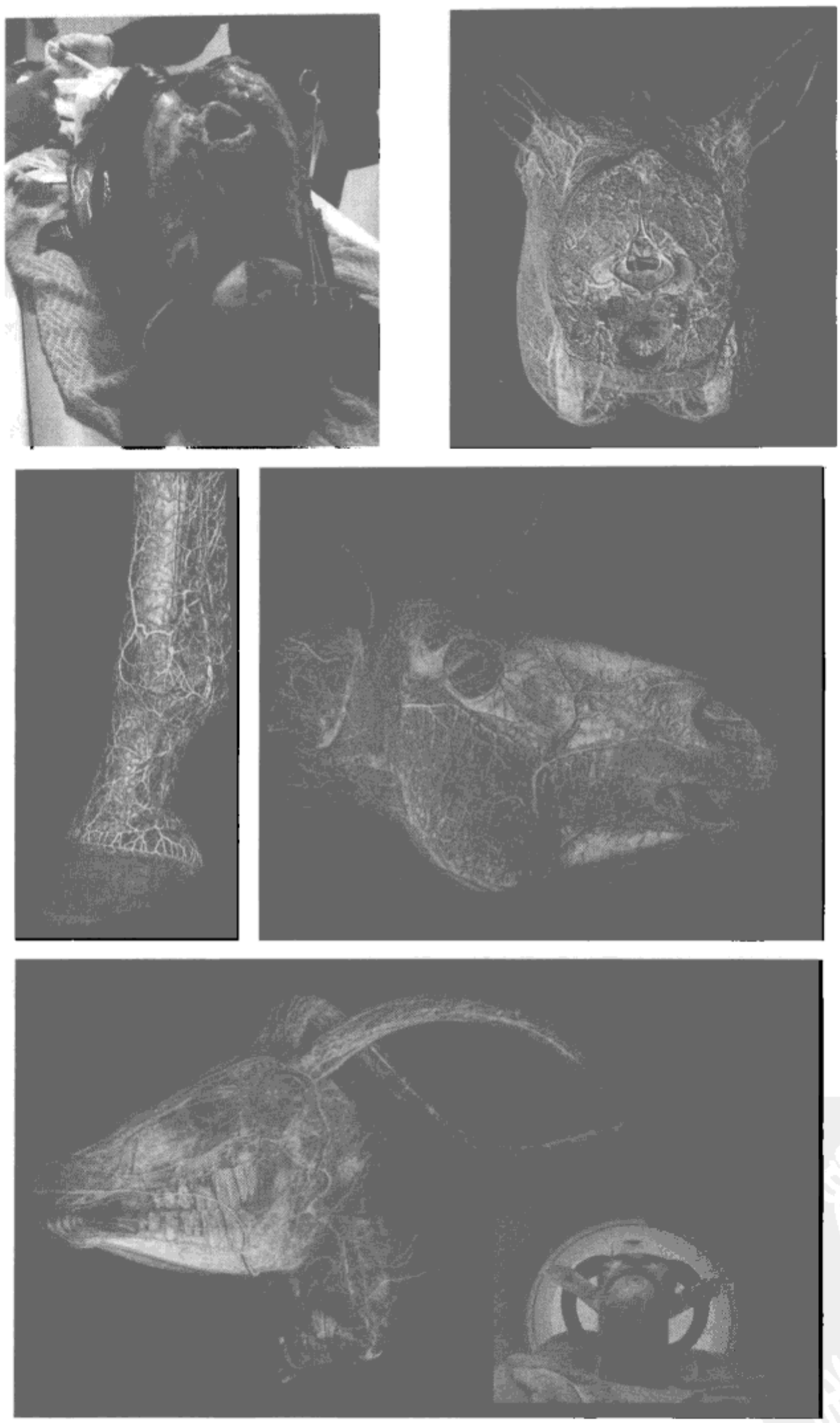


图18-17：在马和羚羊的尸体上执行动脉注射效果良好。数据是通过双能计算断层扫描获取到的（见彩图157）

图18-18：对于传统的尸体解剖，难以检查尸体上不同类型的气体（见彩图158）

数据分析研究包括实现计算机辅助诊断的工具，这些工具一旦应用于尸体数据，将有助于查找和特征化相关的法医调查结果。这些工具还可以提供死者的相关信息，如身高、体重、性别、重大伤害、异物（如子弹），以及自动初步生成的可能的死因、书面虚拟尸检备忘录。

成功解决这些问题之后，可以对虚拟尸检的全过程都有涉及的技术进行改进，推动整个工作流的自动化。这将使得在合理的时间内可以完成大量虚拟尸检。这对于处理一些出现大量伤亡人员的灾难事件非常有用，如2004年的亚洲海啸，当时没有执行任何尸检。

由于恐怖分子时刻都在提高他们的技术，如果法医病理学家不能够利用新兴技术来尽可能地从受害者身上收集更多信息，那将是不可想象的（见图18-19）。如果处在一个没有人真正感觉安全的时代，我们就不应该只着眼于预防灾难，还应该为灾难的发生做好充分的准备，当灾难发生时可以及时处理。

为了真正进入数字尸检的新时代，各种力量必须通力协作。医学专业人士和执法权威人士必须确定扫描和存储数据的标准草案。世界各地的法律制度必须确定成像证据在分辨死亡原因和方式时的可接受性。此外，还需要对新领域的专家进行培训，如尸体放射学。放射科医生通常受到的培训是解释病人的图像，但是死者和病人不同；严重创伤或者解剖分解的效果可以取代器官。理解这些差异，需要知识和专长，而这些知识目前尚未普及。

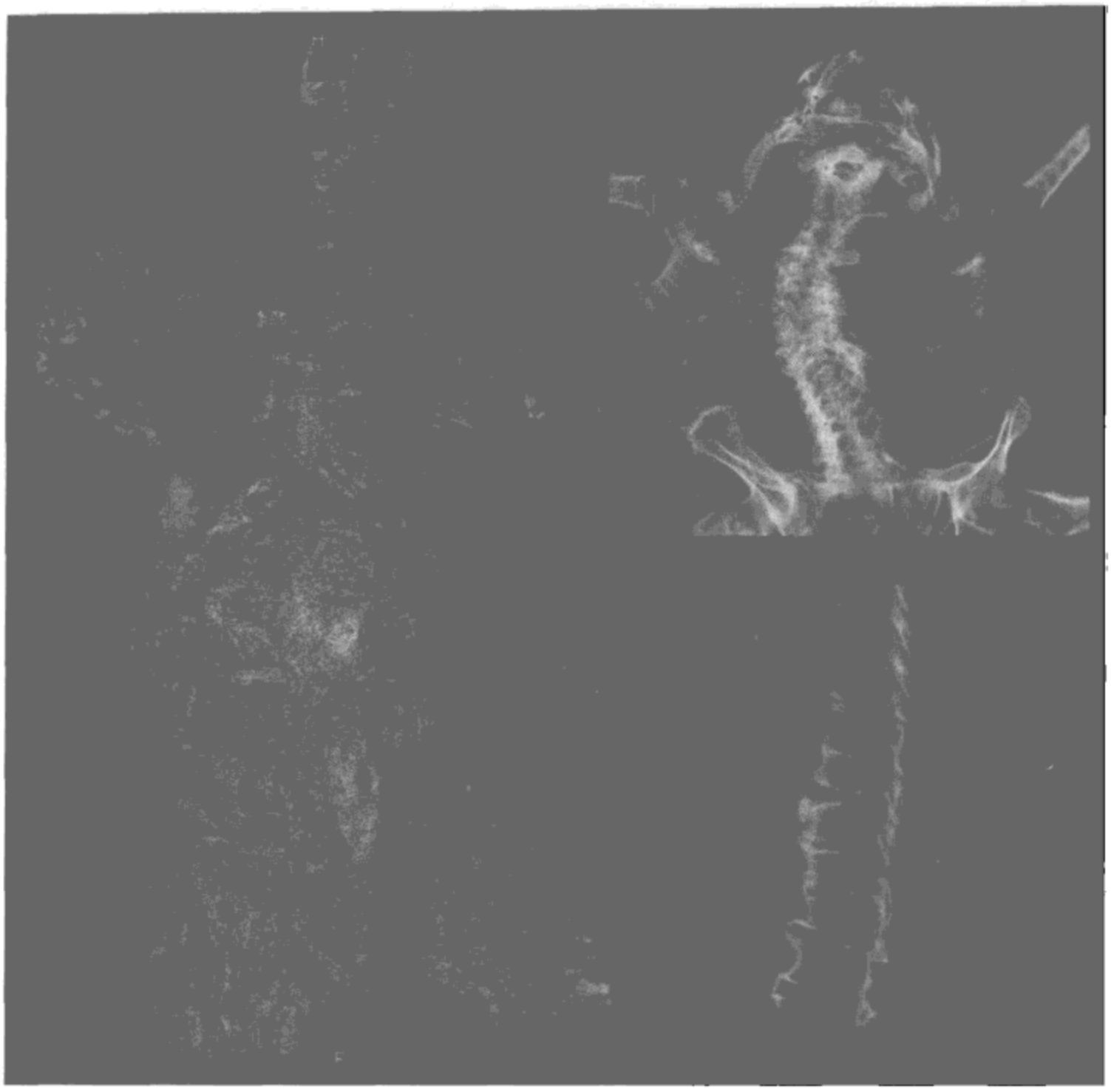


图18-19：对被烧毁的尸体计算断层扫描。体内的金属片使得核磁共振成像无法使用。在执行计算断层扫描之前，没有任何谋杀嫌疑，但是某些无法解释的骨折给调查员指明了方向——谋杀（见彩图159）

创伤性的尸体解剖至少在几年之内还将普遍存在。但是，在某些情况下，我们可能发现传统的尸体解剖可以被非创伤性的虚拟尸检取代，后者只在必要的时候执行微创性的、图像引导的组织抽样。和传统的尸体解剖相比，虚拟尸检有可能获得较高的接受度，使得在法医和传统医学中持续保持高水平的质量控制成为可能。

结束语

虚拟尸检是尸检流程中新增加的一个处理过程，它可以提高传统尸检技术，从而能够得到更为可靠的结果。在某些情况下，虚拟尸检能够取代普通的尸体解剖。然而，关于尸

体放射学独有的研究必须能够识别出应用这种技术时最为受益的案例，并且要验证新的流程。显然，新的尸检方法的引入可能会对法医学、司法系统、警察和普通医药学造成重要影响。

参考文献和扩展阅读

1. Donchin, Y., A.I. Rivkind, J. Bar-Ziv, J. Hiss, J. Almog, and M. Drescher. 1994. "Utility of postmortem computed tomography in trauma victims." *Journal of Trauma* 37, no. 4: 552–555.

2. Etlik, Ö., O. Temizöz, A. Dogan, M. Kayan, H. Arslan, and Ö. Unal. 2004. "Three-dimensional volume rendering imaging in detection of bone fractures." *European Journal of General Medicine* 1, no. 4: 48–52.

3. Jackowski, C. 2003. "Macroscopical and histological findings in comparison with CT- and MRI- examinations of isolated autopsy hearts." Thesis, Institute of Forensic Medicine. O.-v.-G.-University of Magdeburg.

4. Jackowski, C., A. Persson, and M. Thali. 2008. "Whole body postmortem angiography with a high viscosity contrast agent solution using poly ethylene glycol (PEG) as contrast agent dissolver." *Journal of Forensic Sciences* 53, no. 2: 465–468.

5. Jackowski, C., W. Schweitzer, M. Thali, K. Yen, E. Aghayev, M. Sonnenschein, P. Vock, and R. Dirnhofer. 2005. "Virtopsy: Postmortem imaging of the human heart in situ using MSCT and MRI." *Forensic Science International* 149, no. 1: 11–23.

6. Jackowski, C., M. Sonnenschein, M. Thali, E. Aghayev, G. von Allmen, K. Yen, R. Dirnhofer, and P. Vock. 2005. "Virtopsy: Postmortem minimally invasive angiography using cross section techniques—Implementation and preliminary results." *Journal of Forensic Sciences* 50, no. 5: 1175–1186.

7. Kerner, T., G. Fritz, A. Unterberg, and K. Falke. 2003. "Pulmonary air embolism in severe head injury." *Resuscitation* 56, no. 1: 111–115.

8. Ljung, P., C. Winskog, A. Persson, C. Lundstrom, and A. Ynnerman. 2006. "Full-body virtual autopsies using a state-of-the-art volume rendering pipeline." *IEEE Transactions on Visualization and Computer Graphics* 12, no. 5: 869–876.

9. Oliver, W.R., A.S. Chancellor, M. Soltys, J. Symon, T. Cullip, J. Rosenman, R. Hellman, A. Boxwala, and W. Gormley. 1995. "Three-dimensional reconstruction of a bullet path: Validation by computed radiography." *Journal of Forensic Sciences*, 40, no. 2: 321–324.

10. Ros, P.R., K.C. Li, P. Vo, H. Baer, and E.V. Staab. 1990. "Preautopsy magnetic resonance imaging: Initial experience." *Magnetic Resonance Imaging* 8: 303–308.

11. Thali, M., W. Schweitzer, K. Yen, P. Vock, C. Ozdoba, E. Spielvogel, and R. Dirnhofer. 2003. "New horizons in forensic radiology: The 60-second digital autopsy-full-body examination of a gunshot victim by multislice computed tomography." *The American Journal of Forensic Medicine and Pathology* 24: 22–27.

12. Thali, M., U. Taubenreuther, M. Karolczak, M. Braun, W. Brueschweiler, W. Kalender, and R. Dirnhofer. 2003. "Forensic microradiology: Micro-computed tomography (Micro-CT) and analysis of patterned injuries inside of bone." *Journal of Forensic Sciences* 48, no. 6: 1336–1342.

13. Thali, M., K. Yen, W. Schweitzer, P. Vock, C. Boesch, C. Ozdoba, G. Schroth, M. Ith, M. Sonnenschein, T. Doernhoefer, E. Scheurer, T. Plattner, and R. Dirnhofer. 2003. "Virtopsy, a new imaging horizon in forensic pathology: Virtual autopsy by postmortem multislice computed tomography (MSCT) and magnetic resonance imaging (MRI)—a feasibility study." *Journal of Forensic Sciences* 48, no. 2: 386–403.

14. Yen, K., P. Vock, B. Tiefenthaler, G. Ranner, E. Scheurer, M. Thali, K. Zwygart, M. Sonnenschein, M. Wiltgen, and R. Dirnhofer. 2004. "Virtopsy: Forensic traumatology of the subcutaneous fatty tissue; Multislice Computed Tomography (MSCT) and Magnetic Resonance Imaging (MRI) as diagnostic tools." *Journal of Forensic Sciences* 49, no. 4: 799–806.

第19章

动画可视化：机遇和缺点

Danyel Fisher

动画是否有助于创建更丰富、更生动和更易于理解的可视化，或者只是让人更为困惑？

随着Java、Flash、Silverlight和JavaScript等在Web上的广泛使用，使得动画式的具有交互功能的可视化的发布愈加容易。很多可视化人员开始思考如何在他们的可视化中引入动画功能，从而让他们的可视化变得更有吸引力。如何使静态可视化更为有效方面有很多好的指南，很多应用也可以很好地支持交互。但是，动画可视化仍然是一个新领域，对于如何评价一部动画可视化作品还没有达成基本的共识。

从直觉角度来看，动画应该足够清晰：如果一张二维图像的效果已经很不错，那么一张能动的图像的效果应该只会更好。运动，我们都很熟悉：我们早已习惯于现实世界中的各种运动，也习惯于看着事物平滑地运动。在我们周围，事物在以我们切实理解的方式运动、生长、改变色彩。

在可视化中，动画能够显示中间的步骤和转换过程，也能显示数据是如何随着时间的变化而收集起来的，这可能有助于观察者深入理解某个观点背后的逻辑。移动的图片可能提供的是一种崭新的视角，也可能是更能吸引用户从而促使用户更深入地观察数据。动画还可以使两张视图之间的变化更平滑，甚至在不存在平滑数据变化的临时组件的情况下也可能做到这一点。

作为例子，我们一起了解一下Jonathan Harris和Sep Kamvar的“We Feel Fine”的动画可视化（*http://wefeelfine.org*）。在这个可视化中，提到情感的博文被显示成气泡。在不

同的视图内，气泡被组织成直方图和其他模式。举个例子，一个屏幕显示男性和女性的博文的相对分布，而另一个屏幕显示博文中流露的情绪的相对分布。虽然气泡在屏幕上自由移动，但是在屏幕上的气泡的数量一直是恒定的。这种恒定性有助于强化样本以不同方式组织的理念。动画还可以用于唤起情感：气泡的能量值不同则运动也不同，表示“幸福”的气泡的运动方式和表示“悲伤”的气泡的运动方式是不一样的。

但是，并非所有的动画都是成功的。有太多的应用是对PowerPoint的动画的滥用，数据点在屏幕上漫无目的的到处乱飞；各种组件只是在屏幕上毫无意义的空间中横扫、扩展和旋转，这样通常只是导致一片混乱。

我已经多次创建过动态可视化。在2000年，我和几个研究生一起创建了GnuTellaVision，它是对不断增长的Gnutella对等网络的可视化。从那以后，我就一直从事采用了动画可视化的很多项目：比如在一个应用了动画散点图的项目中，以观察员的身份密切关注DynaVis项目，关注不同可视化之间的转换效果。在本章，我将交流一些经验并尝试给出动画可视化的一些基本原则。

如果使用得当，动画将是一种非常强大的技术，但当使用不当时，其效果也会非常差。有些动画虽然提高了可视化的视觉吸引力，但是可能复杂化了对数据集的探索；其他类型的可视化对于探索可能更为合适。本章试着建立一个有效的动画可视化的设计框架。我们首先一起了解一些背景材料，然后探讨最知名的动画可视化之一——Hans Rosling的GapMinder。我参与过一个类似于GapMinder的探索动画散点图的项目；它可以作为讨论动画可视化的成功和失败之处的一个不错的开始。正如我们将看到的，成功的动画可以展示多种转换类型。DynaVis项目会为我们展示其中一些转型和转换是如何实现的。在本章的最后将可视化的一些设计原则作为结尾。

动画原则

本质上来看，任何动画都是向观众展示一系列快速、连续的图像。观众对这些图像进行组装，试着把各个图像上发生的事情贯通起来。感知系统会注意到帧之间的变化，因此动画可以被理解为不同帧之间的一系列视觉变化。当变化很少时，可以很容易理解发生了什么变化，而且观众也可以很容易追踪这些变化。但当有大量的变化时，理解就会变得很复杂。

Gestalt的“共同命运”（common fate）感知原则指出观众会把大量的事物组合在一起，如果这些事物以相同速度、沿相同方向运动，它们就会被看作同一组。个别沿着自己的轨迹运动的对象将会被看作“游离点”，在视觉上会很明显。但是，如果所有事物都沿着不同的方向运动，观众将无法应付。感知研究人员已经证实了观众难以对超过4个或5

个独立运动的对象进行追踪——他们将放弃追踪所有的，转而只追踪几个物体，把其他的作为“噪音”看待（Cavanagh、Alvarez 2005）。

科学可视化中的动画

在一年一度的IEEE VisWeek会议——可视化的研究峰会上，与会人员被分成了两组：信息可视化研究人员和科学可视化研究人员。这两组的演讲不同，坐在不同的会议室，有时吃饭也是坐在不同的餐桌边。观察这些演讲，很快就可以注意到，在科学可视化会议室里大约有一半的文章是关于动画的，而在信息可视化会议室里几乎没有一篇文章是关于动画的。你可以认为这两个分组之间的区别在于科学可视化研究人员是真正理解x、y、z轴含义的人：他们善于绘制图片的各个维度，理解深度和距离的涵义。他们通常研究动态过程，比如大风吹过飞机机翼、飓风席卷整个地图、血液沿着静态流动，此外往往还涉及另外一个维度：时间。因为难以把时间强加到其他三个维度（x、y、z轴）中，动画是显示该过程的一个不错的选择。

与此相反，数据可视化就没有如此简单了。信息可视化研究人员通常致力于抽象的数据空间，其各个轴并未与现实世界对应（如果这些轴有任何涵义的话）。观众需要适应他们能够看到的各个维度，然后学会解释它们。因此，在信息可视化领域，与动画有关的文章相对较少。（我们后面将讨论其中几个例子。）

从卡通中学习

当然，动画在可视化以外的领域很流行。电影和卡通所遵循的一些的原则和计算机动画相同，因此有人会问卡通技术是否会给创建动画可视化带来一些有用的认知。早在1946年，比利时的心理学家Albert Michotte就提出“因果性知觉”（perception of causality）（Michotte 1963）。人们很容易相信动画中的运动是有目的的：一个点是在追逐另一个点（而不是认为一个点在沿着相同的轨迹在其后面运动）；或者是一个球击中了另一个球（而不是“这个点在位置A停下，而另一个点从位置A运动到位置B”）等。因此，我们可以把原因归于媒介和因果性，虽然实际上二者都不存在。

当然，在卡通里，我们希望表达因果关系。传统的漫画家描述了为了表达情感，如何给漫画赋予“生命幻觉”（illusion of life）（Johnston和Thomas 1987），还有一些研究论文（Lasseter 1987; Chang和Ungar 1993）曾尝试探索如何为计算机动画和可视化提炼思想。

传统漫画家采取一系列技术，有些和真实生活不完全一致。举个例子，挤压和拉伸，在事物运动时对它进行扭曲，把眼睛画成和运动方向一致：事物在以最快速度飞行时可能会拉伸，而挤压表达的含义是停止运动、收集能量或改变方向。沿着弧线运动意味着运

动更自然；沿着直线的运动看起来目的很明确。在事物开始运动前，他们预测即将发生的运动，而以持续性告终。“渐进和渐出”（ease-in, ease-out）是定时动画技术的一门技术：动画缓慢开始以强调方向，中间加速，最后速度又缓下来。复杂的运动分解为多个阶段来吸引人们要特别注意的个别部分。

可视化研究人员以不同的热情改造过这些技术并获得了不同程度的成功。举个例子，信息可视化研究框架（Card、Robertson和Mackinlay 1991），一个早期的三维动画框架，其中集成了部分原则，包括预期、弧线和后续跟进。另一方面，其中的某些原则看起来非常不恰当。比如，对一个数据点的挤压或拉伸会扭曲它，改变可视化的本质；因此，我们不再认为可视化在动画的每个帧维持一致性原则“高度映射这个，宽度映射那个”。Zongker和Salesin（2003年）在用幻灯片展示的研究成果中，提醒人们很多动画技术会分散注意力或具有欺骗性，其推导所展示出的因果性可能都不存在。此外，这些动画技术往往会给人们一种错觉，它可能非常不适合数据可视化。（一个例外是“We Feel Fine”，在该可视化中，运动表示传达情感，而且它有效地使用该技术实现了这个效果。）

动画的负面效应

动画在数据可视化中的应用不如科学可视化成功。二元研究查看了不同类型的动画——过程动画和算法可视化——发现这两类动画在帮助学生学习更复杂过程时对记录的追踪参差不齐。

心理学家Barbara Tversky发现，让她有些失望的是，动画看起来并不利于过程可视化（也就是说，显示如何使用工具或技术的可视化）。她在文章《Animation: Can It Facilitate?》（Tversky、 Morrison和Bétrancourt 2002）中讲述了对接近100部的动画和可视化作品的研究。没有任何一部动画的研究证明动画的效果超过信息丰富的统计图，虽然它确实优于文本表示，也优于没有过渡状态而只显示开始和结束状态的简单展示。

算法动画在很多方面类似于过程可视化：可以通过演示各个步骤来说明算法。例如，有些排序算法非常适合于动画：可以把一组值描绘成条形序列图，排序操作就是移动条形图。这些动画可以很容易地演示冒泡排序和插入排序之类的算法。Christopher Hundhausen、Sarah Douglas和John Stasko (2002年)试着了解算法可视化在教室里的有效性，但是在他们的研究中，有一半表明动画无法帮助学生理解算法。有趣的是，预测成功的最强因子是动画背后的可视化。包含建构理论的可视化是最有用的——也就是说，当学生实现代码或算法，查看自己的作品的可视化，或者向学生提问，让他们试着通过可视化回答这些问题。相比之下，动画在传授知识方面效果不好；被动地观察动画并不比其他方式的教学效果好。

GapMinder和动画散点图

动画可视化最近的一个例子是Hans Rosling的GapMinder（*http://www.gapminder.org*）。Rosling是瑞典研究全球健康的教授， 2006年2月他在关于“科技、娱乐、设计（TED）”的会议[注1]中首次和现场观众交互，之后和很多网友进行了交互。他从国际资源中收集了公众健康统计数据，在他的演讲中，这些数据被绘制成了散点图。在可视化中，一个点代表一个国家，其中*x*和*y*值表示如寿命和平均孩子数之类的统计，而且每个点的面积都和其表示的国家的人口数对应。Rosling首先显示的是单个帧——在某一年的国家统计——在通过时间追踪可视化进展前，使用动画对每年的图像进行显示。

图19-1显示了类似于GapMinder动画的3个帧。*x*轴表示出生时的预期寿命，*y*轴表示婴儿死亡率。气泡大小和人口数对应，对每个州进行颜色编码；最大的两个点是中国和印度。

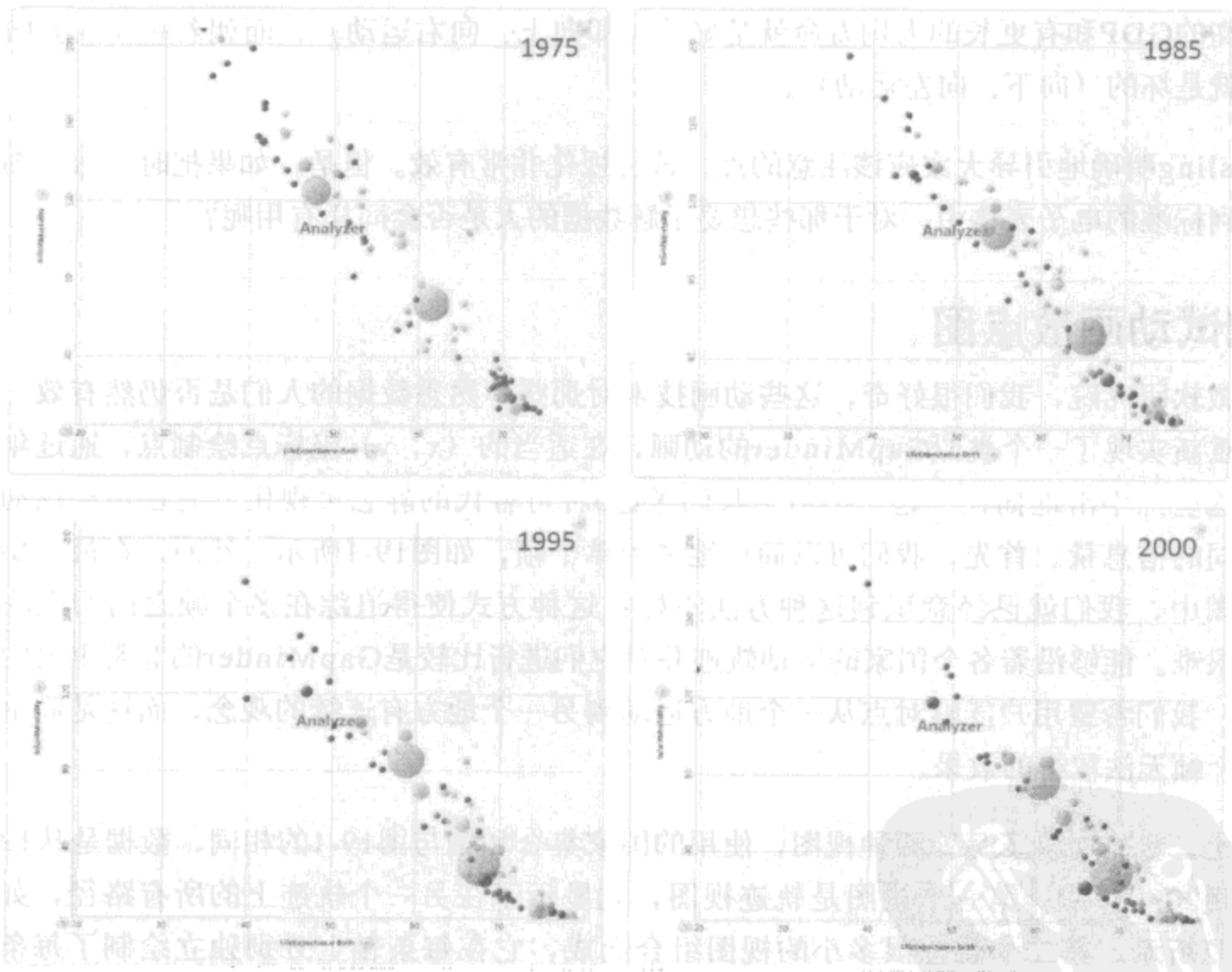


图19-1：类似GapMinder的可视化说明了在1975、1985、1995、2000这4年间75个国家的信息；该图对寿命（x轴）和婴儿死亡率（y轴）进行绘图。在左上角的国家，其婴儿死亡率高，寿命短（见彩图160）

注1： 在网上*http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html*可以得到相关资料。Rosling在TED 2007和TED 2009两次会议中做了类似的探讨。

Rosling的动画很吸引人：他使用了点的运动，描述了他们的相对进展。中国提出了适当的公共健康规划，其所代表的点就向上运动，而其他国家也尝试实践了相同的策略。一个国家的经济飞速增长，其所代表的点就迅速向右运动。Rosling使用该动画很有力地说明了我们对公共健康问题的理解以及发达国家和发展中国家之间的差别，动画帮助观众理解他的观点。

太多点吗

之前提到的感知心理学研究表明人们最多同时跟踪4个运动的点。在展示过程中，Rosling可以引导观众，说明应该查看哪里，而且他的讲述告诉了人们应该关注哪些点。借助很长的指示棒，他描述了一个国家的进步；应该查看哪里。这降低了混乱。

另一个优势在于他使用的二维散点图中“好”和“坏”的指示非常明确：一个国家走向更高的GDP和有更长的人均寿命就是好的（即向上、向右运动），而朝着相反的方向运动就是坏的（向下、向左运动）。

Rosling明确地引导大家应该注意的点，该可视化非常有效。但是，如果把时间散点图结合到标准的电子表格中，对于那些想要了解数据的人是否会同样有用呢？

测试动画散点图

在微软研究院，我们很好奇，这些动画技术对那些不熟悉数据的人们是否仍然有效。我们重新实现了一个类似GapMinder的动画，在适当的（x，y）坐标点绘制点，通过年份把这些点平滑地插在一起。然后，我们考虑3种可替代的静态可视化，它包含和该动画相同的信息量。首先，我们可以简单地采用单个帧，如图19-1所示。然而，在最开始的草图中，我们就已经意识到这种方法不好：这种方式使得追踪在多个帧之间的点的运动很难。能够沿着各个国家的运动轨迹并对它们进行比较是GapMinder的非常重要的部分。我们希望用户能够对点从一个地方运动到另一个地方有连续的观念，而这是简单的单个帧无法取得的效果。

因此，我们实现了另外两种视图，使用的国家集合和轴与图19-1的相同，数据是从1975年到2000年的。第一个视图是轨迹视图，它显示了在另一个轨迹上的所有路径，如图19-2所示。第二个图由很多小的视图组合而成，它在每条轴上分别独立绘制了每条路径，如图19-3所示。在第一个视图中，我们使用透明度描述时间；在第二个视图中，我们通过点的大小表示时间。

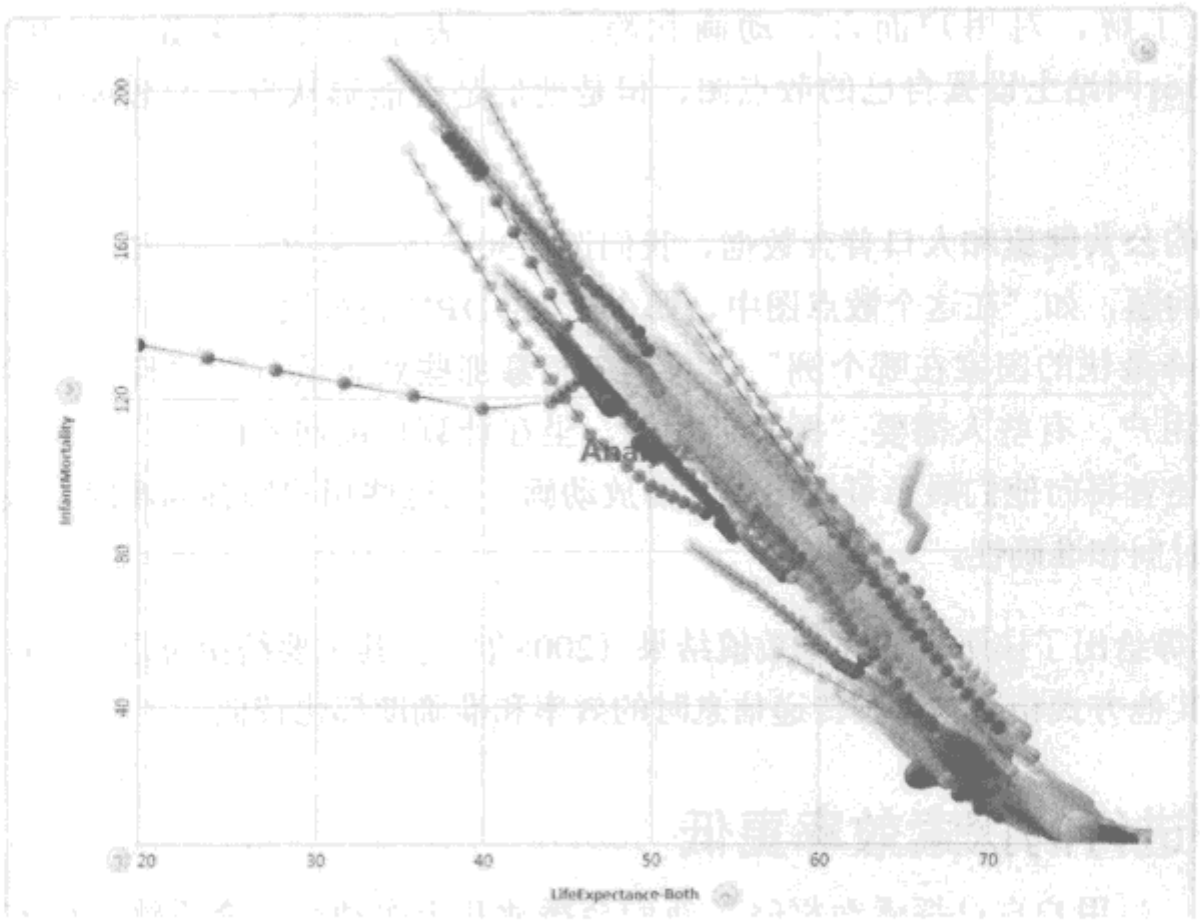


图19-2：轨迹视图，其中每个国家被表示成一系列点，这些点随着时间而变得更模糊；年份和“褪色”的点组成的线相连（见彩图161）

图19-3：多个小视图的组合，其中每个国家在它自己的小的坐标系统中：点的逐渐变大被用来表示时间的前移（见彩图162）

我们希望了解，对用户而言，动画和静态图形表示相比效果如何。用户可以在GapMinder网站上设置自己的散点图，但是他们是否能够从自己的数据中学到新的东西？

基于美国的公共健康和人口普查数据，我们选择30组不同的（x，y）值，向用户展示非常简单的问题，如“在这个散点图中，哪个国家GDP增长最快”、“在这个散点图中，结婚率下降最快的国家在哪个洲”。我们还招募那些熟悉散点图并且在日常工作中处理数据的用户。有些人需要“探索”数据，坐在计算机前回答问题。其他人得到“讲稿”，叙述者将向他们展示可视化或者播放动画。当这些用户回答问题时，我们会记录其回答的时间和准确性。

Robertson等给出了该研究的详细数值结果（2008年）。其主要结论可以非常简单地表述如下：与其他方式相比，动画传递信息时的效率和准确度都比较低。

用动画进行的探索效率更低

我们发现，当用户自己探索数据时，他们会播放几十次动画，查看哪个国家是准确答案。相反，那些观看讲稿并且不能自己控制动画的用户的回答则要快得多：他们必须马上选择一个答案。因此，动画在用于数据探索时是效率最低的，而动画在用于展现时则是效率最高的。有趣的是，这可能正好解释了为什么Tversky的过程动画如此不成功。在我们的测试中，用户显然想要能够快进和后退；可能在过程动画中也存在同样的问题。和一系列静态图片相比，要从动画中获取到相同的信息需要付出更多的努力，因为需要全部重播，而不是直接跳到你想要看的那部分。

动画准确率更低

虽然用户在动画上花费了更多的时间，但那些观看静态可视化的用户的回答往往更准确。也就是说，动画看起来分散了用户注意力，使他们不能给出正确的答案。他们回答问题的准确性和回答速度无关：观看动画的用户花费更多的时间探索数据，但似乎并没有驱动他们产生更好的结果。

这一点似乎是动画的缺点：传递信息的效率和准确率都更低。另一方面，我们发现动画的吸引力更强，更容易打动人心：一名飞行员看到一个饱受战争蹂躏的国家的人均寿命大幅下跌了30年，他震惊得喊了出来。通常，用户更希望接触动画，认为它比其他方式更让人愉快和兴奋。虽然有时他们发现动画更让人沮丧：“那个点要去哪？”有个用户愤怒地喊道，因为有个稳步上升的点突然下降。

这些结果表明Rosling的演讲和我们的用户体验有些区别。至关重要的是，Rosling知道答案：他已经对数据进行了研究，知道自己想要表达的观点，然后带领观察者找到答案。

他在相同的轴集合上表示，这样用户就不会迷失方向。数据相当简单：在静态图中，他只突出极少的几个国家，这些国家在趋势中变化很大，而当他同时对很多国家动画显示时，显示上过于紧凑，运行方向相同。他选择轴表示，使得那些国家可以沿着一致的方向运动，使得用户可以很方便追踪起源和目标。利用“Gestalt共同命运原则”（Gestalt principle of common fate）对这些国家进行了分组，可以最清晰地表达这些国家间的过渡。

相比之下，我们的用户需要及时抓住很短的片段，记住那些突然变化的国家，而且没有解说员来对他们将要看到的进行解释；不是从解说员那里找到答案，而是需要他们自己去找到它。这对我们来说意味着我们的用户需要做的和Rosling所做的区别很大——非常不一样，实际上，这些不同之处甚至可以独立写成一章。

展现不是探索

面对着一张电子表格的分析师事先并不知道数据要说明什么，因此需要从多个角度来分析数据，寻找可能隐藏在数据中的关联、连接和思想。这个过程相当于觅食——快速地查看一张给定图形或视图来确定是否存在一些可以调查的有趣的方面，随后是移动新的过滤方式或发现新的图片。

相反，讲演者非常了解自己的数据。他们已经从数据集中清除了脏数据，可能已经删除了一些游离点或者重点强调了支持自己想要表达的核心思想的数据。他们选择能够很好地表达自己的观点的轴和时间区间，并且引导观众查看数据。最重要的是，他们不太可能需要像我们的用户那样，为了确认自己有没有忽略掉了某个点，需要快退或快进查看数据。对于这些情况，动画有其非凡的意义：它使得演讲者可以生动有趣地表达其观点。

对数据进行探索和对它进行展现是不同的。人们很容易忘记这一点，因为有很多工具把这两者混合在了一起。也就是说，许多软件包提供了很多方式来使得图形看起来更绚丽且适于展现，而且这些工具和那些用于增强图形可读性和适用于分析的工具很难完全区分开。举个例子，在Microsoft Excel表中，同一个菜单，它既控制轴是否有日志规模，而且帮助决定是否使用很绚丽的色彩来完成条形图。对于这两种工具，前者对于数据探索是至关重要的；而后者主要是用于展现。当完成在Excel表中的数据分析后，我可以直接把图形复制到PowerPoint中。由于这种无缝性，使用该软件的人们很少会有人讨论展现和探索之间的重要区别。

表19-1总结了探索和展现在需求上的主要区别。

表19-1：探索和展现的区别

	探索	展现
特征	存在意想不到的数据 可能存在脏数据 数据可能会变得难以预测 数据观察者控制如何交互	展示者对数据非常了解 数据已经清洗过 观看者是被动的
目标/过程	可以一次性分析多个维度 多次改变映射关系 寻找趋势和漏洞	为了推出某个观点，只展示较少的维度 清晰地逐个说明维度 突出关键点 把各点组织起来，说明趋势和运动

当然，探索和展现并不是完全分离的。很多交互的Web应用允许用户去探索一些维度而同时并没有暴露原始数据。展现和探索之间的关系意味着设计师需要考虑他们的可视化的目的。设计上存在权衡，使之不仅仅适用于动画而且适用于更为普遍的方方面面。

动画类型

某些类型的动画最适合于展现，而其他的可能更适合探索。在本章中，我们将讨论不同类型的转换，从改变可视化的视图到改变可视化的轴来改变可视化中的数据。我们首先一起来探讨一些系统，它们需要管理两种不同类型的变化。

动态数据，动画中心重定位

在2001年，对等网络（P2P）文件共享成为被广泛探讨的一个话题。Gnutella系统是其中最早的大规模网络之一，我认为其值得研究。Gnutella和其他的P2P系统不同。更早的Napster系统为网络中的所有东西都保留了一个非常详细的索引；BitTorrent后来完全去掉了索引。Gnutella在不同对等体（peer）之间传递搜索请求，把问题发到各个对等体，然后等待回复。当我使用P2P搜索来查找一首歌，到底会查找多少台机器？我的顾客会看到多大的网络规模呢？

我们利用Gnutella的可视化客户端，来表示整个网络。我们很快发现一些问题：首先，新的节点不断在网络上出现； 其次，知道这些节点的位置是非常有意思的。新节点的不断出现意味着我们需要能够使可视化稳定。系统中可能总会有新的数据进来，而且重要的是，随着新数据进入系统，用户不会由于可视化中数据点的变化而受到干扰。另一方面，我们不希望在有新数据时暂停可视化来添加数据和重新绘制可视化：我们希望有一个系统，新的数据可以简单而且优雅地添加到可视化中。

由于Gnutella网络使用基于P2P的发现协议，专注于单个节点及其邻居节点的研究往往可以发现一些有趣的结果。这个节点是否连接到一个中心“超级节点”?它是否发送很多

请求？我们希望能够重点查看单个节点及其邻居节点，并且能够很容易地估算节点之间的跳数。这就要求在不改变布局的其余部分的情况下能够改变视觉效果（viewpoint）。

我们的工具被命名为GnuTellaVision，或GTV（Yee等 2001）。为满足前文所述的两个需求我们使用了两种动画技术。这个可视化采用了径向布局的方式，既可以揭示数据的变化过程——随着连接的不断增加而不断向外伸展——又有利于估计中心节点和其他节点之间的跳数。径向布局的优点是拥有定义良好的中心点和一系列向外伸展的层次。在发现新的节点时，就把新节点添加到从起点开始的跳数的对应的环中。当有新的节点需要添加时，只需要移动少量的邻居节点（可视化中的多数节点不需要移动）。在运行过程中，这个可视化会随着新数据的到来而不断更新，动画也会随着改变（见图19-4）。

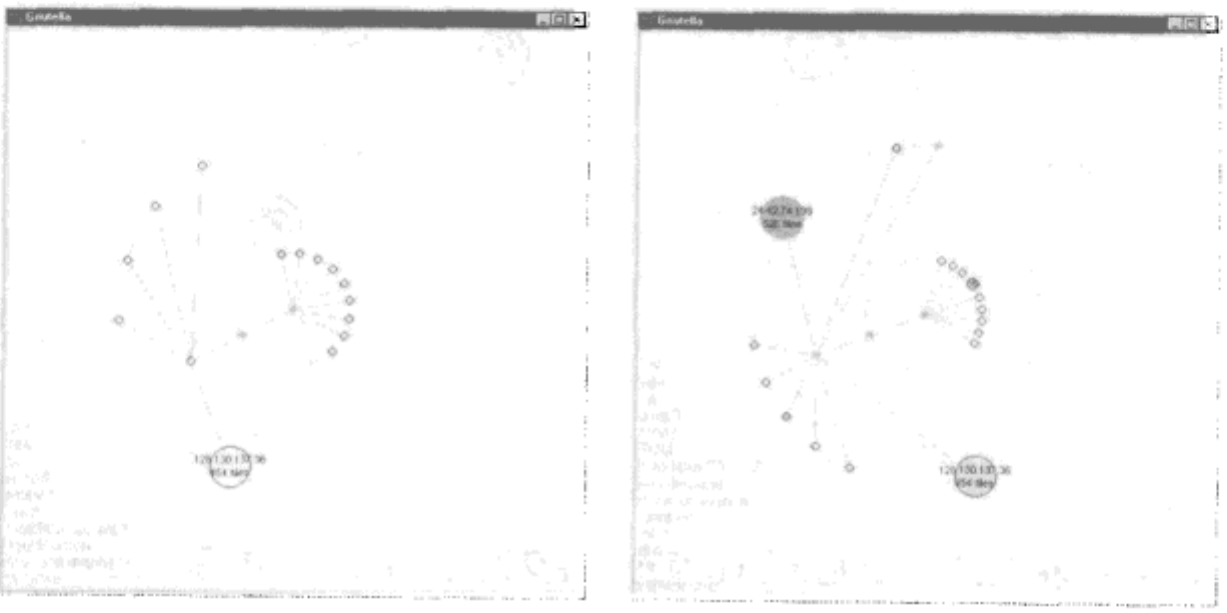


图19-4：网络中新节点出现前的GTV（左图）和新节点出现后的GTV（右图）——由于节点生成了更多的信息，它们的大小和颜色也会变化（见彩图163）

当用户查看一个节点时，GTV会重新调整画面，把选中的节点放在中心。在我们的第一个设计中，以尽可能简单的方式实现了这个功能：我们计算了一种新的径向布局，然后线性移动节点，从原来的位置移动到新位置。使用这种方式生成的结果非常令人困惑，很多节点从旧位置移动到新位置时会产生交叉。第一种解决方式是让节点沿着极坐标运动而且是始终顺时针运动。在绘制时，节点会一直保持在相同的位置，然后平滑地移动到新的位置（见图19-5）。GTV是面向检查节点（对于用户而言可能是全新的），需要不断发现新的信息，因此，使动画能够帮助用户跟踪节点的路径从而促进探索就非常重要了。

采用径向布局拥有较大的自由度：节点可以沿着半径以任何次序出现，而且任何节点都可以出现在最上面。如果我们不对这些维度进行限制，有时节点甚至会从屏幕下方运动到上方。我们希望节点尽可能少地运动，因此我们添加了一组约束条件：尽可能让节点保持相对方位和次序。相对方位保持稳定，意味着维护从可视化旧中心到新中心的连接线的相对位置。相对次序保持稳定，意味着节点的邻居在环上的次序需要保持不变。图19-6说明了这两点。

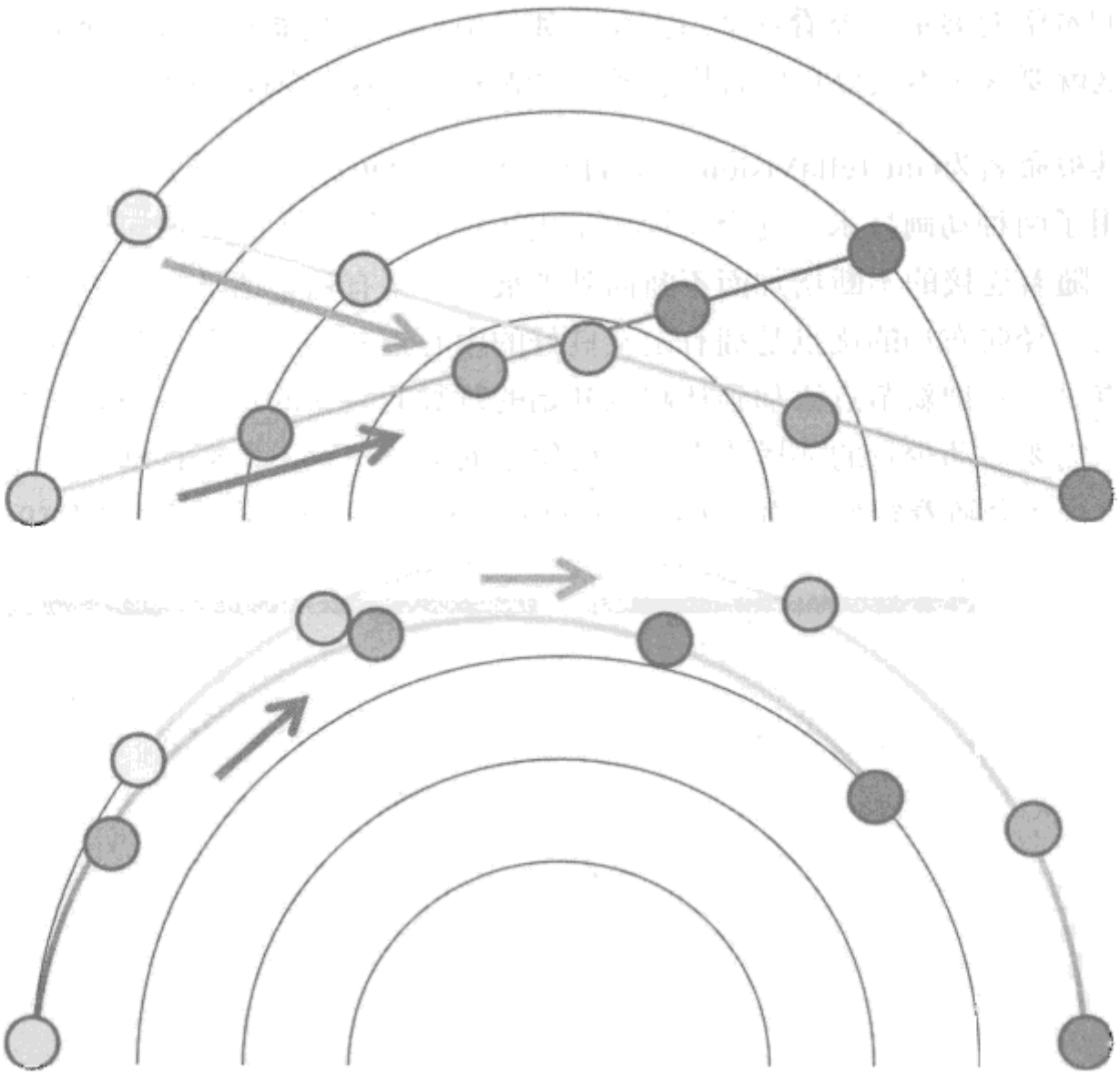


图19-5：直角坐标系（上图）的插值使节点的路径交叉在一起；极坐标系的插值（下图）使得运动变得平滑（见彩图164）

最后，为了帮助用户查看运动是如何发生的，我们借鉴了卡通中的“渐进－渐出”方式。

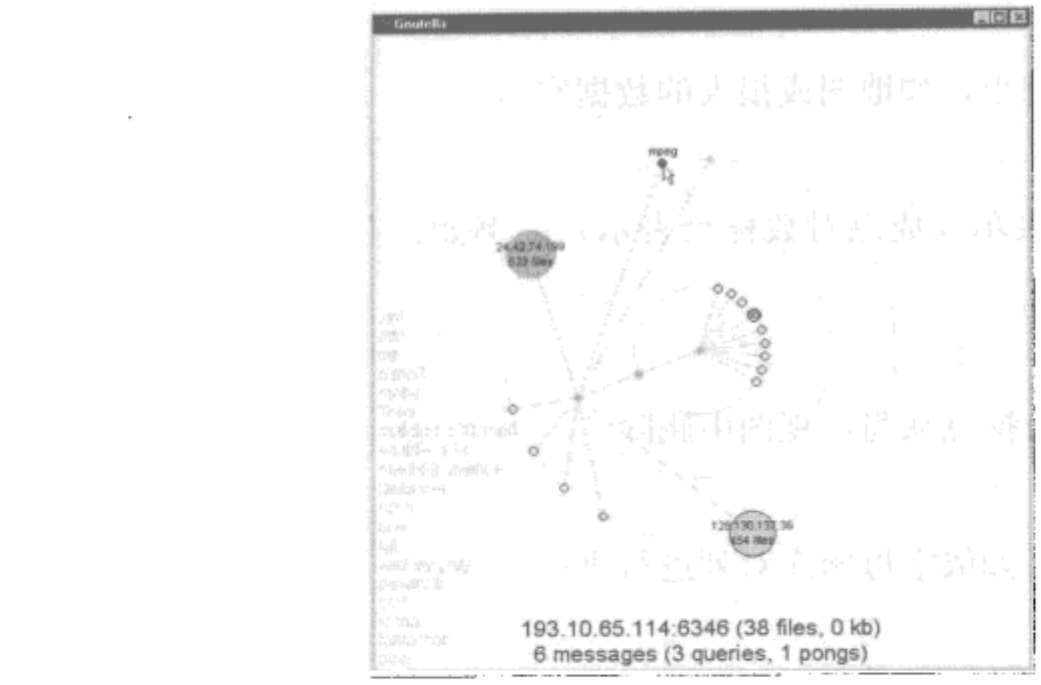
本章说明了一些值得遵循的有用的原则：

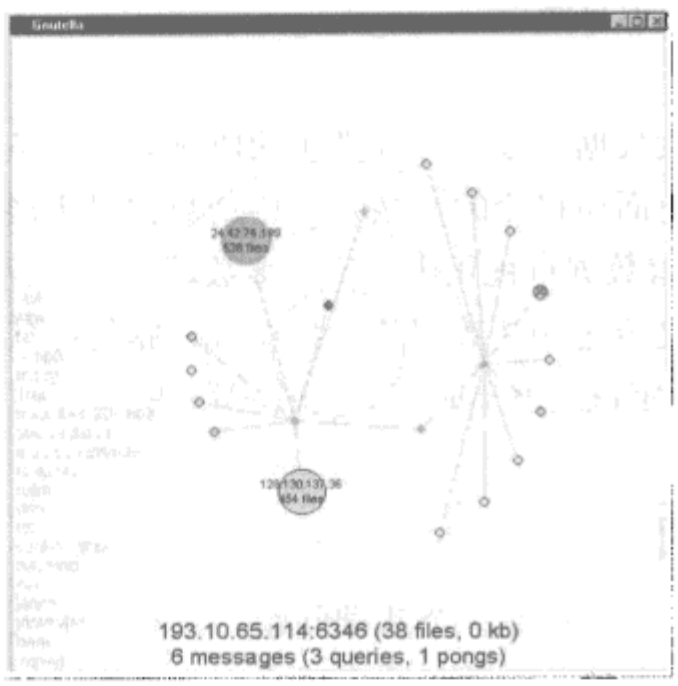
兼容性

选择一款和动画兼容的可视化。在GTV中，修改径向布局很容易；可以通过在图上放置新的节点的方式实现最小化变化的效果，而且像很多树形表示一样，可以对不同的节点进行重定位。

坐标运动

运动应该出现在一个有意义的坐标可视化空间中。我们希望用户在可视化的动画过程中始终能够定位，这样他们可以更好地预测和追踪运动。举个例子，在GTV中，使用直角坐标进行转换会让用户难以预测并深感困惑，相反地，径向坐标意味着用户可以对过渡进行追踪，可视化依然是有意义的。





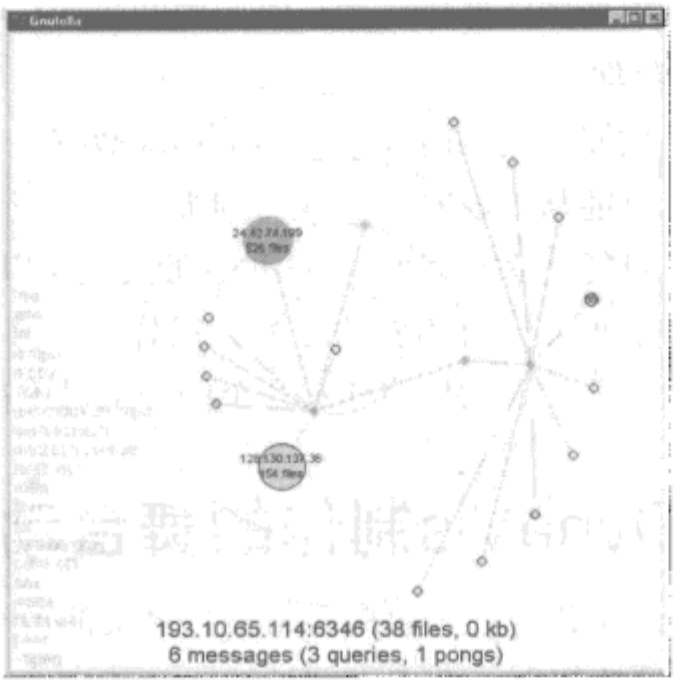


图19-6：动画中心重定位：紫色突出显示的节点变成中心，而其他节点集的相对位置和顺序保持不变（大的蓝色节点在后面，而一组小的黄色节点沿着外环依序散开，见彩图165）

有意义的运动

虽然动画是关于物体的运动的，但不必要的运动会让人困惑。通常情况下，在给定的转换中，运动的物体越少越好。对GTV动画的自由度进行限制，使可视化尽可能少地变化，使物体基本保持在相同的位置。

动画的分类

在可视化中可能会存在很多种变化。在讨论GapMinder时，我们讨论了数据的变化；在GTV的例子中，我们说明了数据和视图的变化。然而，人们希望增加的转换类型可能很多。下面这个列表是基于Heer和Robertson（2007年）的理论所做的一些修改。每种转换都是独立的；应该只改变一个元素。很多这类观点适用于数据展现和探索：

改变视图

对图片进行放大和缩小，如地图或很大的数据空间。

在图上改变绘图表面

改变轴（如从线性表示改成以对数标尺表示）。例如，在地图上，把Mercator投影[译注1]改成球形。

过滤数据

把满足特定标准的数据点从当前视图中删除。

重新排序数据

改变数据点的次序（如依字母顺序对列进行排序）。

改变展现方式

条形图改成饼状图；改变图的布局；改变节点颜色。

改变数据

按照一定的时间步长向前移动数据，修改数据，或者更改描述的值（如一个条形图可能从“赢利”变成“损失”）。正如前面所讨论的，按时间移动数据对于展现很可能是更有用的。这6种过渡类型可以涵盖数据可视化中的绝大多数动画。过程可视化可能会有不同的分类，正如科学可视化传达的是数据流（如穿过翅膀的空气）。下一步，我们将通过几个例子来讨论在给定该过渡集时该如何管理这些动画。

用DynaVis制作的舞台动画

在一台计算机上一起探索某个数据集的两个人之间存在一个基础问题：只有一个人有鼠标。虽然其中一个点击“过滤”可能完全出于直觉，但是另一个用户可能无法追踪发生了什么事情。这一点介于探索和展示之间：动画的一个主要目标是促使第二个用户能够跟上第一个用户做出的改动；但是，第一个用户可能并不清楚自己想要做什么。动画可以是多个可视化之间的一种过渡，使得第二个人（或者是一个观众）能够跟上。在过去几年，我们一直在尝试以不同的方式来显示数据转换和对已知图表的展现，如散点图、条形图甚至是饼图。

DynaVis是一个动画可视化框架，我们采用了该框架。Jeff Heer，现在是斯坦福的教授，他暑期对我们做了访问，使我们有机会对很多可能的应用方式进行了尝试。在他发表的论文中比较详细地描述了这次讨论（Heer和Robertson 2007）。

在DynaVis框架中，每个条形栏、每个点或每条线都代表三维空间中的一个对象，因此

译注1：Mercator投影，又称正轴等角圆柱投影，是地图投影中影响最大的，如想要了解更多，可以访问：*http://baike.baidu.com/view/301981.htm*。

我们可以让本章前面描述的所有转换平滑运动。很多转换都很清晰：举个例子，从散点图中过滤一个点，只需要把这个点隐去。但是，在一些情况下可以用更为有趣的方式实现：展示类型变化的，在某个时刻有多个改变同时发生的。当展现改变时，我们尽力遵守几条基本原则。以下是最重要的两条：

一次做一件事

确保可视化不需要同时做出多个变化。这意味着该可视化被分解成了多个步骤，每个阶段可以确保在下一步开始之前已经完成。

确保有效映射

在每一步的任意时刻都需要确保可视化是有意义的，即确保存在一个从数据到可视化的映射。举个例子，对条形图的条形栏进行重命名将会是无效的：映射的基础是每个条形栏代表一个*x*轴值。

图19-7是将条形图转换为饼状图的首次尝试。通过这次转换发现了很多积极方面。举个例子，条形图的各个条形栏不会每次全部移动，因此人眼能够很容易地跟上运动，而在动画过程中条形栏保持其特征值不变。虽然在条形栏因为运动而互相交叠时存在一些问题，但它们可以按照平滑轨迹运动，这样预测轨迹将在哪里结束也是可行的。最后，动画制作过程进行了良好的分解：所有的楔子首先会全部生成好，然后才组合成一个完整的饼状图。

但是，这种可视化有一个非常大的缺陷。条形栏的长度被换算成楔子的周长，因此，条形栏越长，其对应的楔子的周长也越长。但是，在最终的饼图中，长度越长的条形栏，则其对应的楔子将更宽。这意味着条形栏将变得又宽又长，或者又窄又短。这反过来意味着可视化并没有适应于一个恒定规则（如“像素数和数据值成比例”）。

由此可以引出下一个原则：

保持不变

虽然第一条规则提到了数据元素和显示标志之间的关系，但是该规则限定的是数据值和可视化之间的关系。如果数据值不变，则在整个可视化过程中系统应该保持这些值不变。举个例子，如果每个条形栏的高度和与其对应的数据点的值一致，条形栏应该在动画中保持相同高度。

图19-8中的更为成功的条形图和饼图动画阐明了这两个原则。该图说明了绘图实体（条形栏、折线或者楔子）和底层数据之间的一一对应关系。编排方式从未改变：最左边的条形栏（“A”）对应最左边的饼图切片（也是用“A”表示）。不变量对应于条形图的各个条形栏的长度，它和数据值保持一致。尽管我们不准备在这里详细说明从条形图到折线图的转换，但是，实际转换时也遵循了相同的原则：最左上方条形栏收缩成一条折线之后，将仍保留在最左上方。

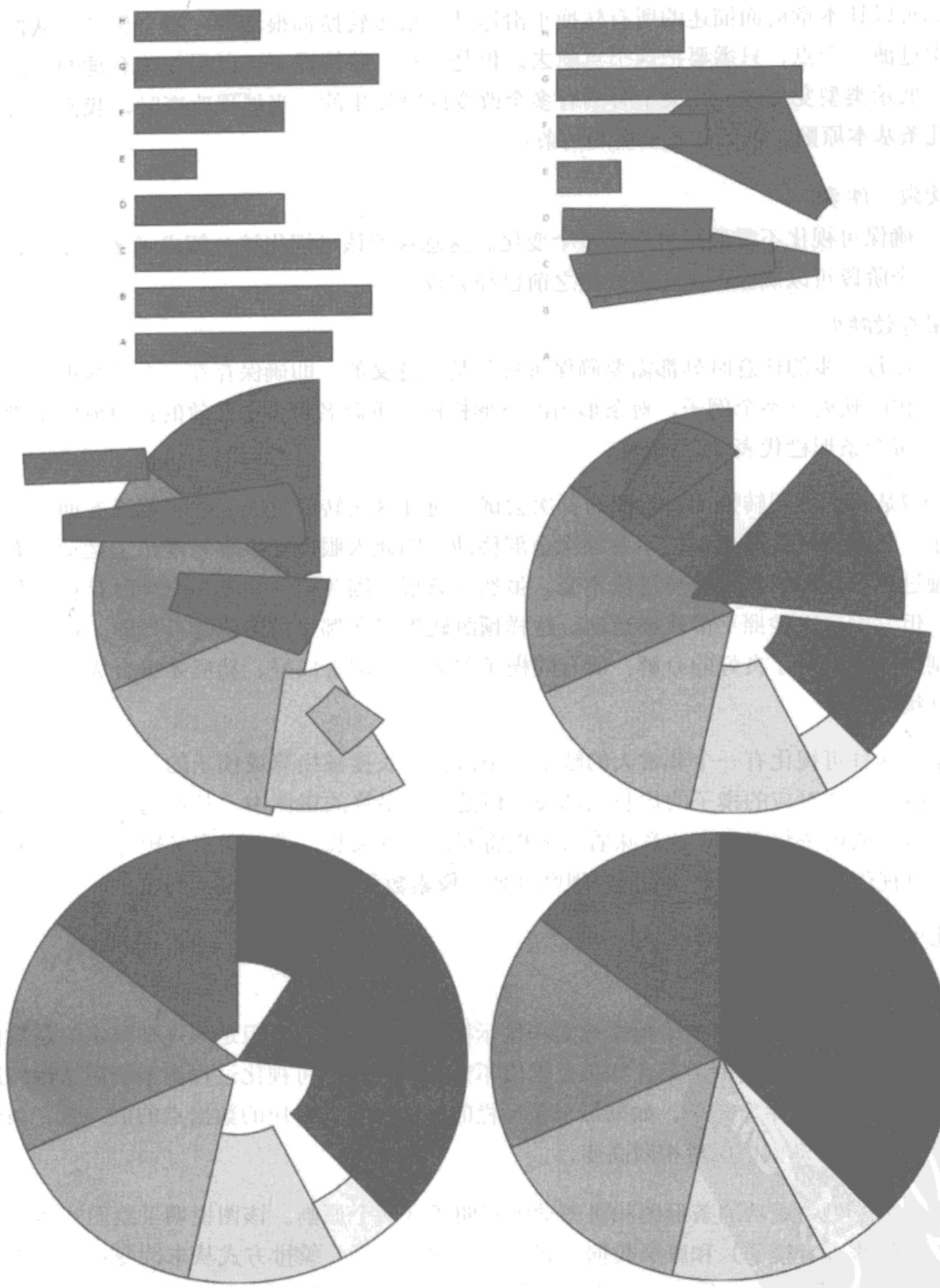


图19-7：条形图到饼图的不太成功的动画：条形图中的较长的条形栏在饼图上变成了又胖又长的楔子；短条形栏变成了又瘦又短的楔子；然后所有的楔子组合在一起形成了一张饼图（见彩图166）

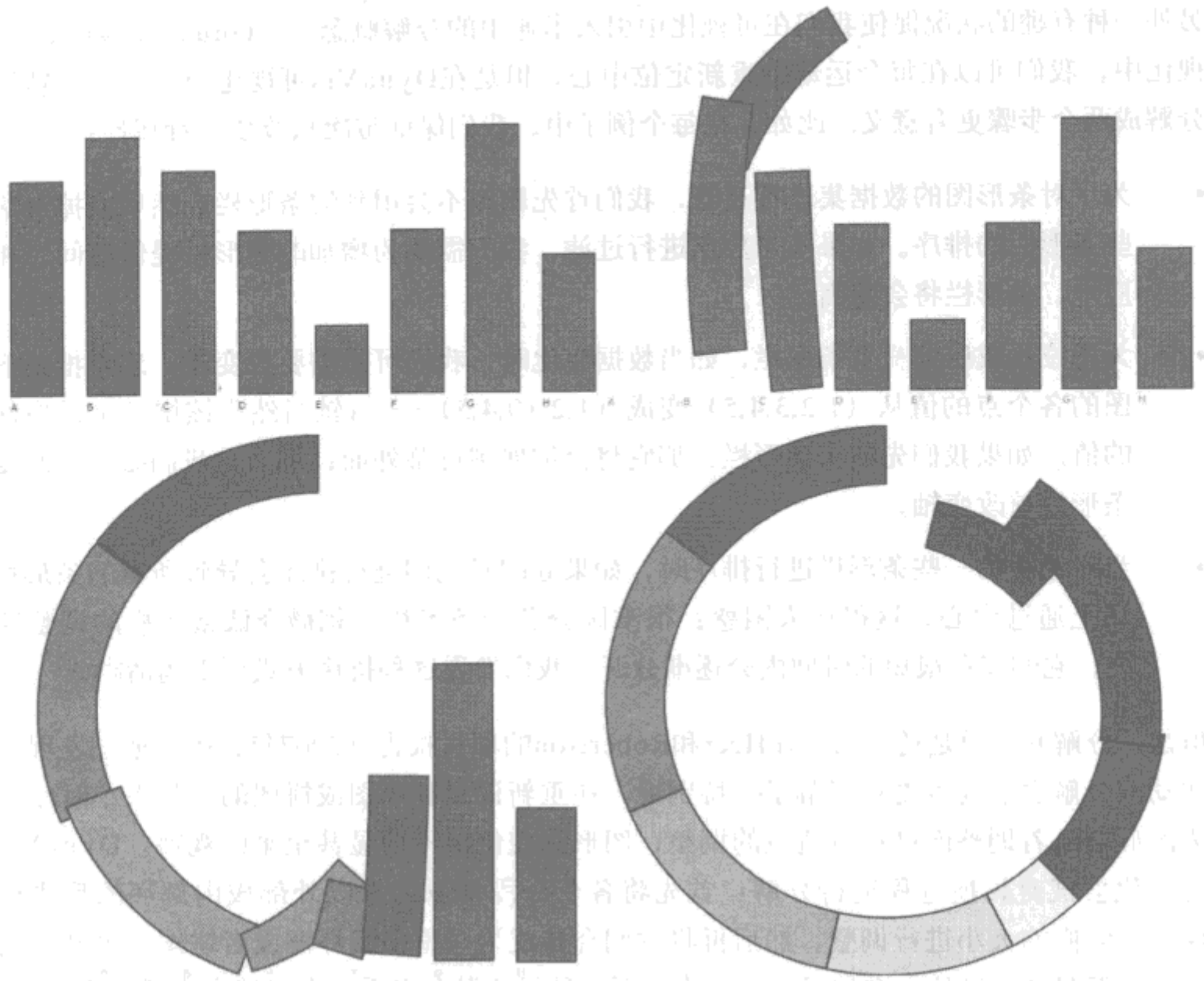


图19-8：条形图到饼图的相对比较成功的动画：条形栏的长度保持不变，首先变成弧形；其次合在一起成为一个环，最后组合成为一张饼图（见彩图167）

另外一种有趣的状况促使我们在可视化中引入卡通中的分解概念。在GnuTellaVision可视化中，我们可以在每个运动中重新定位中心，但是在DynaVis可视化中，把一次转换分解成两个步骤更有意义。比如，在每个例子中，我们保证每次只改变一种属性：

- 为了对条形图的数据集进行过滤，我们首先删掉不会用到的条形栏，然后去掉对这些条形栏的排序。如果不对数据进行过滤，我们需要为增加的条形栏提供空间，相应地，条形栏将会变高。
- 为了拉长或压缩一个条形栏，如当数据变化时，我们可能需要改变轴。想象把条形图的各个点的值从（1,2,3,4,5）变成（1,2,10,4,5）——*y*轴当然应该增长来适应新的值。如果我们先增大条形栏，那它将会扩展到屏幕外面；那么，我们必须在改变条形栏前改变轴。
- 当对选中的一些条形栏进行排序时，如果立即对它们进行排序会导致所有的条形栏马上通过中心。这很让人困惑：很难区分各个条形栏。稍微降低条形栏的调整速度，它们将在很短的时间内分逐渐分开，我们发现这种排序方式要更为清晰。

但是，分解并不总是适当的。在Heer和Robertson的项目报告（2007年）中，他们发现一些动画分解之后变得更难理解了。特别地，在重新设置圆环图或饼图的分片大小时，因为图形会随着调整而进行自适应的调整，图形的变化很不明显甚至难以观察。DynaVis可视化尝试对转换过程进行分解：首先将各个分段抽取出来以外部或内部环的形式展现，对它们的大小进行调整，然后再将它们合并成为一整张圆环图或者饼图。虽然这使得变化更明显，但是它增加了一种行为，而这种行为潜在来看也有可能令人感到困惑。

Heer和Robertson同时收集了定性结果和定量结果。定量结果，即有多少用户喜欢动画；定性结果，即找出哪些动画能够使用户最为准确地回答问题。他们发现，使用动画方式时，用户会更易于回答取值随时间变化的问题；此外，分成多个阶段但每个阶段只有一种转换的动画比那些同时执行了很多种转换的动画的效果要好得多。

基于这些原则，显然这类动画更有可能有利于帮助用户理解转换：相比于演讲者抛出一系列图表并强迫观众适应一个又一个的幻灯片， DynaVis的框架允许用户在整个过程中都保持集中。

动画原则

关于动画的原则，已经有一些有益的尝试。Tversky、Morrison和Bétrancourt (2002年)在他们论文的最后给出了两条通用的指导原则：可视化应该保持一致性和易理解性。前者意味着屏幕上的标志必须总是和底层数据相关联。后者表示可视化必须易于理解。我们前面给出的几条原则也在这两条原则的范畴之内。Heer和Robertson（2007年）对

DynaVis框架的可视化的讨论中另外给出了一些相关的指导原则，Zongker 和 Salesin（2003年）在幻灯片中给出的是动画相关的讨论，Freidrich和Eades（2002年）给出的是图形相关的讨论。

我们在本章中已经讨论过的原则如下：

分段展示

一次性转换太多会分散人们的注意力。如果可以每次只改变一件事，那就只改变一件事。另一方面，有时多个变化必须同时发生，这时，可以将它们分解为多个步骤逐步展示。

兼容性

动画造成困扰的可视化都是因为用户难以跟踪变化。举个例子，给一个条形图增加一个条形栏并不会困扰用户（整个集合可以平滑变动），而在条形栏中另外增加一个序列就可能让人困惑了。但是，一个四方形的树形图是根据尺寸按照贪婪的方式布局的；一个矩形的变大可能需要改变所有矩形的位置，这会让人深感困惑。

必要的移动

特别地，避免不必要的移动。这意味着我们希望确保移动都是有意义的——也就是说，我们将只对变化进行动画展示。总的来说，图像应该总是可理解的。正如对DynaVis框架的用户测试结果所表明的，过渡的动作——即使是有意义的动作，也会让人困惑。

有意义的移动

移动的坐标空间和类型应该是有意义的。这也说明了之前提到的两点：保留有意义的映射并维持不变性。

确定自己坚持了这些原则会帮助你确保动画是在沿着正确的方向演化。

结束语：是否采用动画

在本章，我们讨论了数据展现和探索之间的区别、可视化中可能会变动的各种层次，以及一些确保动画可视化有效的原则。

因此，到了现在这个阶段，你可能正在盯着自己的可视化，试着决定是否要采用动画的方式来展示。本章不断询问的问题是：该动画的功能是什么？如果是为了使用户在多个视图之间能够平滑过渡，那么很可能是有用的。如果用户是为了比较“之前”和“之后”，动画很可能没有用处。

用户希望理解变化的原因和变化的具体内容。如果屏幕上的所有东西都需要移动，能够自动切换到新的图像可能会更好；这可以让户可以更易于追踪不同之处。最后，动画意

味着可视化的打印会更困难。各个帧应该都是有意义的，这样用户可以捕获并融会贯通这些图片。动画增加了复杂性，该复杂性应该有所回报。

扩展阅读

以下是和本章内容相关的一些动画数据可视化项目，你可能会希望进一步了解：

- 很多研究人员开始使用Pad++中的缩放和拖拽作为可视化的基本操作，Pad++是一个在大空间中放置数据的可伸缩架构（Bederson 和 Hollan 1994）。
- Scatterdice（Elmqvist、Dragicevic和Fekete 2008）发现了一种通过旋转第三维度来实现散点图之间转换的方式。
- 树形图数据可视化包括ConeTrees（Card、Robertson和Mackinlay 1991）、CandidTree（Lee等 2007）和Polyarchy（Robertson等 2002）。研究人员通过缩放（扭曲）树形图（Blanch和Lecolinet 2007）来探索树形图动画和在三维空间中的运动（Bladh、Carr和Kljun 2005）。
- 图像布局往往用动画展示布局的过程；在过去10年中，图像绘画社区开始考虑基于底层数据来更新图像。除了前面提到的作品（Friedrich和Eades 2002），还有GraphAEL（Erten等 2003）。

致谢

感谢斯坦福大学的Jeffrey Heer教授，我们在同一个办公室时，他在2007年Infovis论文（Heer 和 Robertson 2007）和他的斯坦福教程笔记中就这些话题提了很多有价值的见解以及这些概念的概括抽象。Jeff在本书的姊妹篇《数据之美》中写了一章，是关于sense.us的数据探索的。此外，还要感谢同事Steven Drucker、Roland Fernandez、Petra Isenberg 和 George Robertson，它们为本章提了很多反馈和意见。

参考文献

1. Bederson, B.B., and J.D. Hollan. 1994. "Pad++: A zooming graphical interface for exploring alternate interface physics." In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*. New York: ACM Press.
2. Bladh, Thomas, David A. Carr, and Matjaz Kljun. 2005. "The effect of animated transitions on user navigation in 3D tree-maps." In *Proceedings of the Ninth International Conference on Information Visualization*. Washington, DC: IEEE Computer Society.

3. Blanch, Renaud, and Eric Lecolinet. 2007. "Browsing zoomable treemaps: Structureaware multi-scale navigation techniques." *IEEE Transactions on Visualization and Computer Graphics* 13, no. 6: 1248–1253.

4. Card, Stuart K., George G. Robertson, and Jock D. Mackinlay. 1991. "The information visualizer, an information workspace." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press.

5. Cavanagh, Patrick, and George Alvarez. 2005. "Tracking multiple targets with multifocal attention." *TICS* 9: 349–354.

6. Chang, Bay-Wei, and David Ungar. 1993. "Animation: From cartoons to the user interface." In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. New York: ACM Press.

7. Elmqvist, N., P. Dragicevic, and J.-D. Fekete. 2008. "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation." *IEEE Transactions on Visualization and Computer Graphics* 14, no. 6: 1141–1148.

8. Erten, C., P.J. Harding, S.G. Kobourov, K. Wampler, and G. Yee. 2003. "GraphAEL: Graph animations with evolving layouts." In *Proceedings of the 11th International Symposium on Graph Drawing*. Springer-Verlag.

9. Fisher, Danyel A. 2007. "Hotmap: Looking at geographic attention." *IEEE Transactions on Visualization and Computer Graphics* 13, no. 6: 1184–1191.

10. Friedrich, C., and P. Eades. 2002. "Graph drawing in motion." *Journal of Graph Algorithms and Applications* 6, no. 3: 353–370.

11. Heer, Jeffrey, and George G. Robertson. 2007. "Animated transitions in statistical data graphics." *IEEE Transactions on Visualization and Computer Graphics* 13, no. 6: 1240–1247.

12. Hundhausen, Christopher D., Sarah A. Douglas, and John T. Stasko. 2002. "A metastudy of algorithm visualization effectiveness." *Journal of Visual Languages & Computing* 13, no. 3: 259–290.

13. Johnson, Ollie, and Frank Thomas. 1987. *The Illusion of Life*. New York: Disney Editions. 14. Lasseter, John. 1987. "Principles of traditional animation applied to 3D computer animation." In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press.

15. Lee, Bongshin, George G. Robertson, Mary Czerwinski, and Cynthia Sims Parr. 2007.

"CandidTree: Visualizing structural uncertainty in similar hierarchies." *Information Visualization* 6: 233–246.

16. Michotte, A. 1963. *The Perception of Causality*. Oxford: Basic Books.

17. Robertson, George, Kim Cameron, Mary Czerwinski, and Daniel Robbins. 2002. "Polyarchy visualization: Visualizing multiple intersecting hierarchies." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press.

18. Robertson, George, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. 2008. "Effectiveness of animation in trend visualization." *IEEE Transactions on Visualization and Computer Graphics* 14, no. 6: 1325–1332.

19. Tversky, Barbara, Julie B. Morrison, and Mireille Bétrancourt. 2002. "Animation: Can it facilitate?" *International Journal of Human-Computer Studies* 57: 247–262.

20. Yee, Ka-Ping, Danyel Fisher, Rachna Dhamija, and Marti A. Hearst. 2001. "Animated exploration of dynamic graphs with radial layout." *In Proceedings of the IEEE Symposium on Information Visualization*. Washington, DC: IEEE Computer Society.

21. Zongker, Douglas E., and David H. Salesin. 2003. "On creating animated presentations." *In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York: ACM Press.

第20章

带索引的可视化

Jessica Hagy

可视化：是一头“大象”

可视化，在第一个人眼里是图形图表和投资回报率（ROI）；在第二个人眼里却是插图、生动的隐喻以及画廊开幕；在第三个人眼里它只是奇妙的冗余的复合词：信息图形。可视化，这个术语就像一个抽象的太妃糖一样需要人们不断咀嚼、反复品味。它就像三个盲人摸象的故事。一个摸到大象的尾巴，说“大象像一条绳子”。另一个摸到大象的脚，说“大象像一棵树桩”。第三个人摸着大象的鼻子，说“大象像一条蛇”。他们都不是完全错误的，但是也没有一个是完全正确的，因为没有一个人可以看到大象的整体，如图20-1所示。

可视化只是你能够看到的某些部分（全部）。它既是整个马赛克，也是单个闪闪发光的镶嵌物。它不仅仅是图表，也不仅仅是视觉隐喻；它不仅仅是取代子弹点的可工作的图形设计，也不仅仅是描绘思想；同样，它不仅仅是数据分析。这些都是更大的概念中的一小片。

真正优秀、美丽、强大的可视化，即触及思想和内心深处的可视化，不仅仅是关于图像、快照和通过玻璃窗的查看，如图20-2所示。强大的可视化在大象测试中能够通过：几乎无法形容，但是一眼就能够识别。本章将讨论这头“大象”的方方面面。总之，这些讨论将有助于绘出一幅能够整体上清晰描述可视化对象的图像。

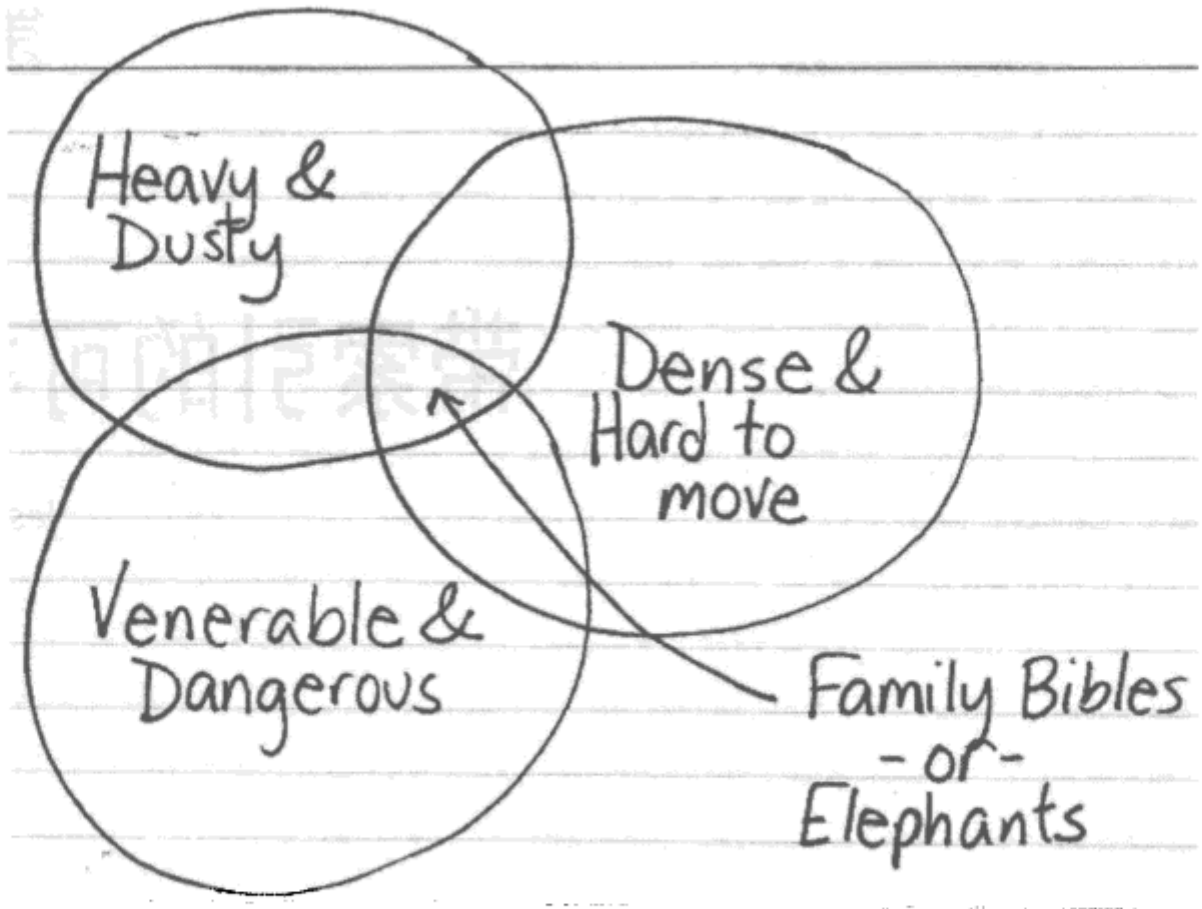


图20-1：总是还有更多

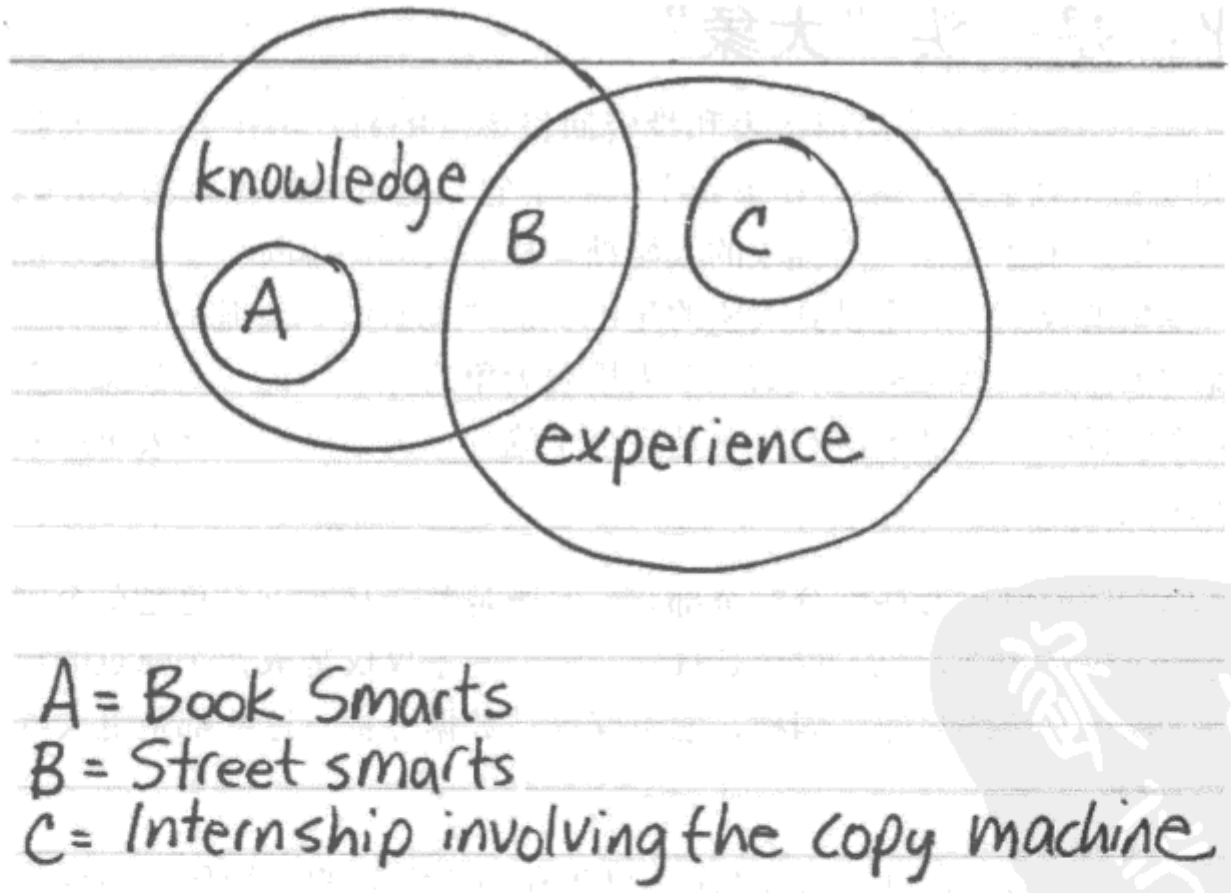


图20-2：知识和实践齐头并进

可视化：是一门艺术

可视化中有一张图片，还有一条信息。人们需要“审问之”、“慎思之”并“明辨之”。可视化的创作者们也因此而获得了更长的工作时间。质量是主观的，而美学总会有争议——但是内在的艺术性却是显而易见的，好比色情。艺术是只有当你看到时才能知道，而且无法提早知道，如图20-3所示。而可视化是一门艺术已经广为认可。

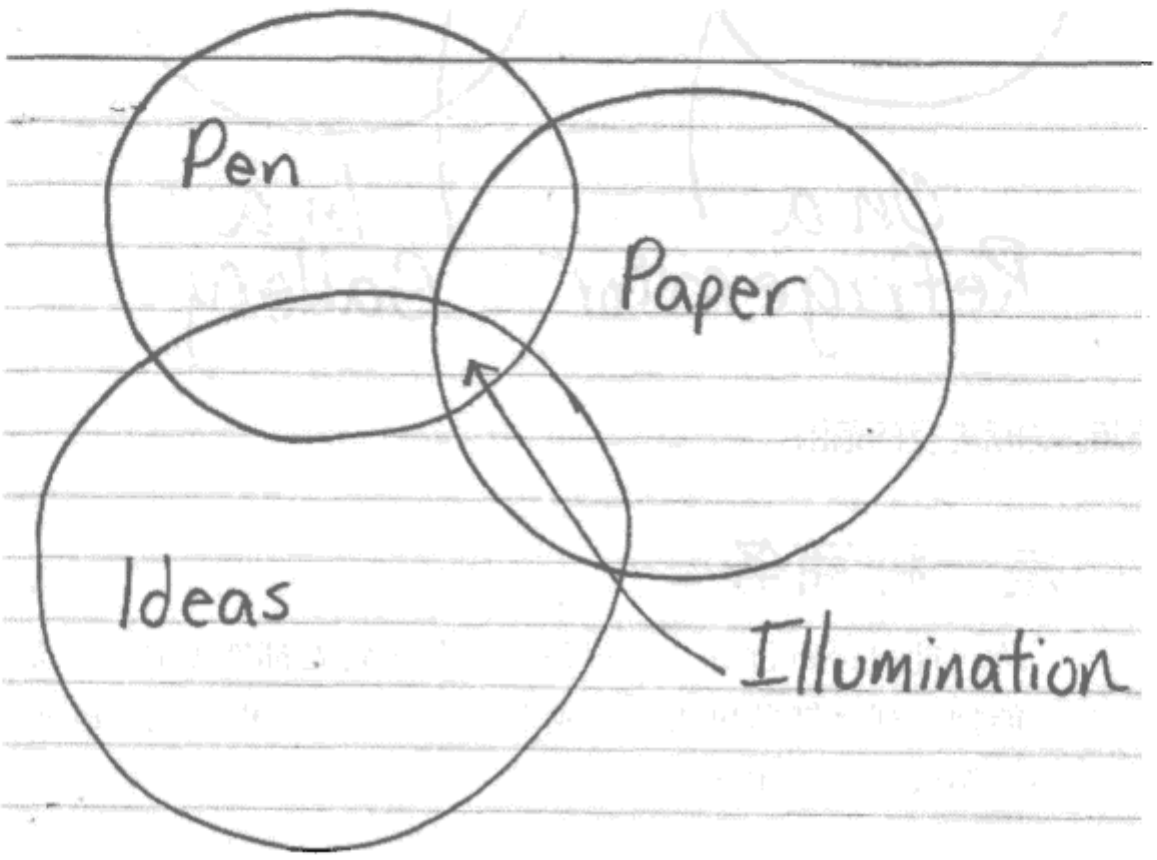


图20-3：可视化的“恍然大悟”时刻

可视化实践者往往富于创意：他们描出、画上或者带着有厚厚的黑色镜框的眼镜。当然，只要一件事情被贴上艺术创作的标签，其进入门槛就会变高。那些认为自己不具备创造力以及从未认为自己具有“创造性”的人会因此而回避可视化。这太糟糕了——因为你不需要成为Rembrandt，只要你有一些想法就可以画几笔并能为人们所了解。

可视化内在的美（有争议的）在于图像背后的思想：你的视网膜杆和视网膜锥看到的线条和形状所要说明的概念。从技术上说，只要有黏土，任何人都可以进行雕刻，而任何人只要有可以可视化表达的思想就可以创建可视化，如图20-4所示。雕塑或可视化的质量往往都是有争议的。任何艺术品或图像的质量都是值得商榷的。

可视化，从观察者的眼里得到思想。

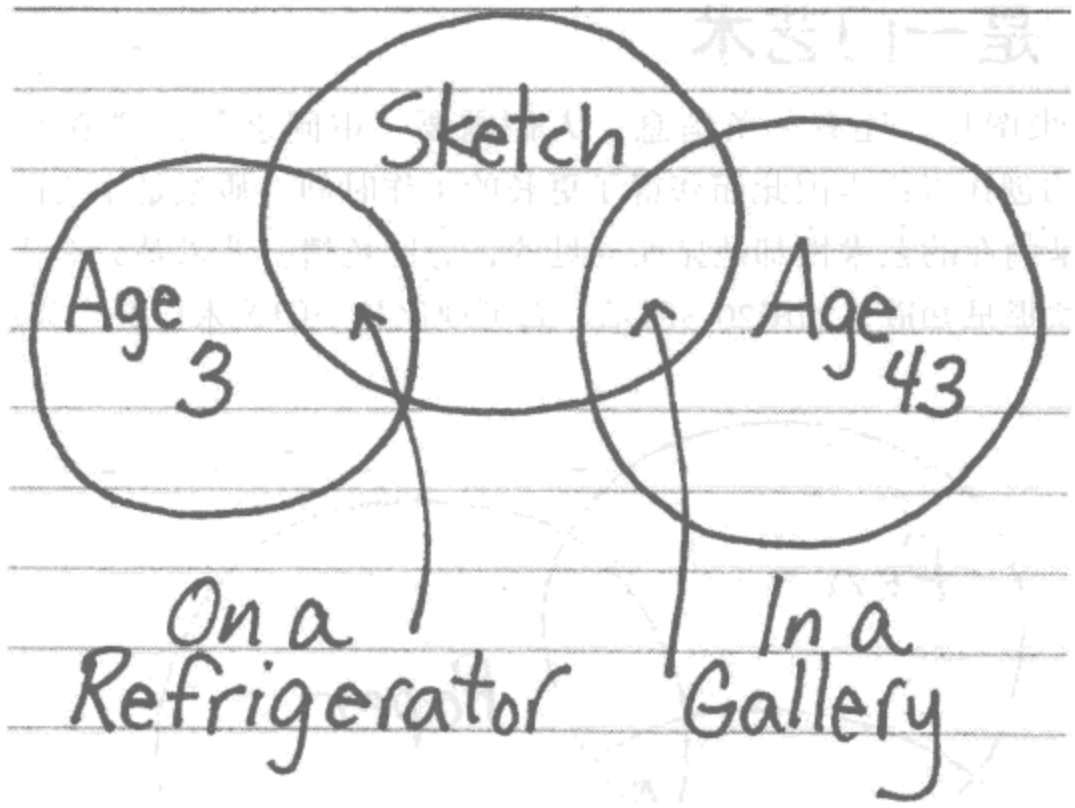


图20-4：不要超出力所能及的范围

可视化：是一种商务

有一个小程序——你可能已经听过它。它很便宜，几乎到处都能见到，它是一个可爱的中间管理件工具，能把可视化辅助想法转换成卡其色。这个小程序就是PowerPoint，把可视化转换到了商业领域，如图20-5所示。

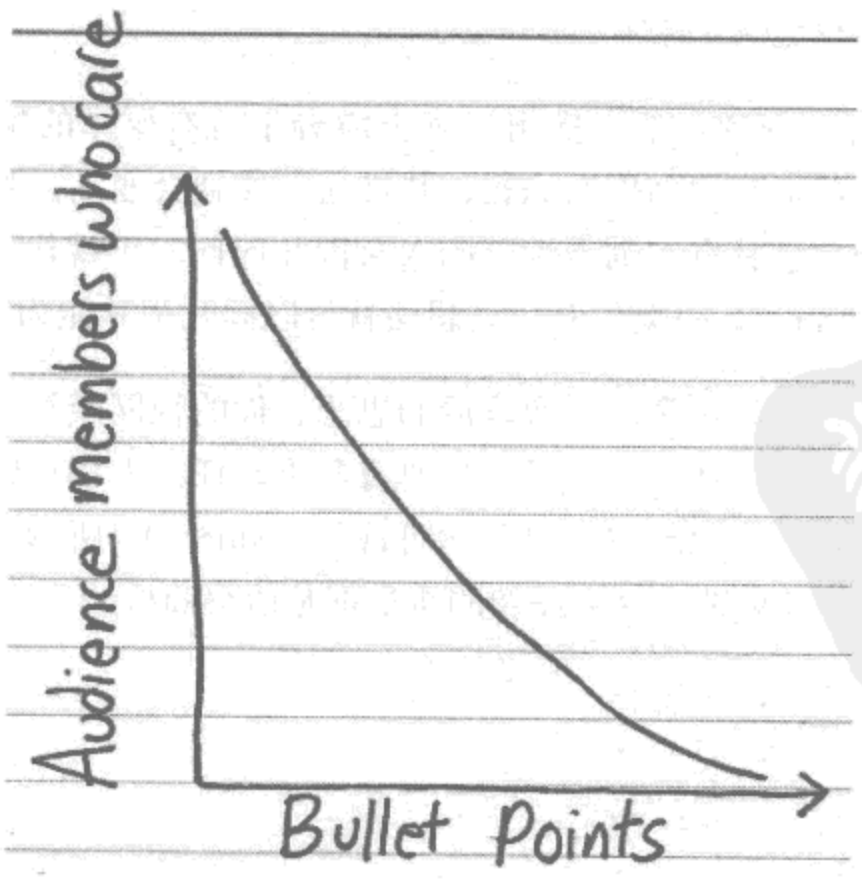


图20-5：Power point =自相矛盾用语

不可否认：可视化是吸引人的。希望别人忘记你讲述的内容的枯燥乏味？如果你在大学中教授数学，一定要确保在你的讲稿中包含了很多图片。当向董事会、一位潜在客户或中年级的MBA同学做报告时，如果没有PowerPoint，在最好的情况下会被认为很奇怪，在最差的情况下会被认为准备不充分（见图20-6）。这是什么原因呢？因为可视化是一款优秀的说服工具，说服是销售的另外一种表达方式。

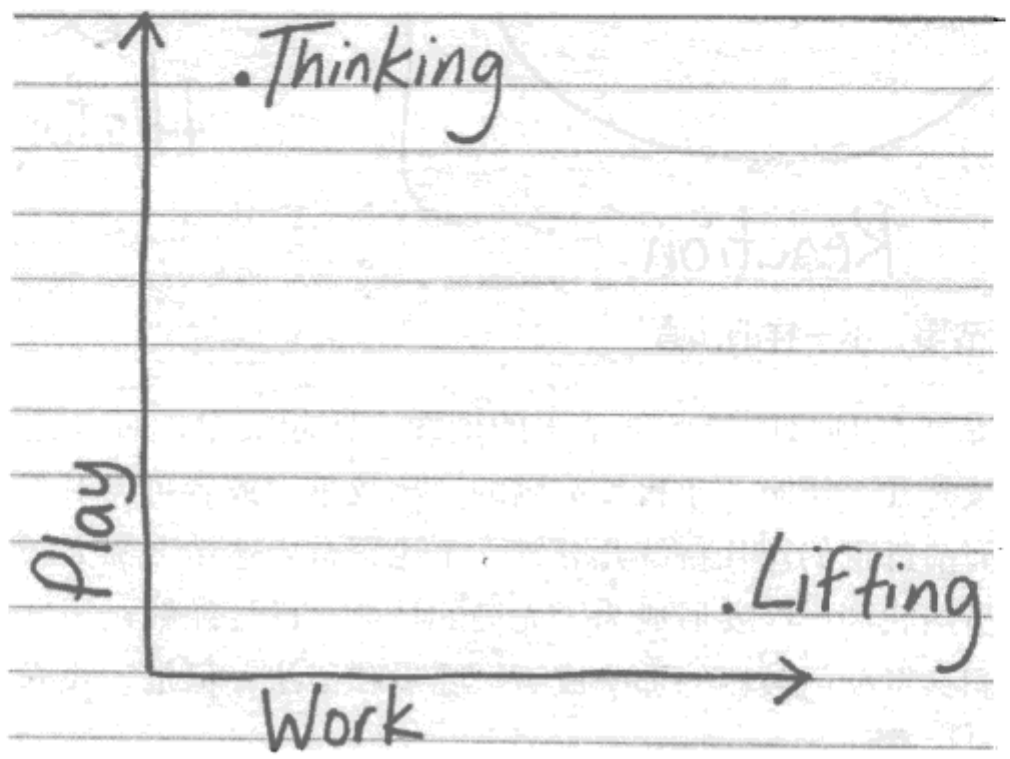


图20-6：想法可以为你工作

合并、收购、谈判、广告和宣传。人们每天都在传递商务通信信息。我的餐巾纸在这里。这是我在刚刚的四小时会议上提出的战略涂鸦。

眼见为实。相信之后，人们才会认同。你觉得公司总部、政治“王朝”和辉煌的教堂都是如何建立起来的？

可视化：是永恒的

法国那些著名的洞穴壁画不是待做事情列表、句子、单词、甚至也不是字母。它们是图像。几千年前，象形字中的每个字符都是用图像表示的。现在的书面汉语也是如此。我们在学会单词之前先学会了微笑。语言再强大，也比不上可视化直接或形象，如图20-7所示。

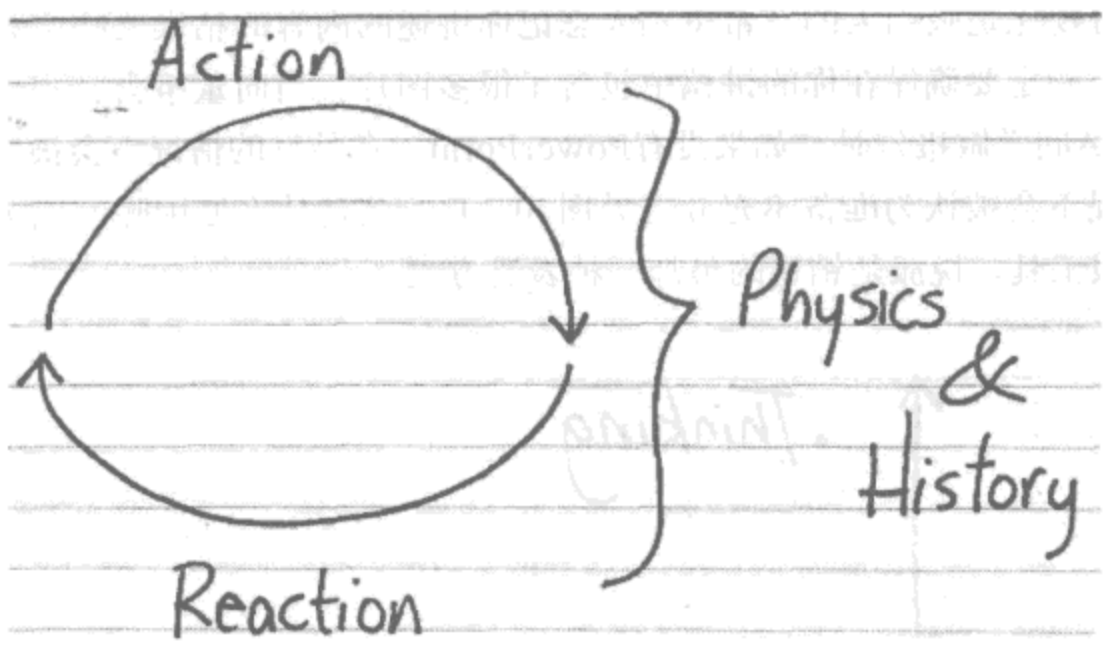


图20-7：一样的老故事，不一样的作者

照片、油画或者是天气预报的绿色屏幕上的地图，比起文字，可以让我们学得更多更快。我们可以连续听几个小时关于极度穷困的介绍故事，或者可以对着一幅一只秃鹰在一个瘦弱的小孩面前徘徊的图片持续几分钟。不论语言表达多么强大，使用图像可以更快地分享故事。虽然我们已经发展成为了先进的社会，已经能够运用复杂的词汇、语言以及修女们在我们孩童期间教授的那些成语、隐喻和语法，但是，我们仍然可以不通过语言而只是通过图片进行交流，如图20-8所示。

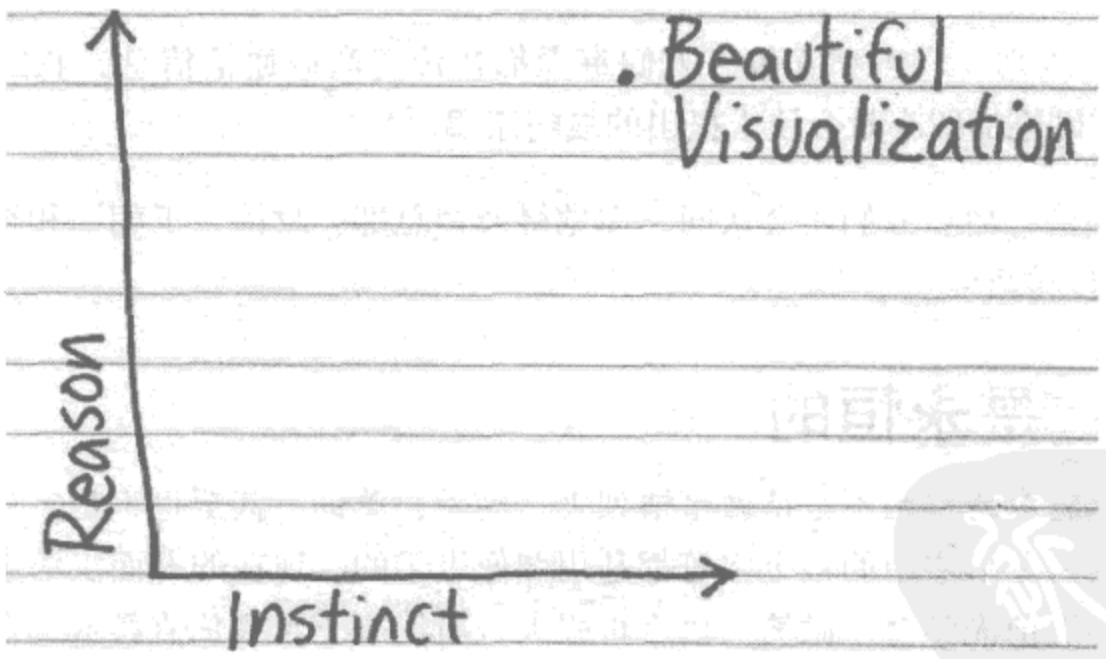


图20-8：眼见为实

想象一下：古代用泥土画的洞穴壁画和形状。没有梵文，没有诗词，没有PowerPoint。

可视化：此时此刻

哪一种含义更深刻：名字还是标识？人们如何认出你：是你的头像还是你的简历？不动产的最珍贵之处在哪里：是一个著名的网址（URL）还是在一个著名邮政区域的一个地段？今天，标识可以讲述史诗故事。屏幕名字等同于人们的身份。Web致力于资助创新，购买牧场、岛屿和街区。

与以往相比，我们现在是在信息海洋中畅游。我们在对数据进行清洗。每天生成的信息是人类世界从未有过的或者从未期望能够理解的，如图20-9所示。因此，我们把可视化作为收集、浓缩和传递信息的工具。

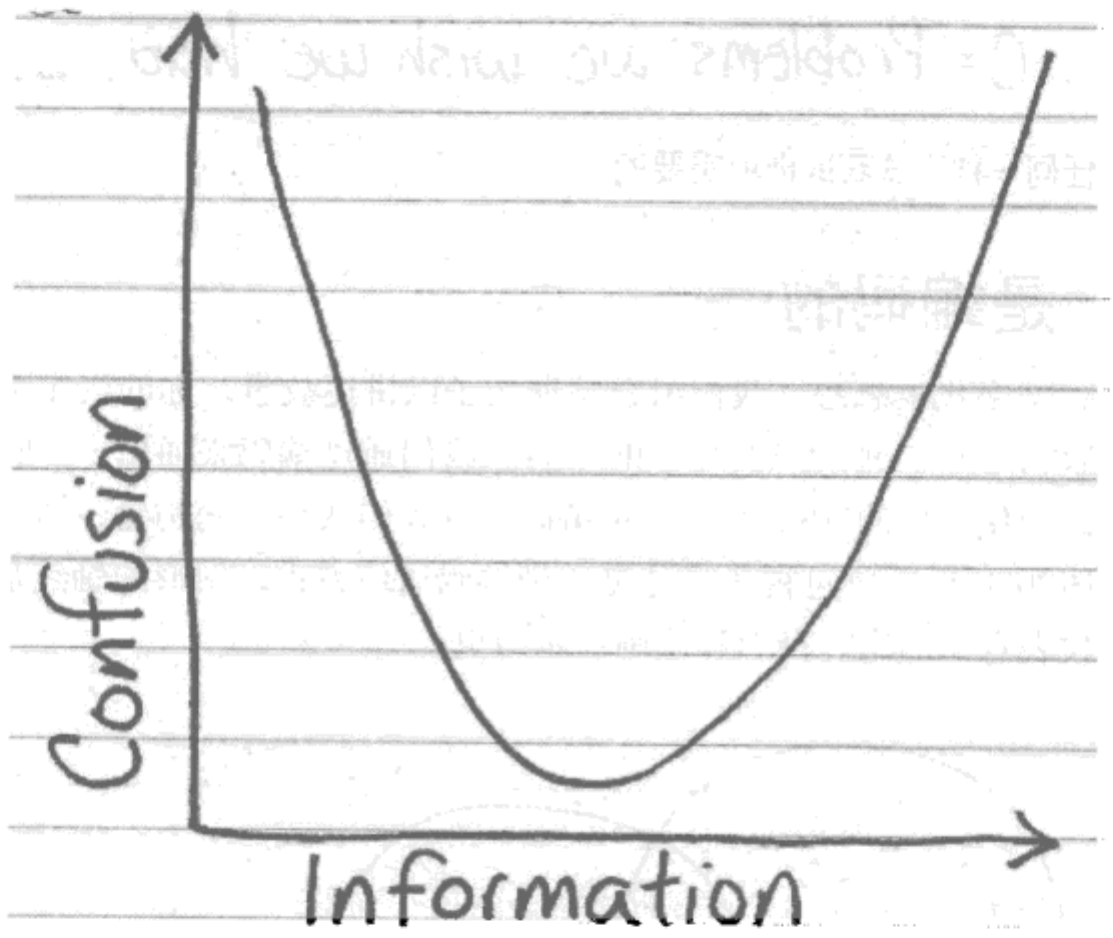


图20-9：水，还是水，到处都是水

视觉处理数据。视觉把大量矮胖、笨拙、黑色和白色的电子表格压缩成光滑、彩色的图形。视觉能够揭示大量数据中的模式；它们运用复杂、难以理解的理论，可以优雅地解释数据，如图20-10所示。把数据点想象成冰分子。可视化能够生成“雪花”：它是对很多小的信息片的华丽、有机的编排布局。

当我们想要弄懂身边的信息海洋时，我们需要制作可视化。这是信息时代。因此，也可以说，这也是一个可视化时代。

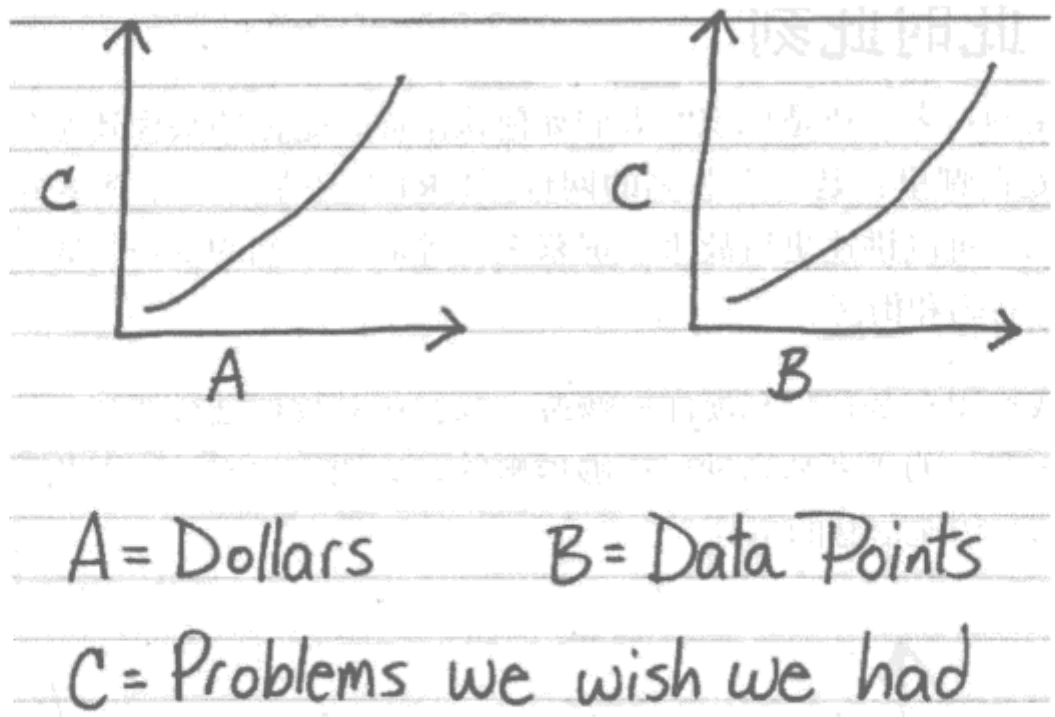


图20-10：使用任何一种方式获取你所需要的

可视化：是编码的

字母代表发音；文字代表思想。我们组合整理句子以讲述故事。你的汽车引擎罩的装饰能够揭秘你的收入水平。皱纹能揭秘你的年龄。我们通过编码来通信——听觉、视觉、触觉和社会性。即使我们的DNA都是一种编码——我们从头开始构建，通过数据的表现位来通信，如图20-11所示。可视化只是另一种编码通信方式，图形的轴线是关联简写，编辑卡通字符代表意识形态。摄影和绘画表示历史。

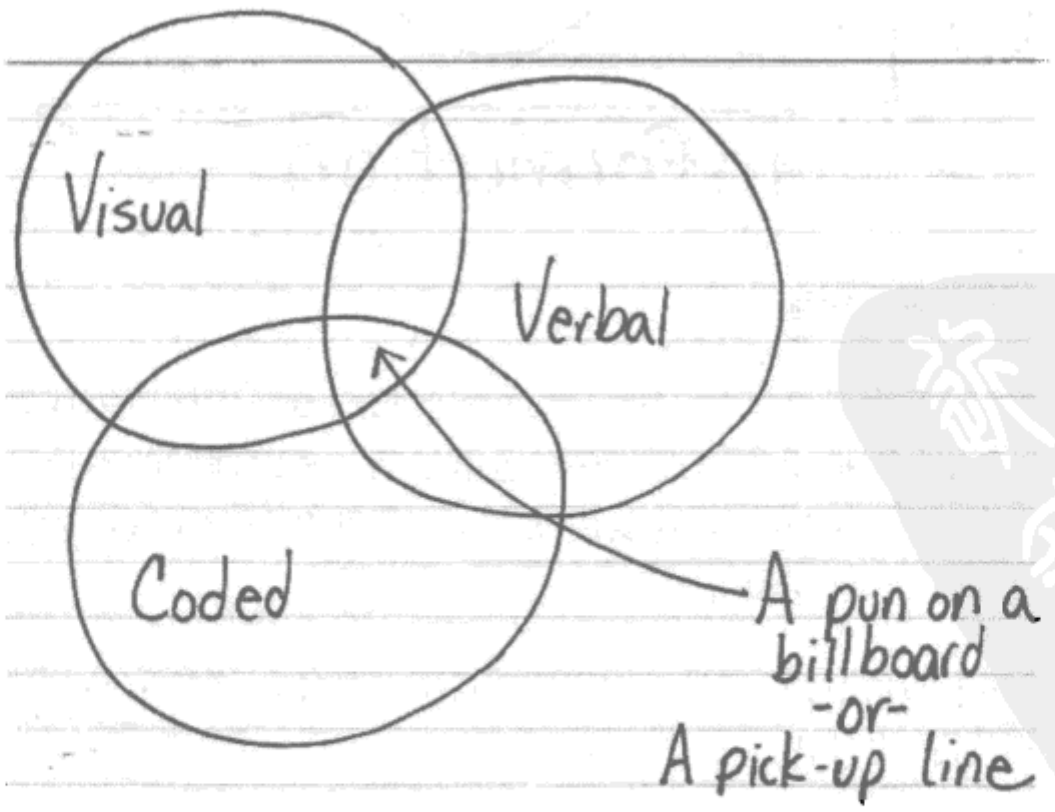


图20-11：眨眼眨眼，轻挪轻挪

由于可视化变成了更大的调查领域——在大学、艺术工作室和留言板——符号学的概念将会被人们更频繁地提及。当我们更仔细地查看标识和符号，我们会发现使用可视化进行的交流和用文字的几乎一样多。我们使用符号来表达自己，从竖起一个手指向闯红灯的司机发出警告、到心理爱情短信、到使用日益陈腐的表情符号。

隐喻、成语、内在的笑话（或者是文学典故，如果你是英语专业）。我们的交流涉及符号的很多层面，每次交谈都需要对很多编码进行翻译。可视化是表现思想的另一种方式；是另一种不包含很多秘密的编码方式。可视化展示越清晰，能够破解该编码的人就越多。

帮派纹身、Rorschach测试[译注1]，包含很多解释的各种艺术作品——这些只是那些包含很多隐藏的（有时深远的）涵义的众多可视化中的几个例子，如图20-12所示。

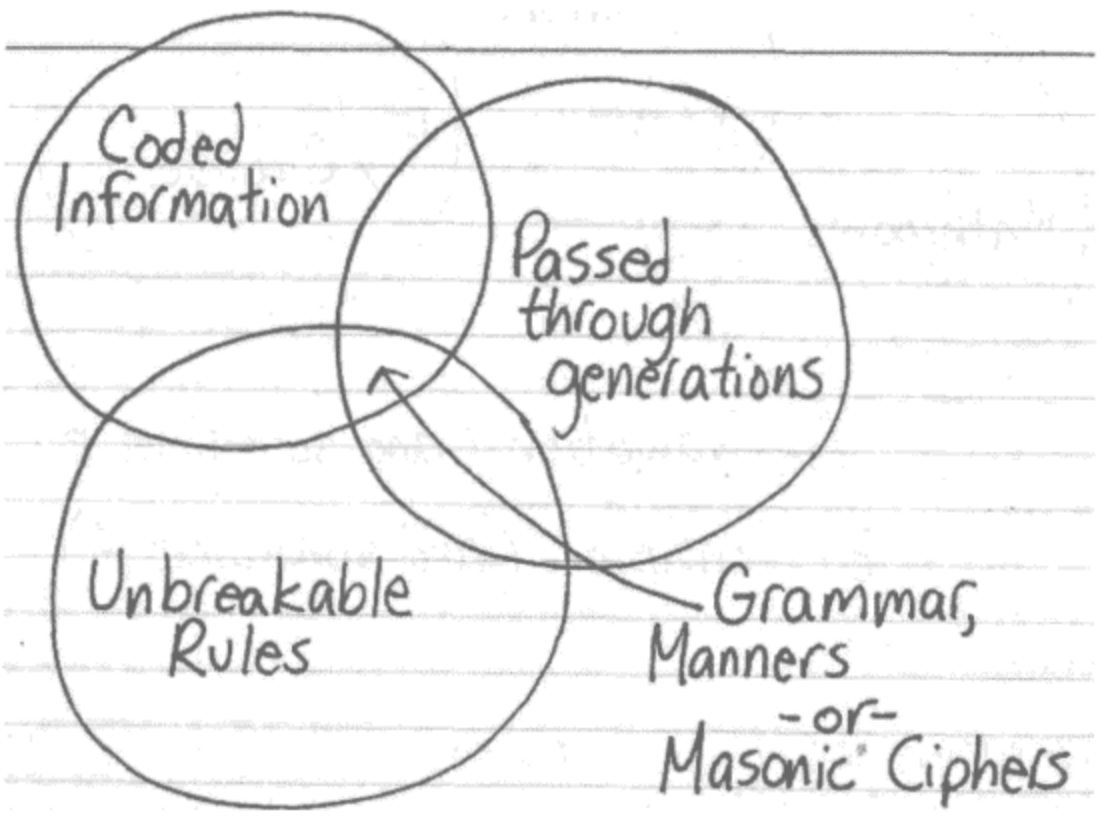


图20-12：秘密和/或社会

可视化：是清晰的

可视化的美丽之一在于其简单性。纯粹的清晰！无比的清晰！超级简单！图像可以为演讲、专题文章和年度报告设置基调。我们期待、查看并理解。从第一眼到“啊哈，我明白啦！”，只是经过了几秒钟的时间。

译注1：Rorschach测试指的是根据人们对墨渍图案的反应来分析其性格的实验。

我们并非总是有时间来剖析涵义或读10页的总结。我们想要查看一个图形，查看一年又一年的结果。图像非常适合快速表达信息。清晰可以使我们能够理解并坚持。模棱两可需要花费时间去琢磨——我们没有那么多时间。

我们见到一个人时，在最开始的10秒内了解到的信息要比花费了几个小时使用百度搜索到的信息还要多得多。我们可以通过封面来评判一本书，通过外观评判一幢房子。看到自由女神像上缠绕着绞索的图片，我们能够推测出存在着不公正的现象。我们看到总统的竞选海报上绘制着魔鬼的角，就知道有人不喜欢他。视觉所传递的信息非常清晰明确，如图20-13所示。但只是由于该信息是显而易见的，可能并不总是真实的。

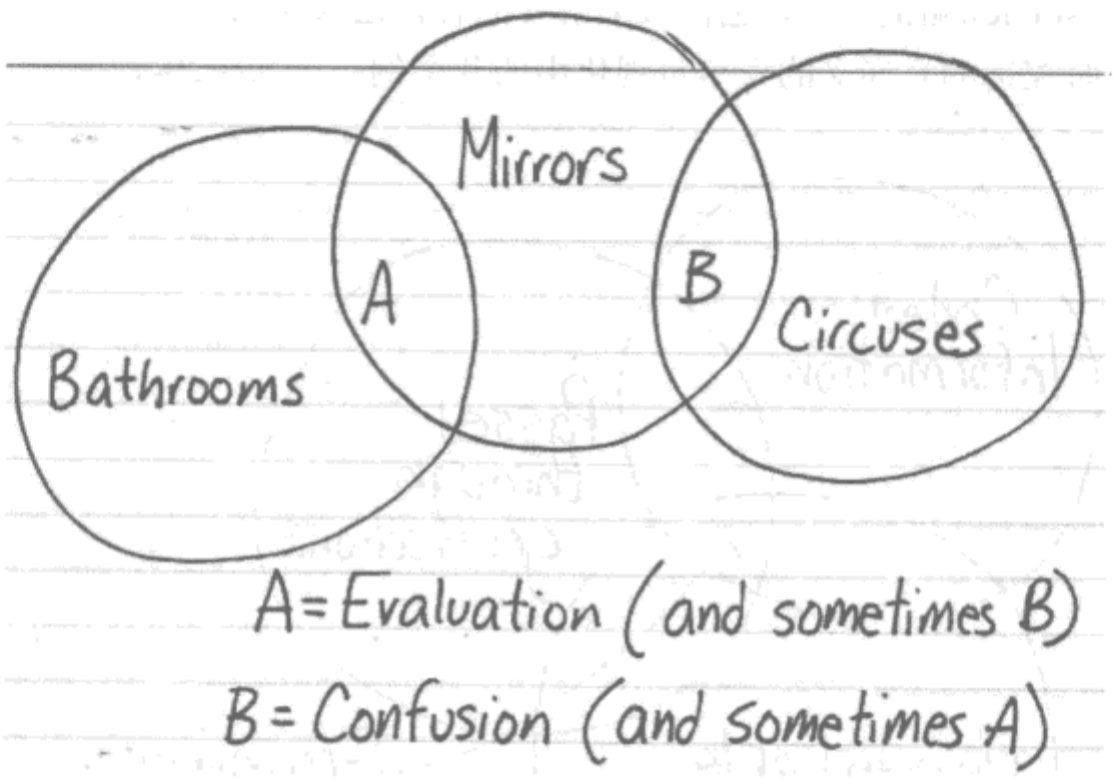


图20-13：都是场景

我们不信任包含偏见的新闻来源。当一个提议听起来很不错，其书面材料往往都很密很长，不利于我们。广告的真实性一直不只是个神话。当凝视一个美丽的可视化时要记住这一点。其信息可能很清晰明显，但是其背后的动机可能需要花费更多的时间来观察（见图20-14）。

可视化：是可学习的

任何形式的信息展示，都是面向所有人的，供所有人创造和消费的。从你的发型到外套颜色，你都在发送视觉信号和传达视觉信息。每个人都可以拿起一支笔，在墙上或纸上画一条线。类似地，像素可以重新布局，来表达任何会用电脑的人的想法。

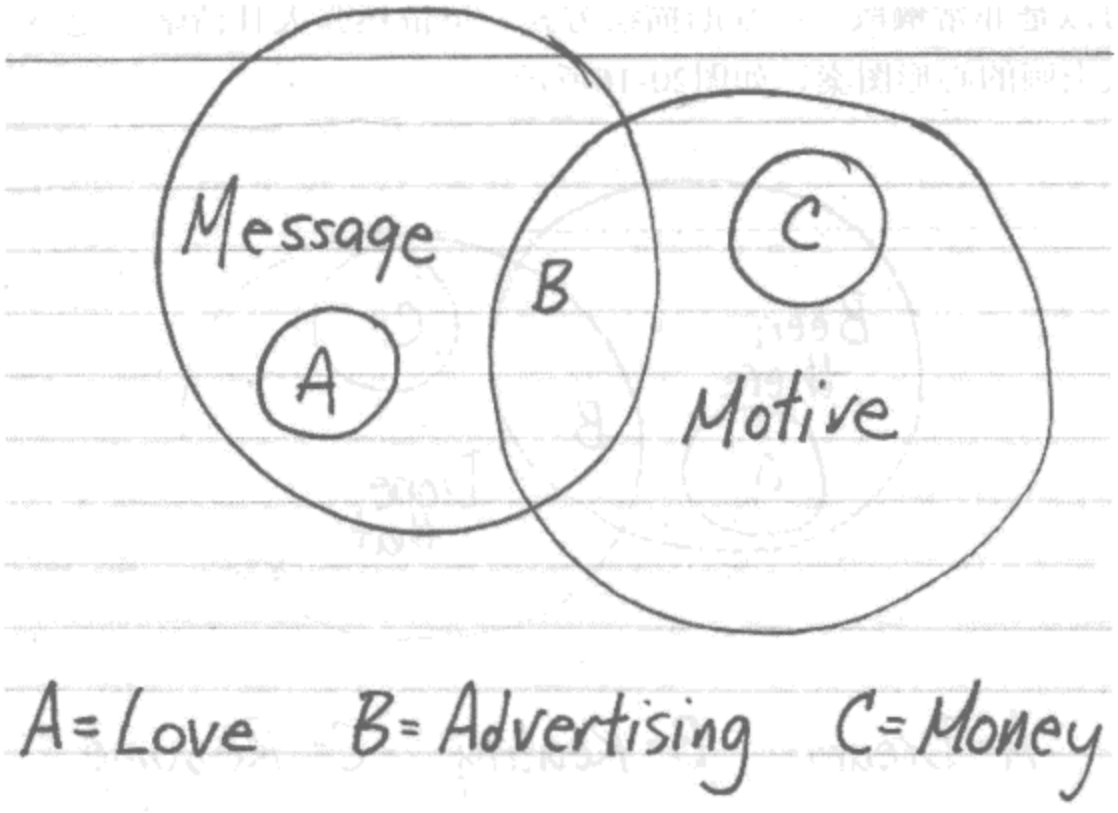


图20-14：问问自己为什么要看眼前的事物

你不必会讲意大利语也可以欣赏米开朗基罗的艺术品。任何参观罗浮宫的人，都深受启发。同样，婴儿一字不识，也可以识别出人脸和表情。

就像通过学习单词来掌握阅读和沟通，通过实践成为专业的视觉沟通者也是可能的。绘画是在纸上把情景翻译出来的能力——属于直接翻译。可视化是把思想表达在纸上——采用数据，并把它提炼成一个概念。不要把这二者混为一谈。思考过程是不同的，即使笔纸能把这两项技能结合起来。思想（概念、理论、等式、意见、过程）和一篮水果的静物画表现不同，如图20-15所示。

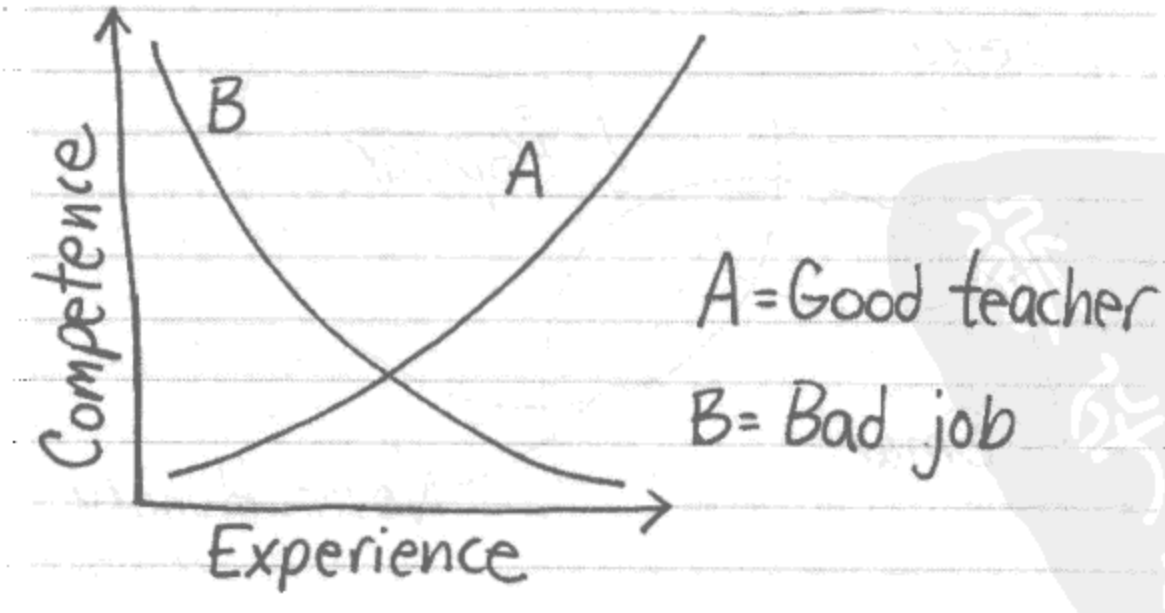


图20-15：知道越多，做得越多

符号和隐喻可以是非常懒散、混乱的描绘方式，但依然强大且清晰。记下下一次你在窗玻璃上的蒸汽上画的心形图案，如图20-16所示。

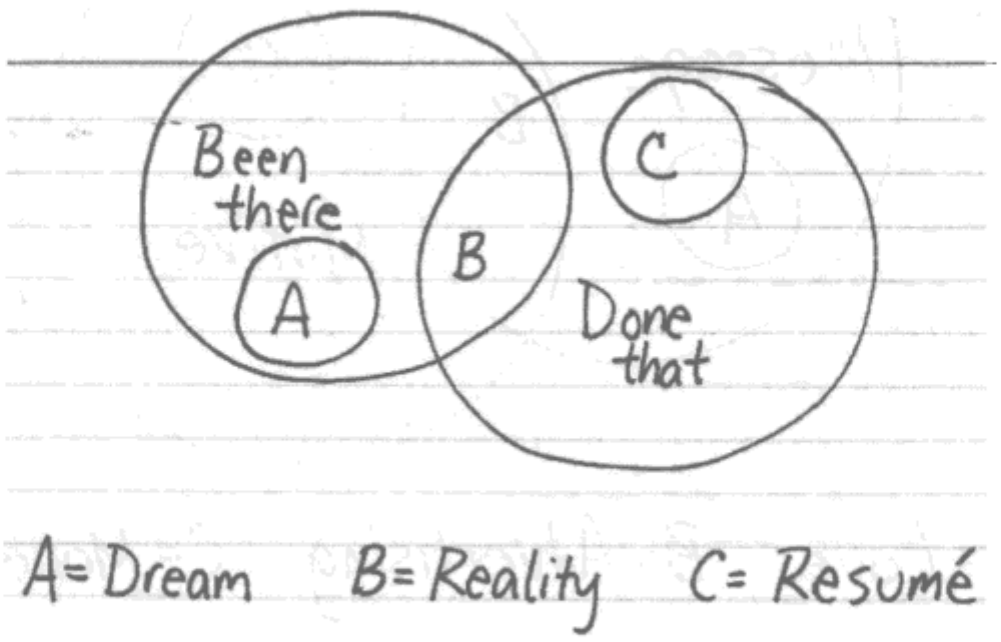


图20-16：你的所作所为决定你是谁

可视化：是一个流行语

因此，它是一个主题吗（见图20-17）？可视化仅仅是席卷商业杂志、招标书和学科教学大纲的最新的流行语吗？它是营销人员为了听起来显得智慧而吹捧的另一个流行语吗？还是它并没有那么潮流，而是对我们当前的数据饱和状况的一个反应？

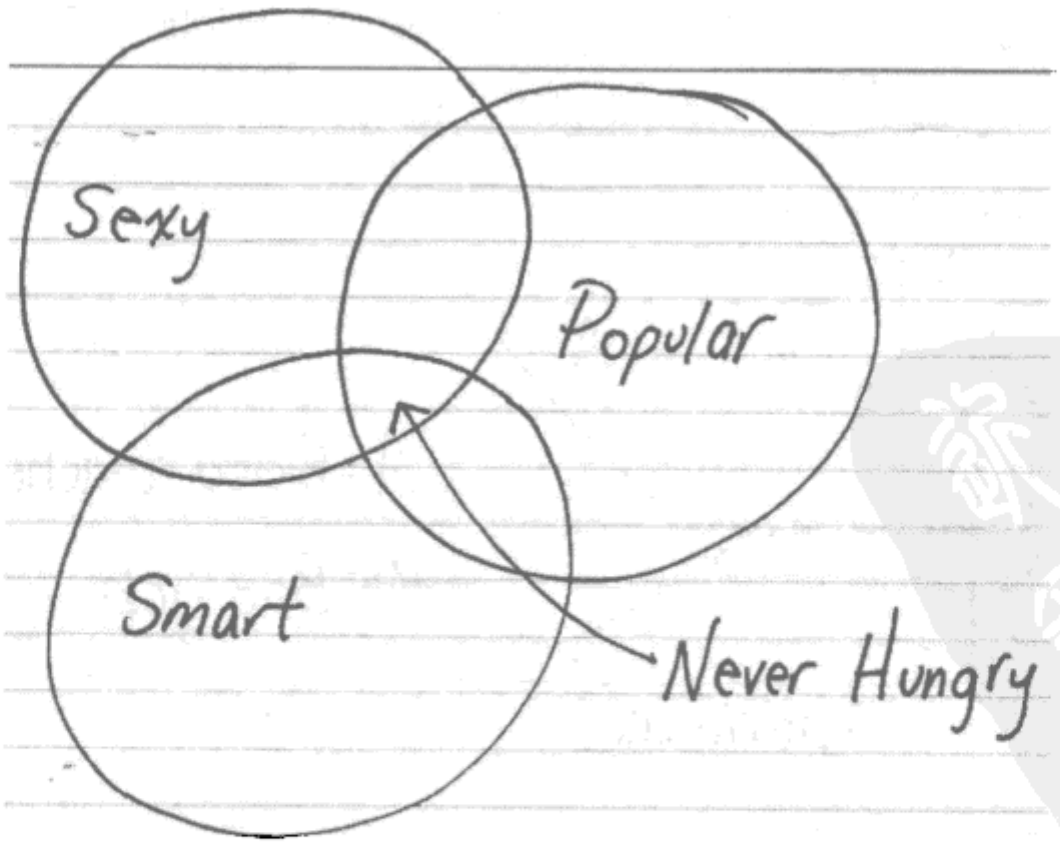


图20-17：欢迎来到因特网

可视化得到了人们的广泛关注：它帮助人们处理信息过载从而节约时间，理解可视化是我们与生俱来的能力。精心设计的可视化引人注目，看起来美丽优雅，让人享受。而且有非常多的可视化软件可以利用，现在是把想法变成图像的最佳时期。因此，看起来可视化的流行是必然的：我们需要筛选的数据越多，就越容易把数据转化成图像，也越容易把图像和文字并列显示，我们越想说服别人提升自己，在我们身边看到的可视化就越多。

可视化这个词本身很受欢迎，其思想很受欢迎，其应用也很受欢迎。可视化帮助我们交流。它能够促使进联系。只要这两个观点是正确的，我们只需要祈祷可视化会向“甲壳虫”一样受欢迎，而不是像“顽童合唱团”（Monkees）一样，如图20-18所示。

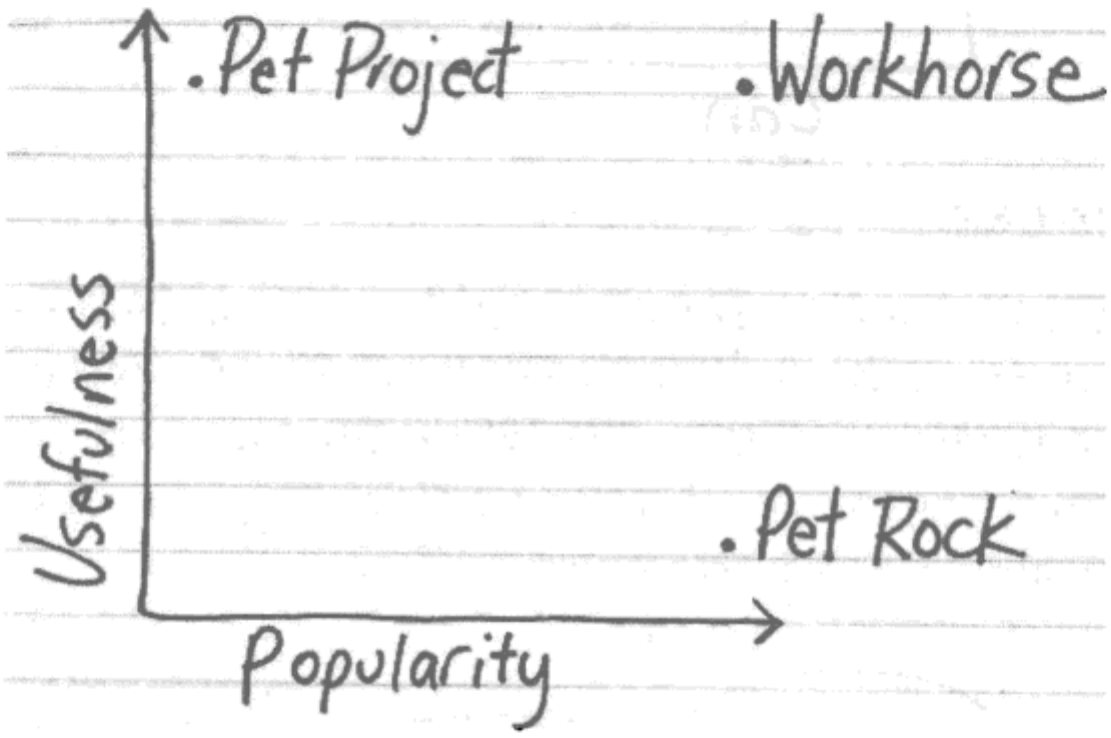


图20-18：你参与的是一场变革还是只是一个时潮？

可视化：是一个机遇

如果你要连接、强迫或交流，你需要使用视觉效果。你可以结合艺术和业务。通过视觉效果，你可以快速、有力且充满感情地和人们交流。即使你并不觉得自己有创意，不认为自己可以像艺术家那样，你依然可以成为可视化制作人员，如图20-19所示。

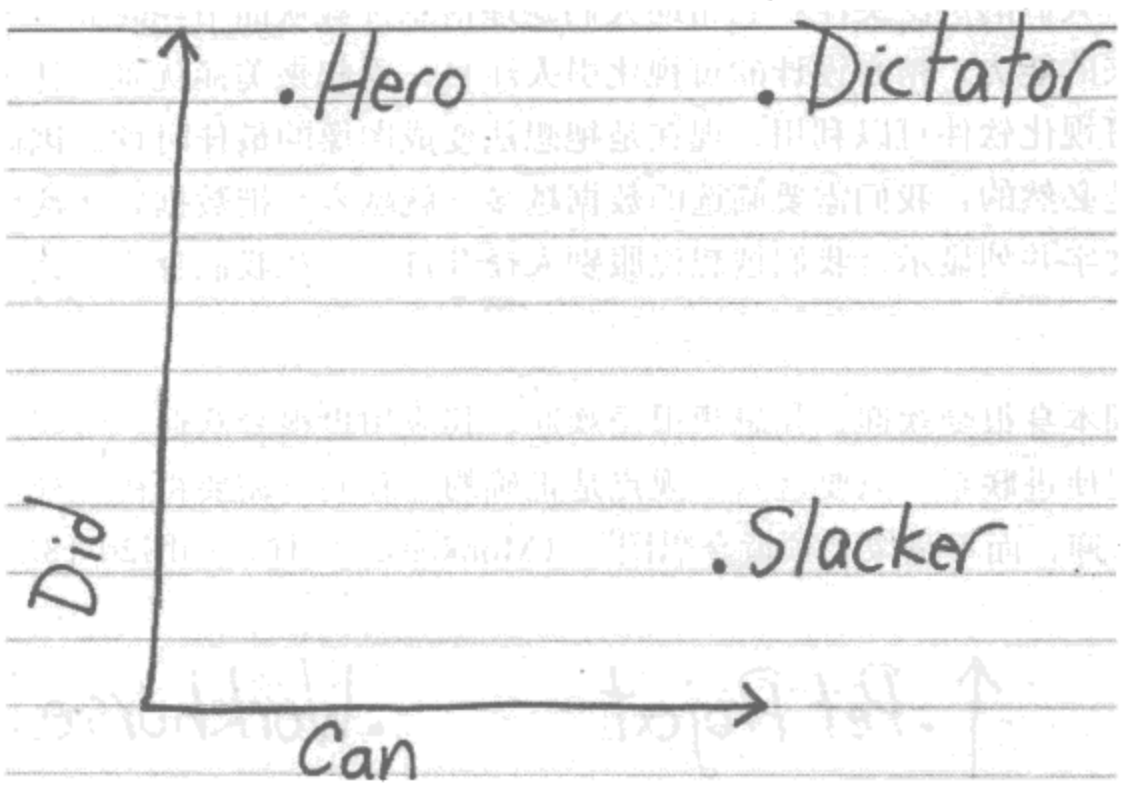


图20-19：借口是无效的

正如作家通过阅读来提升它们的技术，可视化制作人员可以通过观察来增强他们的技能。他们专注地查看，紧紧盯住别人宁可忽略的某些地方。他们不仅仅查看图片，而且观察事件。他们紧紧盯住事件的起因、影响、动机和手段。有时，他们闭上眼睛，思考如何在一个Word文档中说明宇宙，或者如何通过邮件说明自己的感情有多深，或者如何在一个幻灯片里说明自己的业务范畴，如图20-20所示。

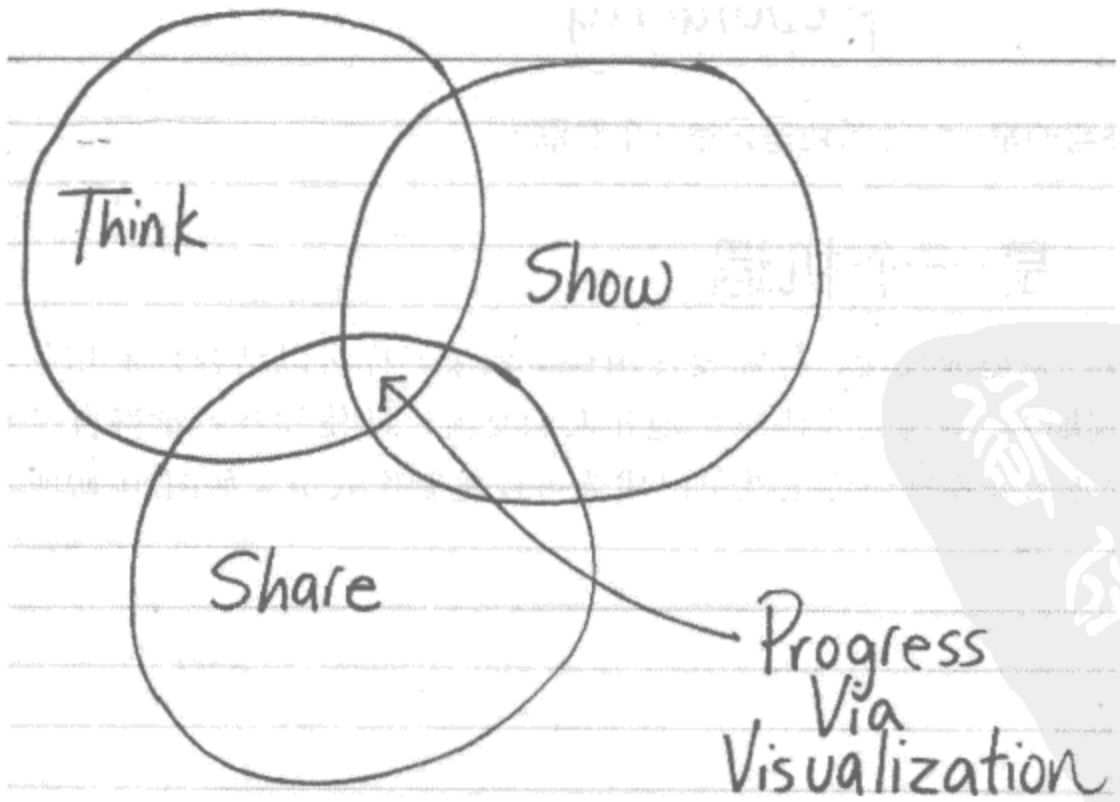


图20-20：看得更贴切、更深远

观察是可视化的第一步，而且此时此刻你正在观察。如果你可以思考它，你就可以对它进行可视化。如果你可以对它进行可视化，你就可以分享它。而如果你可以分享它，你就可以改变这个世界。

但是首先：请环顾一下你的四周。机会正在等着你。

作者简介

Dennis Adderton是一名具有科研仪器设计背景的电气工程师。他目前是加州大学圣巴巴拉分校的AlloSphere项目的研究工程师，并师从JoAnn Kuchera-Morin博士学习媒体艺术。

Basak Alper是加州大学圣巴巴拉分校的媒体艺术和技术项目的博士候选人。她在过去5年致力于计算机图形和可视化。她的研究成果是专注于以人类为中心的虚拟现实环境中的多模式可视化。

Nick Bilton是《纽约时报》的Bits博客的技术作家带头人。他在设计、用户界面、新闻、硬件改装、纪录片和编程上都有背景经验。他之前是《纽约时报》研发实验室的研究人员，在媒体领域探索了10年之久。除了在《纽约时报》工作，Nick还共同成立了NYCResistor，NYCResistor是在纽约布鲁克林的硬件改装空间。他还是纽约大学交互远程通信计划的兼职教授。

Michael Driscoll作为10多年前的Human Genome Project项目中的一名软件工程师，就爱上了数据可视化。他是Dataspora的创始人兼首席数据科学家，在旧金山作为分析顾问。

Jonathan Feinberg是一名计算机程序员，他和妻子以及两个儿子住在麻省的Medford。你可以给他mail：*jdf@pobox.com*，尤其是如果你知道在布鲁克林的Greenpoint的波士顿地区有任何泰国餐厅。

Danyel Fisher是微软研究员的VIBE（可视化和交互）团队的一名研究员。他的研究兴趣主要是信息可视化和在线写作，以及如何联合使用可视化。Danyel在2004年从加州大学欧文分校获得博士学位。他过去的研究主要是反映社会计算活动，可视化电子邮件信息和通信，并通过地图和地理软件进行着色。他是图形绘画软件包JUNG的作者之一；你可以通过以下网址访问其当前的项目：*http://research.microsoft.com/~danyelf*。

Jessica Hagy是一名作家、演讲家和顾问。对于需要表达清晰的公司，她可以把模糊、复杂的思想提炼成“鲜美的”视觉“调味酱”。她是著名的网站thisisindexed.com的作家，她的作品在《纽约时报》、BBC杂志在线、《Paste》、《高尔夫文摘》、《红皮书》、《纽约杂志》、加拿大《国家邮报》、《卫报》、《时代周刊》以及很多其他新老媒体上刊出。

Todd Holloway对于信息可视化、信息检索、机器学习、数据挖掘、网络科学以及人工智能非常痴迷。他曾就读过Grinnell College大学和印第安纳大学。

Noah Iliinsky花了近几年的时间在思考创建信息可视化图表和其他类型的有效方式。他同时在设计界面和交互，都是从功能和以用户为中心的角度。在成为设计师之前，他做了几年的程序员。他在华盛顿大学获得通信技术硕士学位，从Reed学院获得学士学位。他的博客是*http://ComplexDiagrams.com*。

Eddie Jabbour是纽约城市的KICK设计公司的创始人和创意总监。在过去20年，KICK设计公司已经和世界上最知名的品牌共同通过视觉冲击创造欣喜和创新。

Haru Ji是一个雕塑家、跨行业的艺术家和研究人员，通过人工生命创造世界作为计算艺术来探索艺术生活的主题。目前，她是加州大学圣巴巴拉分校的博士候选人，是加州纳米系统研究院的AlloSphere项目组的研究员。她在世界各地的展览会和艺术节展示了计算设备、数字雕塑、虚拟建筑、视频设备、雕塑物体和三维动画，包括ISEA、EvoWorkshops、SIGGRAPH以及相应的出版物。她还完成了一半的协作研究项目和虚拟生态系统“人工自然”，探索扩大媒体艺术走向世界的艺术，网址是：*http://haru.name*。

Valdean Klump居住在加州的旧金山，是Google创意实验室的作家。

Aaron Koblin是加州旧金山的艺术家，他由于数据可视化项目而闻名，如“Sheep Market”（绵羊市场）、“Ten Thousand Cents”（一万个硬币）和Radiohead的“House of Cards”音乐视频。目前，他是Google创意实验室的技术带头人。

Robert Kosara是夏洛特、北卡罗来纳州的计算机科学的助理教授。他的研究兴趣包括分类数据可视化、可视化数据通信和可视化的理论基础。Robert的网站是：*http://EagerEyes.org*。

Valdis Krebs是俄亥俄州克里夫兰市Orgnet.com网站的首席科学家。Orgnet.com网站为公司组织、社区提供社交网络分析软件和服务，并提供咨询。

JoAnn Kuchera-Morin博士是一名作曲家，她是加州大学圣巴巴拉分校的媒体艺术&技术专业以及音乐专业的教授，研究多模式媒体系统、内容和配套设施的设计。她超过25年的数字媒体领域带头人经验，在加州大学圣巴巴拉分校创立、设计和开发了一个数字媒体中心，她目前的最佳设计是Allosphere研究实验室，把三层的金属球体置于无回声的工作室内，其设计目的是为了对多维数据集进行沉浸式、交互式的科学和艺术探索。JoAnn Kuchera-Morin博士是Allosphere研究中心主任。

Andrew Odewahn是O'Reilly媒体的商务发展部门主任，他帮助那些热衷于新领域的观众技术爱好者参与公司活动。他发表了两本关于数据库开发的著作，是*tagcaster.com*的创始人，纽约大学的斯特恩商学院的研究生，是Appalachian Trail的徒步旅行者。

Anders Persson博士是瑞典Linkoping大学的医学图像科学和可视化中心的副教授和主任（CMIV；*http://www.cmiv.liu.se*）。该中心专注于多学科项目内的前线研究，为今后的临床问题提供解决方案。其任务是为卫生健康和医学研究应用的图像分析和可视化制定方法和工具。

Adam Perer博士是以色列海法（Haifa）IBM研究院的研究科学家。他的研究兴趣包括设计新的可视化技术来帮助人们理解复杂数据。可以通过其网站访问更多信息：*http://perer.org/*。

Lance Putnam是一位作曲家和研究人员，调查计算机生成的声音和图像环境下的频率和空间的关系。他目前是加州大学圣巴巴拉分校（UCSB）媒体艺术和技术专业的博士候选人。他拥有麦迪逊威斯康星州大学的电子和计算工程专业的学士学位，以及UCSB的电子音乐和声音设计的硕士学位。他被选为8个国际学生之一，在纽约IBM T. J. Watson研究中心的2007年新兴多媒体会议展示其在媒体信号处理方面的研究。他的工作成果“S Phase”曾在北爱尔兰贝尔法斯特的2008年国际计算机音乐会议和2009年意大利Parma的Traiettorie节日上展示。

Maximilian Schich是一名DFG的艺术历史学家，作为BarabásiLab实验室的访问研究科学家——在波士顿东北大学的复杂网络研究中心，他和网络科学家协作，研究艺术历史和考古学的复杂网络。Maximilian在2007年获得博士学位，有10多年的顾问咨询经验，致力于艺术研究的网络数据，作为项目合作者、用户、程序员和客户四者之间的经纪人。

他花了几年的时间致力于Projekt Dyabola项目、Bibliotheca Hertziana（艺术历史Max-Planck研究所）、Munich Glyptothek和Zentralinstitut für Kunstgeschichte。可以通过以下网址查到更多：*http://www.schich.info*。

Matthias Shapiro是一名软件设计师，并且是基于犹他州盐城的信息可视化爱好者。他通过Silverlight创建了绝大部分的可视化，并兼职作为信息可视化的独立的传播者，向参议员、CNN主持人、微软会议参与人以及任何“不够智慧逃避其发言的人”来说明可视化的重要性。

Julie Steele是O'Reilly媒体的一名编辑，她对于把人们和思想连接起来感兴趣。她从发现新的方式来理解复杂系统中找到美丽，并且对于和组织、存储和可视化数据方面相关的主题感兴趣。她在罗格斯大学获得政治科学学位，并正在为O'Reilly开发Gov 2.0内容，由于该空间继续增长。Julie还致力于Python、PHP和SQL相关的主题工作，而且是纽约尚未学习Python小组的创始人。

Moritz Stefaner是介于信息可视化和设计之间的研究人员和自由职业者。他的主要兴趣是信息可视化和数据挖掘如何帮助我们组织和发现信息。他在认知科学和界面设计上都获得学位。他的作品曾在SIGGRAPH和电子艺术节上展览。最近，他被提名为德国2010年联邦共和国设计奖。可以在*http://moritz.stefaner.eu*和*http://well-formed-data.net*得到更多信息。

Jer Thorp是来自加拿大温哥华的艺术家和教育家。作为前遗传学家，其数字艺术实践探索了在科学和艺术之间的多个方面。最近，他的作品体现了《纽约时报》、《卫报》、加拿大广播公司的特征。Thorp的基于软件的获奖作品曾在欧洲、亚洲、北美、南非、澳大利亚和整个Web上展览。Jer是有线英国电台的特约编辑。

Fernanda Viégas和**Martin Wattenberg**是Flowing媒体公司的创始人，Flowing媒体公司是麻省剑桥的可视化设计视频。他们在2003年决定对维基百科进行可视化时组成一个团队，生成第11章中所描述的历史流项目。在成立Flowing媒体公司之前，他们是IBM的视觉通信实验室的带头人，他们在该实验室探索可视化作为多媒体的强大，以及其促使了数据分析的社会形势。

Viégas因其在描述聊天历史和邮件上所做的开创新工作而著名。Wattenberg对股票市场和婴儿名字的可视化被认为是互联网的经典。Viégas 和 Wattenberg还由于其基于可视化的艺术作品而著名，其作品曾经在纽约的当代艺术展览馆大道、当代艺术伦敦研究所和美国艺术的Whitney展览馆展出。

Graham Wakefield通过从生物系统和由生物哲学启发的灵感来探索计算艺术的开放自主权。他是加州大学圣巴巴拉分校的媒体艺术和技术的博士候选人，并且从伦敦大

学Goldsmiths学院获得音乐作曲学位，从Warwick大学获取本科学位。除了作为CNSI AlloSphere的一名研究人员（AlloBrain，Cosm，LuaAV），他还是自行车‘74比赛（Max/MSP/Jitter）的软件开发人员，并且是南加州建筑学院（SCI-Arc）的一名讲师。他的作品和发表的文章在国际会议上展出和演示，如SIGGRAPH、ICMC、ISEA。

Martin Wattenberg和**Fernanda Viégas**是Flowing媒体公司的创始人，如上文所述。

Michael Young是《纽约时报》公司的研发组的一位富于创意的技术工程师。他带领了一个较小的技术团队，设计和探索在多平台和设备上的内容消费特征。其更多信息可以通过*http://81nassau.com*访问。

[General Information]
书名=数据可视化之美　　通过专家的眼光洞察数据
作者=（美）斯蒂尔等编
页数=357
出版社=机械工业出版社
出版日期=2011
SS号=12811942
DX号=
URL=http://book1.duxiu.com/bookDetail.jsp?dxNumber=&d=20401835 0E153ABB9B73FEBC82B56715