Data Visualization (CSCI 627/490)

Aggregation & Focus+Context

Dr. David Koop
Overview: Reducing Items & Attributes

Filter

- Items
  - [Original set of items]
  - [Reduced set of items]

- Attributes
  - [Original set of attributes]
  - [Reduced set of attributes]

Aggregate

- Items
  - [Grouped set of items]
  - [Non-grouped set of items]

- Attributes
  - [Grouped set of attributes]
  - [Non-grouped set of attributes]

[Munzner (ill. Maguire), 2014]
Restaurant locations are derived from the New York City Department of Health and Mental Hygiene database. Due to the limitations of the Health Department's database, some restaurants could not be placed.

By JEREMY WHITE

Source: New York City Department of Health and Mental Hygiene

© 2013 The New York Times Company

New York Health Department Restaurant Ratings Map

The New York City Department of Health and Mental Hygiene performs unannounced sanitary inspections of every restaurant at least once per year.

Violation points result in a letter grade, which can be explored in the map below, along with violation descriptions. The information on this map will be updated every two weeks. For menus and reviews by New York Times critics, visit our restaurants guide.

Related Article »

Gracie's Cafe
Grade pending
Violation points
27
Click for details

Gracie's Cafe
Grade pending
Violation points
27
Click for details

Item Filtering on Maps

FIND A RESTAURANT

FIND A LOCATION

FIND A RESTAURANT

FIND A LOCATION

FILTER

All grades

All violations

All cuisines

Item Filtering on Maps

[F. White, New York Times]
Attribute Filtering on Star Plots

(a)  
(b)  
(c)  
(d)  

[Yang et al., 2003]
Aggregation: Histograms

- Very similar to bar charts
- Often shown without space between (continuity)
- Choice of number of bins
  - Important!
  - Viewers may infer different trends based on the layout

[Aggregation: Histograms](#)

Aggregation: Histograms

- Very similar to bar charts
- Often shown without space between (continuity)
- Choice of number of bins
  - Important!
  - Viewers may infer different trends based on the layout

[Aggregation: Histograms](#)

Aggregation: Histograms

- Very similar to bar charts
- Often shown without space between (continuity)
- Choice of number of bins
  - Important!
  - Viewers may infer different trends based on the layout

[Aggregation: Histograms](#)
Bininning

- 2D Histogram is a histogram in 2D encoded using color instead of height
- Hexbin advantages:
  - Bins are more circular so distance to the edge is not as variable
  - More efficient aggregation around the center of the bin
Spatial Aggregation

In cartography, changing the boundaries of the regions used to analyze data can yield dramatically different results.
Boxplots

- Show **distribution**
- Single value (e.g. mean, max, min, quartiles) doesn't convey everything
- Created by John Tukey
- Show **spread** and **skew** of data
- Best for **unimodal** data
- Variations like vase plot for multimodal data
- Aggregation here involves many different marks
Four Distributions, Same Boxplot…

[Image of four histograms labeled Normal, Bimodal, Peaked, Skewed, and a box plot labeled Box plot]
Project Design

• Feedback:
  - Data Manipulation?
  - Questions lead, not technique!
  - Be creative! (interaction too) https://xeno.graphics

• Work on turning your visualization ideas into designs

• Turn in:
  - Two Design Sketches (like sheets 2-4 from 5 Sheet Design)
  - One Bad Design Sketch (like sheets 2-4: here, justify why bad)
  - Progress on Implementation

• Due Friday
Assignment 5

- Map of Citi Bike trips
  - Multiple Views
  - Linked Highlighting
  - Filtering
  - Aggregation

- Due Monday, Nov. 23
Linked Highlighting Example
K-Means

Run

[C. Polis, 2014]
K-Means Issues

D. Koop, CSCI 627/490, Fall 2020

[D. Robinson, 2015]
Dimensionality Reduction

- Attribute Aggregation: Use fewer attributes (dimensions) to represent items
- Combine attributes in a way that is more instructive than examining each individual attribute
- Example: Understanding the language in a collection of books
  - Count the occurrence of each non-common word in each book
  - Huge set of features (attributes), want to represent each with an aggregate feature (e.g. high use of "cowboy", lower use of "city") that allows clustering (e.g. "western")
  - Don't want to have to manually determine such rules
- Techniques: Principle Component Analysis, Multidimensional Scaling family of techniques
Principle Component Analysis (PCA)

Original data space

Gene 1

Gene 2

Gene 3

PC 1

PC 2

Component space

Gene 1

Gene 2

Gene 3

PC 1

PC 2

PCA

[M. Scholz, CC-BY-SA 2.0]
Principal component analysis (PCA) is a technique used to emphasize variation and bring out strong patterns in a dataset. It's often used to make data easy to explore and visualize.

First, consider a dataset in only two dimensions, like (height, weight). This dataset can be plotted as points in a plane. But if we want to tease out variation, PCA finds a new coordinate system in which every point has a new (x, y) value. The axes don’t actually mean anything physical; they're combinations of height and weight called “principal components” that are chosen to give one axes lots of variation.

Drag the points around in the following visualization to see PC coordinate system adjusts.

17 dimensions to 2

For more explanations, visit the Explained Visually and Great Britain, see:

the only of the four countries not on the island of Great Britain. (If you're confused about the differences among England, the UK and alcoholic drinks. It's a good sign that structure we've visualized reflects a big fact of real-world geography: Northern Ireland is

in the table, this makes sense: the Northern Irish eat way more grams of fresh potatoes and way fewer of fresh fruits, cheese, fish

Now, see the first and second principal components, we see Northern Ireland a major outlier. Once we go back and look at the data

Here's the plot of the data along the first principal component. Already we can see something is different about Northern Ireland.

Non-linear Dimensionality Reduction

\[ \Phi_{\text{gen}} : \mathcal{Z} \to \mathcal{X} \]

\[ \Phi_{\text{extr}} : \mathcal{X} \to \mathcal{Z} \]

original data space \( \mathcal{X} \)

component space \( \mathcal{Z} \)

[M. Scholz, CC-BY-SA 2.0]
Dimensionality Reduction in Visualization

5.2.2 Layout Quality

Fig. 8 shows the visual quality, normalized stress, and timing of Glimmer, Hybrid, and PivotMDS layouts on four data sets with known structure. In the case of grid, the correct shape is known. In the other three cases, the correct partitions of the points into clusters are available with these benchmark data sets, so the extent to which the color coding matches the spatial grouping created by an algorithm is a measure of its accuracy.

Qualitatively, with cancer, the Glimmer and PivotMDS algorithms indicate these two color-coded groups clearly with spatial position. Quantitatively, the stress of Glimmer is an order of magnitude lower than that of PivotMDS. Hybrid does separate the two groups but produces misleading subclusters in the orange group.

With shuttle_big, Hybrid produces a readable layout separating the red cluster from the other two but is slower by several hundred percent. Glimmer and PivotMDS both produce useful and qualitatively comparable layouts separating the clusters. The PivotMDS layout is twice as fast but has noticeable occlusion and much higher stress than the Glimmer layout.

The 10,000-point grid is accurately embedded by Glimmer and PivotMDS in comparable times. Hybrid is again slower but nevertheless terminated too soon, suffering from very noticeable qualitative distortion and with a much higher quantitative stress metric compared to that of the other layouts.

The Glimmer layout of the docs data set is qualitatively better than the other three. It shows several spatially distinguishable clusters, color coded by blue, red, orange, and green. The green cluster is split into three parts. It took approximately 2 seconds with normalized stress of 0.157. Hybrid suffers from cluster occlusion. The stress is nearly twice as high as that of Glimmer, and the spatial embedding does not clearly separate any of the given clusters. PivotMDS is very fast but almost completely fails to show
Tasks in Understanding High-Dim. Data

Task 1

What?
- In High-dimensional data
- Out 2D data

Why?
- Derive
- Produce

In
HD data
→
Out
2D data

Task 2

What?
- In 2D data
- Out Scatterplot
- Out Clusters & points

Why?
- Identify
- Discover
- Explore

How?
- Select
- Navigate
- Explore

In
2D data
→
Out
Scatterplot
Clusters & points

Task 3

What?
- In Scatterplot
- In Clusters & points
- Out Labels for clusters

Why?
- Annotate
- Produce

In
Scatterplot
Clusters & points
→
Out
Labels for clusters

Munzner (ill. Maguire), 2014

D. Koop, CSCI 627/490, Fall 2020
Probing Projections

Abstract—We introduce a set of integrated interaction techniques to interpret and interrogate dimensionality-reduced data. Projection techniques generally aim to make a high-dimensional information space visible in form of a planar layout. However, the meaning of the resulting data projections can be hard to grasp. It is seldom clear why elements are placed far apart or close together and the inevitable approximation errors of any projection technique are not exposed to the viewer. Previous research on dimensionality reduction focuses on the efficient generation of data projections, interactive customisation of the model, and comparison of different projection techniques. There has been only little research on how the visualization resulting from data projection is interacted with. We contribute the concept of probing as an integrated approach to interpreting the meaning and quality of visualizations and propose a set of interactive methods to examine dimensionality-reduced data as well as the projection itself. The methods let viewers see approximation errors, question the positioning of elements, compare them to each other, and visualize the influence of data dimensions on the projection space. We created a web-based system implementing these methods, and report on findings from an evaluation with data analysts using the prototype to examine multidimensional datasets.

Index Terms—Information visualization, interactivity, dimensionality reduction, multidimensional scaling.

1 INTRODUCTION

A primary goal of information visualization is to find patterns and relationships in multivariate datasets. Many visualization techniques have been developed towards this goal such as multiple coordinated views \cite{2}, parallel coordinates \cite{14}, scatterplot matrices \cite{28}, and dimensionality reductions such as multidimensional scaling (MDS) \cite{5}. Dimensionality reductions are a particular class of techniques that synthesise high-dimensional data spaces onto projection spaces with much fewer dimensions, typically the two dimensions of the plane. While most visualization techniques juxtapose the different data dimensions as matrices or columns, dimensionality reductions integrate them into a planar canvas. The projection results in a so-called spatialisation (i.e., embedding) of data elements that approximately represents similarity as proximity and in turn dissimilarity as distance. Considering that the human perceptional system comprises a well-developed capacity for spatial reasoning, the assumption is that spatialisation would be a more natural way \cite{31} to analyse high-dimensional datasets since groupings, separations, and other patterns among data elements become immediately discernible.

However, there are two major caveats linked with dimensionality reduction: first, it can be challenging to interpret the positions of projected elements, and second, the errors that occur with any projection technique are not exposed to the viewer. Previous research on dimensionality reduction focuses on the efficient generation of data projections, interactive customisation of the model, and comparison of different projection techniques. There has been only little research on how the visualization resulting from data projection is interacted with. We contribute the concept of probing as an integrated approach to interpreting the meaning and quality of visualizations and propose a set of interactive methods to examine dimensionality-reduced data as well as the projection itself. The methods let viewers see approximation errors, question the positioning of elements, compare them to each other, and visualize the influence of data dimensions on the projection space. We created a web-based system implementing these methods, and report on findings from an evaluation with data analysts using the prototype to examine multidimensional datasets.
Probing Projection Goals

• Examining the Projection
• Exploring the Data
• Design Goals:
  - Show and correct approximation errors
  - Allow for multi-level comparisons
  - Spatial orientation
  - Consistent design
• Allow grouping of samples
  - Selections
  - Classes
  - Clusters

[J. Stahnke et al., 2015]
**Tooltips with statistics**

The image shows a visualization of a map with various countries highlighted, including the United States, United Kingdom, Slovenia, Poland, Austria, Israel, Luxembourg, and Brazil. The map also includes a heatmap for Portugal, displaying various statistics such as educational attainment, employees working, life expectancy, life satisfaction, self-reported health, student skills, time devoted to leisure, and years in education. Each statistic is represented with a value and deviation, such as educational attainment with a value of 35 and a deviation of -2.4σ, and life expectancy with a value of 80.8 and a deviation of +0.39σ.

[J. Stahnke et al., 2015]
Comparing Two Groups

![Heatmap example](image_url)

**South America** 3 samples

**Northern Europe** 9 samples

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Mean</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Educational attainment</td>
<td>50</td>
<td>77</td>
</tr>
<tr>
<td>Employees working ve...</td>
<td>18</td>
<td>6.2</td>
</tr>
<tr>
<td>Life expectancy</td>
<td>75</td>
<td>81</td>
</tr>
<tr>
<td>Life satisfaction</td>
<td>7.1</td>
<td>7.4</td>
</tr>
<tr>
<td>Self-reported health</td>
<td>65</td>
<td>77</td>
</tr>
<tr>
<td>Student skills</td>
<td>420</td>
<td>500</td>
</tr>
<tr>
<td>Time devoted to leisure...</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Years in education</td>
<td>16</td>
<td>19</td>
</tr>
</tbody>
</table>

[J. Stahnke et al., 2015]
Heatmap from Dimension Hover

![Heatmap from Dimension Hover](image)

[J. Stahnke et al., 2015]
Fig. 5. Comparing an individual sample with a class from the well-known Iris flower data set. In addition to the distribution of dimensions of the class and the value of the sample, the visualization also indicates sample-centric distortions using grey and white halos. When hovering a dimension in the list (see Figure 4), this allows both for an overview to spot dimensions with interesting patterns, and for a more detailed view together with the dots.

The grid is constructed by dividing up the projection space into a number of cells, with their size based on the mean of distances for the closest three points from each point. A minimum value is used to prevent the grid from getting too small to be useful, and calculations taking too long.

The values for each grid cell are calculated by averaging the values of the points in the cell, or, if there are none, the three closest points for the cell, weighting them according to their distances from the cell. This ensures smooth transitions over large gaps in the projection space, while being responsive to abrupt changes at the same time.

4.4.2 Density plots

While the heatmaps show how the values are spatially distributed in the projection space, kernel density plots in the list of dimensions show their value distributions. In the prototype, currently the plots are generated roughly equivalent to R’s bw.nrd0 function which uses Silverman’s ‘rule of thumb’ [24, p. 48]. Percentiles are indicated on the density plots to support the visual assessment. Used together with brushing and linking, it is possible to assess how a sample, or a group of samples, relates to the whole dataset.

Markers or sub-plots for selected elements are shown on the density plots in the list of dimensions, providing dynamic highlights of samples being examined (see Figure 5, lower right). Additional markers display the dimensional values mouse position in the projection space, based on the calculations done for the heatmap, making it possible to gradually track value progressions for multiple dimensions.

4.5 Examining projection errors

Besides exploring the distribution of samples and dimensions, the visualization environment allows for the integrated examination of projection errors by providing per-sample halos, distance corrections, and a dendrogram.

4.5.1 Error halos

The cumulative distance error for each point is displayed as a halo around the dot, with the radius corresponding to the relative amount of error and the value indicating the direction of the error (see Figure 6). This is intended to help the user visually understand the quality of the projection and find potentially unreliable spots. Hovering over a dot shows the errors in relation to the hovered point, to check if the distances between certain points are correct or just projection artefacts, and learn which points should be closer together or further apart.

Halos were chosen because their visual properties are a good match for the properties of the error they represent. The error does not belong to the sample, but to the projection, and as such should be displayed by the projection. A halo is clearly connected to the dot, but also part of the projection. Size was chosen as a very intuitive metric to display the amount of error, with points with a large error standing out easily. The brightness of the halo displays the direction of the error. If the other points are farther away in the projection than in the high-dimensional space, the halo is white; if they are too close in the projection, the halo is dark. White was chosen for points that should be nearer because it stands out more, and while using the prototype ourselves, we often ended up looking for ‘similar’ samples.

Size and brightness were chosen over colour or shape, as using colour would have clashed with the colouring of the dots, and different shapes were not as glanceable as changes in brightness.

4.5.2 Distance correction

After examining the approximation errors, the viewer might decide that the errors of a certain point warrant more attention. They can then visualize them by selecting to view the high-dimensional distances between the selected point and all others. This removes all projection errors when it comes to distance, but for the selected sample only. The new, corrected positions for each point are calculated by taking the vector between the selected and the other point and multiplying it by the distance error ratio between the two. The angle between them is kept as is. As a result, the other point moves directly towards or away from the specified point, correcting the distance.

Showing Error via Sample-centric Halos

[J. Stahnke et al., 2015]
White: higher levels of similarity
Gray: lower levels of similarity

Showing Projection Errors

Fig. 6. Halos represent the cumulative error for the respective samples. White indicates that a majority of samples is more similar than indicated by their distance to the given sample; grey indicates the opposite. The paths travelled by the points are shown as lines, leading from the points' original positions in the projection to the new, corrected positions (see Figure 8). This connects them to their original positions in the projection, and displays the size of the distance error at the same time. Resembling the brightness encoding of the halos, the brightness of the lines indicates whether they've moved closer or farther away. A problem with this solution is that it introduces new distortions in the spatial relationship between all other points. Only the distances directly between the selected point and the other points are reliable, whereas all the other distances are distorted, and the new positioning might lead to wrong assumptions about potential clusterings. To mitigate this problem, the correction paths are shown. Another solution would be to recompute the projection while preserving the distances from and to the selected point and being more generous with distance errors among the remaining points. This would somewhat reduce the introduced distortions. However, in a recomputed projection, the positions of the points might change significantly, most likely leading to completely different positions for all points, possibly confusing the observer even if an animation is used.

Fig. 7. Dendrograms mapped onto the projection. Left: projection with low projection error. Right: high projection error.

4.5.3 Dendrogram In addition to the visualization of errors and corrections, a dendrogram can visualize the samples with regard to their position in the clustering hierarchy. Such a dendrogram (using the same agglomerative algorithm as the clusters) overlaid onto the projection may also help to visualise high-dimensional distances on the projection space. It graphically emphasises clusters by connecting close dots through dense lines. Interestingly, the dendrogram is a surprisingly good indicator of goodness of fit: if many thick, long lines intersect, it is likely that the projection is of low quality.

To illustrate the functionality of the interface we visualize the dataset of OECD countries in the prototype (see Figure 9). The dataset contains 8 dimensions for 36 countries. First, the viewer is drawn to the projection and notices Turkey that seems to be a clear outlier, far away from all other countries. To explore why this is, the viewer can examine this sample by hovering over it. A tooltip relating Turkey to the rest of the dataset appears, showing that it deviates strongly from the mean in nearly every dimension. This indicates the positioning as outlier is probably correct.

To test this assumption and build up trust in the visualization, the viewer selects 'correct distances', showing the high-dimensional distances between Turkey and the other countries. This reveals that Turkey should be even farther apart from several of the other countries. Having confirmed that Turkey is an outlier in this dataset, the viewer uses the built-in clustering to get a sense of how the countries are grouped. Playing around with the number of clusters, they notice that there seem to be seven clusters roughly corresponding to the geographical and geopolitical placement of the countries.

Taking a closer look at the positioning of the clustered countries, they realise that the arrangement seems to roughly correspond to geographic directions: Northern and Southern countries are roughly distributed along the vertical axes, East and West along the horizontal. To find out if or how this correlates with the dimensions, the viewer first compares the different clusters. Here the differences along the dimensions are very much pronounced. Interestingly though, life expectancy is lower in Latin America than Asia, while the self-reported health is higher for the former than the latter.

After a few more comparisons between the clusters, the viewer becomes interested in the dimension life satisfaction and turns towards the heatmaps. They notice that the values for life satisfaction and self-reported health seem to be higher in the Western countries, whereas the value for employees working very long hours seems to be especially high in the countries of the far East and the South.

[http://www.oecdbetterlifeindex.org/]

[J. Stahnke et al., 2015]
User Study & Results

- Types of Questions:
  - How would you try to characterize the type X?
  - In what way are X and Y different in their properties?
  - Are the projections of X and Y correct or do they deviate? How do you interpret this?
  - Can you discover which parts of the cluster combinations are A, B, and C?

- Discussion:
  - Learnability: need more effective mechanisms for grasping the concepts behind dimensionality reduction
  - Manipulation: What happens with results?
  - Large data: What about text corpora?

[J. Stahnke et al., 2015]
Focus+Context

- Show everything at once but compress regions that are not the current focus
  - User shouldn't lose sight of the overall picture
  - May involve some aggregation in non-focused regions
  - "Nonliteral navigation" like semantic zooming
- Elision
- Superimposition: more directly tied than with layers
- Distortion
Focus+Content Overview

Embed

- Elide Data

- Superimpose Layer

- Distort Geometry

Reduce

Filter

Aggregate

Embed

[Munzner (ill. Maguire), 2014]
Elision

• There are a number of examples of elision including in text, DOI Trees, …
• Includes both filtering and aggregation but goal is to give overall view of the data
• In visualization, usually correlated with focus regions
Degree of Interest Function

- DOI = I(x) - D(x,y)
  - I: interest function
  - D: distance (semantic or spatial)
  - x: location of item
  - y: current focus point (could be more than one)
- Interactive: y changes
Elision: DOI Trees

- Example: 600,000 node tree
  - Multiple foci (from search results or via user selection)
  - Distance computed topologically (levels, not geometric)
Superimposition

• Different from layers because this is restricted to a particular region
  - For Focus+Context, superimposition is **not global**
  - More like overloading
• Lens may occlude the layer below
Superimposition with Interactive Lenses

(a) Alteration

(b) Suppression

[ChronoLenses and Sampling Lens in Tominski et al., 2014]
Superimposition with Interactive Lenses

(c) Enrichment

[Extended Lens in Tominski et al., 2014]
It can be difficult to observe micro and macro features simultaneously with complex graphs. If you zoom in for detail, the graph is too big to view in its entirety. If you zoom out to see the overall structure, small details are lost.

Focus + context techniques allow interactive exploration of an area.
Distortion Choices

• How many focus regions? One or Multiple
• Shape of the focus?
  - Radial
  - Rectangular
  - Other
• Extent of the focus
  - Constrained similar to magic lenses
  - Entire view changes
• Type of interaction: Geometric, moveable lenses, rubber sheet
Overplotting
Cartesian Distortion
Stretch and Squish Navigation

Principle: multiple views are most effective when coordinated through explicit linking.

The principle of linked views [15] is that explicit coordination between views enhances their value. In LiveRAC, as the user moves the cursor within a chart, the same point in time is marked in all charts with a vertical line. Similarly, selecting a time segment in one chart shows a mark in all of them. This technique allows direct comparison between parameter values at the same time on different charts. In addition, people can easily correlate times between large charts with detailed axis labels, and smaller, more concise charts.

Assertion: showing several levels of detail simultaneously provides useful high information density in context. Several technique choices are based on this assertion. First, LiveRAC uses stretch and squish navigation, where expanding one or many regions compresses the rest of the view [11, 17]. The accompanying video shows the look and feel of this navigation technique. The stretching and squishing operates on rectangular regions, so expanding a single chart also magnifies the entire row for the device it represents, and the entire column for the parameters that it shows. The edges of the display are fixed so that all cells remain within the visible area, as opposed to conventional zooming where some regions are pushed off-screen. There are rapid navigation shortcuts to zoom a single cell, a column, an aggregated group of devices, the results of a search, or to zoom out to an overview. Users can also directly drag grid lines or resize freely drawn on-screen rectangles. Navigation shortcuts can also be created for any arbitrary grouping, whose cells do not need to be contiguous. This interaction mechanism affords multiple focus regions, supporting multiple levels of detail.

Second, charts in LiveRAC dynamically adapt to show visual representations adapted in each cell to the available screen space. This technique, called semantic zooming [13], allows a hierarchy of representations for a group of device-parameter time-series. In Figure 3, the largest charts have multiple overlaid curves and detailed axis and legend labels. Smaller charts show fewer curves and less labeling, and at smaller sizes only one curve is shown as a sparkline [24]. On each curve, the maximum value over the displayed time period is indicated with a red dot, the minimum with a blue dot, and the current value with a green one. All representation levels color code the background rectangle according to dynamically changeable thresholds of the minimum, maximum, or average values of the parameters within the current time window. The smallest view is a simple block, where this color coding is the only information shown.

Third, aggregation techniques achieve visual scalability by ensuring dense regions show meaningful visual representations. Given our target scale of dozens of parameters and thousands of devices, the size of the matrix could easily surpass 100,000 cells. Stretch and squish navigation allows users to quickly create a mosaic with cells of many different sizes.

[McLachlan et al., 2008]
The concerns we discuss in this paper emerged in the design and evaluation of fisheye interfaces that aim to support programming [21,23]. With the specific goal of helping programmers navigate and understand source code, we have integrated a fisheye view with the Java editor in Eclipse, an open source development platform.

Basically, the fisheye view works by assigning a degree of interest (DOI) to each program line based on its a priori importance and its relation to the user's current focus in the file. Then, lines with a DOI below a certain threshold are diminished or hidden, resulting in a view that contains both details and context.

Below, we discuss the fisheye interface design used in an initial controlled experiment [21], and the design used in a later field study [23], arguing for the changes made to the initial design.

Fig. 2. The fisheye interface initially studied [21] contains an overview of the entire document shown to the right of the detail view of source code. The detail view is divided into a focus area and a context area (with pale yellow background color) that uses a fixed amount of space above and below the focus area. In the context area, program lines that are less relevant given the focus point are diminished or hidden.

Fisheye Distortion in Programming

[Jakobsen and Hornbaek, 2011]
Fig. 3. The fisheye interface evolved for use in a field study [23]. Less interesting lines are hidden in the context area by using a magnification factor of 0. However, all lines with a degree of interest above a given threshold are included in the context area. In the example shown here, the bottom context area contains more lines than can be shown simultaneously. The context can be scrolled to view lines that are not initially shown. The motivation for this change is that all the lines may be important to the user. This design thus aims to guarantee users that the context area contains all the lines they expect to find (e.g., all occurrence so far a variable that the user has selected).

3.3 Findings from User Studies

Overall, the results from our studies attest to the usefulness of fisheye interfaces to programmers. Participants in a controlled experiment preferred the fisheye interface to a linear source code interface [21]. Participants in a field study adopted and used the fisheye interface regularly and across different activities in their own work for several weeks [23]. The fisheye interface does not seem useful in all tasks and activities, however. Participants in the experiment completed tasks significantly faster using the fisheye interface, a difference of 10% in average completion time, but differences were only found for some task types. Although the results indicate usability issues, they also suggest that some tasks were less well supported by the fisheye interface. In addition, data from the field study showed periods where programmers did not use the fisheye interface, and debugging and writing new code were mentioned as activities for which the fisheye interface was not useful.

Distortion vs. Hide

[D. Koop, CSCI 627/490, Fall 2020] [Jakobsen and Hornbaek, 2011]
Research Questions

• Is a priori importance useful (and for what)?
• What does the user focus on?
  - predictability of view changes when focus changes
  - how direct user control is
  - task & context
• What interesting information should be displayed
  - degree of interest function may produce varied result sizes
• Do fisheye views integrate or disintegrate?
  - interference with other interactions; allow on-demand use?
• Are fisheye views suitable for large displays?

[Jakobsen and Hornbaek, 2011]
Distortion Concerns

- Distance and length judgments are **harder**
  - Example: Mac OS X Dock with Magnification
  - Spatial position of items changes as the focus changes
- Node-link diagrams not an issue… why?
- Users have to be made aware of distortion
  - Back to scatterplot with distortion example
  - Lenses or shading give clues to users
- **Object constancy**: understanding when two views show the same object
  - What happens under distortion?
  - 3D Perspective is distortion… but we are well-trained for that
- Think about **what** is being shown (filtering) and method (fisheye)