Fisheye Views  
G. Furnas, CHI 86

```c
int i;  
for(i=0;i<n;i++)
    t[i] = s[i];

int x;  
for(i=0;i<k;i++)
    t[i] = (t[i] + 10000) / x[i];

int y;  
for(i=0;i<k;i++)
    t[i] = (int) (1 - t[i]/10000);  

int z;  
for(i=0;i<k;i++)
    t[i] = t[i-1] % 10000;

if(!s0[i])
    t[i] = t[i-1];

if(t[i] == 0)
    t[i] = 0;

for(i=n-1;i>0;i--)
    if(t[i] < 0)
        t[i] = 0;
```

Conventional view of C program
Fisheye Views G. Furnas, CHI 86

```c
#define DIG 40
#endif

int c, i, x[DIG/4], t[DIG/4], k = DIG/4, noproject = 0;

while((c = getchar()) != EOF) {
    if (c >= '0' && c <= '9') {
        switch(c) {
            case 'e':
                case 'E':
                    for (i = 0; i < k; i++) t[i] = x[i];
                    break;
            case 'q':
                default:
                }
                if (!noproject) {
                    }
                    noproject = 0;
                    }
                    }
                    }
                    }

```

First-order fisheye view of C program (same number of lines, redrawn using compaction)

---

Fisheye Views G. Furnas, CHI 86

- "Code folding" (e.g., Flash Builder)
  - Manual or automatic
  - Based on syntax, indentation, tokens

Unfolded: Click on (-) to fold

Folded: Click on (+) to unfold

Folded: Mouse over to expand temporarily

First-order fisheye (underlined code) vs. conventional view (boxed code)
Fisheye Menus

- Apple OS X dock with “Magnification” enabled
  - But, remember Fitts’s Law!

Data Types: 2D Spatial

- Inherently spatial data
  - Maps
Data Types: 2D Spatial

- Use of distortion viewing to provide “focus+context”

Simple magnification loses context

Robertson and Mackinlay, UIST 93

Bederson et al., TOCHI 04

Data Types: 2D Spatial

Inherently spatial data: Distortion viewing applied to a map

http://holisticsofa.com/category/visualization/page/3/

http://www.cambooth.net/washington-metro-diagram-my-last-word/
Data Types: 2D Spatial

- Abstract data
  - Need to select bindings to XY coordinates
  - Can use semantic zoom
    - Zooming (magnification/minification) that changes the representation (e.g., shape, format, level of detail) instead of or in addition to geometric scale

---

Data Types: 2D Spatial

- Using image deformation to explore visual data
  - J. Brosz et al., Transmogrification: Casual Manipulation of Visualizations, UIST 2013
    http://www.youtube.com/watch?v=S1Roi2N0mx8
Data Types: Multidimensional

- Abstract data
  - Parallel coordinates (A. Inselberg)
    - $N$ variables represented by $N$ parallel axes
    - Multivariate point depicted as a polyline connecting vertices on axes
  - Interaction
    - Limit range on axis
    - Scale axis
    - Reorder axes

E.g., [http://www.parallelcoordinates.de/paco/](http://www.parallelcoordinates.de/paco/)

Tasks: Overview

- Panning/scrolling over display
- Separate "overview" display with "you are here" marker
  - E.g., Sublime Text 2
Tasks: Overview

- **Focus+context**
  - Handle overview and zoom tasks in same display
  - Focus items receive greater
    - Magnification and/or
    - Level-of-detail
  - Alternatively, two geometrically registered displays can be used
    - Feiner & Shamash, 91
    - Baudisch et al., 01

Tasks: Zoom

- Current example: Prezi
Tasks: Filter

- Narrow scope by eliminating (or de-emphasizing) "uninteresting" items
- Dynamic queries
  - Widgets specify queries
  - Satisfied interactively as widgets are manipulated
- Examples
  - FilmFinder
  - spotfire.tibco.com
  - bluenile.com
  - Google instant

Tasks: Details-on-demand

- Present additional information when requested
- Examples
  - Separate window/display
  - "Popup"/overlay/"tool tip"
  - Modify rendering of item to increase detail
Tasks: History

- Show historical evolution
- Support undo/redo
- Support creation of variants
- Range of examples
  - Multilevel undo/redo
  - Photoshop history (with history brush)
  - Editable Graphical Histories
  - Automatically generated tutorials

Tasks: History

- Show hypertext browsing history