

Weekly Report in Week 2 July 2022

1. Book Notes

- Visualization Analysis and Design
 - Chapter 4
 - 书接上周，对于验证，切忌将不同层级的验证方法和威胁问题对应，这样的不匹配将会使验证毫无意义，看算法的时间没看验证视觉编码的好坏。不过实际构建vis时，也不是非要4层全部验证，根本不太现实。一般的研究基本都是选取部分验证方法来进行，这些方法可以匹配对应设计层级即可。
 - 这部分作者给了很多验证vis各设计层级的很优秀的例子，这些项目也蕴含了许多后续要讲到实现vis的各个步骤途径。
 - Genealogical Graphs（系谱图）该篇文章阅读笔记见papers那一节。上周五浙大网课上讲树、图网络可视化的那个老师也有讲到他们做的一个清代家谱的项目。

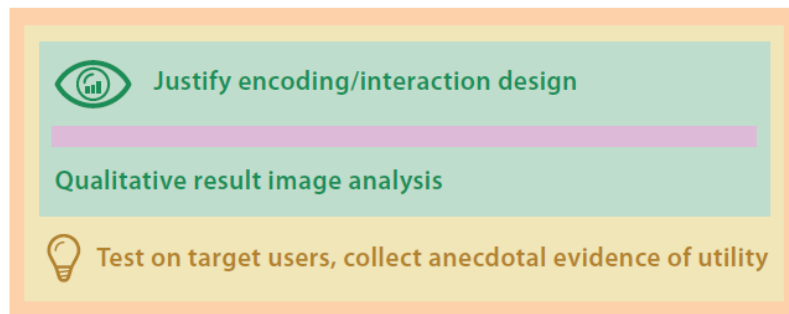


Figure 4.7. Genealogical graphs [McGuffin and Balakrishnan 05] validation levels.

- MatrixExplorer

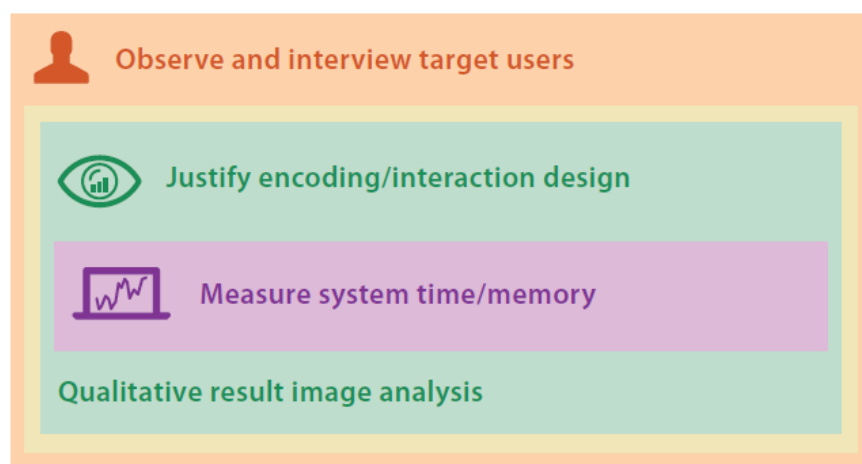


Figure 4.9. MatrixExplorer [Henry and Fekete 06] validation methods.

- Flow Maps

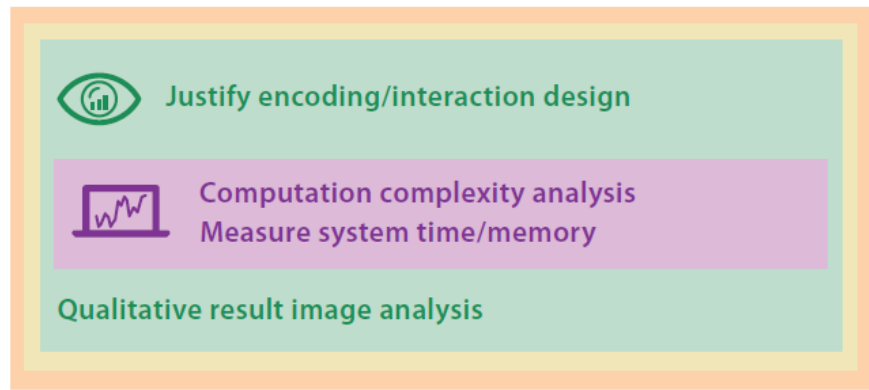


Figure 4.11. Flow map [Phan et al. 05] validation methods.

2. Papers

1 教材中关于验证的第一个例子: Interactive Visualization of Genealogical Graphs

- 面临的主要问题:
 1. 过往的传统系谱图图表都往往很简单
 2. 想要一个能自动生成、布局的系谱图，但是又因规模大，各种交叉的边使得布局合理很困难。
 3. 前人提出了一些结构，但都无法解决系谱崩坏，简单来说就是后代的增长和扩展与其在平面图上的几何布局不匹配。（随着图的扩大，只会越来越挤（即使允许1型通婚的存在，增加单体的复制节点也不行）。而且这么拥挤的图也没法看出哪些是同代人之类的信息。）
 4. 又从图形表示企图解决，但不管怎样，还是无法解决交叉边或过长边的问题（这俩本来就矛盾）。换成嵌套，还是缩进轮廓风格也都不能根治，顶多能优化。
- 解决问题和验证系统的过程:
 1. 文章最开始提到一个hourglass chart，从两个方向出发，一边展示选中根节点的祖宗树，一边展示根节点的子孙树。在这个基础上，通过offset，在根节点基础上再选他的一个祖宗节点为另一个根节点，两个根节点可以同时最大化祖宗树和子孙树。
 2. 有了dual-tree以后，还可以更换风格，比如之前提到的indented outline 风格，可以更节省空间。
 3. 至于系统实现，完成软件部分。分别从设计、交互手段入手。
 - 设计上有两种风格，点线（他又分为横与竖呈现）以及缩进轮廓
 - 交互上主要是如何对树进行展开、折叠
 4. 最后通过下游验证，看最初使用的用户反馈来验证

2 教材中第二个例子: MatrixExplorer: a Dual-Representation System to Explore Social Networks

- 面临的主要问题:
 1. 首先就是在社交网络领域，表示关系网络的结构还是大多是点线关系风格的。很少有人有矩阵来表示（主要是点线可以更明显的表示关系，矩阵可以更好应对大的，密集的图。各有利弊）
 2. 目前分析社交网络的vis有两类：基于编程和基于菜单

- 基于编程的对于零基础的领域专家并不友善，更像为设计者自己设计（这在教材中作者也反复强调）
- 基于菜单的又不够直观，对于探索发现不大支持（需要各种复杂交互）
- 3. 对于基于矩阵的表示，主要关注其重新排序，主要有两种方式：自动排序 和交互系统
 - 有一系列前人关于邻接矩阵重现排列的算法
 - 对于交互系统来辅助矩阵重排序，目前还没有这类系统（这不就明确目标有活可干了）
- 4. 接下来是确认需求的重要步骤，找了许多不同领域专家作为参与者来调查他们在社会科学研究上的需求。共整合了13个需求出来。后文介绍MatrixExplorer也始终在按照如何满足这些要求来行文。（先分三个问题入手调查，三个阶段进行，最后总结列出各项要求，来指导后面的工作。好结构）
- 解决问题和验证系统的思路与过程（这篇文章的思路很清晰，了解一些相关工作后就直接开展了选取不同领域专家参与的调查，来确定完成这个系统的需求。依次满足这些需求，解决主要问题。从而实现了这个MatrixExplorer）

提出了第一个去尝试满足社会科学研究者对于探索系统各项要求的vis， MatrixExplorer。

- 两种表达方式：不管你点线好还是矩阵好，我干脆都用。而且全都同步，交互过程各视图变化一致。
 - 提供各种概览，无论是对于数据集本身有概览，还有其分析上的一些概览供参考。工作区的加入还允许许多窗口，折叠展开都能更好探索，防止空间有限。
 - 连接组件的概览，将整个图形提供宏观概览，还可以直接通过他进行过滤或选取来进入详细视角。
- 总之，概览不但对整体结构全方位概括，还能用于上下文提醒，引导用户的工具。
- 对于视觉变量，可以通过属性动态过滤和排序。
 - 对于交互，作者尽可能使布局创新，且各参数的调整是可交互的。进而引出：点线风格的排序本就有许多好算法可选。**主要是矩阵排序，作者将邻接矩阵改为最短路径矩阵，尽可能将相邻点在矩阵行列上最近。进而重排。**而且重排中还有一些好功能：可以锁定一部分行或列，可以过滤或者遗忘（过滤使结构简单清晰，直指目标。遗忘将新生成的凸显，过去的淡化。二者都为了简化结构，方便用户探索系统并进行分析。
 - 最后就是如何引导用户，并在vis中找到对布局、对集群设定的共识。也就是满足需求，验证系统。（为了方便区分不同集群，系统还提供了比较工具）

总之就是时时刻刻都围绕着参与者提出的需求来跟进，创新本就有的，还做的非常完备。

3 教材中第三个例子：Flow Map Layout

- 面临的主要问题：
 1. 过多的连接关系导致过多的线，过多的线导致视图乱的很。
 - 2.
- 解决问题和验证系统的思路与过程：
 1. 总的来说，提出一个可以自动生成flow map的系统，而且该系统可以最大程度的最小化边（也就是将公共部分统一到一个集群当中），来尽可能减少混乱程度。
 - 对于智能更改（扭曲）位置：
 - 交叉边最小化
 - 流量线才是主体，一要通过粗细体现流量大小，还要合理安排粗细线交叉时候的布局。
 - 对于合并边：通过分级群聚（hierarchical clustering）的方式，将就近的边合并起来。
 - 对于智能边的路线：也通过分级群聚（hierarchical clustering）

2. 从算法上来一次针对：

- 首先对于布局的调整，正常100以内个节点来说，力导向图第一每次呈现的不同，因此选用强制扫描这一更稳定算法，在稳定节点上做flow maps。
- 首先生成主集群。如果先进性根的更换，则需要重组集群，递归更换。
- 空间布局上，通过对分支节点的位置确定，以及尽量避免交叉。
- 边缘路线：对于产生了交叉的，通过新增节点来改变线路，进行重新规划
- 关注改变根节点，导致的图层叠加带来的复杂度增加问题

3. 作者直接通过case study，找数据集来用在他们系统上进行分析验证：

- 对于智能调整扭曲：虽然轮廓清晰可辨，但有的在垂直方向上被扭曲了，东海岸的各州被分散开来，又会把西海岸延伸得太长了
- 对于合并边：若区域内有太多的目标节点，强制分割会引入太多额外的路由节点，导致混乱。
- 对于智能边缘路由：在离开根节点处没执行算法，还是可能会重叠。
- 多个flow map一起分层共享多个节点的分支很有效。
- 对于不是由少数高量级值主导的数据集，其实线性映射还是更吸引眼球。

4. 全文主要在明确当前在flow map系统这里有需求后，尤其是自动生成的算法上时，就着重入手：尤其是算法上，对各部分的复杂度，以及各负载情况进行分析。最后还通过下游验证：几个数据集的真实实验来定性的验证分析系统当前的优劣。

3 Else

浙大暑期夏令营理论课程部分完结撒花。

实践部分由于线上无法参与，自己结合之前浙大20年在慕课上面的内容（虽然也没有直接实验课部分以及实践详细部分，但感觉内容更全），顺便在看b站上把前端的基础内容再熟悉一下。

回家这周预计直接摆烂。。。