The Information Mural: A Technique for Displaying and Navigating Large Information Spaces

Dean Jerding and John Stasko

Haozhen Zhao
iSchool@Drexel
Outline

• Introduction
• Problem
• Algorithms
• Implementation
• Limitations
• Applications
Introduction

• Being able to see some representation of the entire information space provides an initial gestalt overview and gives context to support browsing and search tasks.

• However, the limited number of pixels on the screen constrain the information bandwidth and make it difficult to completely display large information spaces.

• The Information Mural
  – a two-dimensional, reduced representation of an entire information space that fits entirely within a display window or screen.
  – creates a miniature version of the information space using visual attributes, such as gray-scale shading, intensity, color, and pixel size, along with antialiased compression techniques.
Problem

• **Aliasing**
  – Display a visual representation of a large information space whose resolution is $M \times N$ pixels, on a screen in $X \times Y$ pixels. ($M, N >> X, Y$)
  – Overwriting pixels that happen to overlap.

• Information Mural technique is to make the density of overlap visually apparent.
Original Algorithm

1. for each \( x, y \) set \( \text{mural}_\text{array}[y][x] \) to zero
2. for each pixel \( m, n \) in the original representation
   a. compute \( x = m / M * X, y = n / N * Y \)
   b. compute the area of the quadrants defined by the point \( x, y \) and a unit square connecting each of surrounding pixels (\( \text{floor}(x), \text{floor}(y) \); \( \text{floor}(x), \text{ceil}(y) \); \( \text{ceil}(x), \text{floor}(y) \); \( \text{ceil}(x), \text{ceil}(y) \))
   c. add the area of the diagonally opposite quadrant to each \( \text{mural}_\text{array} \) entry:
      \[
      \begin{align*}
      \text{mural}_\text{array}[\text{floor}(y)][\text{floor}(x)]
      
      \text{mural}_\text{array}[\text{ceil}(y)][\text{floor}(x)]
      
      \text{mural}_\text{array}[\text{floor}(y)][\text{ceil}(x)]
      
      \text{mural}_\text{array}[\text{ceil}(y)][\text{ceil}(x)]
      \end{align*}
      \]
   d. update \text{max}_\text{mural}_\text{array}_\text{value} \text{ if one of the four new } \text{mural}_\text{array}[][] \text{ values is a new maximum}
3. for each \( x, y \) in the \( \text{mural}_\text{array} \)
   a. map the value \( \text{mural}_\text{array}[y][x] / \text{max}_\text{mural}_\text{array}_\text{value} \) to a gray-scale or color intensity varying scale, depending on the type of Mural being created
   b. shade the pixel at \( x, y \) of the Mural based on the mapping computed in the previous step
Efficient Algorithm

1. for each \( x, y \) set \( mural_array[y][x] \) to zero
2. for each pixel \( m, n \) in the original representation
   a. compute \( x = m / M \times X, y = n / N \times Y \)
   b. add 1.0 to \( mural_array[floor(y)][floor(x)] \)
   c. update \( max_mural_array_value \) if the new \( mural_array[floor(y)][floor(x)] \) is greater than the existing maximum
3. for each \( x, y \) in the \( mural_array \)
   a. map the value \( mural_array[y][x] / max_mural_array_value \) to a gray-scale or color intensity varying scale, depending on the type of Mural being created
   b. shade the pixel at \( x, y \) of the Mural based on the mapping computed in the previous step
Attribute Color Algorithm

• Color each screen pixel according to the attribute color that occurs most frequently at that pixel in the Mural.

• Treat the intensity at each pixel independently from the attribute colors.
Implementation

- Implemented as an abstract widget that can be used by an application.
  - a global view for more detailed views by providing a “navigation rectangle” that can be panned and zoomed by the user.

- Built in C++ on top of X Windows and Motif and Vz visualization framework
  - Redraw: calls mural primitives.
  - Navigation: MuralValueChangeCB(), MuralZoomCB().
Limitations

• Use of gray-scale shading or density
  – low in ordering of elementary graphical perception tasks

• Potential addition of color
  – color is better suited for portraying categorical data, rather than continuous values.

• Context of color
  – pixel-level detail, it can be difficult to notice a single yellow pixel in a sea of gray.

• Computation expensive when re-display.
Applications

- Browsing or navigating through information spaces with the aid of a global overview.

Execution Mural view of bubble sort algorithm animation built using the Polka animation toolkit.
\subsection{Our Solution}

We have developed a method for displaying and navigating large information spaces using the multiple view technique. This work is derived from our implementation of the message trace visualization.

As Pilehvar, Jard, and Shneiderman propose three important considerations in the design of multiple-view browsers: window-placement strategy, view coordination, and the global view itself. The next section describes the design of the message view and the navigation mechanisms. We call our views of large information spaces "Information Murals," and describe them in Section 3. In Section 4 we discuss several application areas where the information murals are useful, and compare our method with related work in those areas.

\section{The Execution Mural}

As mentioned in the first section, one area of our \cite{Pil93} software visualization research involves visualizing the execution of object-oriented programs. As a component of an integrated tool, we are designing a display of the messages exchanged between objects during the execution of a \texttt{C++} program. This section describes the "Information Execution Mural," focusing specifically on the visual mechanisms used to provide navigation capabilities. While the current state of the design does not contain all the functionality needed for this view, it does provide an effective framework for \texttt{C++} methods. The techniques discussed in this section will be generalized to other information spaces in Section 3.
App2: Discovering attributes of or relationships within multidimensional data from a visualization.

View of sun spots showing focus area and mural of entire data set at the bottom.
App 3: Studying geographic or spatial data to understand its characteristics.

Mural of population density distribution, using data from the 1990 census. (Choropleth map)
• Thanks!