Data Visualization CSE3020

Prof. Ramesh Ragala

January 18, 2018

Visualization pipeline

- Visualization Application
  - Conceptual perspective
  - Practical Perspective
- The conceptual description of the structure of visualization applications includes:
  - Data Importing
  - Data Filtering and enrichment
  - Data Mapping
  - Data Rendering
- The role of visualization is to create images that convey various types of insight into a given process

- Visualization Pipe:

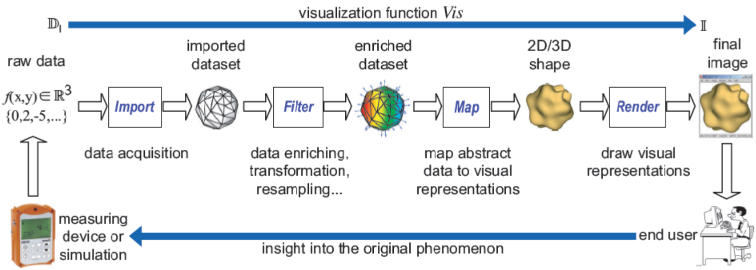

Figure Functional view on the visualization pipeline.

Data Acquisition

- The Visualization Process consists of sequence of steps or operations, that manipulate the data produced by the process under study and ultimately deliver the desired images.
- Divide and conquer strategy in designing process of data visualization $\rightarrow$ manage the complexity of the whole process.
- The visualization process can be treated as a pipeline $\rightarrow$ consisting of several stages $\rightarrow$ each stage modelled by specific data transformation operation.
- Input data $\rightarrow$ flows through many stages in pipeline $\rightarrow$ output image
- the sequence of data transformations that take place in the visualization process is often called the visualization pipeline.

Visualization Pipeline

- The Visualization Pipeline has four stages:
  - Data Importing
  - Data Filtering and enrichment
  - Data Mapping
  - Data Rendering
- Conceptual representation of visualization pipeline:
  - $\mathbf{V}_{is} : \mathbf{D}_i \rightarrow \mathbf{I}$
  - D_i the all possible types of raw input data
  - $I \rightarrow$ set of produced images
- **Insight function:**A process, which maps from the output image to the input data $\rightarrow$ inverse direction to the V_i function.
  - **Insight : $\mathbf{I} \rightarrow \mathbf{D}_i$**
- insight function $\rightarrow$ mapping from images to raw data
- Insight maps from the produced images to the actual questions the user has about the raw data, which are not necessarily one-to-one with the data itself.

- When visually monitoring a live process that generate data, the user may want to steer the process on a given path by changing its parameters.
- i.e loop between the application input and visualization output. $\rightarrow$ if the complete round trip is executed in seconds $\rightarrow$ the user effectively steers the process at hand by means of visual feedback $\rightarrow$ **Computational Steering**.
- some of visualization software, which has computational steering are:
  - SciRUN, CUMULVS and CSE
- For, real-world complex application, we are not able to construct the desired visualization function in single step. it is not easy.
- So the combination of sub-functions in each step in visualization pipeline gives the desired visualization V_{is}.

Visualization pipeline

- simple visualization procedure for complex applications

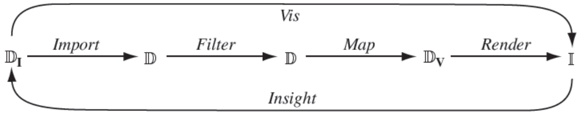

Figure The visualization process seen as a composition of functions.

- some of most important factors in complex visualization of real time scenario for qualitative and quantitative are:
  - Domain Specific Knowledge of user
  - The procedure used in visualization pipeline
  - Type of user interaction $\rightarrow$ interaction of various parameters

- **Data Import:**
  - Identify the representation of original information
    - Continuous
    - Discrete
  - Most probably, the datasets are of type, Uniform, Structured, Unstructured and rectilinear.
  - The function for importing data is: **Import : $\mathbf{D}_I \rightarrow \mathbf{D}$**
  - Practically, importing data means choosing a specific dataset implementation and converting the original information D_I to the representation implied by the chosen dataset type D.
  - Import is a One - to - One mapping $\rightarrow$ reading the input from
    - from External Storage: files or database
    - from Live streams : Measuring devices, scanner, analog-to-digital converter.
  - data importing can imply the translating of
    - different data storage formats
    - resampling the data from the continuous to the discrete domain
    - one resolution to another **Example:** Plot on continuous data

Visualization Pipeline

- The choice made during data import determines:
  - Quality of resulting images
  - Effectiveness of Visualization
- if the import data is incorrect, incomplete, uncertain, low quality $\rightarrow$ restoring the data quality later is difficult
- more information has to be given in data import step.

- **Data Filtering and Enrichment:**
  - decides which are the important feature
  - Imported data is not one - to - one in many cases.
  - Distill the raw dataset into more appropriate representation → Enriched Datasets
  - Enriched datasets → Encode the features of interest in more appropriate form for visualization and analysis. → process called **Data filtering or Data Enriching**
  - simply
    - Data is filtered to extract relevant information
    - Data is enriched with high level information that supports a given task.
  - The function representation of data filtering:
    - **Filter : D → D**
  - Here input and output of filtering function are datasets. → filtering is a data manipulation function.

- Data filtering is necessary in many situations:
  - See what is relevant:
    - **subset of interest** that is relevant for a task.
    - subset of interest → selection can be done in spatial domain or attribute value domain or combination of both.
  - Handle large data:
    - If the data is large → efficient processing is difficult → for interactive visualization applications.
    - Example: Very Big Visualizing 2D dataset which has color attributes specified for cell.
    - If the input datasets exceed a certain size, ???
    - Solution - 1 is **Zooming**: → subsampling of input image → displays a only a subset of its pixels that captures the overall characteristics of the complete dataset
    - Solution - 2: **Panning**: Selecting the subset of the input images at its original resolution.
  - Ease of Use:
    - datasets are transformed from one form to another form, which fits best for visualization.

- **Data Mapping:**
  - Filtering operation produces enriched data $\rightarrow$ directly represents the features of interest for exploration
  - In this data mapping, the task is $\rightarrow$ map the features of interest to the visual domain $\rightarrow$ associating elements of visual domain with data elements in enriched data.
  - The visual domains consists of visual features
  - The mathematical function for this mapping process is
    - **Mapping : $\mathbf{D} \rightarrow \mathbf{D}_V$**
  - The Mapping mainly depends upon,
    - Application
    - Purpose of Visualization
    - Designer's preferences for that visualization

- **Data Mapping:**
  - Reasons for data mapping:
    - Mapping encodes explicit design decisions about what and how, we want to visualize
    - Mapping typically converts "invisible" data to "visible" representations.
    - **Mapping → the visible attributes that encodes actual data**
    - **Rendering → the remaining visual attributes that users can tune according to 3D scene**
    - **simply rendering → 2D or 3D visualization**
    - Modularity → Modularise the visualization pipeline
  - Desirable mapping properties:
    - Data mapping is arguably the visualization pipeline step that is most characteristic for the visualization process.
    - **Importing and Filtering data** → resampling, projection, restriction and many other signal processing operations.
    - **Rendering**→ applying computer graphics techniques → coordinate transformation, lighting, texture mapping and rasterization.

- **Data mapping:** targets $\rightarrow$ **invisible data to visible data** and **multidimensional to low-dimensional**
- To do this Data mapping function should satisfy several desirable properties.
- Data mapping should follow the **injective property**
- Different Values $\rightarrow$ should map different visual variables
- Different values $x_1 \neq x_2 \in$ D (dataset) to be visualized should be mapped to different visual attribute values $Map(x_1) \neq Map(x_2)$ in the visual feature dataset D_V
- This property is essential for the design of an effective visualization.
- After the feedback given by user $\rightarrow$ *invert function* $\rightarrow$ to adjust the data inorder to produce more effective visualization.
- **Filter** and **Import** functions are may not be invertible practically. $\rightarrow$ **Map** and **Render** are invertible.

Visualization Pipeline

- **Inverting the mapping:**
  - Map invertible is not enough in mathematical view
  - We must know how, and be able, to do the inversion mentally when we look at the pictures
  - We must know the significance of the visual attributes used in the rendering.
  - we must know how color, shape, position, texture, and the other visual attributes used in the mapping relate to data attributes of interest.
  - This knowledge can be implicit, encoded in well-established conventions that are assumed to be known by all our users.
  - weather maps $\rightarrow$ wind speed and direction, sun intensity, type and strength of rain fall. $\rightarrow$ uses different icons.
  - cartographic maps $\rightarrow$ North direction on top of the map
  - shades of blue $\rightarrow$ water, lakes ...
  - Color legend $\rightarrow$ color mapping to values $\rightarrow$ color-to-value inverse mapping