

第7章 时变数据可视化

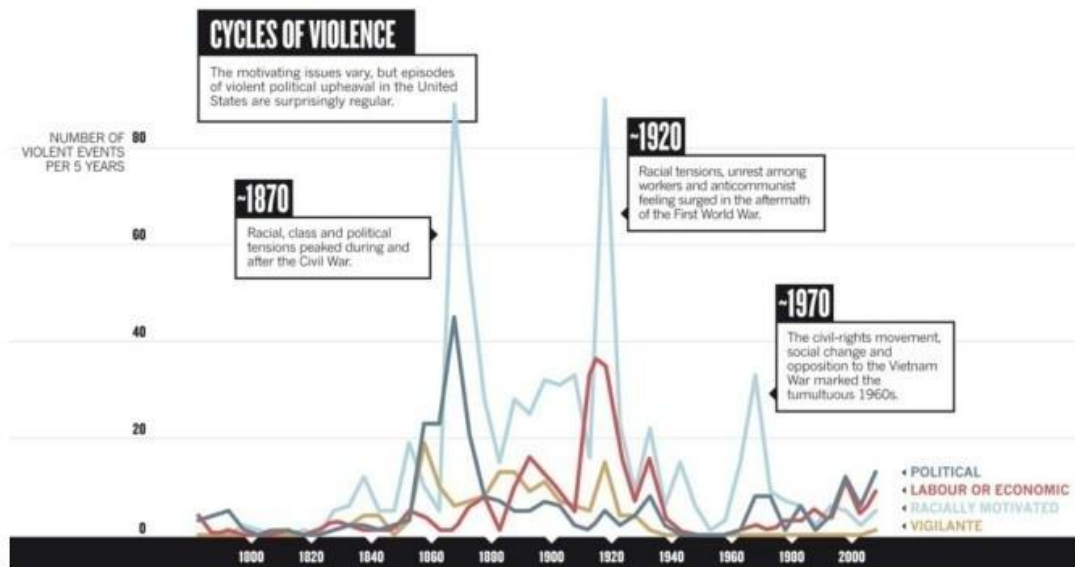
时变型数据

- 随时间变化、带有时间属性的数据称为时变型数据(Time-varying Data)或者 (Temporal Data)。时变型数据的处理方法与顺序型数据(Sequential Data)有相通之处
 - 以时间轴排列的时间序列数据(Time-series Data; Time-varying Data)
 - 具有内在的排列顺序的顺序型数据集，如文本、生物DNA测序
- 在各类传感器网络、移动互联网应用中，以流模式(Streaming)生成的流数据(Streaming Data)是一类特殊的具有无限长度时间轴的时变型数据

时变型数据

➤ 人类社会的微观活动和历史事件也构成了一个随时间变化的数据。

➤ 历史动力学(Cliodynamics)

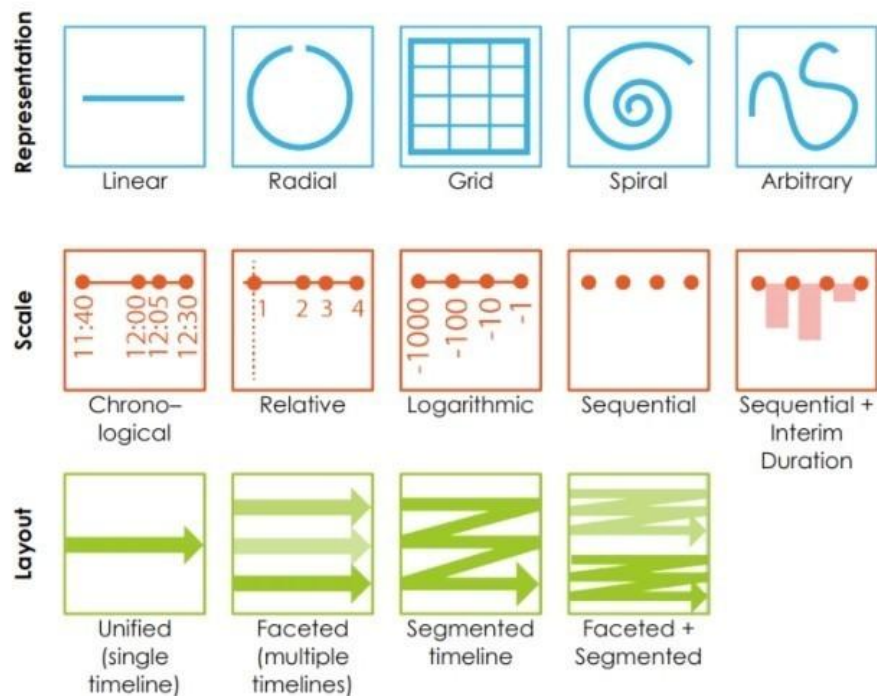


时变型数据的可视化设计

- 分析和理解时变型数据通常可以通过统计、数值计算和数据分析的方法完成。
 - 数据的相似性；数据匹配性；检索；频率和概率；子片段的检测
- 从特征计算的角度看，也可采用传统的数据挖掘方法对时变型数据进行信号分解、模式挖掘和特征预测等处理。针对大尺度的时变型数据，自动的数据挖掘方法往往难以预测

可视化设计空间

► 时变型数据的可视化设计空间涉及三个维度，即表达、比例尺和布局



可视化方法分类

➤ 时变型数据的可视化方法可分为两类

- 一类方法采用静态方式展示数据中记录的内容，不随时间变化，但可采用多视角、数据比较等方法体现数据随时间变化的趋势和规律 [Aigner2008]
- 另一类方法采用动画手法，动态地展示随着时间变化的感觉和过程，因而具有更多的表现空间

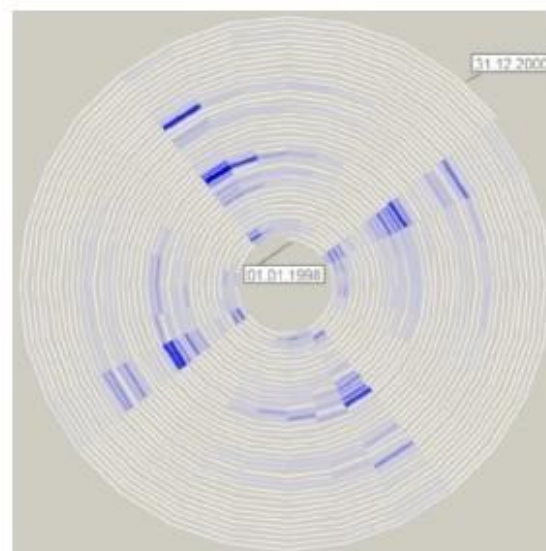
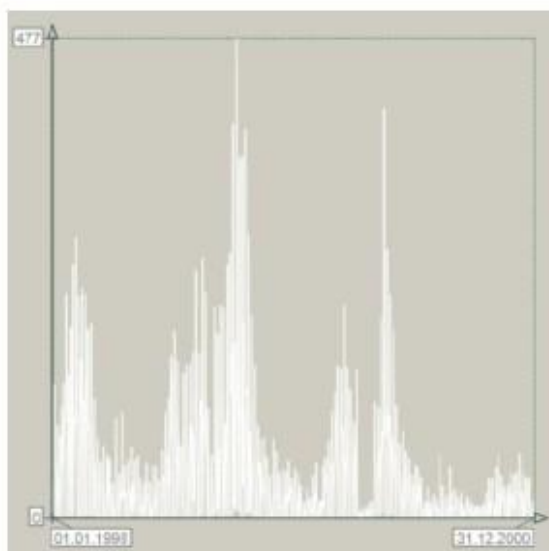
1 时间属性的可视化

时间属性

- 将时间属性或顺序性当成时间轴变量，每个数据实例是轴上某个变量值对应的单个事件。对时间属性的刻画有三种方式
 - 线性时间和周期时间：线性时间假定一个出发点并定义从过去到将来数据元素的线性时域。许多自然界的进程具有循环规律，如季节的循环
 - 时间点和时间间隔：离散时间点将时间描述为可与离散的空间欧拉点相对等的抽象概念。单个时间点没有持续的概念。
 - 顺序时间、分支时间和多角度时间：顺序时间域考虑那些按先后发生的事情。对于分支时间、多股时间分支展开，这有利于描述和比较有选择性的方案

线性和周期时间可视化

- 标准的显示方法将时间数据作为二维的线图显示，x轴表示时间，y轴表示其他变量

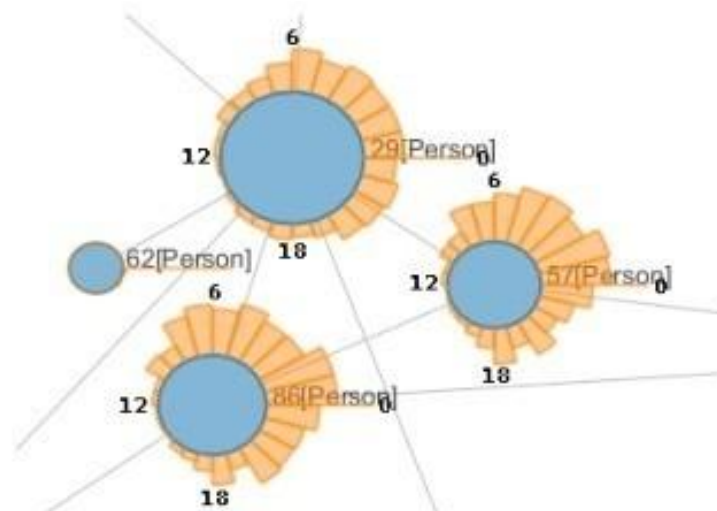
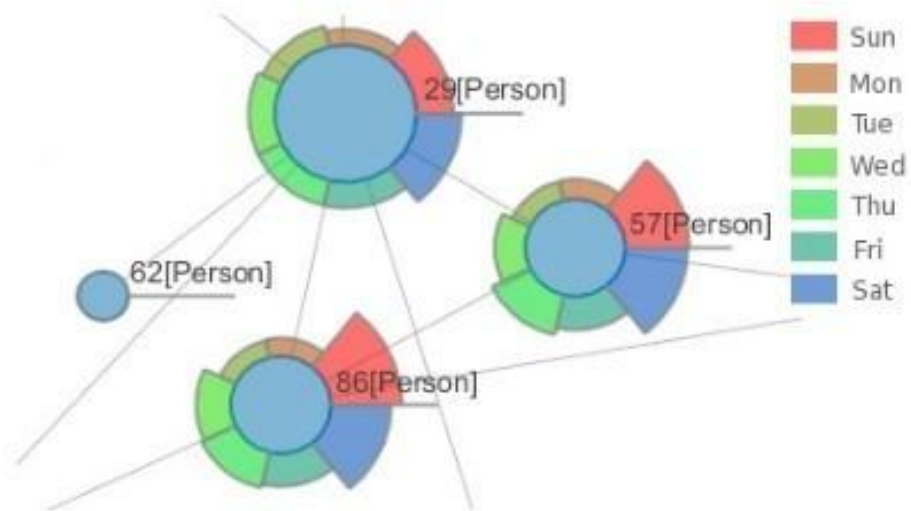


声音时变数据的径向布局可视化

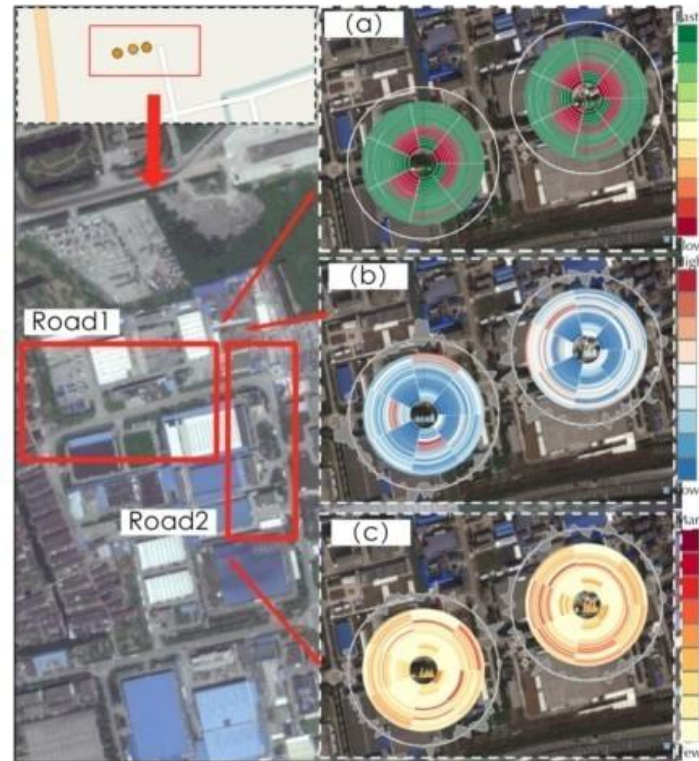
➤ 不同显示周期下的时变序列数据的可视化效果

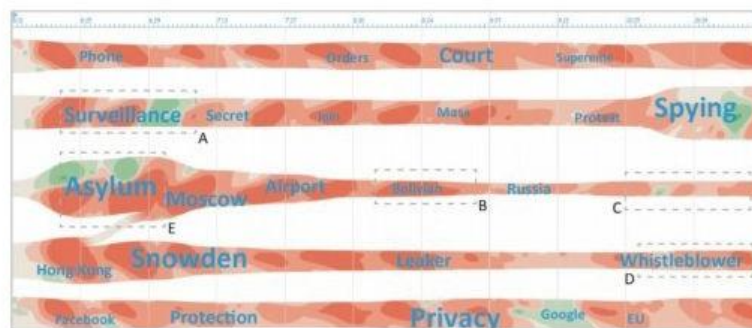
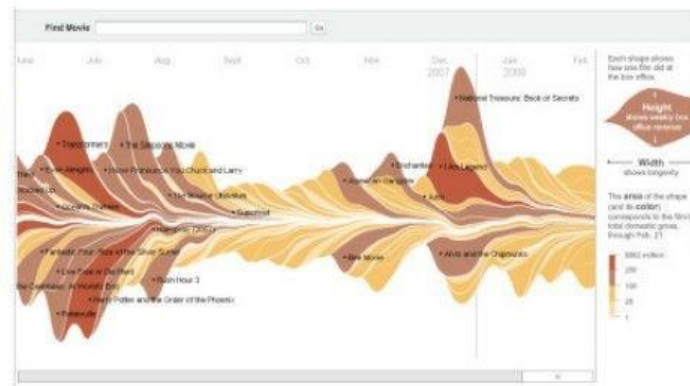


手机用户活动的时间分布



交通流量



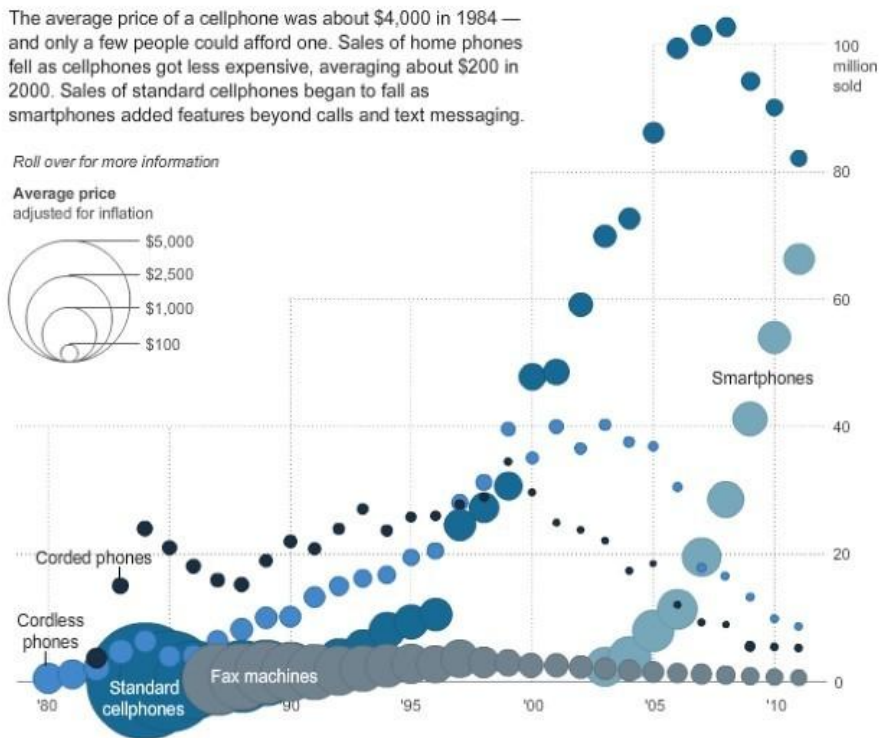


采用不同的可视化通道表达

The average price of a cellphone was about \$4,000 in 1984 — and only a few people could afford one. Sales of home phones fell as cellphones got less expensive, averaging about \$200 in 2000. Sales of standard cellphones began to fall as smartphones added features beyond calls and text messaging.

Roll over for more information

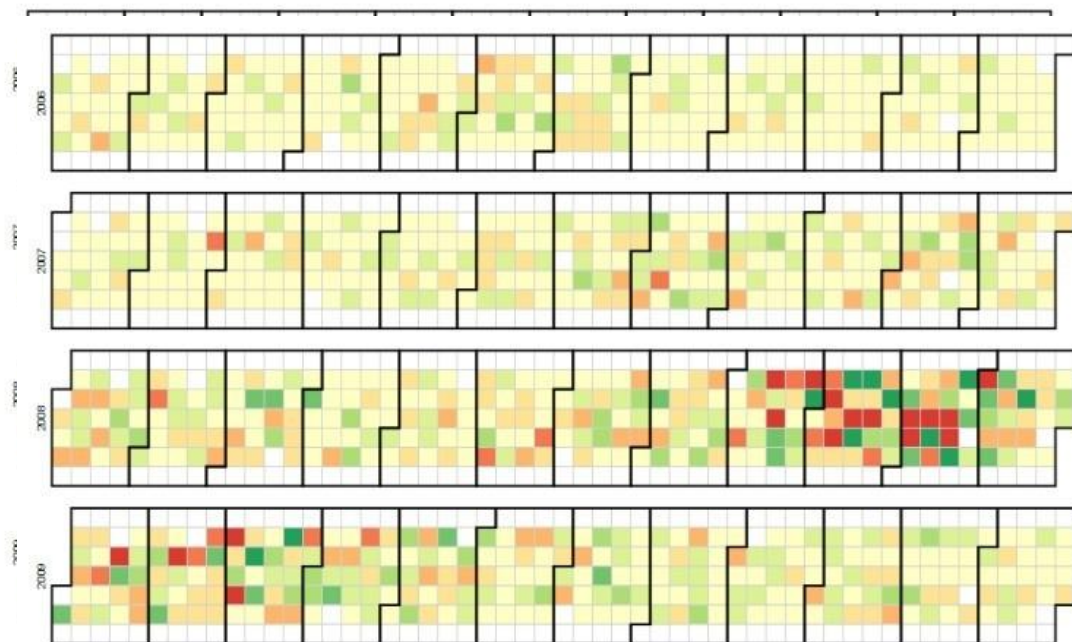
Average price
adjusted for inflation



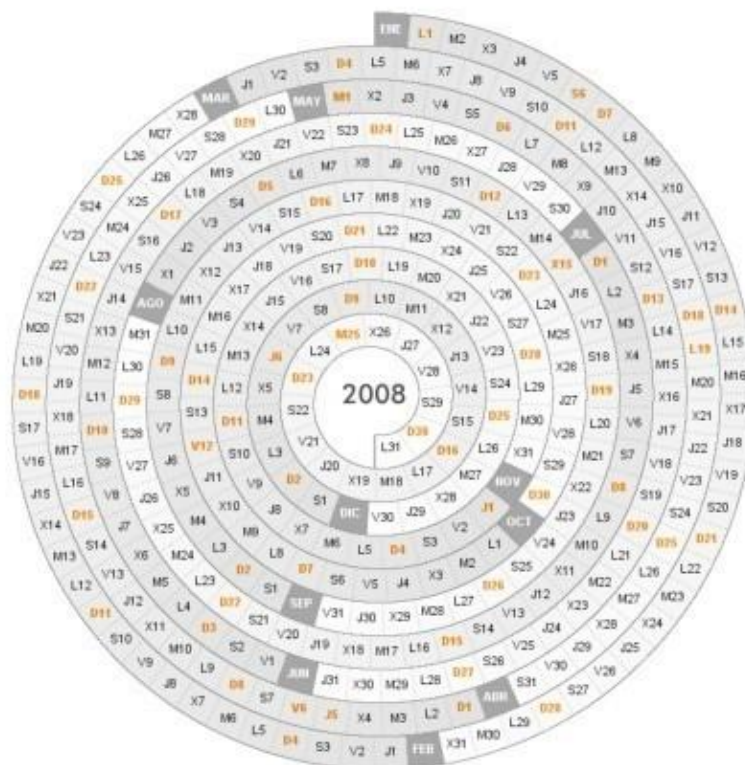
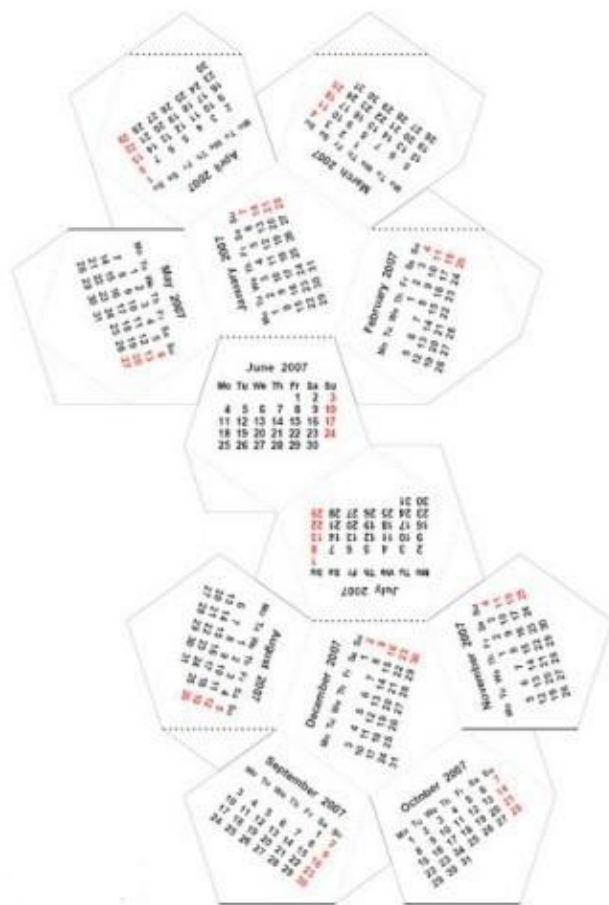
NOTE: 2010 data are estimates and 2011 data are projections. GRAPHIC: Alicia Parlapiano / The Washington Post - January 10, 2011.

日历时间可视化

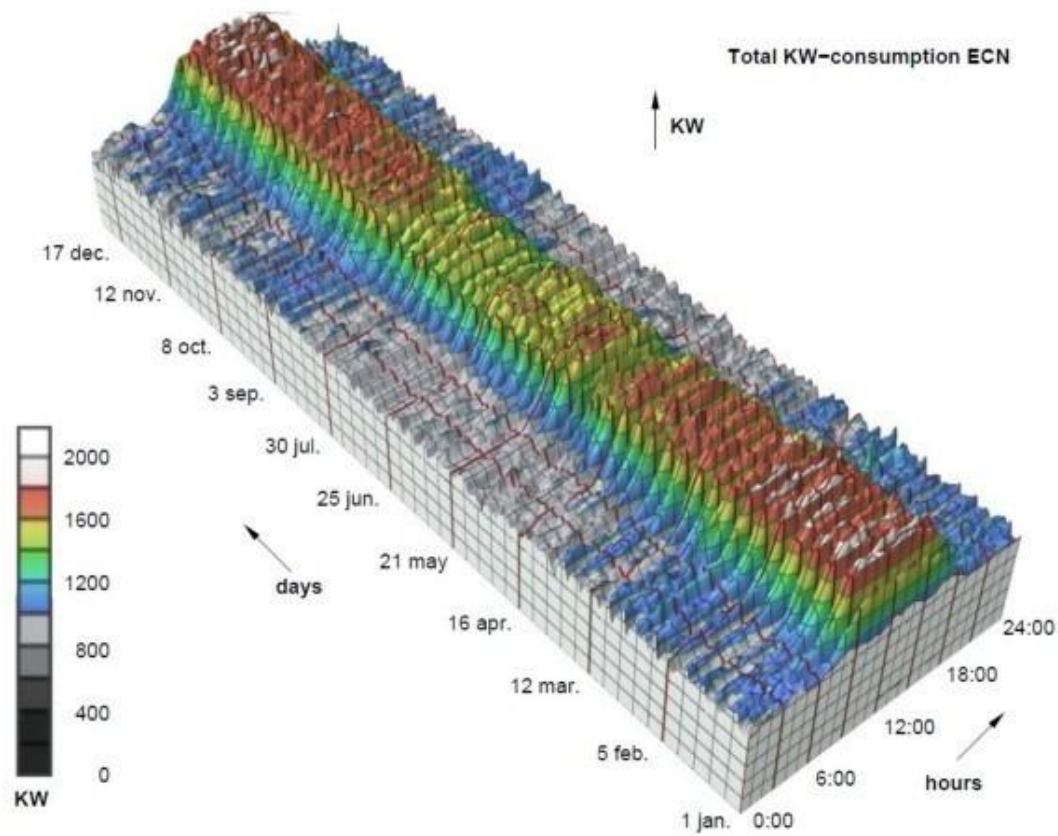
- 对于日历时间的可视化，在表达维度上一般采用表格映射的方式对时间轴进行处理



日历时间可视化

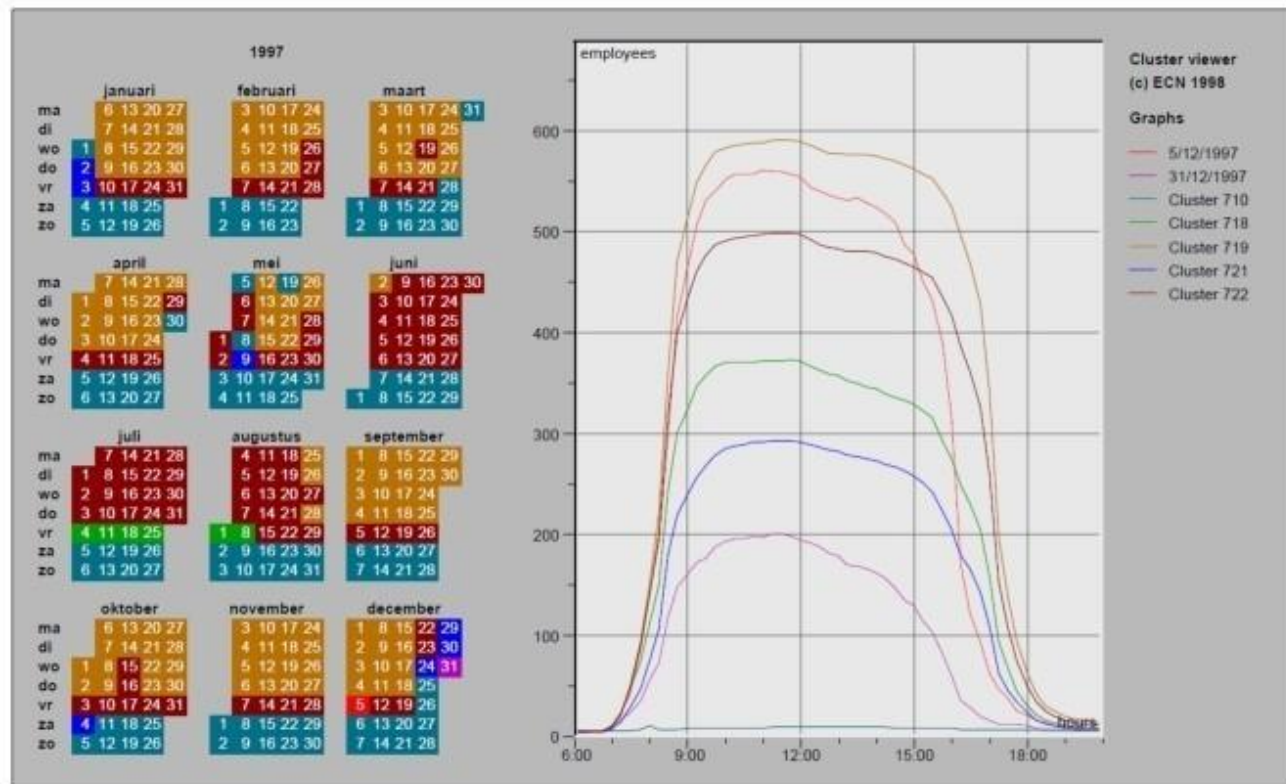


日历时间可视化



日历时间可视化

► 对偶视图

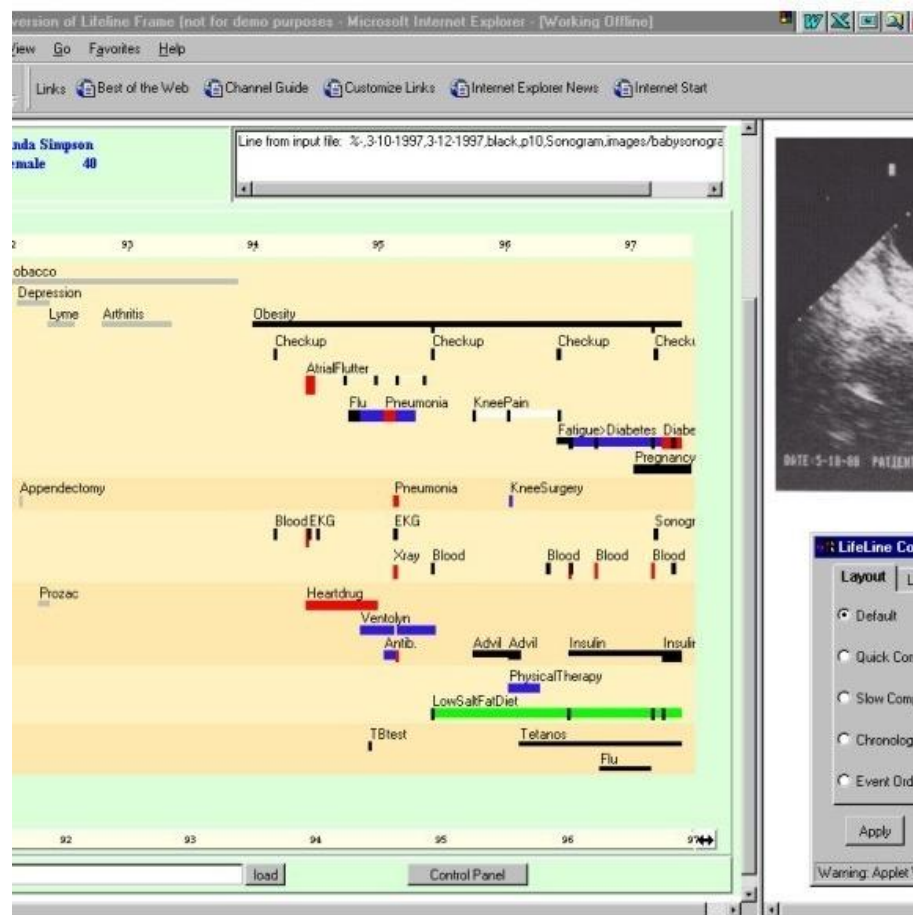


分支和多角度时间可视化

- 时变型数据中蕴含的信息存在分支结构，对同一个事件也可能存在多个角度的刻画
- 按照时间组织结构，可分为线性、流状、树状、图状等类型
- 一般在表达维度上采用线性映射的方式对时间轴进行处理

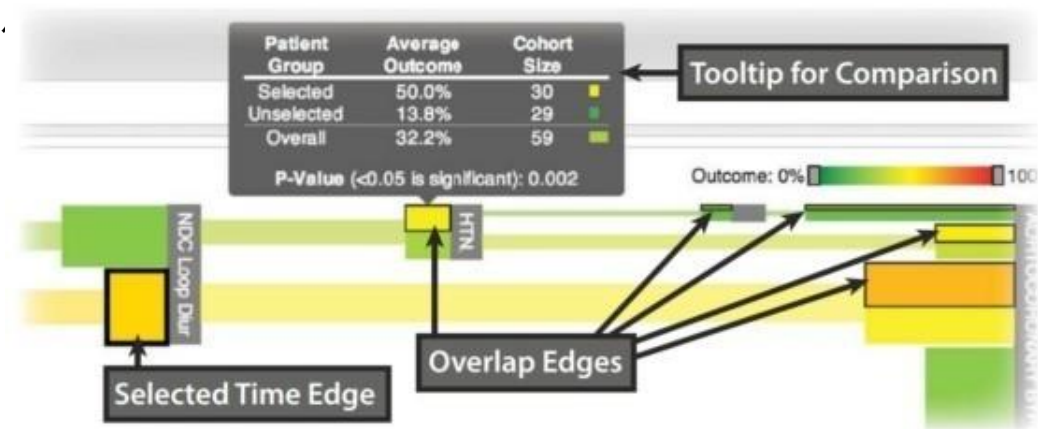
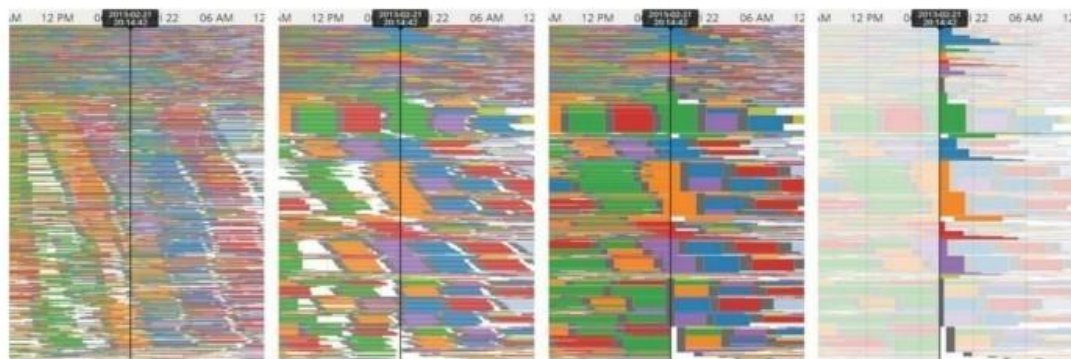
线性多角度时间可视化

- 为了呈现一个完整的事件历程和社会行为(如个人健康记录、历史事件),可采用类似于甘特图(用条形图表进度的可视化标志方法)的方式
- 使用多个条形图线程表现事件的不同属性随时间变化的过程,线条的颜色和厚度都可以编码不同的变量
- 观察者既可以交互地点击某个线程获取详细的细节,也可以直观地得到按时间排列的事件的概括

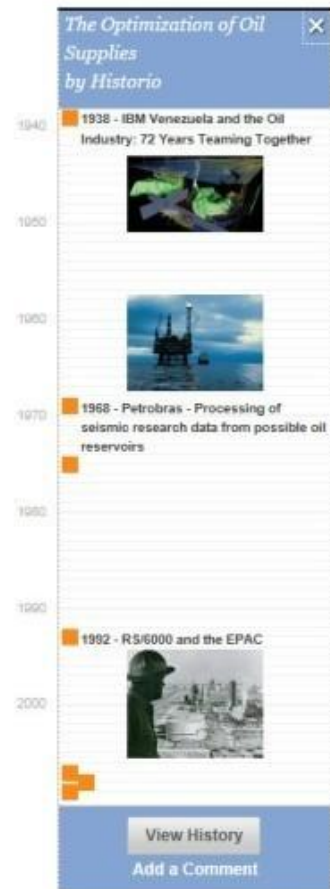
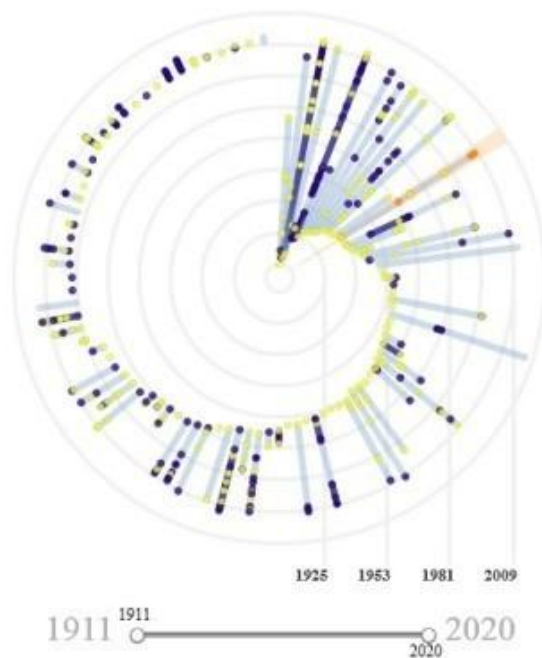


线性多角度时间可视化

- 为了呈现一个完整的事件历程和社会行为(如个人健康记录、历史事件),可采用类似于时间轴(用条形图表进度的可视化标志方法)的方法
- 使用多个条形图线程表现事件的不同属性间变化的过程,线条的颜色和厚度都可以是不同的变量
- 观察者既可以交互地点击某个线程获取详细信息,也可以直观地得到按时间排列的事件概括

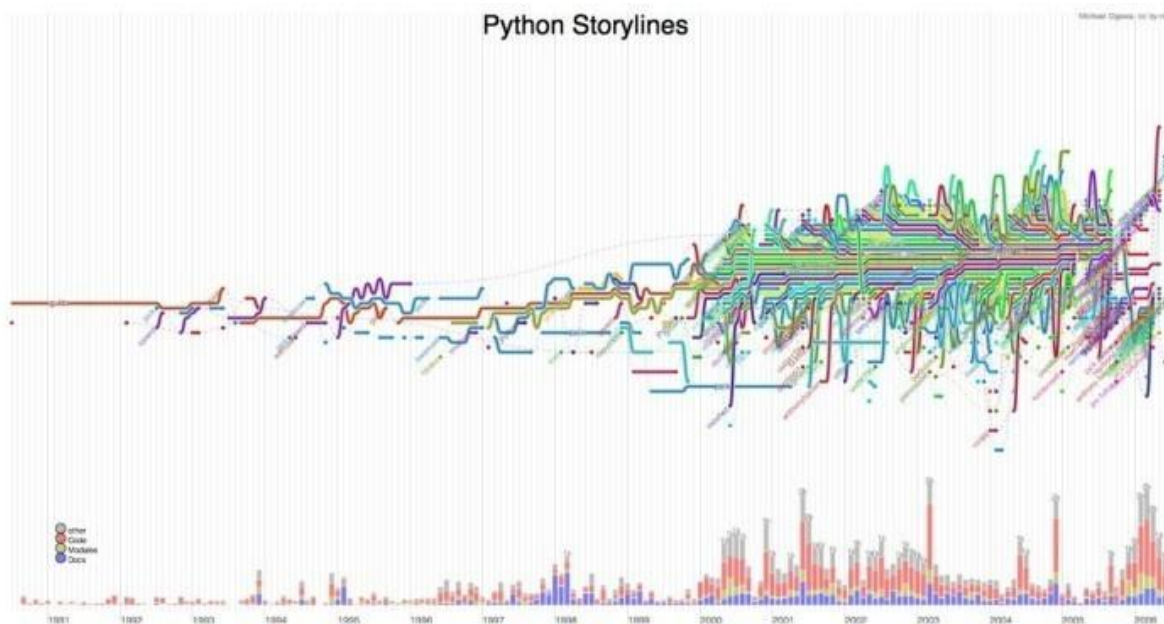


Historio

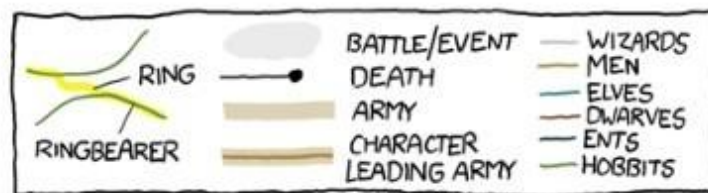
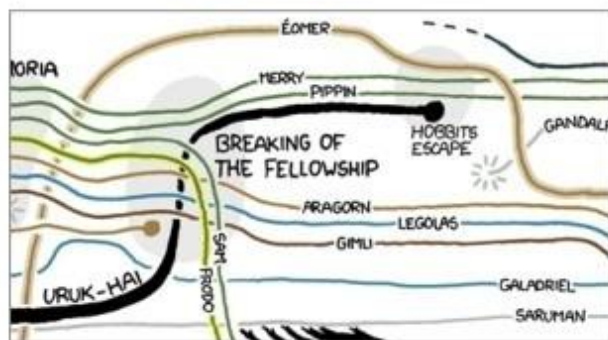
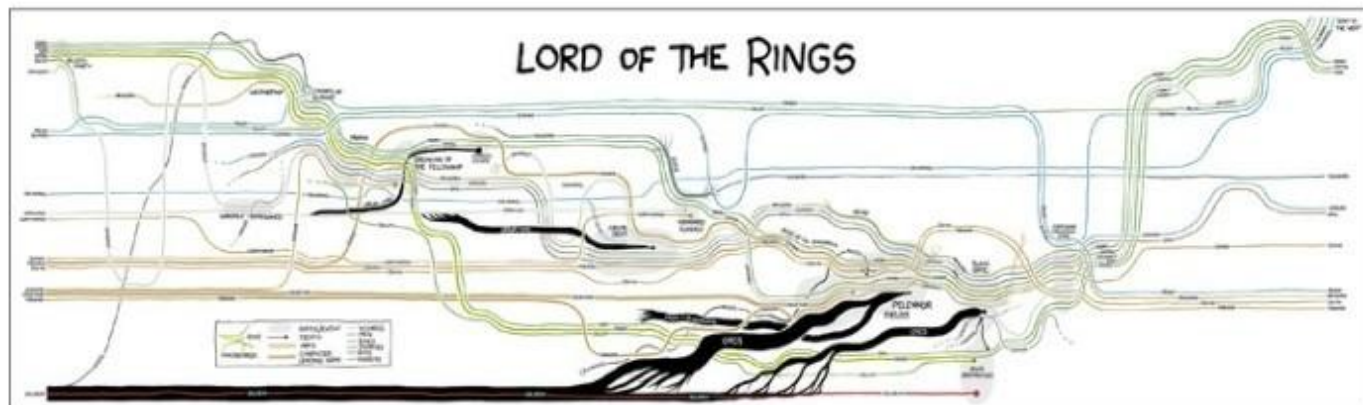


流状分支时间主线可视化

- 基于河流的可视隐喻可展现时序型事件随时间产生流动、合并、分叉和消失的效果，这种效果类似于小说和电影中的叙事主线



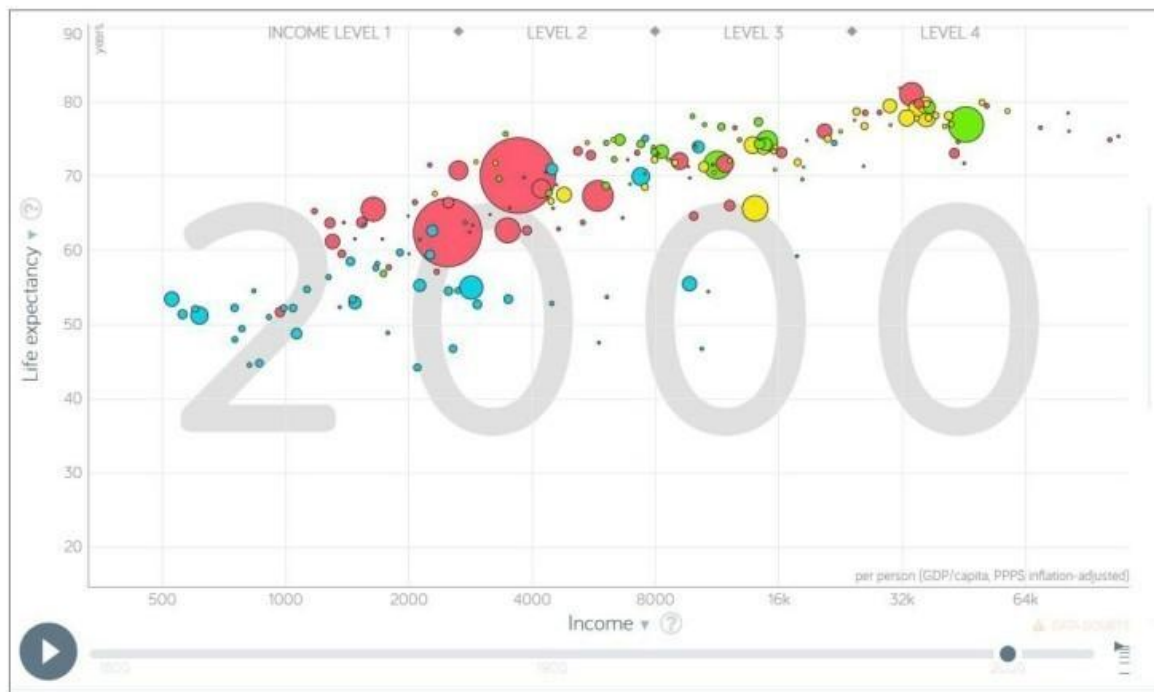
“指环王” 人物关系的演变





时间属性的动态可视化

- 在诠释某些动态事物的过程时，适当地采用动态可视化方法，有助于普通用户以可视的形式了解整个事件过程，达到一图胜千言的效果

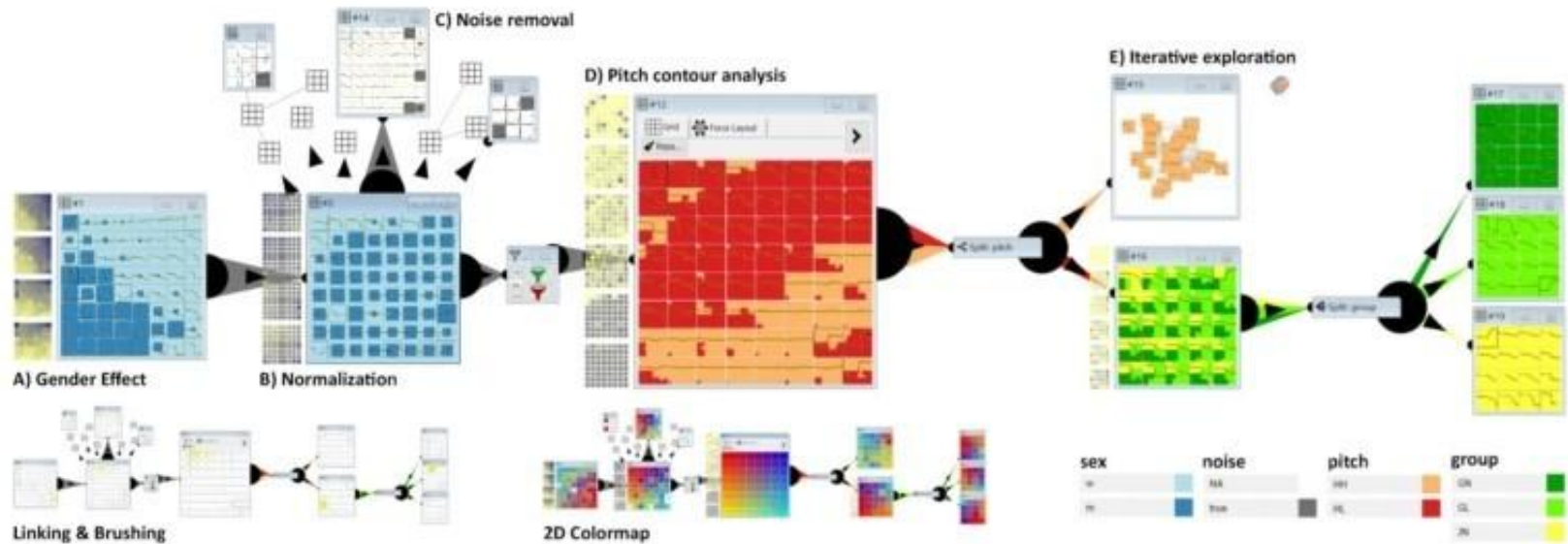


2 多变量时变型数据可视化

概述

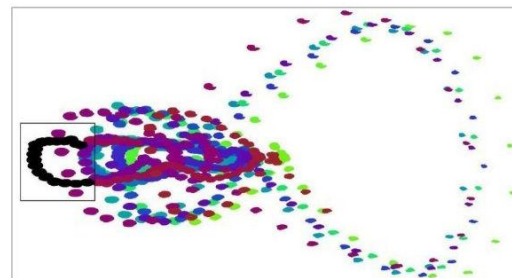
- 面向大尺度数据，首要任务是对数据进行抽象和重构，以便刻画复杂有序数据集的内蕴特征，生成紧凑的概述图像，方便索引和搜索，进而允许用户在交互分析过程中添加其他细节
- 与可视化的基本流程“全局摘要；显示重要部分——缩放和过滤；按要求显示细节，进一步分析”一致，归纳为三类基本方法
 - 数据抽象
 - 通过数据降维、特征选取和数据简化等方法，构建增强关键特征而抑制不相关细节的表达，获得有序数据流的时间相关或无关的内蕴量或隐含的特征模式
 - 数据聚类
 - 将数据集划分为多个具有某种相似性的子集
 - 特征分析
 - 包括特征抽取、语义分析等操作。在一个高维时变型数据中，通常假设其中蕴含某些事件的演化规律。基于事件的可视化技术，通常包含事件定义、事件抽取、基于事件的语义分析等三个步骤

SOMFlow

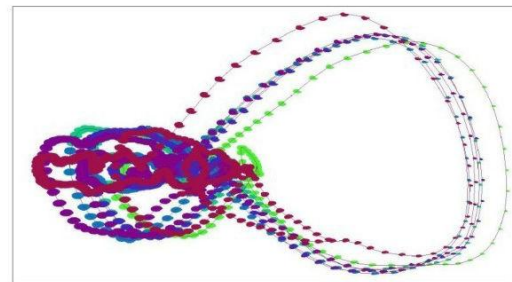


基于线表示的可视化

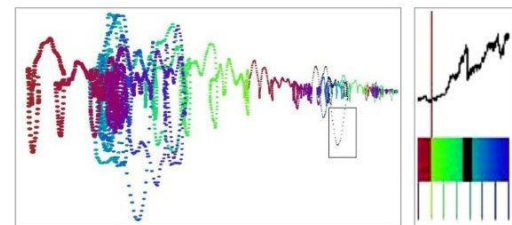
- 将时变序列中的每个数据采样点连接，原时变序列组成一条在高维空间的线，在低维空间可视化这条线可揭示高维空间的时间序列演化趋势



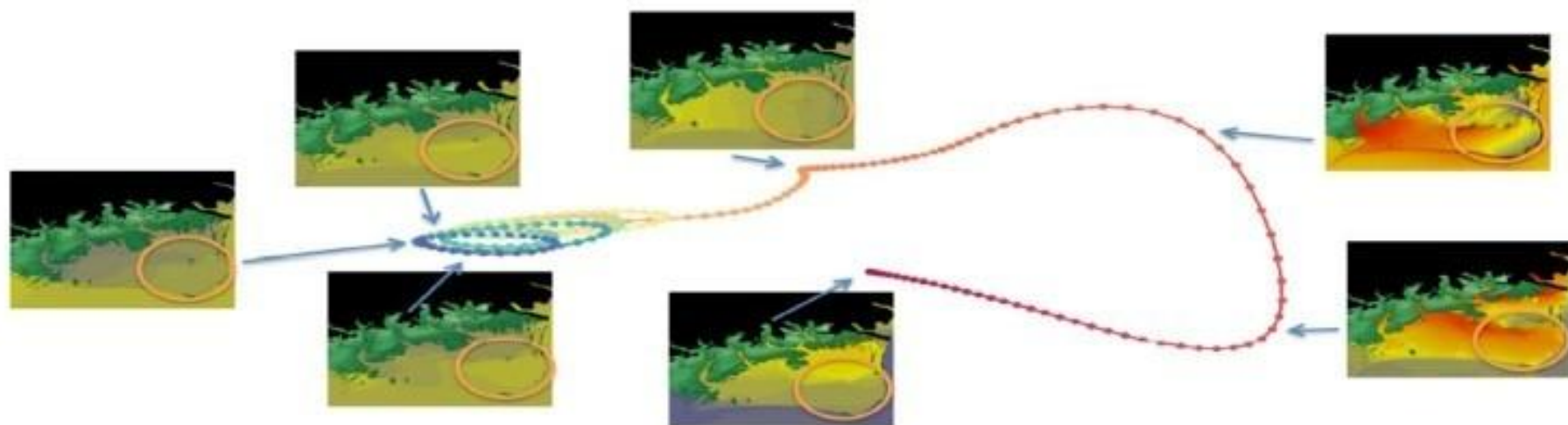
(a)



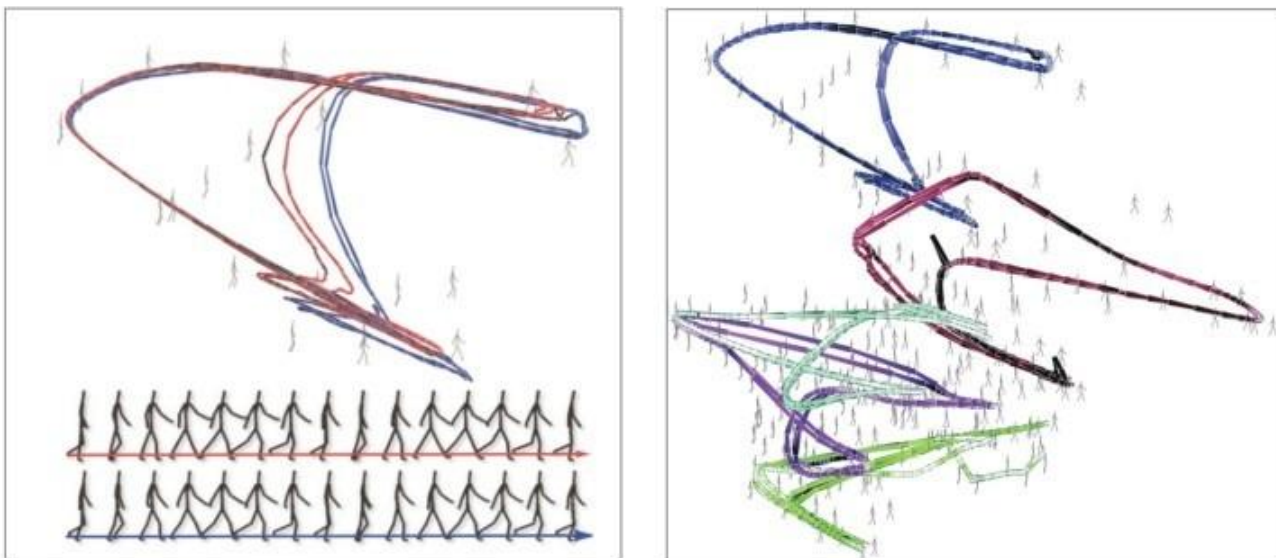
(b)



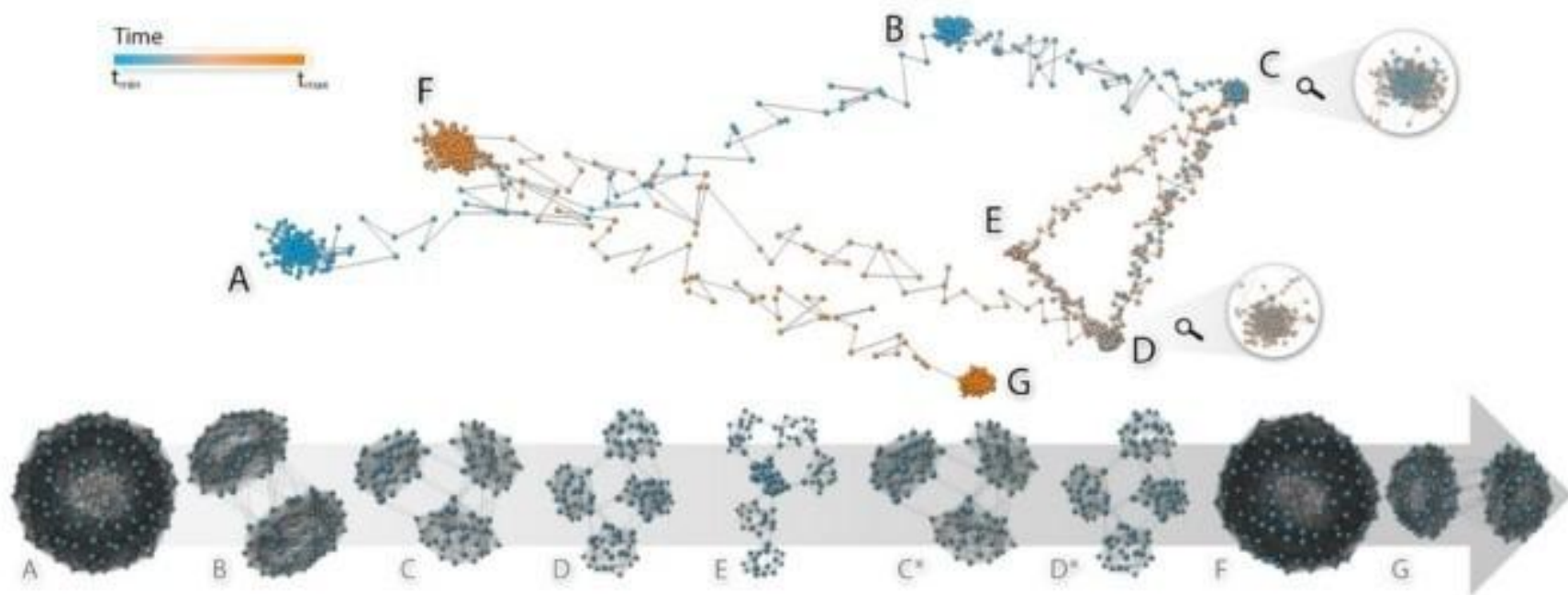
(c)



- 采用动作捕捉(Motion Capture)设备获取的人体动画序列是高维的时变序列

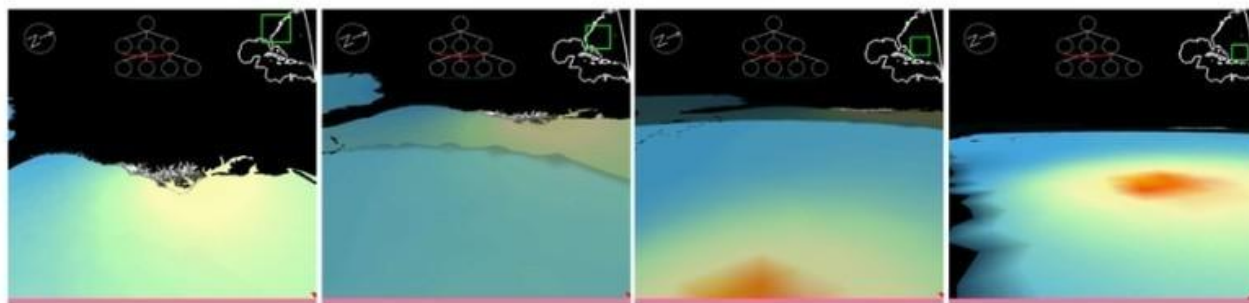


时变网络数据



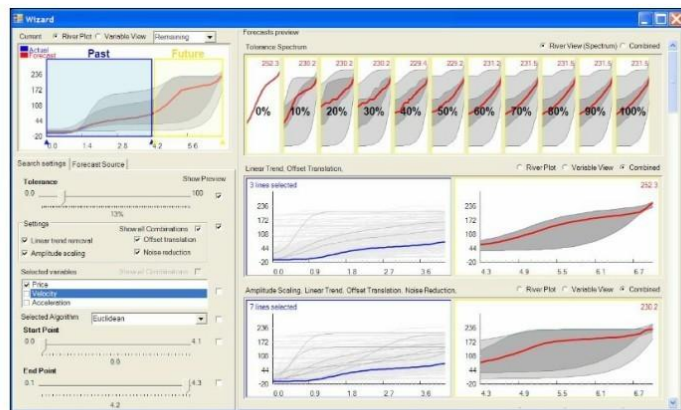
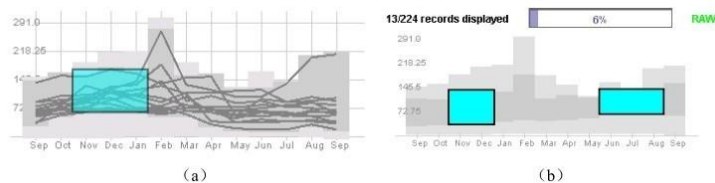
基于图结构的可视化

- 基于事件的时变型或顺序型数据可视化的核心是事件演化的组织
 - 用户根据领域需求和任务描述关注点，并基于此从数据中找到与用户关注点实际匹配的事件，对事件分类获得不同类型
 - 继而，根据事件类型的特征描述，从输入有序数据中检测事件，得到事件实例
 - 最后，通过可视化方法将检测到的事件整合到可视表达中
- 在可视化中，可采用树、图等非线性结构表述事件结构的非线性



时间序列数据的可视化交互

► 设计合适的交互方法表现重要的区域。代表性方法有概览加上下文、层次细节等



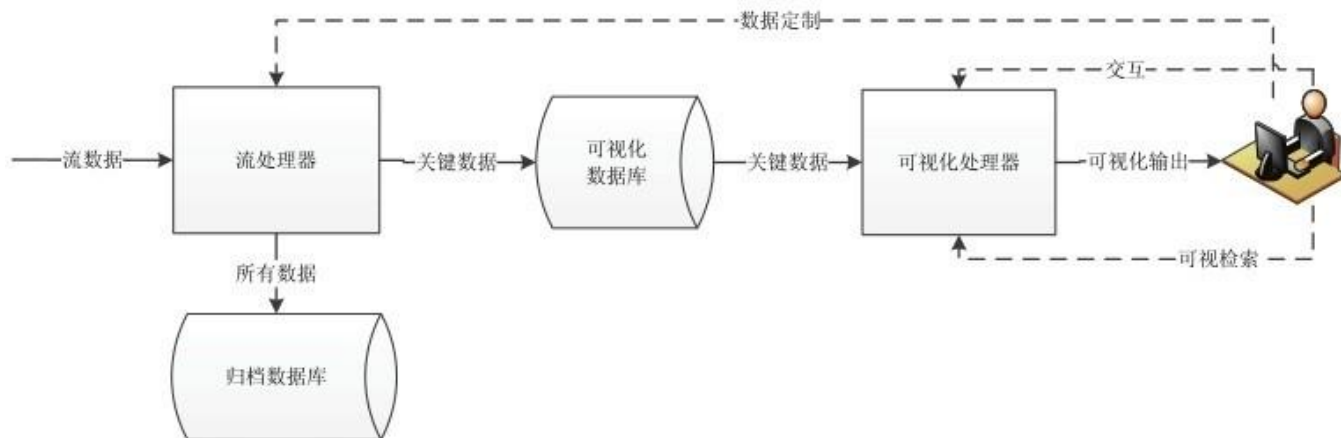
3 流数据可视化

流数据

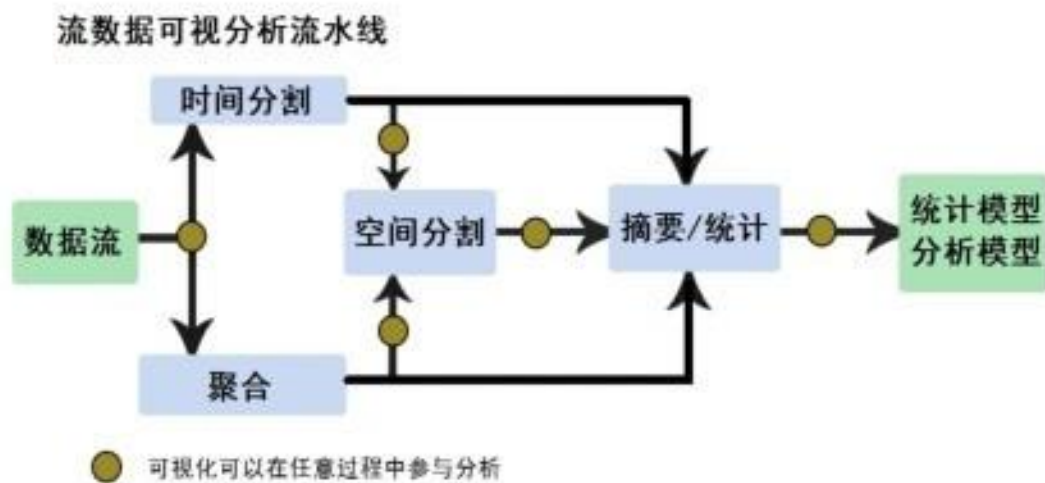
- 流数据是一类特殊以一个或多个“连续数据流”的形式到达的数据
 - 常见的流数据有移动通信日志、网络数据(日志、传输数据包、警报等)、高性能集群平台日志、传感器网络记录、金融数据(如股票市场)、社交数据等
 - 数据流的潜在大小也许是无限的
 - 数据元素在线到达，需要实时处理，否则数据的价值随时间的流逝可能降低
 - 无法控制数据元素的到达顺序和数量，每次流入的数据顺序可能不一致，数量时多时少
 - 某个元素被处理后，要么被丢弃，要么被归档存储
 - 对于流数据的查询异常情况和相似类型比较耗时，人工检测日志相当乏味且易出错

流数据可视化模型

- 把不同的处理方法放在流数据处理器这样一个黑匣子里，综合可视化的过程得到了一个流数据模型。
- 关键数据进入可视化处理器后，经过一系列可视映射和布局等可视化过程转化为可视化输出
- 用户交互包括：输出内容的可视检索；对可视布局的基本交互；自定义的数据定制



流数据分析流水线



流数据处理技术

➤ 流数据挖掘的算法种类繁多

- 包括分类、聚类、频繁模式挖掘、降维等传统数据挖掘算法在流数据中的改进算法，大数据相关的统计方法、采样算法和哈希算法，以及滑动窗口、数据预测等流数据特有的算法

➤ 关注

- 窗口技术，是时序数据特有的技术。
 - 包括滑动窗口、衰减窗口和时间盒，给予不同时间段的数据不同的权重，让最近的数据发挥更大效用。
- 相似性计算，是时序数据聚类、分类、检索、降维以及异常检测的基础
 - 基于形状的(Shape-based)相似度、基于特征(Feature-based)的相似度、基于模型的(Model-based)相似度和基于压缩(Compression-based)的相似度
- 符号技术，将时序数据转化到另一个维度

窗口技术

- 窗口技术在数据的时间上进行限定
 - 滑动窗口、衰减窗口、时间盒(Sliding Window,Decaying Window,Timebox)
- 滑动窗口的设计假设数据带有时效性
 - 随着时间流逝，窗口向前滑动，始终只包含有效时间范围内的数据
- 由于数据流处理有严格的时间与空间限制，对于大多数算法，只能以降低计算结果的精度为代价，从而达到降低算法时空复杂度的目的，实现流数据的实时处理

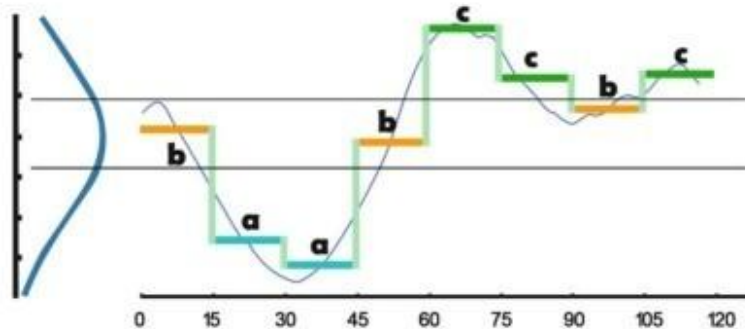
时序数据相似性计算

- 动态时间扭曲(Dynamic Time Warping,DTW)是基于形状的相似性算法。动态规划算法，计算两个序列为矩阵从(1,1)点到(m,n)点的最佳路径，m和n分别为两个序列的长度。距离满足
 - 边界条件，路径要从矩阵的起始位置(1,1)开始计算到矩阵的结束位置(m,n)。
 - 连续性，要求路径的计算是连续的，反映在矩阵中就是最佳路径必须是不间断的一条线
 - 满足单调性，要求路径的计算是单调的，反映在矩阵中就是最佳路径必须是向前向上的一条线。
- 算法步骤
 - 最佳路径函数：定义 $D(i,j)$ 为 $A(1:i)$ 和 $B(1:j)$ 之间的动态时间扭曲距离，对应从(1,1)走到(i,j)的距离。
 - 最佳路径的迭代关系： $D(i,j)=|A(i)-B(j)|+\min\{D(i-1,j),D(i-1,j-1),D(i,j-1)\}$ ，起始条件为 $D(1,1)=|A(1)-B(1)|$
 - 最后的最佳路径则为 $D(m,n)$

3	12	12	12	9	15	2
2	10	10	10	8	9	2
10	9	9	9	8	1	8
1	0	0	0	1	10	12
1	0	0	0	1	10	12
1	0	0	0	1	10	12
A/B	1	1	1	2	10	3

符号技术

- 符号累积近似（Symbolic Aggregate Approximation, SAX）是一种针对时序数据的符号表达。数据经SAX表达转换后可以再用时序数据相似性算法快速得到其相似性
- SAX经过两次离散化将时序数据近似转化为字符串，所有时序数据的聚类、检索等操作都转化为字符串操作，并借助后缀树的数据结构和相关算法加速字符串操作



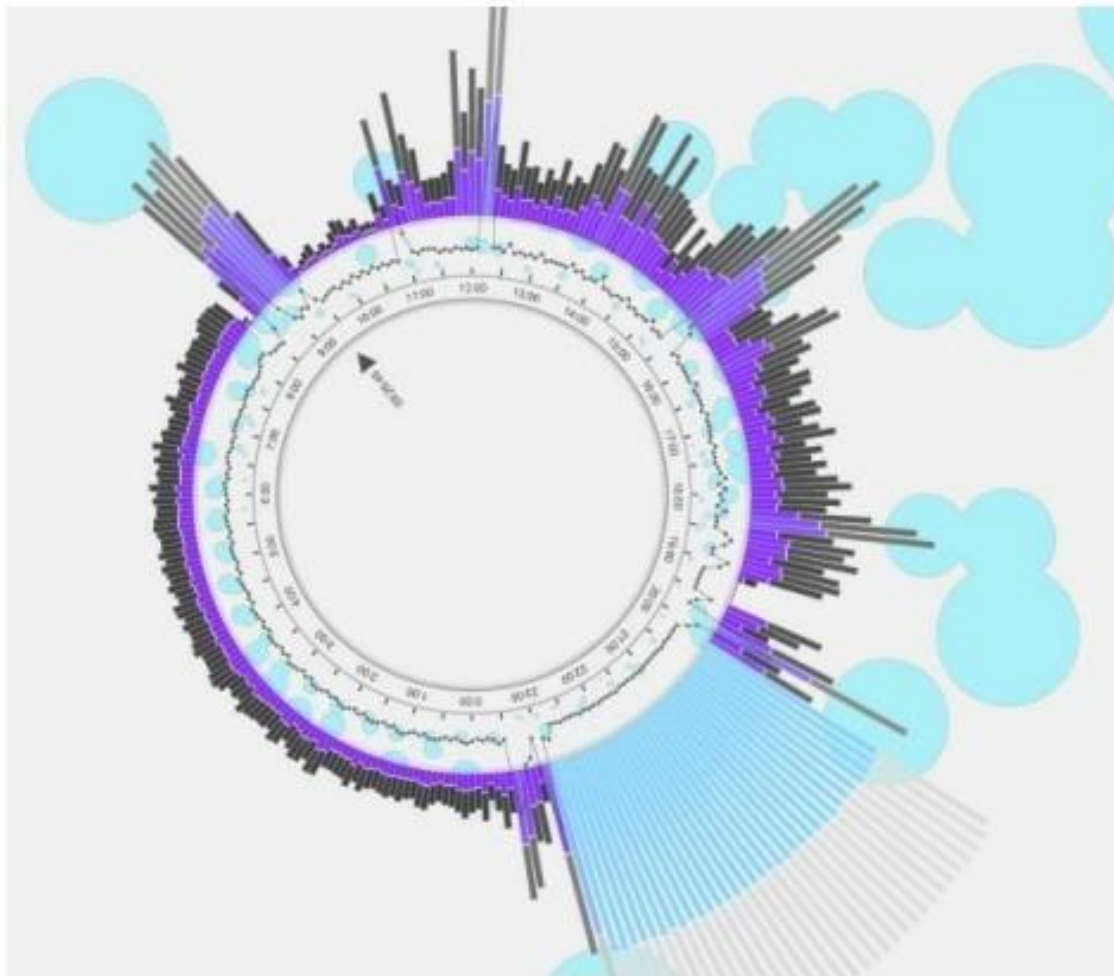
流数据可视化案例

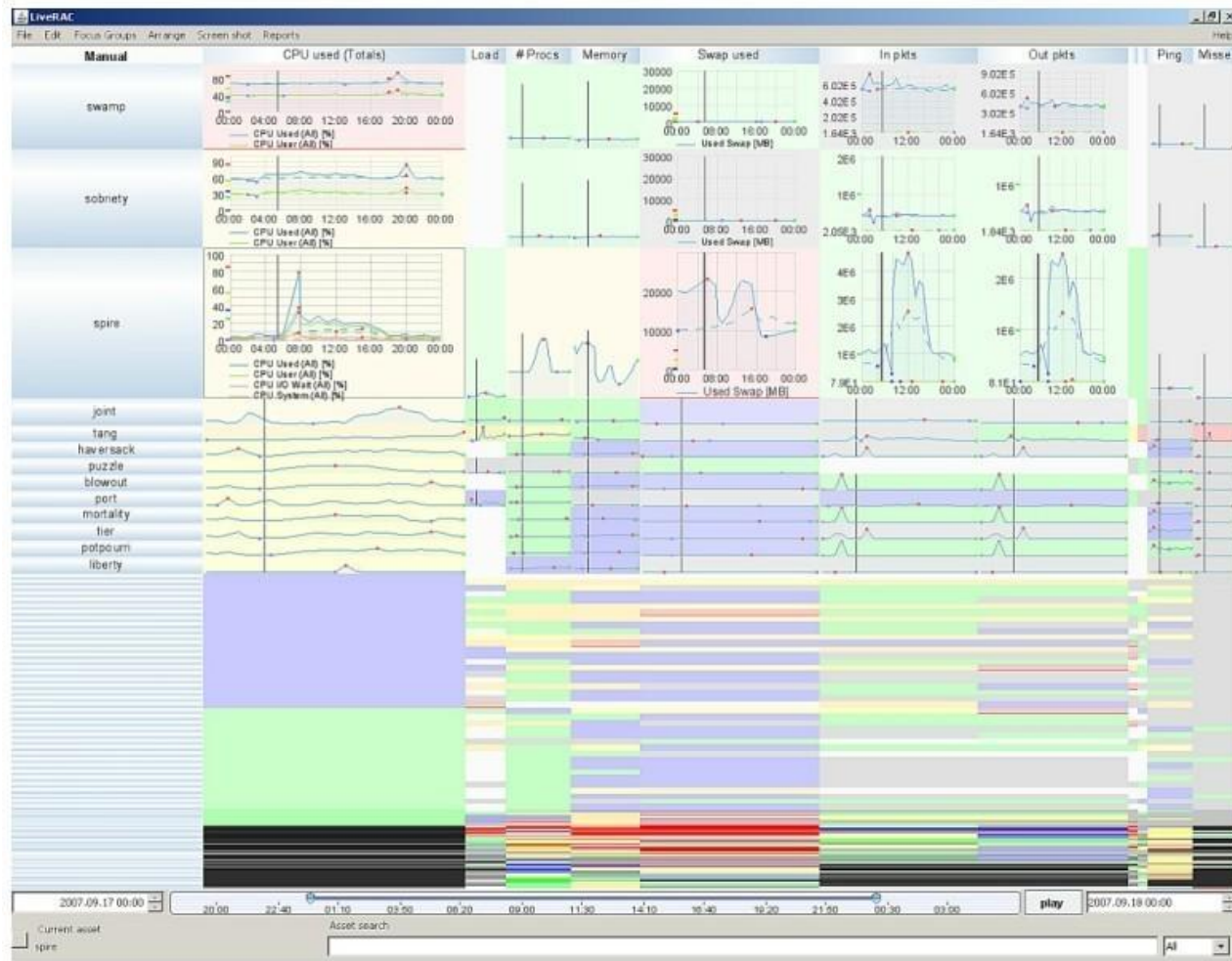
- 流数据可视化按功能可以分为两种可视化类型
 - 监控型，用滑动窗口固定一个时间区间，把流数据转化为静态数据，数据更新方式可以是刷新，属于局部分析
 - 叠加型，或者是历史型，把新产生的数据可视映射到原来的历史数据可视化结果上，更新方式是渐进式更新，属于全局分析
- 局部分分析与全局分析各有侧重点，为了得到更加全面的分析结果，往往将两种可视化结合到一个系统中

系统日志监控流数据

- 系统日志数据反映了一台机器、一个计算集群的系统性能，是商业智能和高性能计算中的重要数据
 - 工业界已经有Splunk[\[1\]](#)、Loggly[\[2\]](#)、Flume[\[3\]](#)等诸多或收费或开源的系统日志监控工具
 - 在系统底层插入脚本获取性能数据，再用基本的条形图、折线图等基本统计图和信息检索工具得到系统性能的概要分析

➤ LogTool



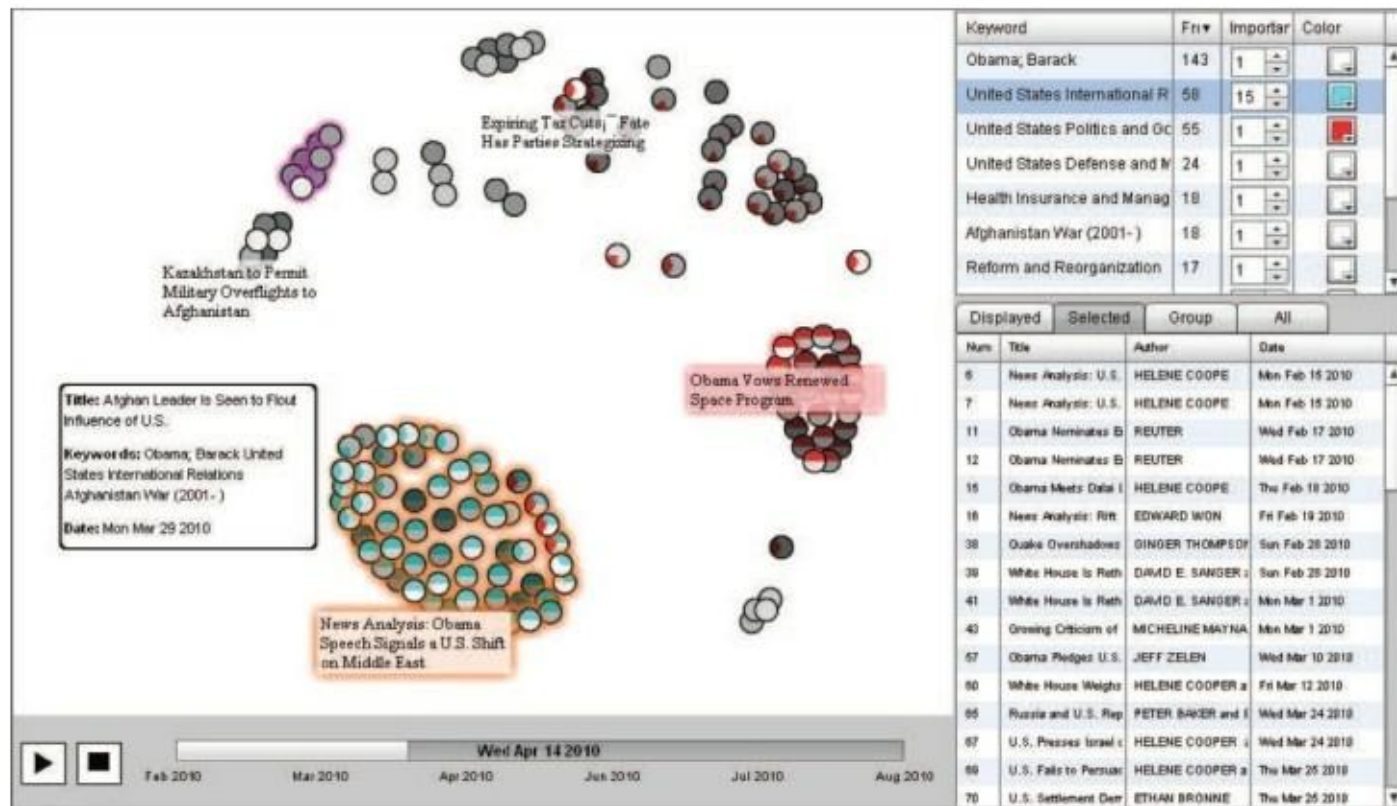


文本流数据

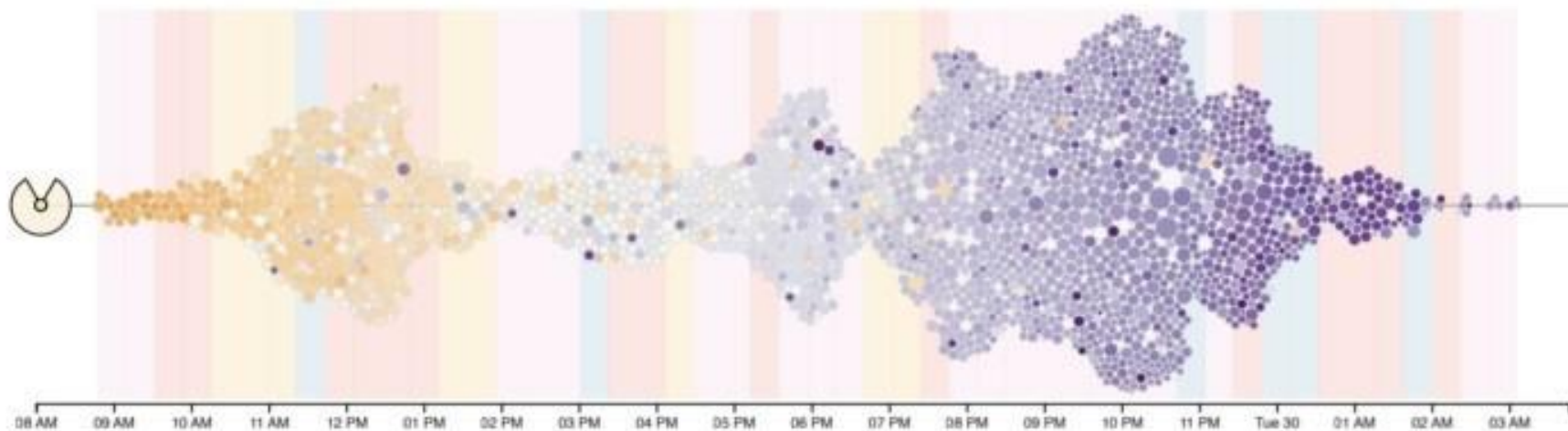
从事件角度对文本进行可视分析，挖掘事件的发生、发展及变化
EventRiver[Luo2012]是一个广播新闻视频集合的交互式探索工具



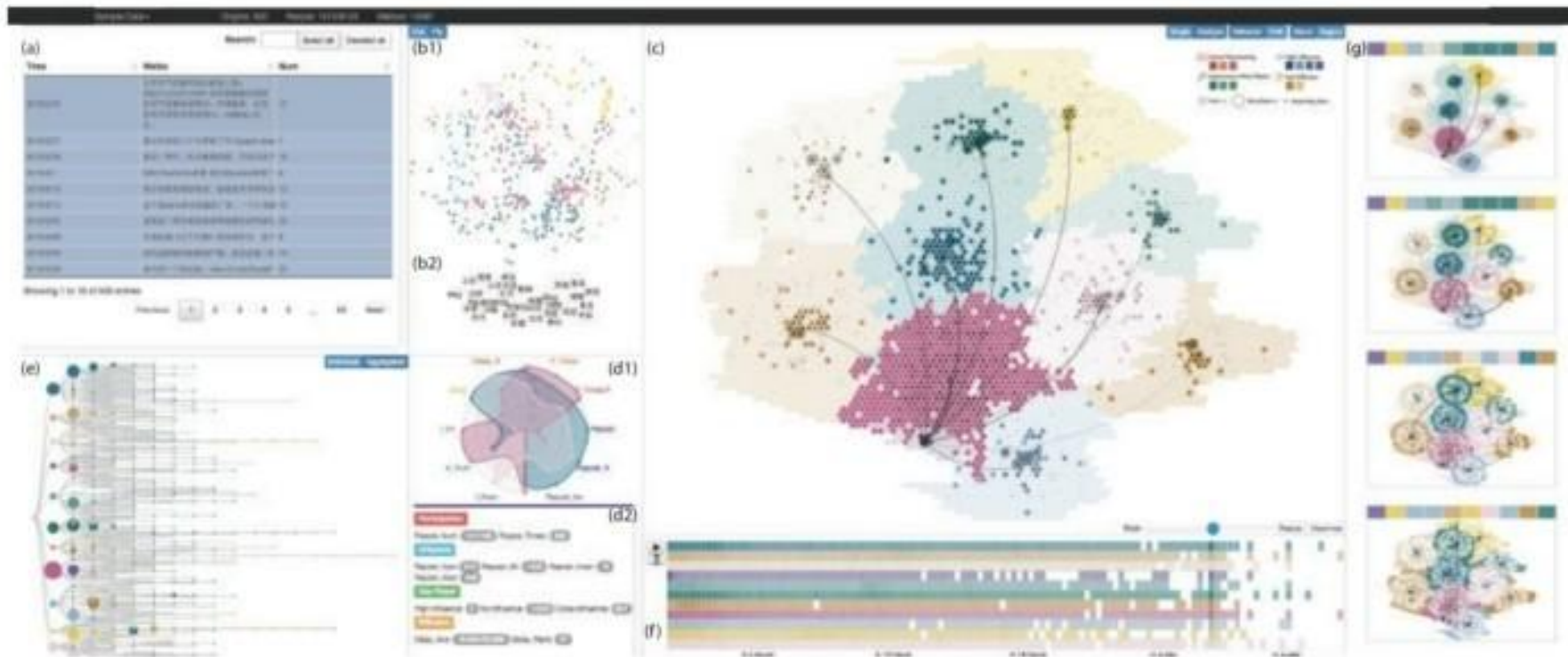
StreamIT



FluxFlow



D-Map: 利用地图的隐喻来对时序文本进行可视化





并行流计算框架

- 由于数据的价值随着时间的递增而递减，所以数据越快处理，结果就越有价值，这也是实时处理的价值所在
- 典型的有Yahoo! 的分布式流计算平台S4
- Twitter的实时数据处理框架Storm
- Facebook的实时数据处理分析平台PUMA[Borthakur2011]
- 斯坦福的流数据管理系统STREAM[Stream2003]