科学数据驱动的演示动画快速构建与自动探索

刘日晨

南京师范大学计算机科学与技术学院/人工智能学院

研究组简要介绍

- **南师大可视化与可视分析研究组**
- **2018年9月成立（三年不到），两位本科生入组**
- **到目前为止，已有约10位本科生入组，2位硕士 (还有今年刚毕业了1位)**
  - 有些参加算法竞赛：ACM竞赛，蓝桥杯，计算机设计大赛，国家级大创等
  - 大部分学生成绩名列前茅

研究组成员相对全的照片 (拍于2019.12，实际为研究组欢送书记外调)

我们今年发表在CHI'21的两篇论文 (其中1篇LBW)

- **Richen Liu**, Min Gao, Shunlong Ye, and Jiang Zhang.
  IGScript: An Interaction Grammar for Scientific Data Presentation.
  CHI Conference on Human Factors in Computing Systems (ACM CHI'21),
  Article No.: 26, pages 1-13, Yokohama, Japan, May 8-13, 2021.

- Xiaohan Wang, Chuyu Zhang, Yu Zhu, Xueyi Chen, Liming Shen, **Richen Liu***, and Rongtao Qian.
  Hybrid Line-Based and Region-Based Interactive Set Data Visualization.
  CHI Conference on Human Factors in Computing Systems (ACM CHI'21 LBW) ,
  Article No.: 411, pages 1-7, Yokohama, Japan, May 8-13, 2021.
  (**Late-Breaking Work：LBW**, 双栏7页+2页附录的poster track论文)

注：以上两篇论文的学生作者全是小组本科生

IGScript: An Interaction Grammar for Scientific Data Presentation

- **Richen Liu**, Min Gao, Shunlong Ye, and Jiang Zhang.
  IGScript: An Interaction Grammar for Scientific Data Presentation.
  CHI Conference on Human Factors in Computing Systems (ACM CHI'21),
  Article No.: 26, pages 1-13, Yokohama, Japan, May 8-13, 2021.

研究背景与动机 (1/2)

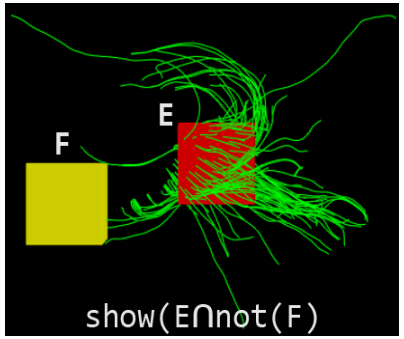
- **主题：科学数据演示动画**的构建与自动探索
- **科学数据驱动的演示动画/数据演示动画** (data presentation animation)：揭示自然科学现象的动态变化 (dynamic changes) 与时变演化过程(evolution processes)；
- **传统构建科学数据动画的方法：手动交互+录屏。缺点：**录制后**无法更改**；录制过程**很难定制**底层的数据计算或可视化渲染的参数；很难**覆盖**所有重要特征；重要特征**特写镜头**不够；多次录制**繁琐**；无法像拍纪录片一样在重要特征之间华丽**切换**；无法做到目标对象与视角变化的**渐变**
- **科学数据演示动画的定制：**数据规模大、变量多，需要大规模科学计算与GPU渲染，但GPU编程**学习曲线陡峭**，编程经验不足的领域专家或普通用户更难

```
sampler2D _MainTex;

struct v2f {
    float4 pos : POSITION;
    float2 uv : TEXCOORD;
};

v2f vert(appdata_full v): POSITION {
    v2f o;
    o.pos = mul (UNITY_MATRIX_MVP,v.vertex);
    o.uv = v.texcoord;
    return o;
}

float4 frag(v2f v):COLOR{
    float4 texColor = tex2D(_MainTex,v.uv);
    return texColor;
}

SubShader {
    Pass{
        CGPROGRAM
        #pragma vertex vert
        #pragma fragment frag
        ENDCG
    }
}
```

GPU编程：学习曲线陡峭

研究背景与动机 (2/2)

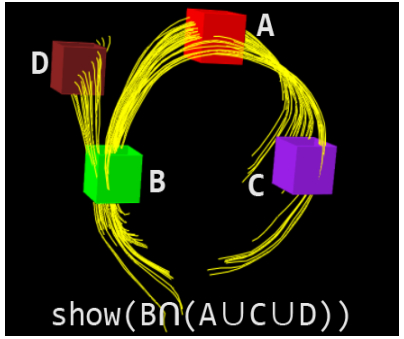
- 科学可视化中常需使用各种**数据查询**或**过滤**用于减少数据规模/减少visual clutter
  - 流场数据的源汇查询 (source-destination query)
  - 轨迹数据的OD查询 (origin-destination query)
- 复杂的数据查询难以通过传统的交互或GUI控件进行定制
  - 难以定制复杂的“交并补非”+优先级查询。例如：DTI数据中水分子弥散运动成像数据中的轨迹查询：“*show* ($B \cap (A \cup C \cup D)$)”



为了解决上述问题，我们提出了 *IGScript*：

- **脚本驱动**：设计文本式的**交互语法 (interaction grammar)** 帮助用户快速定制数据演示动画
  - 融入**自然语言中的语法结构**或使用**领域专业术语**作为关键字：“在大脑白质区放置一个查询box”

    placeANewBox {boxID("A"), at (“**WHITER_MATTER**"), color=green, alpha=0.5};
  - 高效定制化**编辑** (cp & paste & cut & add & delete & append & switch & clip loop)：通过编辑文本来编辑动画片断或定制动画风格
  - 高效直观编写**明确的、无二义性**的复杂查询代码
  - 方便**复用**、**共享**：传统方法通过录制的演示视频太大、且不可编辑
- **数据驱动**：数据驱动的**自动探索**，演示动画自动**全覆盖**数据中所有用户感兴趣区 (ROI) 并仿照类似于纪录片拍摄的**特写**镜头



相关工作

- 本文最相关的工作为DSL (领域自定义库/语言)：从三个角度进行分类调研
  - External Grammars (Ext) v.s. Internal Grammars (Int)
  - Textual Grammars (Tex) v.s. Graphical Grammars (Gra)
  - DSVL, DSML, DSEL
- 现有的DSL大部分集中在流水线的计算 (Vivaldi, ViSlang等) 或渲染 (VolumeShop, Voreen, ShaderFramework等) 部分，或二者的组合 (Diderot等)，几乎没有做交互式动画定制
- *IGScript*更接近于DSVL这一子类，但与其他DSVL工作不同之处在于，*IGScript*更关注数据演示动画的交互式定制而不是可视化本身

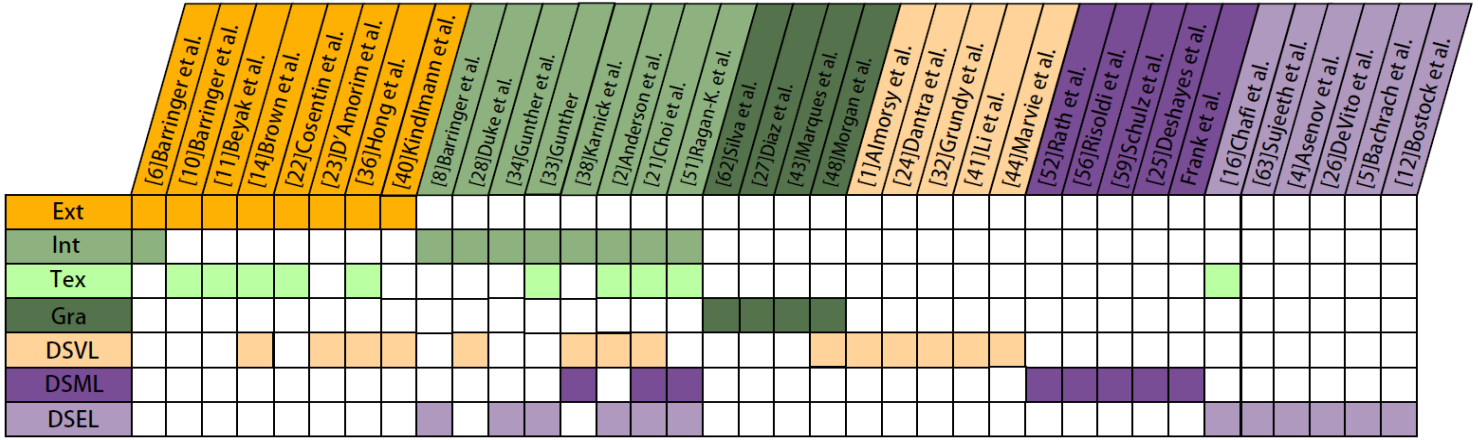
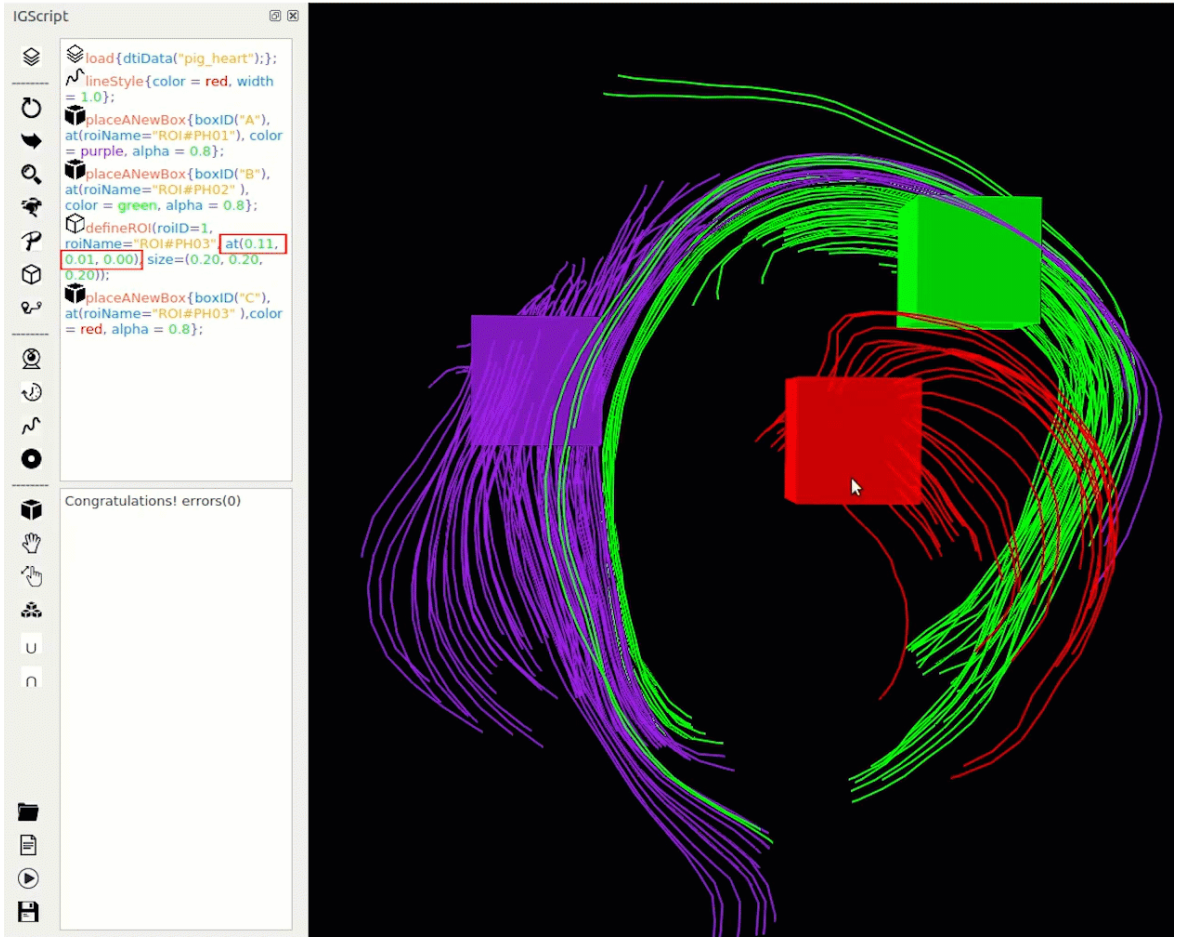


Table 1: Some of related work classified according to the three criteria [60]. They are external (Ext) or internal (Int), and the programming symbols are textual (Tex) or graphical (Gra), and DSVL is designed for visualization libraries, DSML is for modeling libraries, DSEL is designed for embedded libraries.

设计概览 & ROI 定义

- 为了高效定制动画，我们设计了一个**双联动视图** (linked-view)
  - 一个**可视化空间视图** (visualization space view)
  - 一个**编码空间视图** (coding space view)，以提供给用户更好的**可视操纵**(visual steering) 与**可视反馈** (visual feedback) 用于高效的动画编辑、风格定制、参数调节、 ROI定义

- 在双联动视图中定义ROI
  - 在**可视化空间**中通过拖拉拽等操作粗粒度(coarse-grained) 放置与调节ROI的位置、大小、方向
  - 在**编码空间**中通过修改代码文本细粒度调节(fine-grained) 包括**微调**与**深度方向的调节** (勿需刚好转90度再通过平移实现)
  - 两个编码空间**实时联动：**任何在一个空间中的修改都能实时在另一个空间更新以达到可视反馈与可视操纵



技术设计：目前支持的数据类型

- *IGScript*当前支持的数据为**最常见的四大类**科学数据：总结自Charles Hansen和Chris R. Johnson的科学可视化红宝书 “The Visualization Handbook” [28]:
  - 2-D **标量场**数据 (0阶张量，0th order tensors)
  - 3D **标量场**数据，一般被称为标量场**体数据** (0阶张量，0th order tensors)
  - **向量场**数据 (1阶张量，1st order tensors)
  - **张量场**或其衍生数据包括DTI 数据 (高阶张量，higher order tensors)

Grammar Design (1/2)

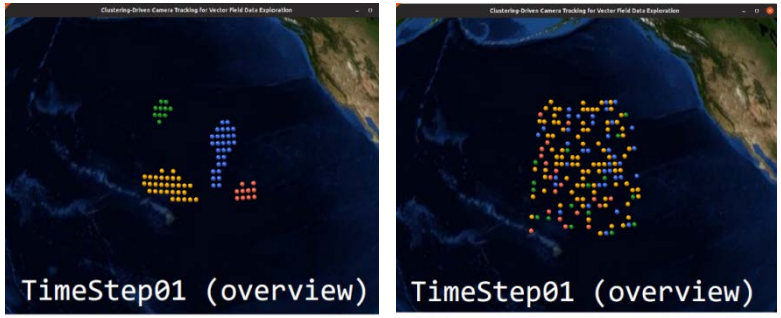
- **通用型语法设计 (General-purpose Grammars)**
  - 定义ROI相关的函数: *defineROI*
  - 数据加载相关的函数: *load*, 类似于C++里的函数重载
  - 摄像头控制相关的函数: *rotate*, *translate*, *scale* (zoom), *parallel*
  - 数据动画生成相关的函数: *animate*, *locate*

```
load { // similar to "overloading" function in C++
    // load a map-based 2-D scalar data
    data(string dataFile); map(string mapFile);

    // or load a 3-D scalar data (volume data)
    volumeData(string dataFile);
    rgbaScheme(string tf);// transfer function

    // or load a vector field data, e.g., "ocean";
    vectorFieldData(string dataFile);

    // or load a DTI data, e.g., "dti";
    dtiData(string dataFile);
};
```

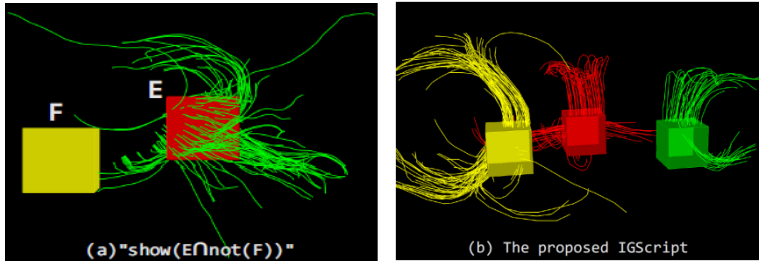
```
rotate {axis=string, angle=string, duration=float seconds};
rotate {axis=(float,float,float), angle=float degrees, duration=float seconds};
translate {to(float,float,float), duration=float seconds};
scale {factor=(float,float,float), duration=float seconds};
animate {speed=string}; // [low, moderate, high]
parallel { // executed concurrently
    rotate {axis=(float,float,float), angle=float degrees, duration=float seconds};
    scale {factor=(float,float,float), duration=float seconds};
    translate {to(float,float,float), duration=float seconds};
};

defineROI(roiID=int, roiName=string, at(float,float,float), size=(float,float,float));
locate {
    roiArray=string[roiName#1,roiName#2,roiName#3,...],
    foreach {
        duration=float seconds,
        interval=float seconds
```

Grammar Design (2/2)

- **专用型语法设计 (Application-specific Grammars): 主要是针对向量场数据与DTI数据的专用语法设计**
  - **向量场**中的基于聚类的源汇查询，**轨迹数据**中基于聚类的OD查询，以及迹线聚类后对不同簇的自动探索 (overview+details)
  - **DTI数据**中的基于Box的交并补非等复杂查询



(a) mode = origin (b) mode = destination

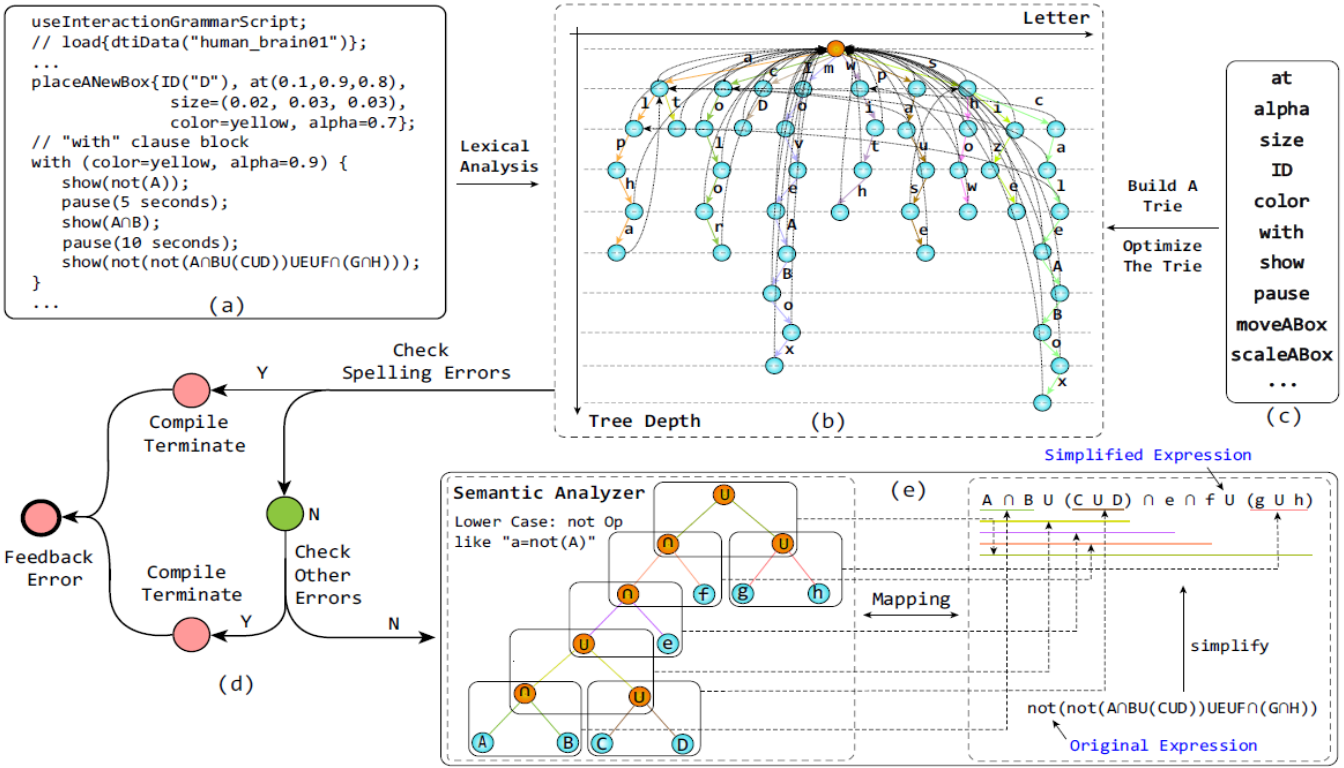



```
trace {
    mode=string, //[destination, origin, realtime]
    clusteringAlg=[dbscan, kmeans, pca, ...]
    lifeTime=float,//lifetime of traced fieldlines
    colorOfCenterPathline=[c1, c2, ...]
};
halfMergeSplit { // overview-to-details
    timeSlots=[overview t1 timeUnits, t2, t3, ...]
}
lineStyle {color=colorL, width=float};
tubeStyle {color=colorT, thickness=float};
```

```
// overloading "placeANewBox": assigned by an ROI
placeANewBox {boxID(string), at(roiName=string),
    color=c, alpha=float};
// overloading "placeANewBox"
placeANewBox {boxID(string), at(float,float,float)
    , size=(float,float,float), color=c, alpha=
    float};
moveABox {boxID(string), to(float,float,float),
    duration=float seconds};
scaleABox {boxID(string), factor(float,float,float
    ), duration=float seconds};

// code block "with": set local color and opacity
with(color=c, alpha=float) {
    show (queryExpr); //e.g., show(not(A∩B)∪(C∩D))
    pause (float seconds);
    show (queryExpr); //e.g., show((A∩C)∪(B∩C))
};
```

Design Details: 双联动视图的实现

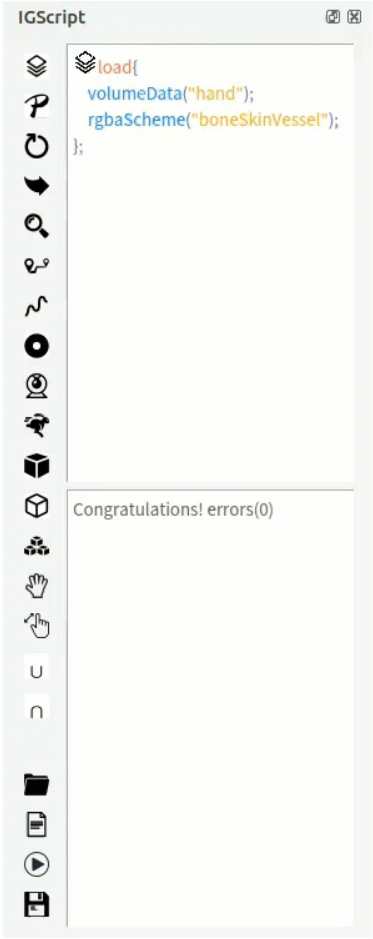
- 我们设计了专用的小型**编译器**：将文本语法代码最终变成数据演示动画
- **反编译器** (代码生成器) ...

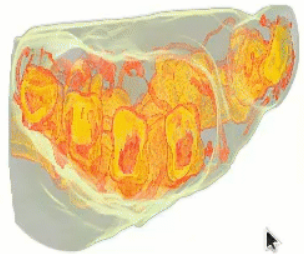


The compiler

双联动视图：反编译器 (代码生成器示例)

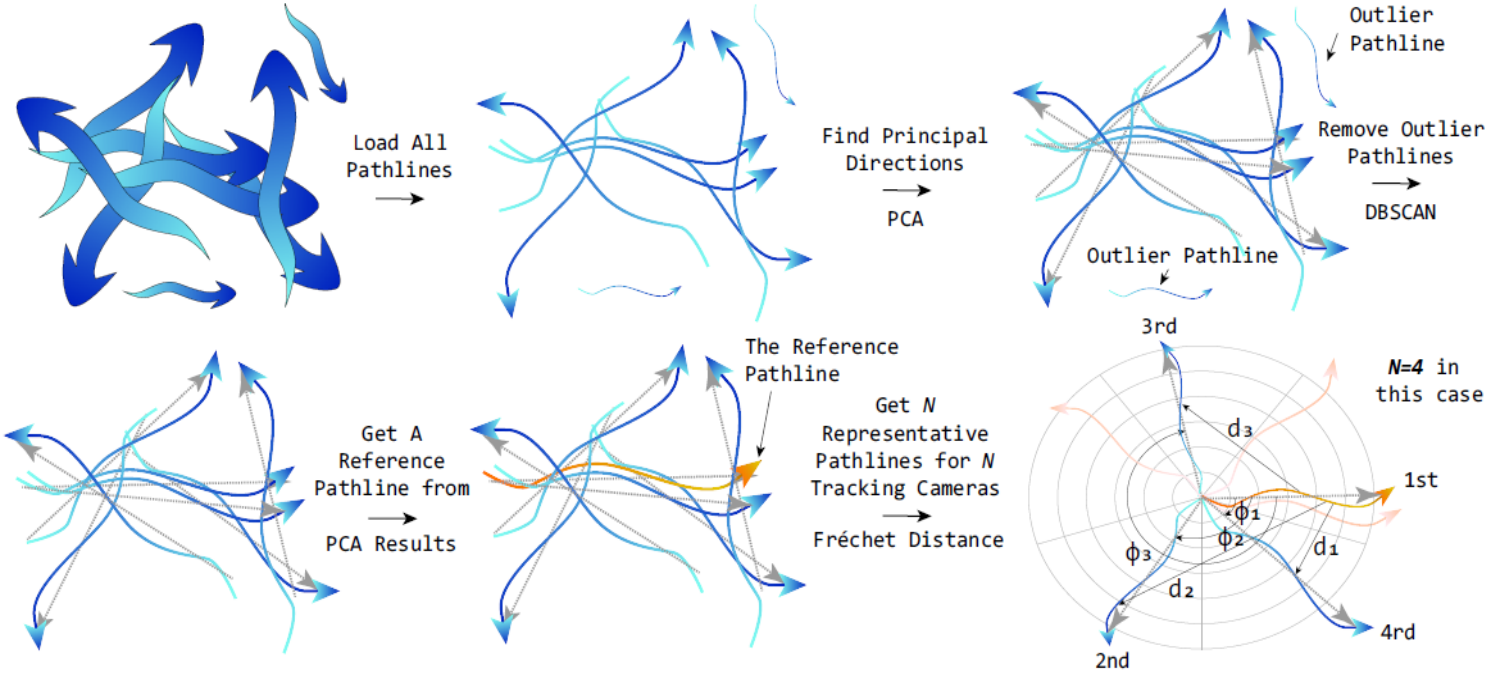
- **反编译器 (代码生成器)：**
  - 通过简单的GUI交互操作来构建代码
  - 在可视化空间中的所有交互操作都可自动转译成对应的*IGScript*代码
  - 代码可复用与可共享
  - 例如：可在另人分享的代码基础上再次编辑，在可视化空间中都可重现编辑后的动画 (cp & paste & cut & add & delete & append & switch & clip loop)





Design Details: 实现

- 针对时变向量场数据的专用语法设计的实现
  - PCA：提取迹线的多条**“主流”**成分，同时去除outlier迹线
  - 聚类：获取迹线不同的簇 (聚类个数为动画构建的虚拟摄像头个数)，并得到每个类的“**中心迹线**”(虚拟摄像头自动跟踪拍摄)



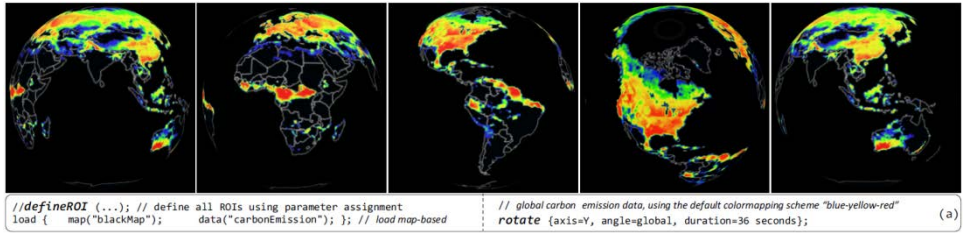
Camera tracking for pathline clusters (PCA components)

Demo of IGScript

论文相关展示下载地址：https://dabigtou.github.io/richenliu/

Case Study 01: 2-D标量场数据

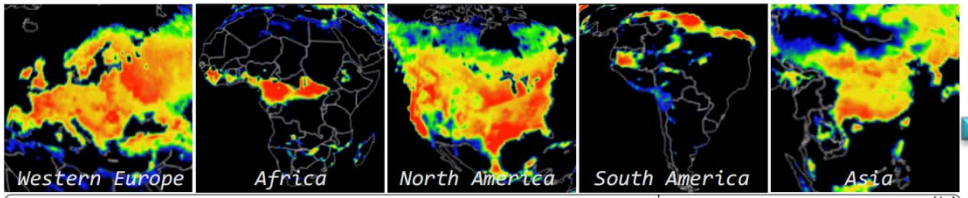
- 卫星观测全球碳排放数据



```
//defineROI (...); // define all ROIs using parameter assignment
load {    map("blackMap");      data("carbonEmission"); }; // load map-based
// global carbon  emission data, using the default colormapping scheme "blue-yellow-red"
rotate {axis=Y, angle=global, duration=36 seconds};
```

(a)

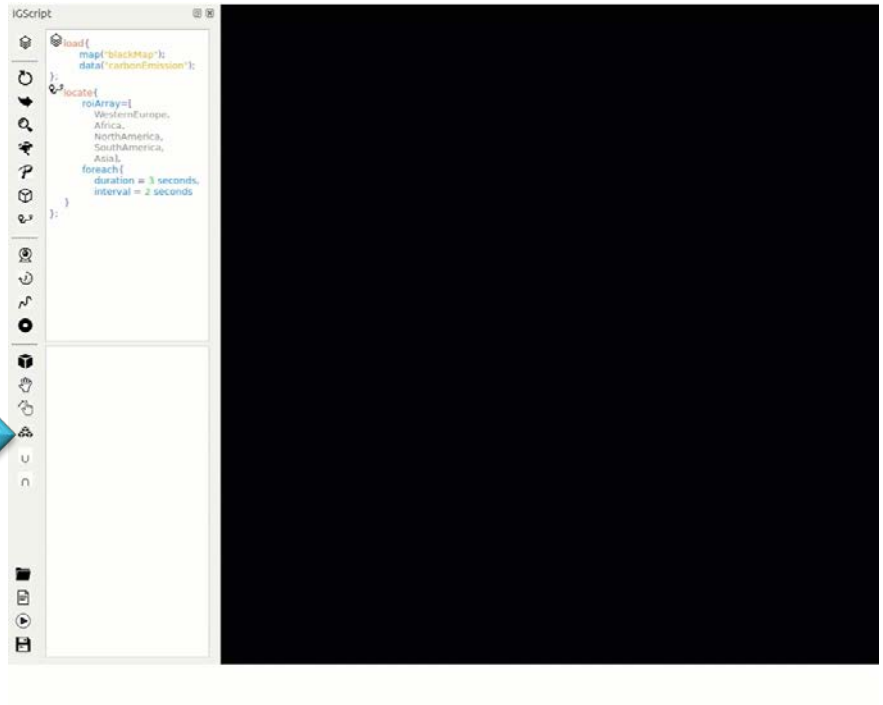
A 36-sec animation by the statement *rotate*




```
//defineROI (...); // define all ROIs using parameter assignment module via vis.
locate {          // space and coding space (all ROIs are defined only once)
roiArray = string["WesternEurope", "Africa",  "NorthAmerica", "SouthAmerica", "Asia"],
foreach {duration=3 seconds,
         interval=5 seconds}
}; // end of locate
```

(b)

An animation with transitional scene switching across ROIs

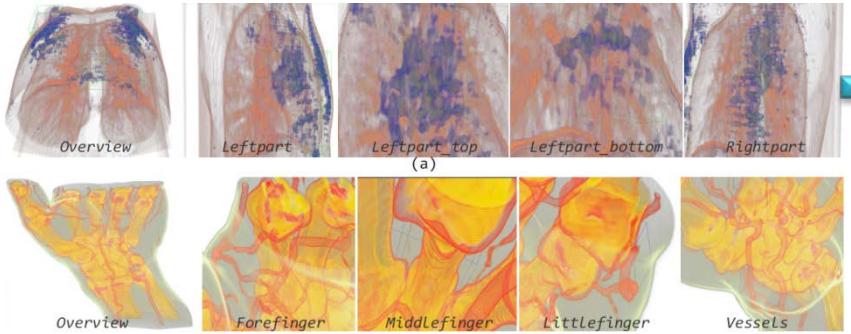


Case Study 02: 3-D标量场体数据

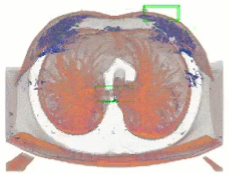
- CT扫描的医学切片数据堆叠成三维体：LUNG (左下角上图) 和HAND (左下角下图)
- 脚本驱动的数据演示动画会在不同ROI之间切换，并给每个ROI特写镜头，切换时间，特写时间均可定制
- ROI之间镜头切换仿照了纪录片拍摄时的手法：(1) 渐变切换；(2) 根据ROI区域大小，在旋转时自动缩放

```
//defineROI (...);
load { volumeData("human_lung");
rgbaScheme("rgba_lesions"); };
locate {
   roiArray=["Overview", "Leftpart",
       "Leftpart_top", "Leftpart_bottom",
       "Rightpart"],
   foreach {
       duration=25 seconds,
       interval=2 seconds }
}; // end of locate
```

```
load { volumeData("hand");
rgbaScheme("boneSkinVessel");};
locate {
   roiArray=["Overview", "Forefinger",
       "Middlefinger", "Littlefinger",
       "Vessels"],
   foreach { duration=20 seconds,
       interval=2 seconds
   }
}; // end of locate
```

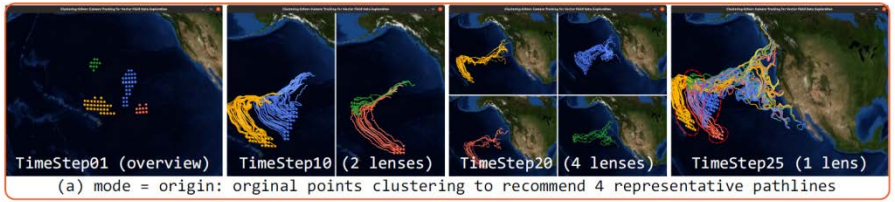


Snapshots of overview-to-details animations



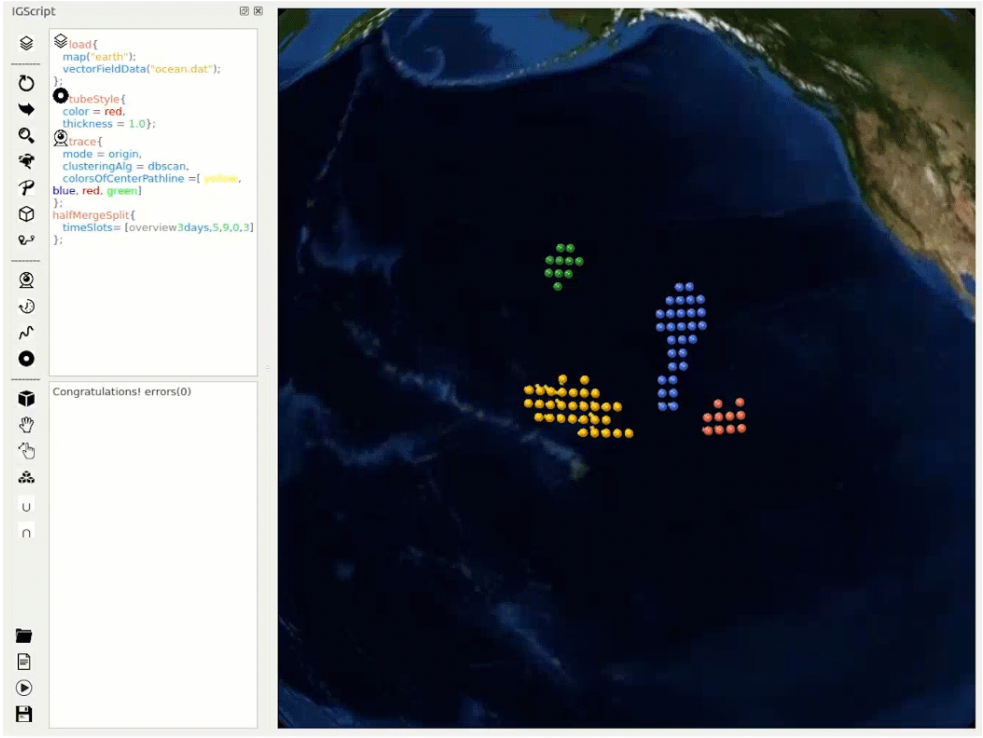
Case Study 03: 时变全球洋流流场数据

- 聚类驱动：获取迹线不同的簇 (聚类个数为动画构建的虚拟摄像头个数)，并得到每个类的“中心迹线”(虚拟摄像头自动跟踪拍摄)
- OD模式：可以以origin模式 (左上图) 或destination模式 (左下图) 生成自动探索动画
- 允许用户定制：在哪些time step进行镜头拆分 (不断二分往下走)、镜头合并；线/管风格；迹线起始时间步 (lifeTime)；迹线运动时的动画速度 (animate)

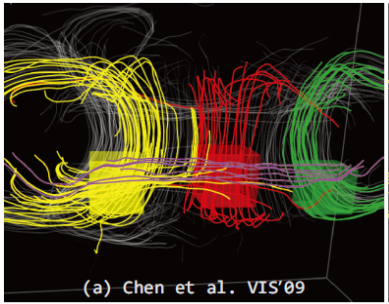


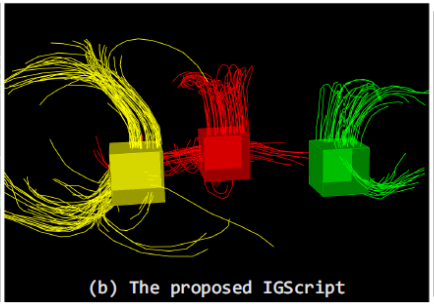

```
load {vectorFieldData("ocean.dat")};
animate {speed=moderate};
trace { mode=origin, //or destination
    clusteringAlg=[pca, dbscan],
    lifeTime = 20 days,
    colorsOfCenterPathline
      = [yellow, blue, red, green]
}; //4 colors means 4 lenses (max)
halfMergeSplit {
    timeSlots
      =[overview3days,5,9,0,3] };
//manually cyclic camera splitting
```

Animation snapshots with four time-steps (#01, #10, #20, #25) for the ocean data



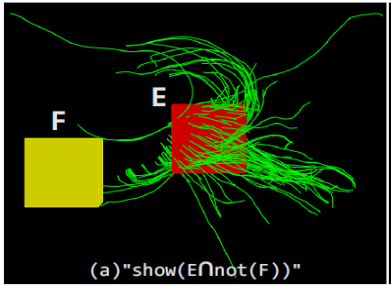
Case Study 04: DTI数据

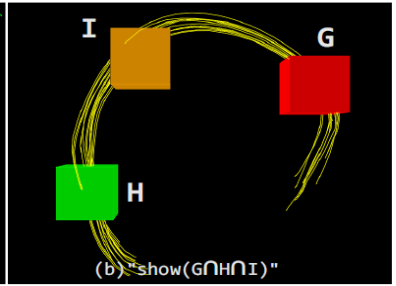


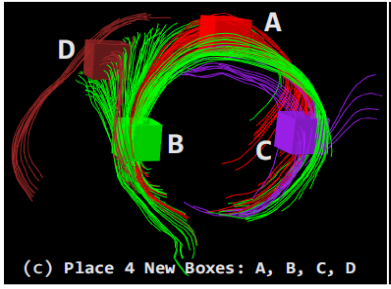


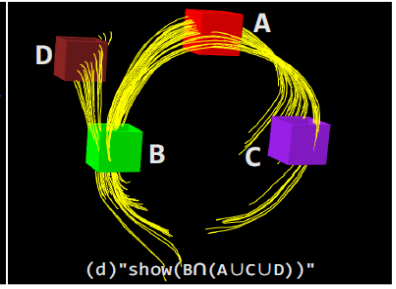

```
// define all ROIs in parameter assignment module via visualization
// space and coding space, which will be invoked only once for each ROI
// defineROI (...); // defined ROIs are global variables

load {dtiData("human_brain01")};

lineStyle {color=green, width=1.0};

placeANewBox {boxID("A"), at("ROI#HB11"), color=green, alpha=0.5};
placeANewBox {boxID("B"), at("ROI#HB12"), color=red, alpha=0.6};
placeANewBox {boxID("C"), at("ROI#HB13"), color=yellow, alpha=0.6};
```

勿需修改任何渲染代码即可快速重现浙大陈为老师09年的VIS工作 [14] (左图a)










```
// defineROI (...); // defined ROIs are global variables
load {dtiData("human_brain02")};
placeANewBox {boxID("E"), at("ROI#HB21"), color=red, alpha=0.8};
placeANewBox {boxID("F"), at("ROI#HB22"), color=yellow, alpha=0.8}; (a)
with (color=green,alpha=1.0) {
   show(E∩not(F));          }; // end of with
```

```
load {dtiData("pig_heart")};
placeANewBox {boxID("G"),at("ROI#PH11"), color=red,alpha=0.8};
placeANewBox {boxID("H"),at("ROI#PH12"), color=green,alpha=0.8};
placeANewBox {boxID("I"),at("ROI#PH13"), color=orange,alpha=0.8}; (b)
with(color=yellow,alpha=1.0) {
   show(G∩H∩I); }; // end of with
```

```
load {dtiData("pig_heart")};
lineStyle {color=red, width=1.0};
placeANewBox {boxID("A"), at("ROI#PH14"), color=red,alpha=0.8};      (c)
placeANewBox {boxID("B"), at("ROI#PH15"), color=green,alpha=0.8};
placeANewBox {boxID("C"), at("ROI#PH16"), color=purple,alpha=0.8};
placeANewBox {boxID("D"), at("ROI#PH17"), color=brown,alpha=0.6};
```

```
load {dtiData("pig_heart")};
with (color=yellow,alpha=1.0) {
   show(B∩(A∪C∪D));
   pause(5 seconds); // after 5-sec pause,
   // show(...);    // then show other queries
}; // end of with
```

$show(B \cap (A \cup C \cup D))$

基于集合的交并补非、及使用括号表示优先级的复杂查询

Discussion and Future Work

- 数据类型的选择：主要参考的是Charles Hansen 和 Chris R. Johnson “The Visualization Handbook” [28]) 中总结的四类最为常见的科学数据
  - 2-D/3D 标量场数据 (0th order tensors)
  - 向量场 数据 (1st order tensors)
  - 张量场及其衍生的数据 (higher order tensors)
- 数据类型的可扩展性：scalability on data types
  - 只要是属于最为常见的这四类科学数据，*IGScript*都适用
  - 针对除这四大常见科学数据外的其他数据：需要实现其数据加载APIs，以及针对该类型数据的一些特定应用的APIs (application-specific functions)，并加到*IGScript*的支持库里
- 将来工作：自动推荐ROI
  - 机器学习方法自动推荐ROI
  - 根据根据眼动仪等收集用户可能的ROIs
- 领域专家反馈
  - 提供更多的sample codes让用户能快速上手，或者套用
  - 定义更多专业术语作为语法关键字，易于理解与使用
  - 使用可视编程方法或GUI界面辅助用户快速构建*IGScript*的代码

Conclusions

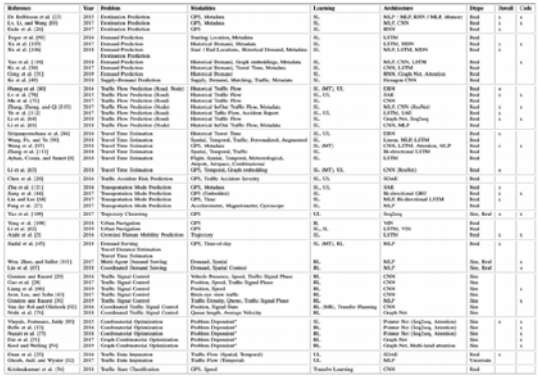
- **该工作设计并实现了一个基于文本式交互语法的数据驱动演示动画构建工具 *IGScript***
- **设计了一个双联动视图：可视化空间与代码空间**
  - 在调参、动画风格定制、ROI定义时提供实时的可视反馈和可视操纵
- **实现上构建了专用编译器与反编译器 (代码生成器)**
  - 编译器：将用户在**代码空间**构建的代码编译成复杂计算与GPU渲染的代码
  - 反编译器 (代码生成器)：将用户在**可视化空间**的动画定制操作自动生成*IGScript*的代码
- ***IGScript*代码: 仿照自然语言的语法结构，尽量使用不同领域的专业术语**
  - 动画的高效编辑 (cp & paste & cut & add & delete & append & switch & clip loop)：通过编辑文本来编辑动画片断/定制动画风格
  - 方便复用、共享：传统方法通过录制的演示视频太大、且不可编辑

第二个工作简要介绍

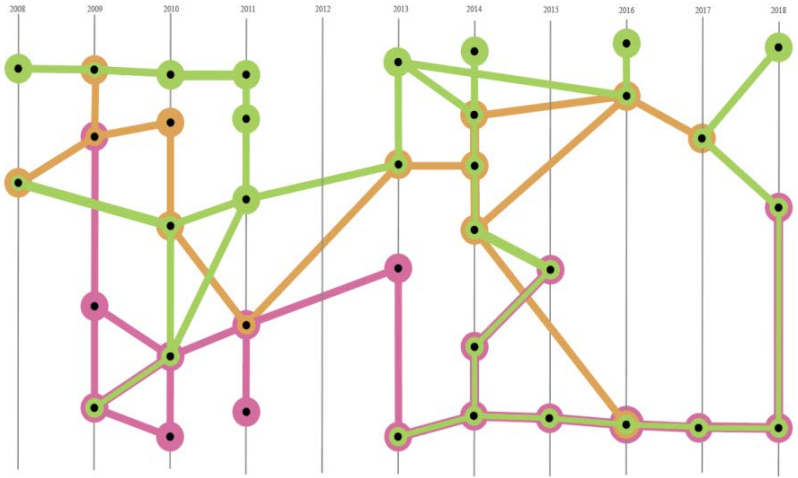
- Xiaohan Wang, Chuyu Zhang, Yu Zhu, Xueyi Chen, Liming Shen, **Richen Liu***, and Rongtao Qian.
  Hybrid Line-Based and Region-Based Interactive Set Data Visualization.
  CHI Conference on Human Factors in Computing Systems (ACM CHI'21 LBW) ,
  Article No.: 411, pages 1-7, Yokohama, Japan, May 8-13, 2021.
  (**Late-Breaking Work：LBW**, 双栏7页+2页附录的poster track论文)

Motivation

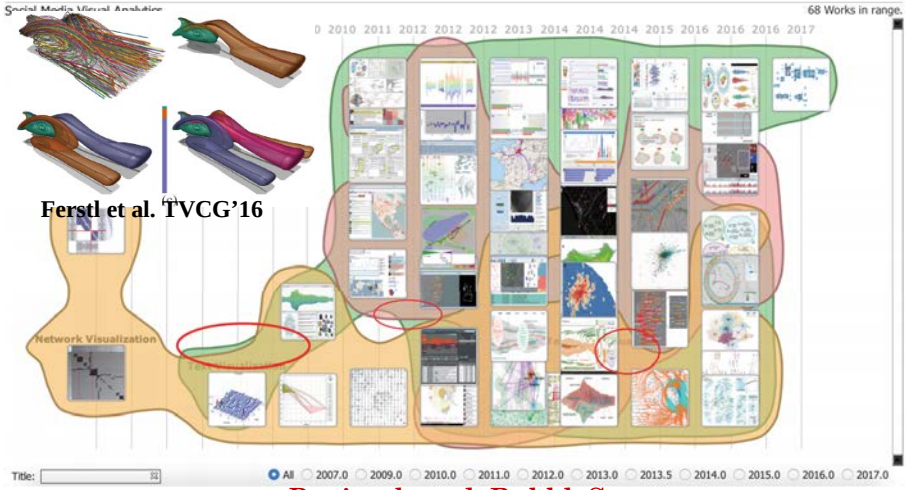
- **集合数据 (set data) 可视化一直具有很大的挑战**
  - 难以一次性将所有集合关系展现出来
  - **集合关系复杂**：一对多，多对一，多对多。如文献数据一篇论文可以用到多个方法 (PCA，DBSCAN)，一个方法 (例如CNN) 可以被多篇论文共同使用
- **现有的方法的不足:**
  - **表格法**：很难感知多对多的集合关系
  - **基于线的方法 (line-based):** 无法得到每个节点的语义信息，如文献数据中每篇论文的代表性图片
  - **基于区域的方法 (region-based):** 可扩展性差，视觉遮挡严重等等



Traditional: Tabular



Line-based: Kelp Diagram

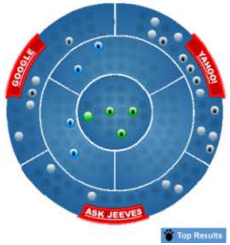


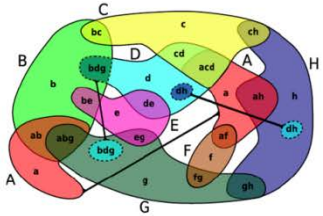
Ferstl et al. TVCG'16

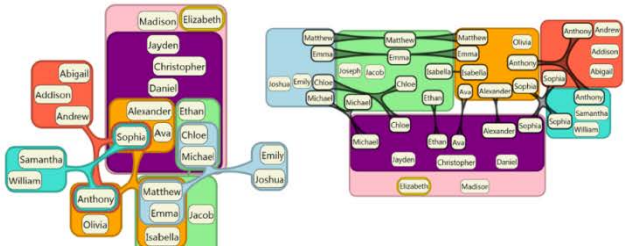
Region-based: BubbleSets

背景与相关工作

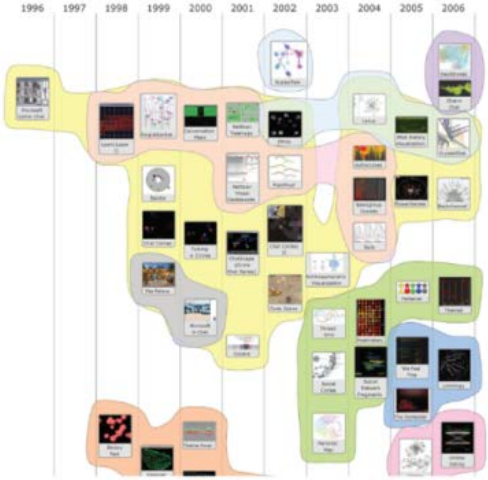
- 相关工作
  - **Euler and Venn** diagrams
  - 基于区域的方法
  - 基于线的方法

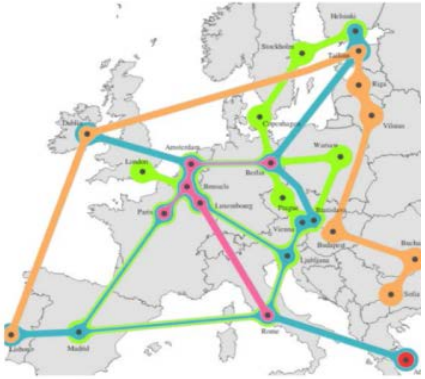






欧拉图的多种形式



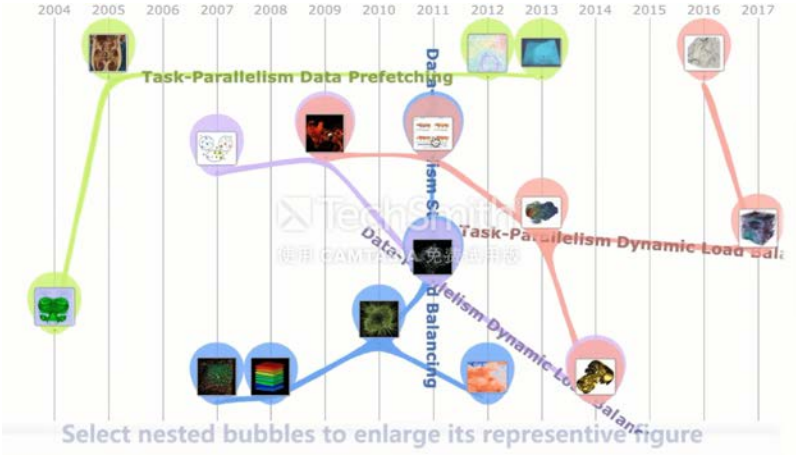


Region-based overlays： BubbleSets的timeline应用 (左) 和地图应用 (右)

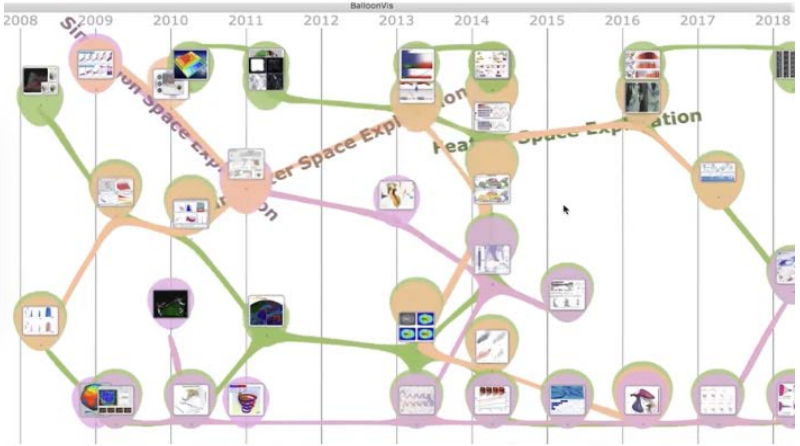
Line-based overlays

设计思路与方法

- 气球的隐喻：嵌套气球表示一个节点有多个属性，气球连线表示节点间共享属性
- 设计思路
  - 混合策略：基于线与基于区域的混合，既减少空间，又可使用代表性图片表示节点语义或上下文信息
  - 将多对多的集合关系清晰表现出来，哪些论文使用到相同方法，一篇论文使用到多个
- 优化目标
  - 减少交叉：线球交叉，线线交叉
  - 充分利用空间：减少空白区域
  - 保持原有拓扑结构的情况下尽量减少视觉遮挡
- 方法：基于能量最小化的优化，不断迭代，直到算法收敛



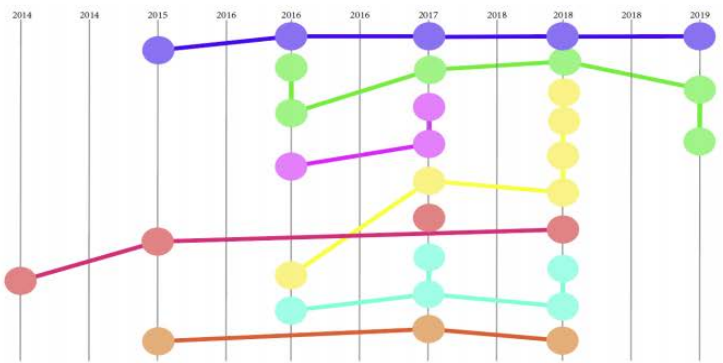
Show representative image



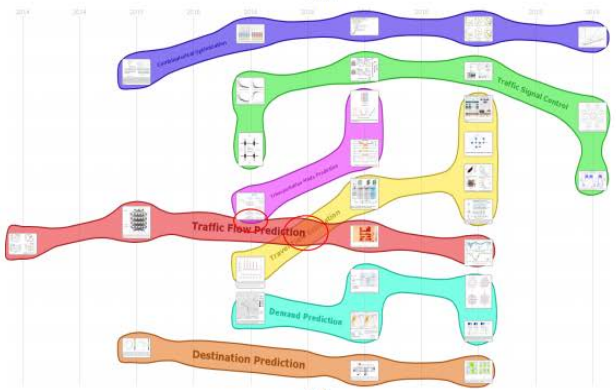
Drag items to re-layout

Case 01: 一篇深度学习方面的综述文献数据

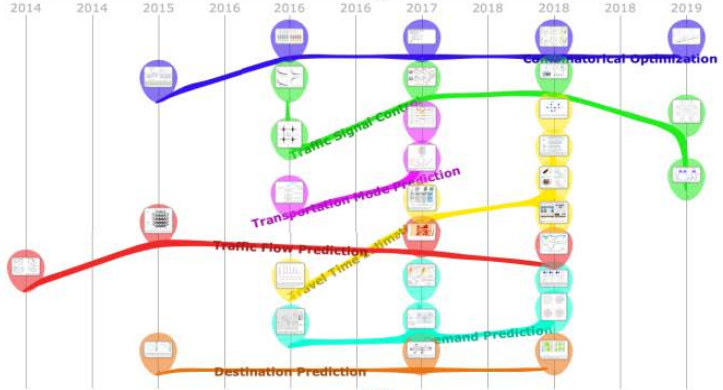
(a)



(b)



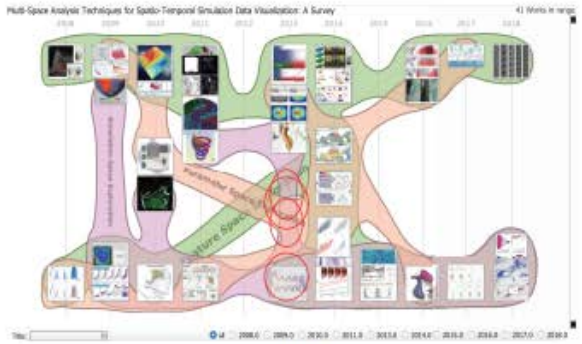
(c)



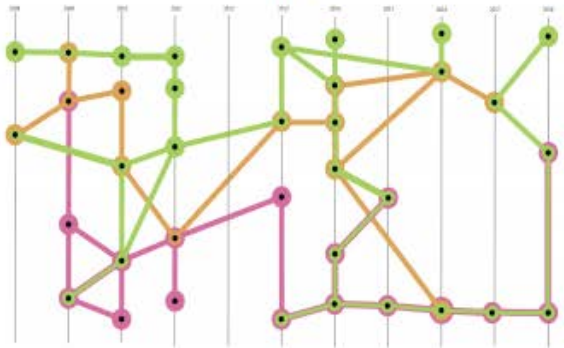
(d)

32篇文献，7个不同类别： A survey paper “emerging trends of deep learning methods” [Veres et al. TITS’19]

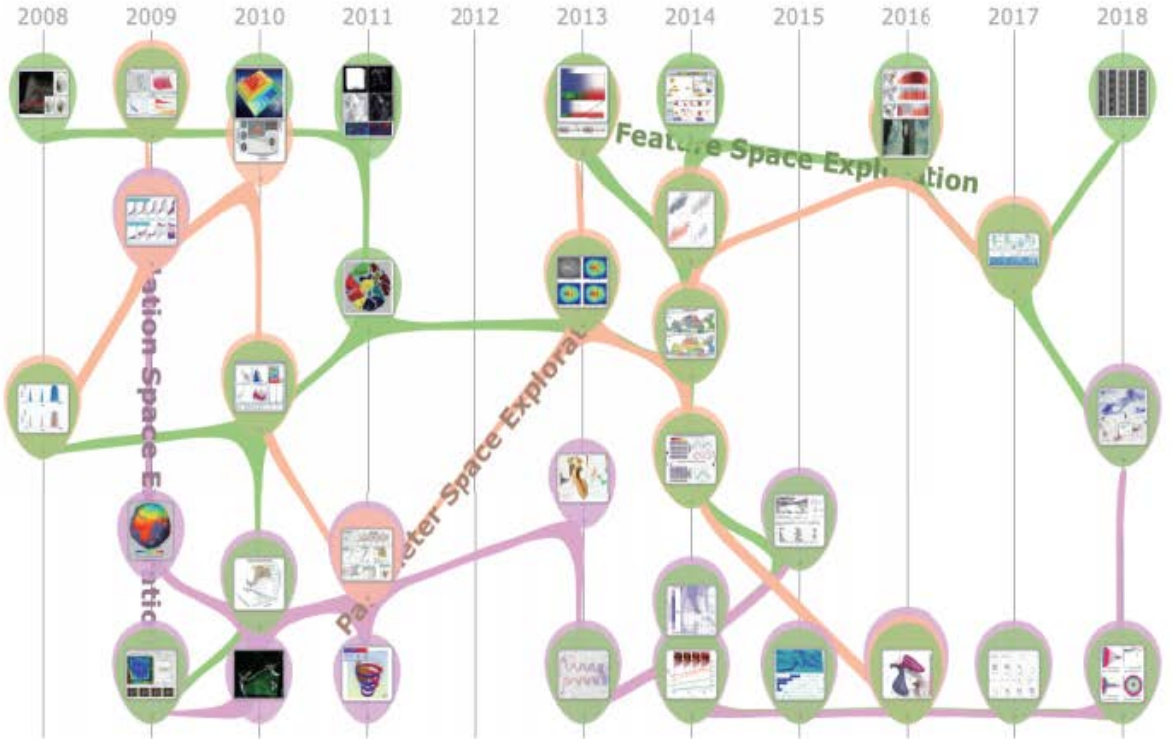
Case 02: 一篇集合模拟可视化方面的综述文献数据



(a)



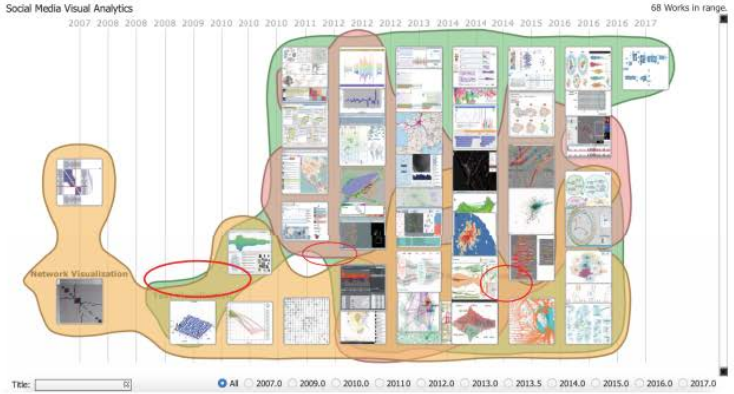
(b)



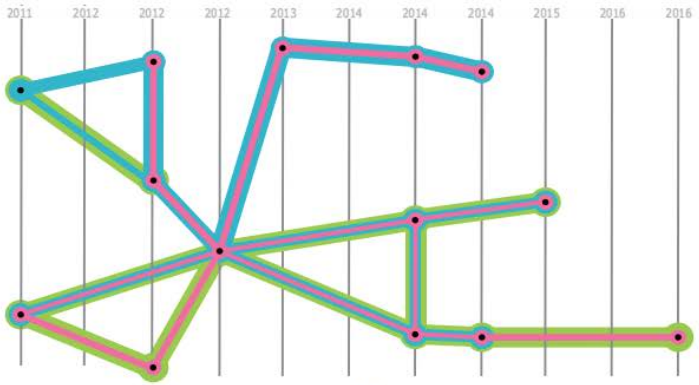
(c)

41篇文献，3个不同类别： A survey paper on “Simulation data visualization” [Chen et al. Visual Informatics’19]

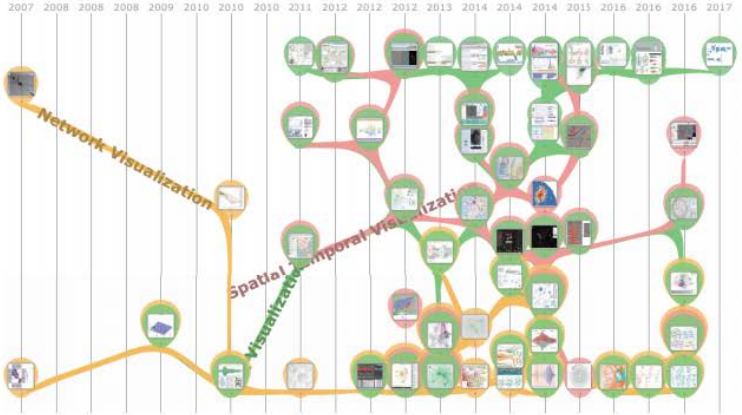
Case 03: 社交媒体数据可视化方面的综述文献数据



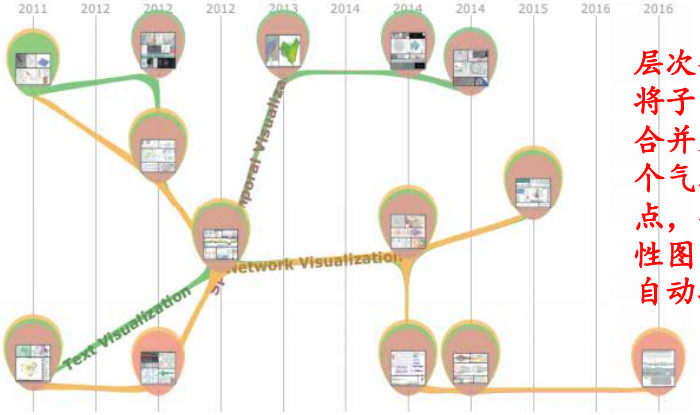
(a)



(b)



(c)



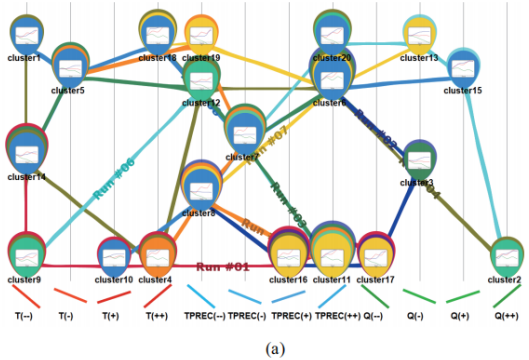
(d)

层次合并：将子类别合并成一个气球节点，代表性图片也自动拼合

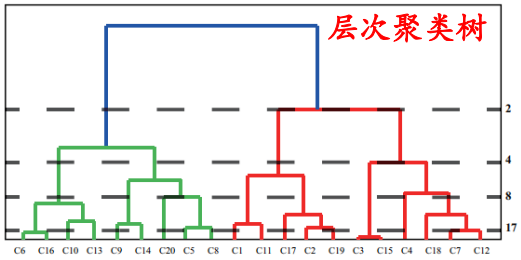
68篇文献，3个不同类别： A survey on “Social media data visualization” [Chen et al. EuroVis STAR’17 (CGF)]

Case 04: 集合模拟数据

GEOS-5

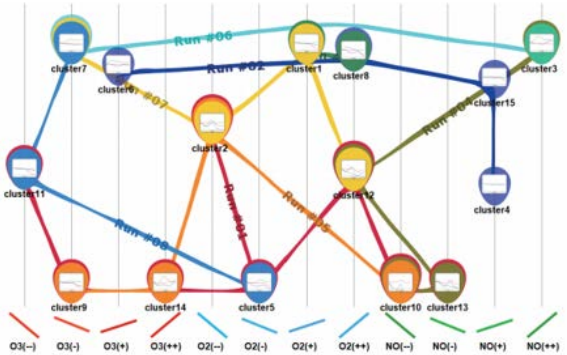


(a)



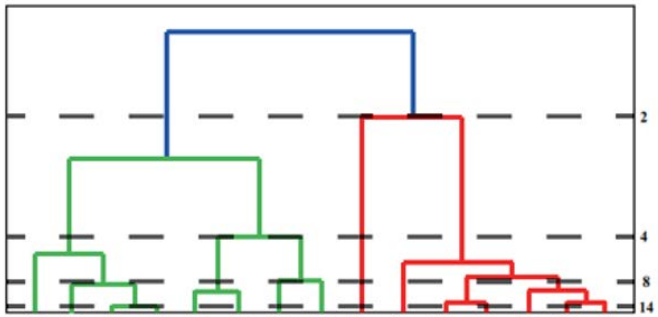
(c)

- 两个集合模拟数据
  - Goddard Earth Observing System Model, Version 5 (GEOS-5)
  - Model of Ozone and Related Tracers, Version 4 (MOZART-4)
- 将集合模拟数据中的成员 **(simulation run)** 与聚类 **(cluster)** 之间的多对多集合关系可视化



(b)

层次聚类树



(d)

MOZART-4

Demo of BalloonVis

论文相关展示下载地址：https://dabigtou.github.io/richenliu/

Thank You!